**Detecting Government Fraud Using Semi-Supervised Machine Learning**

Dissertation Manuscript

Submitted to National University

School of Technology and Engineering

in Partial Fulfillment of the

Requirements for the Degree of

DOCTOR OFPHILOSOPHY

by

WILLIAM ISAAC SAPP

San Diego, California

July 2025

**Abstract**

This study addressed the lack of intelligent semi-supervised fraud identification in the government domain, with a specific focus on the Paycheck Protection Program, a federally administered pandemic-era loan program that disbursed over five hundred billion dollars with limited oversight controls. While existing literature on fraud detection in machine learning has largely focused on supervised methods in credit card and healthcare contexts, there is a notable gap in research related to unsupervised and semi-supervised approaches for detecting fraud in public-sector programs.

The purpose of this study was to develop and evaluate a semi-supervised learning methodology for identifying potentially fraudulent Paycheck Protection Program loan records. This study employed a Classification Through Clustering approach using only publicly available loan data and limited court filings for labelling. Rather than depending on fully labeled datasets, the research introduced unsupervised clustering to extract structural patterns from the data, which were then incorporated into a supervised learning pipeline. This approach was designed to evaluate whether meaningful fraud detection could still be achieved in contexts where verified fraud labels are limited or incomplete in emerging fraud domains.

To test this approach, a subset of Paycheck Protection Program loans exceeding $150,000 was analyzed, including 301 manually labeled fraud cases identified from federal court records. The research process involved extensive data preprocessing, feature engineering, and implementation of three clustering algorithms: K-Means, DBSCAN, and Hierarchical clustering. Clustering outputs were evaluated using internal validation metrics & fraud capture metrics and then encoded as features in multiple classification models. The results indicated that, while the XGBoost model consistently outperformed other supervised models, cluster feature configuration did have a significant effect on model performance in several models, notably Neural Network and SVM.

Although the study was constrained by the scope of the dataset, limited label availability, and reliance on publicly sourced data, the findings demonstrate that semi-supervised learning techniques can potentially support fraud detection in high-volume, low-label environments like the PPP. The remainder of this chapter discusses the study findings’ implications for fraud analytics and methodological research within the PPP and government fraud domain and offers recommendations for practical application and future study.

**Dedication**

To my daughter, Clíodhna, my match. Watching you conquer anything in your path, be it reading, Minecraft, or the waterslide at the pool, has inspired me every day to keep persevering. This is for you.

To my son, Ruaidhrí, my bud. You were born about halfway through this process. Watching you explore the world around you continuously inspires me to keep working even if I’m stuck. This is for you.

To my mother, Dr. Janet Baltzell, PhD. I wish you were here to see this. I know you’re proud. This is for you.

To my wife, Dr. Tara Hennessy, PhD. None of this would have happened without you. You have been my rock, not just throughout this process, but since we met in Japan while you were working on your PhD. This is for you.

**Acknowledgments**

I want to start by thanking my family and friends. I’ve been a hermit for quite some time and I truly appreciate your support throughout this process. My children, Clíodhna and Ruaidhrí, thanks for sitting next to me while I work on “daddy-science”. Your smiling faces really got me through some rough patches over the last couple of years. My beautiful, amazing, awesome wife, Tara. You believed in me when I stopped believing in myself. You kept this family and our house going while I disappeared to write and recalculate. You are forever my hero.

I also want to thank Northcentral University, National University, and my dissertation committee. Dr Fulton, Dr Al-Najada, and Dr Tsapara, thank you for sticking with me throughout this entire process. Without your support, guidance, and feedback, I would have never seen this to completion.

Finaly, I want to thank the dedicated federal employees and contractors at the Federal Communications Commission, Office of Inspector General. It has been a privilege to work alongside you all for the last eight years. Your dedication to duty has been an inspiration both in the office and while completing this program. I also want to mention two retired FCC OIG employees; my former bosses. Jay Keithly, thanks for giving a young Marine Captain a chance at changing career directions, and Sharon Diskin, thanks for always having our backs and for giving me the thumbs up to start pursuing this.

**Table of Contents**

[Chapter 1: Introduction 1](#_Toc202815223)

[Statement of the Problem 3](#_Toc202815224)

[Purpose of the Study 4](#_Toc202815225)

[Introduction to Theoretical Framework 5](#_Toc202815226)

[Introduction to Research Methodology and Design (Nature of the Study) 6](#_Toc202815227)

[Research Questions 7](#_Toc202815228)

[Hypotheses 8](#_Toc202815229)

[Significance of the Study 8](#_Toc202815230)

[Definitions of Key Terms 9](#_Toc202815231)

[Summary 10](#_Toc202815232)

[Chapter 2: Literature Review 11](#_Toc202815233)

[Databases and Search Strategy 13](#_Toc202815234)

[Theoretical Framework 16](#_Toc202815235)

[Data Ethics and Legal Frameworks 22](#_Toc202815236)

[COVID-19 and The Paycheck Protection Program 25](#_Toc202815237)

[Fraud in Government Programs 28](#_Toc202815238)

[Machine Learning for Fraud Detection 32](#_Toc202815239)

[Summary 44](#_Toc202815240)

[Chapter 3: Research Method 50](#_Toc202815241)

[Research Methodology and Design Process Diagram 50](#_Toc202815242)

[Research Methodology and Design (Nature of the Study) 51](#_Toc202815243)

[Population and Sample 53](#_Toc202815244)

[Materials and Instrumentation 55](#_Toc202815245)

[Study Procedures 59](#_Toc202815246)

[Data Analysis 65](#_Toc202815247)

[Assumptions 66](#_Toc202815248)

[Limitations 68](#_Toc202815249)

[Delimitations 69](#_Toc202815250)

[Ethical Assurances 71](#_Toc202815251)

[Summary 72](#_Toc202815252)

[Chapter 4: Findings 74](#_Toc202815253)

[Data Preprocessing and Modeling 75](#_Toc202815254)

[Results 88](#_Toc202815255)

[Evaluation of the Findings 114](#_Toc202815256)

[Limitations 116](#_Toc202815257)

[Summary 117](#_Toc202815258)

[Chapter 5: Implications, Recommendations, and Conclusions 119](#_Toc202815259)

[Implications 121](#_Toc202815260)

[Recommendations for Practice 123](#_Toc202815261)

[Recommendations for Future Research 124](#_Toc202815262)

[Conclusions 126](#_Toc202815263)

[References 128](#_Toc202815264)

[Appendix A IRB Approval Letter 137](#_Toc202815265)

[Appendix B Aggregated Classification Model Results: Full Feature-set 138](#_Toc202815266)

[Appendix C Aggregated Classification Model Results: PCA Feature-set 159](#_Toc202815267)

**List of Tables**

[**Table 1** *Research Question and Hypotheses Alignment* 52](#_Toc202814997)

[**Table 2** *Original Variables for The PPP Dataset* 57](#_Toc202814998)

[**Table 3** *Experimental Variables and Hypotheses Alignment* 59](#_Toc202814999)

[**Table 4** *Supervised Classification Models and Libraries* 64](#_Toc202815000)

[**Table 5** *K-Means Clustering Metrics* 95](#_Toc202815001)

[**Table 6** *Hierarchical Clustering Metrics* 96](#_Toc202815002)

[**Table 7** *DBSCAN Clustering Metrics, PCA Feature Set* 96](#_Toc202815003)

[**Table 8** *DBSCAN Clustering Metrics, Full Feature set* 97](#_Toc202815004)

[**Table 9** *Levene's Test Results: PCA Feature-set* 100](#_Toc202815005)

[**Table 10** *Levene's Test Results: Full Feature-set* 101](#_Toc202815006)

[**Table 11** *Two-Way ANOVA: AUC ROC, PCA Feature-set* 103](#_Toc202815007)

[**Table 12** *Two-Way ANOVA: Accuracy, PCA Feature-set* 103](#_Toc202815008)

[**Table 13** *Two-Way ANOVA: Precision, PCA Feature-set* 103](#_Toc202815009)

[**Table 14** *Two-Way ANOVA: Recall, PCA Feature-set* 103](#_Toc202815010)

[**Table 15** *Two-Way ANOVA: F1 Score, PCA Feature-set* 103](#_Toc202815011)

[**Table 16** *Two-Way ANOVA: AUC ROC, Full Feature-set* 104](#_Toc202815012)

[**Table 17** *Two-Way ANOVA: Accuracy, Full Feature-set* 104](#_Toc202815013)

[**Table 18** *Two-Way ANOVA: Precision, Full Feature-set* 104](#_Toc202815014)

[**Table 19** *Two-Way ANOVA: Recall, Full Feature-set* 104](#_Toc202815015)

[**Table 20** *Two-Way ANOVA: F1 Score, Full Feature-set* 104](#_Toc202815016)

[**Table 21** *ANOVA: XGBoost, PCA Feature-set* 105](#_Toc202815017)

[**Table 22** *ANOVA: Neural Network, PCA Feature-set* 105](#_Toc202815018)

[**Table 23** *ANOVA: Random Forest, PCA Feature-set* 105](#_Toc202815019)

[**Table 24** *ANOVA: Naïve Bayes, PCA Feature-set* 106](#_Toc202815020)

[**Table 25** *ANOVA: SVM, PCA Feature-set* 106](#_Toc202815021)

[**Table 26** *ANOVA: Logistic Regression, PCA Feature-set* 106](#_Toc202815022)

[**Table 27** *ANOVA: XGBoost, Full Feature-set* 106](#_Toc202815023)

[**Table 28** *ANOVA: Neural Network, Full Feature-set* 107](#_Toc202815024)

[**Table 29** *ANOVA: Random Forest, Full Feature-set* 107](#_Toc202815025)

[**Table 30** *Welch's ANOVA: Naïve Bayes, Full Feature-set* 107](#_Toc202815026)

[**Table 31** *ANOVA: SVM, Full Feature-set* 107](#_Toc202815027)

[**Table 32** *ANOVA: Logistic Regression, Full Feature-set* 108](#_Toc202815028)

[**Table 33** *Baseline Classification Model Performance Metrics: Full Feature-set* 109](#_Toc202815029)

[**Table 34** *Baseline Classification Model Performance Metrics: PCA Feature-set* 109](#_Toc202815030)

[**Table 35** *Cluster-Feature & Baseline Comparison, Full Feature-set* 113](#_Toc202815031)

[**Table 36** *Cluster-Feature & Baseline Comparison, PCA Feature-set* 113](#_Toc202815032)

**List of Figures**

[**Figure 1** *Classification Through Clustering Framework* 18](#_Toc203064283)

[**Figure 2** *COVID-19 Deaths, by Week, in The United States* 25](#_Toc203064284)

[**Figure 3** *Labor Market Policy Responses to Recessions the United States and Europe* 28](#_Toc203064285)

[**Figure 4** *PPP Loan Application Process Flowchart* 31](#_Toc203064286)

[**Figure 5** *Logistic Regression Response Function* 34](#_Toc203064287)

[**Figure 6** *Graphic Representation of a Random Forest Decision Tree* 34](#_Toc203064288)

[**Figure 7** *SVM Hyperplane* 35](#_Toc203064289)

[**Figure 8** *Illustration of K-Means Clustering Method* 37](#_Toc203064290)

[**Figure 9** *Graphic Depiction of DBSCAN Clustering* 38](#_Toc203064291)

[**Figure 10** *Hierarchical Clustering Dendrogram* 39](#_Toc203064292)

[**Figure 11** *Research Methodology and Design Process Diagram* 51](#_Toc203064293)

[**Figure 12** *Research Methodology and Design Process Diagram (Repeated From Figure 11)* 75](#_Toc203064294)

[**Figure 13** *Correlation Matrix, Full Feature Set* 77](#_Toc203064295)

[**Figure 14** *Scatterplot Showing currentapprovalamount vs initialapprovalamount* 78](#_Toc203064296)

[**Figure 15** *Scatterplot Showing currentapprovalamount vs payroll\_proceed* 78](#_Toc203064297)

[**Figure 16** *Scatterplot Showing forgivenessamount vs currentapprovalamount* 79](#_Toc203064298)

[**Figure 17** *Scatterplot Showing forgivenessamount vs jobsreported* 79](#_Toc203064299)

[**Figure 18** *Scatterplot Showing payroll\_process vs jobsreported* 80](#_Toc203064300)

[**Figure 19** *Correlation Matrix, Key Feature Subset* 81](#_Toc203064301)

[**Figure 20** *Scree Plot for Key Feature Subset* 82](#_Toc203064302)

[**Figure 21** *PCA Loading Magnitude, Key Feature Subset* 83](#_Toc203064303)

[**Figure 22** *Scree Plot for Full Feature Set* 84](#_Toc203064304)

[**Figure 23** *PCA Projection Scatterplot and Feature Loadings (Full Set, 3Components)* 85](#_Toc203064305)

[**Figure 24** *Cluster Metrics: K-Means* 90](#_Toc203064306)

[**Figure 25** *Cluster Metrics: Hierarchical* 90](#_Toc203064307)

[**Figure 26** *Cluster Metrics: DBSCAN* 91](#_Toc203064308)

[**Figure 27** *ROC AUC Metrics per Classification Model* 92](#_Toc203064309)

[**Figure 28** *Accuracy Metrics per Classification Model* 93](#_Toc203064310)

[**Figure 29** *Precision Metrics per Classification Model* 93](#_Toc203064311)

[**Figure 30** *Recall Metrics per Classification Model* 94](#_Toc203064312)

[**Figure 31** *F1 Score Metrics per Classification Model* 94](#_Toc203064313)

[**Figure 32** *Q-Q Plot: AUC ROC* 98](#_Toc203064314)

[**Figure 33** *Q-Q Plot: Accuracy* 99](#_Toc203064315)

[**Figure 34** *Q-Q Plot: Precision* 99](#_Toc203064316)

[**Figure 35** *Q-Q Plot: Recall* 100](#_Toc203064317)

[**Figure 36** *Q-Q Plot: F1 Score* 100](#_Toc203064318)

[**Figure 37** *F1 Scores for PCA and Full Feature-set Derived Cluster Features, Full Feature Dataset* 110](#_Toc203064319)

[**Figure 38** *F1 Scores for PCA and Full Feature-set Derived Cluster Features, PCA Feature Dataset* 111](#_Toc203064320)

# Chapter 1: Introduction

The Paycheck Protection Program (PPP), signed into law in April 2020, was designed to provide financial relief to American small businesses during the early stages of the COVID-19 Pandemic (Demko et al., 2021). The PPP enabled small businesses, including sole proprietorships and self-employed persons, to take low-interest business loans to continue paying their employees and covering other business operating costs during the widespread lockdowns put in place at the time. The PPP facilitated nearly $660 Billion in uncollateralized loans, approximately 90% of which were ultimately forgiven (Emmons & Dahl, 2022).

While many government subsidy or financial assistance programs are vulnerable to fraudulent activity, initial reporting suggests that at least $200 Billion in forgiven PPP loans were likely fraudulent (USSBA OIG, 2023). Government fraud investigations are primarily informed by whistleblowers and conducted by attorney investigators (Council of the Inspectors General on Integrity and Efficiency, 2011). Government agencies are increasingly interested in employing machine learning within their oversight framework (West, 2021). Although the use of machine learning tools in fraud detection is well-established within the financial sector, published government applications outside Medicare fraud are limited.

Two machine learning tasks are supervised and unsupervised learning. Within the fraud detection domain, supervised machine learning typically involves using a training set of previously identified or labeled data that is flagged fraudulent or not fraudulent. This labeled training dataset is used to train a fraud detection model, which will ultimately be used against unlabeled data to determine whether it is fraudulent or not. The quality of the training data is largely deterministic of the quality of the trained algorithm. For example, the bias introduced in the training data (e.g., the training data only contains fraudulent loan applications from a particular minority group) will be present in the model and applied to any future data analyzed by the algorithm (Dridi, 2022a).

Unsupervised machine learning does not require previously labeled training data for model development. Unlike supervised machine learning, unsupervised machine learning tools can identify relationships and discover insights about a dataset without the supervision of training or prior knowledge of independent variables. In fact, one subset of unsupervised machine learning, principal components analysis (PCA), is used to identify the most influential independent variables within a dataset. Other common forms of unsupervised machine learning include clustering, which groups similar records together, and anomaly detection, which identifies outliers (Dridi, 2022b).

Supervised, semi-supervised, and unsupervised ML techniques hold promise for detecting PPP fraud. Machine learning is used extensively in the financial fraud detection sector (Ashtiani & Raahemi, 2022). Financial institutions and other private sector organizations regularly employ supervised machine learning techniques to detect fraudulent credit card |transactions, fraudulent loan applications, and other electronic transactions (A. Ali et al., 2022). Similarly, the healthcare sector uses unsupervised learning to identify anomalies which aid in diagnosis and treatment plans (Nassif et al., 2021). While typical fraud detection using ML employs supervised machine learning which rely on training datasets, unsupervised machine learning does not require labeled data. Given the hundreds of billions in PPP funds and the lack of previously labeled fraudulent training data (Bailey et al., 2021), there is a growing need for fraud detection procedures that are still effective when employed using previously unstudied PPP loan application data. In this case, a semi-supervised approach using unsupervised machine learning can be employed to cluster loan applications and identify anomalous loan applications which could then be flagged for further investigation.

## Statement of the Problem

The problem to be addressed in this study is the lack of intelligent unsupervised fraud identification in the government domain, specifically in the Paycheck Protection Program. As outlined by Ali et al. (2022) and Minastireanu & Mesnita (2019), a substantive number of studies focus on supervised machine-learning techniques, resulting in a notable gap in the literature on unsupervised learning in fraud detection. Similarly, Ali et al. (2022) and Dridi (2022a) note that much of the recently published work on fraud detection using machine learning focuses on the credit card or healthcare fraud domains.

Specifically within the government fraud domain, fraud within the United States PPP is a high-profile issue that affects the integrity of the program and the welfare of the American economy. According to Bailey et al. (2021) and Ma & McKinnon (2020), over five hundred billion dollars in PPP loans were disbursed with little regulatory or oversight controls put in place. Initial law enforcement investigations revealed pervasive fraud in the program. Bailey et al. ( 2021) suggest that completed PPP fraud investigations and data from previous relief programs (e.g., transfer learning) could be used to develop machine learning models aimed at detecting PPP fraud. However, fraud detection is typically conducted using supervised machine learning techniques that rely on labeled data, which depends on Department of Justice (DOJ) indictments or court filings for PPP fraud. Reliance on the limited number of confirmed fraud cases will introduce models trained using only the high-profile cases DOJ investigated. This may introduce significant bias, potentially resulting in a high false negative rate (Benala & Tantati, 2022). Short of an effective and unbiased fraud detection scheme, PPP loan fraudsters will likely continue to benefit at the expense of the United States taxpayer. Additionally, without precise tools to detect PPP fraud, opponents of the program are free to make baseless claims against its general effectiveness and use them as ammunition to oppose future disaster relief funding.

## Purpose of the Study

The purpose of this study is to develop intelligent semi-supervised fraud identification methodologies in the government domain, specifically in the Paycheck Protection Program. The practical application of semi-supervised learning for fraud detection within the PPP will increase program integrity and public trust in disaster relief efforts Additionally, there is an opportunity to apply models developed using PPP data to similar government program datasets via transfer learning. Unsupervised machine learning algorithms do not rely on labeled training data; therefore, the lack of reliably identified fraudulent data within a real-world dataset is not a limiting factor, as identified by Benala & Tantati (2022).

While previous work incorporating Unsupervised ML with PPP loan fraud detection focused on risk mitigation for financial institutions using proprietary methods and non-public financial records such as clustering using existing bank records for businesses with similar profiles (Crowe, n.d.), this study presents a novel approach to PPP fraud detection using a classification through clustering methodology intended to support government-led fraud detection. This study will use the publicly available, open-access PPP Loan dataset to evaluate the effectiveness of unsupervised learning techniques such as clustering, anomaly detection, and PCA. While the population for this study is the complete record of PPP loan applications, this study will limit analysis to an (albeit large) sample of PPP loan applications over $150k. This sample reflects the data population, including all instances over the predetermined threshold, resulting in nearly one million records. This study leverages secondary data, is location agnostic, and will be conducted at the researcher's home location.

Data gathering and collection will consist of downloading the open-access dataset from its government data publisher, in this case the Small Business Administration (SBA). Once data is loaded and requisite transformations and normalizations are conducted, it will be analyzed using a series of unsupervised machine learning algorithms encompassing PCA, clustering, and anomaly detection. Additionally, the limited number of labeled fraudulent applications will be used as part of a training dataset to conduct supervised machine learning and binary classification for cross-validation.

While the dependent variable in this study is the binary classification likely fraudulent vs not likely fraudulent, independent variables will include loan application features such as dollar amount, credit history, or employment status. The finalized list of independent variables will be determined as a result of PCA. For example, unsupervised clustering will group fraudulent and non-fraudulent activity. The measures of effectiveness for unsupervised learning models will occur in two methods; clustering effectiveness will be measured with common unsupervised evaluation metrics such as cohesion and separation, while the small number of previously labeled data (e.g., from guilty pleas and verdicts on alleged fraud) enable measuring of typical supervised machine learning metrics such as precision and recall (Dridi, 2022b)

## Introduction to Theoretical Framework

This study will utilize the classification through clustering framework proposed by (López et al., 2012). In their work, the authors conducted classification via unsupervised machine learning using clustering algorithms to sort records into a finite number of clusters, enabling labeling or classification of the data. In addition to clustering, the authors also conducted PCA to reduce the dimensionality of the dataset.

Through the classification through clustering framework, previously unlabeled PPP loan data can be analyzed for both key independent variables (i.e., via PCA) and sorted into clusters based on likely fraudulent and likely non-fraudulent applications. The fraud identification problem is ultimately a binary classification problem (i.e., fraudulent vs non-fraudulent), which is typically performed using a training dataset and supervised machine learning algorithms such as logistic regression. However, the noted lack of previously labeled data, as identified Bailey et al. (2021), precludes the use of effective binary classification via supervised machine learning alone.

## Introduction to Research Methodology and Design (Nature of the Study)

This study will use a quantitative experimental design to identify key features and methodologies to identify PPP loan fraud using unsupervised and supervised machine learning. According to Barroga & Matanguihan (2022), quantitative research is the appropriate methodology for comparing relationships among variables. Similarly, experimentation is the most efficient research design to establish cause and effect between independent and dependent variables (Stoner et al., 2023). By systematically manipulating independent variables, such as clustering techniques or preprocessing methods, while measuring their effects on fraud detection accuracy, experimentation provides a structured approach to validate hypotheses. This design enables the evaluation of specific methodologies, such as the integration of PCA and semi-supervised learning, under controlled conditions to assess their impact on model performance. Furthermore, experimentation ensures reproducibility, allowing this study to contribute practical insights to both the theoretical development of semi-supervised learning and its application in government fraud detection.

In this study, existing unsupervised learning algorithms and methodologies such as clustering, anomaly detection, and PCA will be the independent variables, while the dependent variables are the resulting performance metrics when applied to the PPP loan dataset. Additionally, unsupervised machine learning model effectiveness will be compared to previously identified PPP loan fraud using supervised learning performance metrics. The quantitative research methodology is appropriate as the goal is to evaluate the relationship among variables, in this case, the relationship between the various unsupervised machine learning models and the performance evaluation metrics when applied to the PPP loan dataset. Experimentation will enable the control and manipulation of the independent variable (the choice of algorithm) to achieve the highest performance evaluation metrics.   
 This quantitative experimental study will use publicly available PPP loan data and existing unsupervised and supervised machine learning algorithms to detect fraudulent PPP loan applications. Collection of this data will be performed electronically via direct download from the United States Small Business Administration website. All experimentation will be conducted within a Google Colab environment using Python.

## Research Questions

### RQ1

What are the key features or variables associated with fraudulent loan applications within the PPP?

### RQ2

What novel combination of existing unsupervised and supervised learning models can effectively identify fraudulent activity within the PPP?

## Hypotheses

### H10

PPP Loan applications are best clustered and further classified given the complete list of values of each feature or variable in the dataset.

### H1a

PPP Loan applications are best clustered and further classified given the values of specific features or variables in the dataset.

### H20

All combinations of unsupervised and supervised models perform identically in detecting fraud: Model1=Model2=Modelk.

### H2a

Not all unsupervised learning models used in conjunction with supervised learning models perform identically in detecting fraudulent activity. At least two model combinations differ.

## Significance of the Study

Much of the published research on ML-driven fraud detection relies on supervised ML given a robust training dataset (Dridi, 2022a). When there is a significant imbalance in the dataset (i.e., when there is a disproportionate number of records labeled not fraudulent versus fraudulent) several imbalance compensation techniques are typically applied to reduce their effects on algorithm development. However, when presented with a unique dataset in an underrepresented fraud detection domain (e.g., government fraud) solely supervised ML techniques include either incorporating unrelated training data from previous fraudulent activity investigations (Bailey et al., 2021), whereas solely unsupervised ML often require significant third party data such as bank records to increase dimensionality and variability (Crowe, n.d.). Incorporating unsupervised ML for anomaly detection, PCA, and clustering for fraud detection using solely the government published PPP loan dataset in conjunction with supervised learning techniques, and then comparing findings to the limited known fraudulent activity (as determined by fully prosecuted cases) will provide the framework for a novel classification through clustering fraud detection methodology not limited by the traditional requirement of training datasets.

## Definitions of Key Terms

### 18 U.S.C. §§ 1001, 1342

Title 18, United States Code (U.S.C.) contains the laws of the United States pertaining to crimes. § 1001, Statements or Entries Generally, covers false statements made in matters involving each branch of government (*18 USC 1001: Statements or Entries Generally*, 2004). § 1342, Fraud by Wire Radio or Television, covers the use of electronic means to defraud or obtain money under false pretense (*18 USC 1343: Fraud by Wire, Radio, or Television*, 2008).

### Paycheck Protection Program

Funded by congress in the spring of 2020, the Paycheck Protection Program (PPP) made available nearly $670 billion to eligible small business via government backed loans which could eventually be partially or completely forgiven (Humphries et al., 2020).

### Unsupervised Machine Learning

Unsupervised machine learning is a form of machine learning used to identify patterns and anomalies in data without the need for previously labelled data (Dridi, 2022b).

### Semi-Supervised Machine Learning

Semi-supervised learning combines elements of both supervised and unsupervised learning, which is useful when a small set of labeled data is available, but most of the data remains unlabeled (Dridi, 2022a).

### Supervised Machine Learning

Supervised machine learning is a form of machine learning used to identify patterns and anomalies in a dataset. Supervised machine learning encompasses a series of algorithms which are trained against previously labelled data (Dridi, 2022a).

## Summary

There is extensive research exploring the application of supervised machine learning techniques to aid in fraud identification in the private and financial sectors. However, since supervised machine learning requires previously labelled data, these techniques will not be effective against a novel dataset or fraud identification problem. To address this gap in literature this study will investigate the application of semi-supervised machine learning techniques to aid in fraud detection using the PPP loan dataset. Comparing results of various supervised adnd unsupervised machine learning algorithms using established measures of effectiveness, this study aims to develop a novel methodology for fraud identification when presented with an unlabeled dataset.

# Chapter 2: Literature Review

The purpose of this literature review is to establish a foundational understanding of machine learning methodologies, ethical considerations, and the unique operational context required for effective fraud detection within the Paycheck Protection Program (PPP). The rapid deployment of PPP funds during the COVID-19 pandemic exposed significant vulnerabilities within public sector programs to fraud and misuse. This study specifically addresses these challenges by exploring semi-supervised machine learning models tailored to the imbalanced datasets characteristic of government-administered financial aid programs, which frequently feature scarce instances of fraudulent activity relative to the vast number of legitimate claims (Itri et al., 2019; Zhao et al., 2024).

This chapter explores relevant literature across key areas to address the complexities involved in this type of fraud detection. These areas include foundational data science methodologies, theoretical frameworks that guide the interpretation and design of fraud detection models, ethical and legal considerations necessary to ensure responsible AI deployment, and the specific operational context of the PPP. Each section builds upon a synthesis of studies, integrating key findings and highlighting critical research gaps that inform the methodology choices outlined in Chapter 3 (Debener et al., 2023; Gui et al., 2024; Rixom et al., 2021).

The structure of this literature review is organized as follows:

1. **Theoretical Framework**: This section examines clustering techniques and dimensionality reduction methods as applied to fraud detection, particularly within imbalanced datasets. The Fraud Triangle Theory and its expanded versions offer insights into the behavioral drivers of fraud, connecting technical approaches to behavioral risk indicators and guiding the study's design (Awang et al., 2020).
2. **Data Ethics and Legal Frameworks**: Ethical considerations, such as fairness, accountability, and privacy, are critical when deploying machine learning in public sector contexts. This section discusses the implications of frameworks like GDPR and CCPA, ensuring the study aligns with legal standards and ethical expectations in government applications (Emilio Ferrara, 2023; Koreff et al., 2023).
3. **COVID-19 and the Paycheck Protection Program**: The PPP’s rapid deployment during the pandemic provides a unique case study of fraud vulnerabilities in government-administered relief programs. This section addresses the inherent risks and operational challenges within the PPP, using comparative studies to highlight the need for fraud detection strategies adaptable to high-volume, high-urgency scenarios (Bozza, 2024; Miller & Bertozzi, 2024).
4. **Fraud in Government Programs**: Broadening the context, this section reviews challenges across other public sector programs, discussing macro-level governance and meso-level organizational controls that influence fraud detection. This comparison underscores the importance of adapting fraud detection methods to meet the regulatory and operational needs specific to public sector programs (A. Ali et al., 2022; King et al., 2023).
5. **Machine Learning for Fraud Detection**: This section examines the application of supervised, unsupervised, and semi-supervised learning techniques for fraud detection, with an emphasis on semi-supervised models that effectively handle the imbalanced data typical in fraud cases. Evaluation metrics are reviewed to clarify performance considerations specific to imbalanced datasets in fraud detection (Debener et al., 2023; Gui et al., 2024).
6. **Summary**: The chapter concludes with a synthesis of research gaps, setting up a direct lead-in to Chapter 3, where these insights inform the selection of methodologies. This includes the rationale for focusing on semi-supervised models and clustering techniques, which align with the identified challenges and operational demands of fraud detection within the PPP.

To ensure a rigorous approach, this literature review relied on peer-reviewed studies and comprehensive database searches that included EBSCOhost, ProQuest, Google Scholar, and arXiv. This multi-source approach supports a balanced review, incorporating established methodologies and emerging trends to ground this study’s design choices in theoretical and practical insights.

## Databases and Search Strategy

To ensure a comprehensive and academically rigorous literature review, this study employed a multi-database search strategy, drawing on diverse sources to cover machine learning methodologies, fraud detection in government programs, and data ethics frameworks. Primary databases accessed include the Northcentral University (NCU) Library with EBSCOhost and ProQuest platforms, Google Scholar, and arXiv for preprints and cutting-edge research. Each database contributed unique insights, with academic journals providing validated studies and arXiv supporting recent developments in machine learning. This multi-source approach ensured a well-rounded review that includes both foundational theories and emerging trends in fraud detection.

### Search Terms and Keywords

The following search terms and combinations were used to ensure a thorough review across each main section of this chapter. Specific terms were selected based on relevance to fraud detection, machine learning, and government program oversight, with search parameters customized by section and sub-section to target the most pertinent literature.

**1. Data Science Methodologies for Fraud Detection**

* **Supervised Learning Techniques**:
  + Search terms: “supervised learning for fraud detection,” “classification in fraud detection,” “decision trees fraud detection,” “random forests in fraud detection,” “logistic regression for fraud”
* **Semi-supervised Learning Techniques**:
  + Search terms: “semi-supervised learning in fraud detection,” “imbalanced data semi-supervised,” “semi-supervised machine learning fraud,” “credit card fraud semi-supervised learning,” “PPP fraud semi-supervised models”
* **Unsupervised Learning Techniques**:
  + Search terms: “unsupervised learning fraud detection,” “clustering for fraud detection,” “anomaly detection unsupervised learning,” “autoencoders fraud detection,” “PCA in fraud detection”

**2. Theoretical Frameworks**

* **Clustering and Dimensionality Reduction**:
  + Search terms: “clustering for fraud detection,” “K-means fraud detection,” “hierarchical clustering in finance,” “dimensionality reduction techniques fraud,” “principal component analysis (PCA) fraud detection”
* **Fraud Triangle and Related Theories**:
  + Search terms: “Fraud Triangle theory,” “expanded Fraud Triangle,” “behavioral theories fraud detection,” “opportunity pressure rationalization fraud,” “capability as a factor in fraud”

**3. Data Ethics and Legal Frameworks**

* **Ethical Implications of Machine Learning**:
  + Search terms: “data ethics in machine learning,” “fairness and bias in AI,” “AI accountability in fraud detection,” “transparency in machine learning fraud detection,” “privacy concerns machine learning government programs”
* **Legal Standards and Regulations**:
  + Search terms: “GDPR machine learning,” “CCPA and fraud detection,” “regulatory standards in machine learning,” “data protection and government programs,” “AI regulations and compliance”

**4. COVID-19 and the Paycheck Protection Program (PPP)**

* **PPP Loan Effectiveness**:
  + Search terms: “COVID-19 PPP program,” “fraud in Paycheck Protection Program,” “PPP loan misuse,” “government relief fraud,” “financial aid fraud detection”

**5. Fraud in Government Programs**

* **Macro-Level Governance and Regulatory Challenges**:
  + Search terms: “government fraud prevention strategies,” “public sector fraud detection,” “macro-level fraud governance,” “regulatory challenges in fraud detection,” “public policy on fraud”
* **Meso-Level Organizational Controls**:
  + Search terms: “organizational controls fraud prevention,” “internal fraud detection controls,” “auditing practices public sector,” “PPP fraud detection organizational practices”

**6. Machine Learning for Fraud Detection**

* **Evaluation Metrics for Imbalanced Data**:
  + Search terms: “fraud detection metrics,” “evaluation metrics imbalanced data,” “precision-recall imbalance fraud,” “AUC-ROC fraud detection,” “confusion matrix fraud detection”

These search terms guided a systematic review process that ensured the inclusion of both foundational and emerging literature across key topics. Boolean operators (AND, OR) were applied to expand or narrow searches as needed, and filters for peer-reviewed publications and recency (primarily 2018-2024) were used to maintain relevance and academic rigor. Each selected source contributes to a layered understanding of fraud detection methods, ethical considerations, and government program oversight, grounding this study in both theoretical and applied research.

## Theoretical Framework

Fraud detection within the PPP presents distinct challenges due to highly imbalanced datasets and limited labeled instances of fraudulent activity. To address these challenges, classification through clustering serves as a primary methodological approach, leveraging unsupervised and semi-supervised learning to identify outliers that may indicate fraud. This approach clusters similar data points, enabling the detection of anomalies as potential fraud cases without the extensive labeling required by traditional supervised learning models (Gui et al., 2024; Miller & Bertozzi, 2024). Supported by dimensionality reduction techniques and the Fraud Triangle behavioral model, classification through clustering provides a robust and interpretable framework that addresses both technical and behavioral dimensions of fraud risk.

### Classification through Clustering in Fraud Detection

Classification through clustering focuses on grouping data points based on similarity, creating clusters that represent "normal" behavior patterns. In the context of fraud detection, these clusters establish baseline behaviors, while outliers—data points that deviate significantly from cluster norms—are flagged as potential fraud indicators. This technique is particularly effective for highly imbalanced datasets, where labeled fraud cases are scarce relative to the vast number of legitimate transactions (Debener et al., 2023; Itri et al., 2019).

Incorporating insights from López et al. (2012), classification through clustering achieves accurate predictive performance in semi-supervised environments, using clustering to effectively categorize outliers even with minimal labeled data. By forming clusters of similar transactions, this method can classify previously unseen cases based on their proximity to identified clusters, enhancing the model’s ability to generalize in scenarios with minimal labeled data. In the PPP, classification through clustering addresses the need for rapid and scalable fraud detection in a high-volume, low-fraud setting, effectively leveraging unsupervised learning to detect anomalies.

**Figure 1**   
*Classification Through Clustering Framework*

A diagram of a clustering process

Description automatically generated

***Note.*** *Source* (López et al., 2012)

### Clustering Techniques for Fraud Detection

K-Means Clustering, DBSCAN (Density-Based Spatial Clustering of Applications with Noise), and Hierarchical Clustering are some of the most used algorithms for detecting outliers and anomalies within large datasets. These methods enable the grouping of loan applications with similar features, and any significant deviations from these clusters may suggest fraudulent behavior.

* **K-Means Clustering**: This algorithm partitions data into a predefined number of clusters based on feature similarity. In the context of the PPP, businesses with comparable payroll sizes, loan amounts, and industry classifications would be grouped together. Outliers within these clusters—such as businesses with inflated payrolls relative to their peers—would be flagged for further investigation (López et al., 2012).
* **DBSCAN**: Unlike K-Means, DBSCAN does not require the number of clusters to be specified beforehand. It detects clusters based on the density of data points, making it particularly useful for identifying smaller, densely packed clusters of fraudulent activity that might otherwise be overlooked in larger datasets (Carcillo et al., 2021).
* **Hierarchical Clustering**: This method creates a tree-like structure of nested clusters. It is useful for detecting fraud at multiple levels of granularity, helping to identify not only individual anomalies but also broader patterns of suspicious behavior within certain segments of the data (Zhao et al., 2024).

### Clustering and Principal Component Analysis (PCA) in Fraud Detection

Clustering is a fundamental technique in fraud detection, offering a means to group similar data points and identify anomalies indicative of fraudulent activities. Unlike supervised methods, clustering operates without labeled data, making it particularly valuable for applications where labeled instances of fraud are scarce. Techniques such as k-means, DBSCAN, and hierarchical clustering have been widely used in the literature.

Carcillo et al. (2021) ilustrated how clustering can be integrated into hybrid supervised-unsupervised frameworks. For example, clustering algorithms can segment loan applications into groups based on shared attributes, flagging outliers as potential fraudulent cases. These flagged cases can then be passed to a supervised classifier for further evaluation, improving overall fraud detection rates.

Dimensionality reduction, particularly PCA, enhances clustering by simplifying high-dimensional data while preserving the most significant variance in the dataset. This is especially important in datasets like those associated with the PPP, where numerous variables such as loan amount, applicant history, and lender behavior contribute to data complexity. Miller & Bertozzi (2024) demonstrated the combined use of PCA and graph-based clustering to identify anomalous patterns in high-volume datasets. By projecting data into a lower-dimensional space, PCA not only reduces computational costs but also enhances clustering performance.

**Advantages of Clustering and PCA in Fraud Detection**:

1. **Scalability**: Clustering and PCA can efficiently handle large datasets, making them suitable for high-volume programs like the PPP.
2. **Flexibility**: These methods are adaptable to different fraud scenarios, from small-scale fraud networks to systemic patterns.

**Challenges and Mitigations**:

1. **Data Quality**: Clustering accuracy depends on clean and well-preprocessed data. Feature engineering and outlier handling are critical steps.
2. **Model Interpretability**: PCA reduces interpretability by transforming original features. However, explainable AI techniques can help bridge this gap.

### Supporting Framework: The Fraud Triangle and Its Relevance

While clustering offers a data-driven basis for detecting anomalies, the **Fraud Triangle** provides a behavioral context for interpreting these outliers. The Fraud Triangle, which identifies pressure, opportunity, and rationalization as key factors driving fraud, complements classification through clustering by linking data patterns to potential motivations for fraudulent activity. In the PPP, where financial pressures from the COVID-19 pandemic and the rapid availability of funds created new fraud risks, the Fraud Triangle aids in contextualizing anomalies by associating certain deviations with underlying behavioral factors (Awang et al., 2020; Bozza, 2024).

Integrating the Fraud Triangle with classification through clustering enhances model interpretability by providing a behavioral rationale for data points that diverge from cluster norms. For instance, anomalies flagged in the clustering process that align with high-risk indicators from the Fraud Triangle may represent cases where financial pressure or opportunity contributed to potentially fraudulent behavior. This integration supports a more adaptive fraud detection model, responsive to the unique characteristics of the PPP.

### Expanded Fraud Triangle: Capability as a Factor

An expanded version of the Fraud Triangle introduces capability as a fourth factor, addressing the role of specialized knowledge or access in enabling individuals to exploit program vulnerabilities. Capability is especially relevant in the PPP context, where individuals with insider knowledge or technical skill may engage in complex fraud schemes, concealing their actions within legitimate-looking clusters. Including capability as a factor informs the interpretation of clustered outliers, recognizing that some anomalies may reflect intentional, concealed fraud strategies rather than random deviations (King et al., 2023; Zhao et al., 2024).

Incorporating capability into classification through clustering enhances the model's predictive power by connecting technical anomalies with behavioral insights, allowing for a nuanced understanding of outliers. This framework supports semi-supervised learning by adding a behavioral layer that aids in categorizing unusual data points, effectively balancing technical accuracy with contextual relevance.

### Conclusion of the Theoretical Framework

The theoretical frameworks outlined in this section—clustering techniques, dimensionality reduction, and behavioral models such as the expanded Fraud Triangle—provide a comprehensive basis for the methodological choices in this study. By combining data-driven approaches with behavioral insights, these frameworks support a nuanced model capable of identifying fraud even within highly imbalanced datasets. This integration of technical methods with behavioral context allows for a robust fraud detection approach that aligns with the ethical, operational, and regulatory considerations specific to the PPP.

## Data Ethics and Legal Frameworks

Deploying machine learning for fraud detection within government programs such as the Paycheck Protection Program (PPP) necessitates adherence to stringent ethical and legal standards. This section examines critical aspects of data ethics, including fairness, accountability, and privacy, alongside legal frameworks like the General Data Protection Regulation (GDPR) and the California Consumer Privacy Act (CCPA), which govern data use and model transparency.

### Ethical Considerations in Machine Learning

Ethical considerations in machine learning are especially important in public sector applications, where models impact not only individual privacy but also public trust. Issues such as fairness and bias mitigation are paramount, as machine learning models trained on imbalanced datasets may inadvertently reinforce existing biases, leading to unfair treatment of certain groups. Studies emphasize the importance of fairness-aware algorithms and bias detection in semi-supervised models, which help prevent unjust classification outcomes (Emilio Ferrara, 2023; Zhou et al., 2022). Transparency in model operations is another essential factor; explainable AI (XAI) methodologies enable stakeholders to understand and assess model decisions, particularly when potential fraud is flagged (Koreff et al., 2023).

For the PPP, achieving ethical AI deployment requires balancing model complexity with interpretability. High-performing models that lack transparency may undermine public confidence and accountability, especially in cases where decisions impact funding or legal repercussions. To address these challenges, recent developments in XAI provide interpretability techniques that make machine learning decisions more accessible to both administrators and the general public, supporting ethical deployment in government settings (Sarker, 2021; Zhao et al., 2024).

### Legal Frameworks for Fraud Detection

Compliance with legal standards such as GDPR and CCPA is fundamental for machine learning models used in government fraud detection. GDPR mandates strict guidelines for data protection, including the rights of individuals to access, rectify, or erase their data, and requires organizations to adopt privacy-by-design principles in their data processing activities. In the context of fraud detection, GDPR's principles of data minimization and purpose limitation are essential, guiding the use of personal data strictly for specific and justified purposes (Emilio Ferrara, 2023).

Similarly, CCPA enhances data privacy rights by granting consumers greater control over their personal information, including the right to opt out of data sales and request the deletion of their information. In government-administered programs like the PPP, adhering to these regulatory frameworks requires careful management of data sources and processes to ensure data use aligns with both ethical and legal standards. Privacy-aware machine learning techniques, which protect individual data during the fraud detection process, play a critical role in regulatory compliance, supporting model development that respects individual privacy while remaining effective for large-scale fraud detection (A. Ali et al., 2022).

### Explainability Requirements for AI

Transparency and explainability in AI are critical for fraud detection in government programs, where decisions carry significant implications for fairness, accountability, and public trust. Machine learning models used to detect fraud within the PPP must balance high accuracy with the need for clear, understandable explanations of their processes. This requirement is particularly challenging in semi-supervised learning, where clustering algorithms, such as K-means or DBSCAN, introduce opacity due to their reliance on unsupervised patterns and the lack of direct interpretability in cluster assignments (Koreff et al., 2023). For example, a cluster flagged as fraudulent might combine unexpected features (e.g., high loan-to-employee ratios or anomalies in business types), making it difficult for auditors or non-technical stakeholders to understand the rationale behind a flagged decision.

This opacity is problematic in government fraud detection because interpretability is essential to justify decisions that can affect individuals or businesses. For instance, in the PPP, the inability to explain why a loan was flagged for investigation could undermine public trust and lead to claims of bias or unfair targeting (Miller & Bertozzi, 2024). Addressing this challenge involves navigating the trade-offs between interpretability and model performance. Highly interpretable models, such as logistic regression, may sacrifice performance in complex fraud scenarios, while advanced models, like semi-supervised ensembles, require explainability techniques to bridge this gap (King et al., 2023).

Emerging explainability tools, such as SHapley Additive exPlanations (SHAP) and counterfactual explanations, offer enhanced insights compared to traditional methods like Local Interpretable Model-agnostic Explanations (LIME). SHAP provides consistent feature importance rankings, while counterfactual explanations help identify minimal changes required to alter a model’s decision (Koreff et al., 2023). These methods are particularly valuable in semi-supervised settings, where both labeled and unlabeled data contribute to decision-making, making model outputs inherently more complex. Regulatory and ethical frameworks further emphasize the importance of explainability, ensuring fairness in detecting fraud patterns and preventing discriminatory practices (Miller & Bertozzi, 2024). By integrating explainability tools with high-performing semi-supervised models, government programs can responsibly deploy AI systems that are both effective and accountable.

## COVID-19 and The Paycheck Protection Program

The PPP was launched in response to the economic crisis of the COVID-19 pandemic, aiming to provide quick financial relief to businesses affected by lockdowns. Administered rapidly, the PPP disbursed over $800 billion in loans, prioritizing speed over stringent vetting processes. This urgency exposed vulnerabilities, as highlighted in the SBA OIG Report 23-09, which estimated over $200 billion in potentially fraudulent PPP and EIDL loan disbursements due to reduced controls and expedited approvals (USSBA, 2023).

The SBA’s challenges reflect broader issues in fraud detection for government aid programs, where traditional models are often insufficient for high-volume, urgent contexts. These circumstances underscore the need for adaptive machine learning models, such as classification through clustering, which can identify fraud even within imbalanced data settings (Bozza, 2024; King et al., 2023). For the PPP, where legitimate applications vastly outnumber fraudulent ones, semi-supervised methods provide an efficient approach to identifying outliers and prioritizing cases for further investigation (A. Ali et al., 2022).

In adapting machine learning models to the PPP’s high-volume, low-fraud environment, this study aims to contribute to enhanced fraud detection frameworks applicable to future government relief initiatives. By implementing techniques capable of detecting subtle patterns associated with fraud, this approach aligns with the SBA’s focus on improving oversight and accountability in federal aid disbursements.

**Figure 2**   
*COVID-19 Deaths, by Week, in The United States*

*A graph of blue lines

Description automatically generated*

***Note.*** *Provisional COVID-19 Deaths, by Week, in The United States, Reported to* (CDC, 2020)

The CARES Act, signed into law in late March 2020, along with the Consolidated Appropriations Act, signed in late December 2020 (Department Of Treasury, 2024), authorized the PPP which has since provided 934 Billion dollars in federally backed loans and grants to small businesses throughout the country (The National Law Review, 2023). Small businesses, typically under 1500 employees, sole proprietorships, and other public interest organizations under 500 employees were considered eligible to take-out two-year loans from private financial institutions (i.e., banks) and could request loan forgiveness or conversion to grants as long as the usage requirements were met. Usage requirements for PPP loans included paying existing employees, rehiring recently released employees, and up to 40% on mortgage, rental, and utility costs (Sabasteanski et al., 2021).

### PPP Loan Effectiveness

While the PPP was created to provide financial lifelines to small businesses during the COVID-19 pandemic, its overall effectiveness has been the subject of debate. The program distributed over $934 billion in loans through various rounds of funding, with the goal of preventing job losses and business closures. According to the Small Business Administration (SBA), more than 11 million businesses received PPP loans (USSBA OIG, 2023). However, despite the program’s scale, questions have been raised about whether it efficiently met its objectives and whether it disproportionately benefited certain businesses.

Independent evaluations have found that while the PPP may have helped many businesses survive, it did so at a significant cost per job saved. Autor et al. (2022) estimate that the PPP preserved between 2 to 3 million jobs annually during the pandemic, but at an exceptionally high cost of $169,000 to $256,000 per job. Much of this inefficiency stemmed from the program's design, which incentivized banks to process as many loans as possible, as the loans were federally guaranteed, and banks earned a 1% interest fee for processing them. As a result, many loans went to larger and more stable businesses that were not at immediate risk of closure (Li & Strahan, 2020).

The program’s reliance on banks to disburse funds, coupled with the absence of robust administrative infrastructure, contributed to this inefficiency. The lack of targeted funding meant that businesses that were relatively unaffected by the pandemic could still qualify for loans, while many struggling small businesses faced challenges in accessing the funds quickly (Autor et al., 2022). Furthermore, studies by the Bureau of Economic Analysis (BEA) suggest that the PPP ultimately functioned as a government subsidy that shifted the financial burden of operating expenses—such as payroll, rent, and utilities—from businesses to the federal government (Bureau of Economic Analysis, 2021).

Comparative analyses also suggest that alternative approaches used by other countries may have been more efficient. For example, Giupponi et al. (2022) compared the PPP to direct wage subsidies implemented by European countries. These subsidies, which were paid directly to workers rather than funneled through businesses, proved to be a more cost-effective way to preserve jobs and avoid administrative complexities.

Despite these critiques, the PPP succeeded in preventing mass layoffs, especially in the early months of the pandemic. The program provided critical liquidity to businesses that were facing unprecedented uncertainty and helped to stabilize the labor market during a period of extreme volatility. However, the trade-off was that a substantial portion of the funds went to businesses that did not urgently need them, and many businesses that did not meet loan forgiveness requirements were allowed to convert their loans into grants, further straining the program’s financial efficiency (Li & Strahan, 2020).

**Figure 3***Labor Market Policy Responses to Recessions the United States and Europe*

*A graph of the united states

Description automatically generated*

***Note.*** *Unemployment insurance and short-time work take-up* (Giupponi et al., 2022)

## Fraud in Government Programs

Fraud within government programs is a longstanding challenge that has been exacerbated by the scale and urgency of relief efforts, such as those seen during the COVID-19 pandemic. The PPP, while well-intentioned, became a prime target for fraud due to its size, speed of implementation, and broad eligibility criteria. Fraudsters exploited the program’s vulnerabilities by submitting false applications, inflating payrolls, and in some cases, creating fictitious businesses to qualify for loans (USSBA OIG, 2023).

The issue of fraud in government programs is not unique to the PPP. Historically, government programs—especially those disbursing large sums of money—have been prone to fraud. According to Kadens (2023), fraud has plagued public administration for centuries. For instance, during the Great Depression, programs like the New Deal also faced widespread fraud as individuals and organizations sought to capitalize on federal assistance without meeting eligibility requirements. These historical examples illustrate that government efforts to distribute relief quickly often result in trade-offs between oversight and efficiency.

The rapid implementation of the PPP created similar vulnerabilities. The program disbursed loans through thousands of private banks, which were incentivized to process loans quickly due to the federal loan guarantees. This decentralized approach meant that many fraudulent applications went unnoticed, as banks did not conduct thorough checks on the legitimacy of the applicants (Li & Strahan, 2020). Furthermore, the urgent need to distribute funds left little time to develop robust anti-fraud measures, resulting in significant amounts of taxpayer money being lost to fraud.

### Macro-Level Governance and Regulatory Challenges

At the macro level, regulatory frameworks often struggle to keep up with the complexities of large-scale government programs, especially during crises. The rapid distribution of PPP funds created challenges for federal regulators, such as the U.S. Treasury and SBA, both of which were tasked with overseeing the program (Department Of Treasury, 2024). A key issue was the lack of existing infrastructure to monitor and verify the legitimacy of millions of loan applications in real time.

Efforts to combat fraud have varied across different countries, with some nations taking more proactive approaches to prevent abuse. Bozza (2024) highlights how the U.K. employed stricter regulatory measures to recover fraudulent pandemic loans, relying more heavily on centralized investigations and automated flagging systems to identify suspicious applications. By contrast, the U.S. approach relied more on whistleblowers and retroactive investigations, which have been less effective in preventing fraud upfront.

### Meso-Level Organizational Controls

At the organizational level, the effectiveness of fraud prevention often depends on the internal controls and governance structures within the agencies administering the programs. In the case of the PPP, banks and financial institutions were responsible for distributing the loans, but they often lacked the incentive to rigorously vet applications, given the federal loan guarantees (Li & Strahan, 2020). Ali et al. (2021) argue that robust internal controls are essential to reducing the risk of fraud, particularly in public sector programs. However, the complexity and scale of the PPP overwhelmed many of these controls, allowing fraud to proliferate.

Organizational dynamics also play a role in how fraud is managed internally. For instance, Harrington & Leslie (2023) highlight how organizations with weak internal governance and poor fraud prevention measures are more likely to fall victim to sophisticated fraud schemes. This insight is particularly relevant to large government programs where decentralized structures, such as the PPP’s reliance on private lenders, complicate the enforcement of anti-fraud policies.

In the context of the PPP, operational constraints presented additional challenges as organizations had to process an unprecedented number of loans swiftly. This environment created a strain on organizational controls, as traditional fraud detection mechanisms could not adapt to the urgency of loan disbursement and were not equipped to handle the unique data profile of PPP transactions. Studies indicate that machine learning, especially semi-supervised models like classification through clustering, can augment organizational controls by detecting subtle anomalies that may otherwise go unnoticed in high-volume datasets. This is essential for programs like the PPP, where legitimate applications vastly outnumber fraudulent ones, creating an imbalance that complicates traditional auditing approaches (Gui et al., 2024; Miller & Bertozzi, 2024).

**Figure 4***PPP Loan Application Process Flowchart*

A diagram of a computer system

Description automatically generated

***Note.*** *Source:* (USSBA, 2021)

### Comparative Insights from Government Programs

Lessons from other government programs, such as Medicaid and unemployment insurance, further illustrate the importance of adaptive fraud detection strategies. Medicaid, for example, has been an ongoing target for fraud due to its extensive, complex data ecosystem and the variety of service providers involved. The program’s reliance on claims-based transactions necessitates sophisticated fraud detection algorithms capable of distinguishing between legitimate claims and fraudulent patterns. Similarly, unemployment insurance programs have faced elevated fraud risks in recent years, particularly as pandemic relief benefits introduced expanded eligibility criteria. These programs highlight the need for fraud detection models that combine automated anomaly detection with human expertise to provide a layered defense against fraud (Emilio Ferrara, 2023; Zhao et al., 2024).

Studies suggest that for public sector fraud detection to be effective, it must integrate both macro-level policy compliance and meso-level operational controls within a flexible, technology-enabled framework. By implementing classification through clustering, this study aims to bridge these levels, providing a model that can adapt to the unique regulatory and operational demands of the PPP. The use of advanced machine learning techniques not only enhances fraud detection accuracy but also allows the program to meet transparency and accountability standards essential in public sector applications.

## Machine Learning for Fraud Detection

Fraud detection in government programs like the PPP presents significant challenges due to the sheer volume of data, complexity of transactions, and constantly evolving fraud strategies. Traditional auditing and detection methods often fall short when dealing with large, high-dimensional datasets, which makes ML an ideal solution. ML models can detect hidden patterns, continuously adapt to new fraud techniques, and process massive amounts of data in real time, helping detect fraud more efficiently.

The PPP disbursed over $934 billion in loans, creating an environment ripe for potential fraud due to the rapid distribution and lack of oversight. This makes it essential to employ advanced data-driven techniques to analyze the millions of loan applications submitted to the program.

### Supervised Learning

Supervised learning models are highly effective when labeled data is available, such as pre-identified fraudulent and non-fraudulent transactions. The model is trained on this labeled data, learning patterns, and features associated with fraud. After training, the model can classify new data points by predicting whether they are fraudulent or not based on these learned patterns.

**Logistic regression** is a simple yet widely used supervised learning algorithm for fraud detection. It operates by estimating the probability that a particular loan application is fraudulent, based on input features such as the size of the loan, payroll data, and the business’s industry. Logistic regression is particularly effective when the relationship between variables is linear (Sarker, 2021).

* **Advantages**: Logistic regression is highly interpretable, meaning that the model’s predictions can be understood by non-technical stakeholders such as auditors and policymakers. It allows for easy identification of which features contribute most to predicting fraud.
* **Limitations**: The model struggles with non-linear relationships and may underperform in complex scenarios, such as detecting sophisticated fraud schemes where features interact in a more intricate manner.

**Figure 5**   
*Logistic Regression Response Function*

A graph of a function

Description automatically generated

***Note.*** *Source:* (Maalouf, 2011)

**Random forest** is an ensemble learning method that builds multiple decision trees during training and averages their outputs to make a final classification. This approach is highly robust and effective in handling both categorical and continuous variables, making it ideal for detecting complex patterns of fraudulent behavior in PPP loan application (Itri et al., 2019).

* **Advantages**: Random forest handles large datasets with many variables and is resistant to overfitting. It can manage unbalanced datasets, which is often the case in fraud detection, where non-fraudulent cases vastly outnumber fraudulent ones.
* **Limitations**: Despite its robustness, random forest models are less interpretable than simpler models like logistic regression, making it harder for human auditors to understand the decision-making process.

**Figure 6***Graphic Representation of a Random Forest Decision Tree*

A diagram of a tree

Description automatically generated

***Note.*** *Source:* (Schonlau & Zou, 2020)

**Support Vector Machines (SVMs)** is another powerful supervised learning algorithm used in fraud detection. SVMs work by finding a hyperplane that best separates fraudulent transactions from non-fraudulent ones in a high-dimensional space. SVMs are especially useful when the data is non-linearly separable, which is common in fraud detection scenarios where fraudsters employ sophisticated methods to mimic legitimate transactions (López et al., 2012).

* **Advantages**: SVMs are effective in high-dimensional spaces and can model complex, non-linear relationships between variables. They are well-suited for identifying subtle patterns in large datasets.
* **Limitations**: SVMs can be computationally intensive, especially when applied to very large datasets, and require careful tuning of parameters to achieve optimal performance.

**Figure 7***SVM Hyperplane*

A diagram of a graph

Description automatically generated

***Note.*** *Source:* (Cervantes et al., 2020)

### Unsupervised Learning and Clustering Techniques

Unsupervised learning is critical when labeled data is sparse or unavailable. In fraud detection, unsupervised learning models are designed to identify hidden structures in the data, grouping similar data points together into clusters and flagging outliers or anomalies that may represent fraudulent activity. This is particularly useful in the PPP, where labeled fraud data is limited but suspicious patterns can still be uncovered by analyzing the inherent structure of the data.

**K-Means Clustering** is one of the most common unsupervised learning algorithms used for fraud detection. It works by grouping data points into clusters based on their similarity. For the PPP, businesses with similar characteristics—such as loan amount, industry, or number of employees—would be clustered together. Outliers within these clusters, such as businesses with inflated payrolls relative to their peers, can be flagged as potential fraud cases (López et al., 2012).

* **Advantages**: K-Means is computationally efficient and scalable, making it suitable for large datasets like the PPP. It is also easy to implement and interpret.
* **Limitations**: K-Means requires the number of clusters to be specified in advance, which can be challenging if the true structure of the data is unknown. Additionally, it is sensitive to outliers, which may skew the clustering results.

**Figure 8***Illustration of K-Means Clustering Method*

*A comparison of different colored dots

Description automatically generated with medium confidence*

***Note.*** *Source:* (Rodriguez et al., 2019)

**DBSCAN** is another popular unsupervised learning algorithm that is particularly effective at detecting fraud in large datasets. Unlike K-Means, DBSCAN does not require the number of clusters to be specified beforehand. Instead, it detects clusters based on the density of data points, identifying areas of high density (potential legitimate transactions) and separating out noise or anomalies (potential fraud) (Carcillo et al., 2021).

* **Advantages**: DBSCAN is highly effective at identifying clusters of irregular or unusual behavior and can detect smaller clusters of fraudulent activity within larger datasets. It also handles noise in the data more effectively than K-Means.
* **Limitations**: DBSCAN may struggle to identify clusters in data with varying density, and it can be sensitive to the choice of parameters for detecting outliers.

**Figure 9***Graphic Depiction of DBSCAN Clustering*

A diagram of a circle with arrows and circles

Description automatically generated

***Note.*** *Source:* (Zhang et al., 2022)

**Hierarchical clustering** creates a tree-like structure of nested clusters known as a dendrogram, providing multiple levels of grouping within the data. This technique is particularly useful for detecting fraud at different levels of granularity, helping to uncover not only individual outliers but also broader patterns of suspicious behavior across various subsets of PPP loan applicants (Zhao et al., 2024).

* **Advantages**: Hierarchical clustering does not require the number of clusters to be specified in advance, making it more flexible than K-Means. It also allows for a more detailed analysis of the relationships between clusters.
* **Limitations**: Hierarchical clustering can be computationally expensive, particularly for very large datasets, and it may not perform well in situations where the true structure of the data is complex and non-hierarchical.

**Figure 10***Hierarchical Clustering Dendrogram*

A diagram of a tree

Description automatically generated

***Note.*** *Source:* (Halkidi, 2009)

### Semi-Supervised Learning

Semi-supervised learning (SSL) combines elements of both supervised and unsupervised learning, making it especially useful in fraud detection scenarios where a small set of labeled data is available, but most of the data remains unlabeled. This approach enables the model to leverage the limited labeled data to make informed predictions about the unlabeled data (Gorle & Panigrahi, 2024).

**Graph-based SSL**, as described by Miller & Bertozzi (2024), leverages graph Laplacian spectral truncation to model complex relationships within data. By representing data points (e.g., loan applicants or transactions) as nodes and their connections as edges, this approach captures relational patterns that traditional clustering methods may overlook. This is particularly effective for datasets like the PPP where interconnected entities naturally form network structures.

* **Advantages**:
  + **Relational Insights**: By encoding both local and global relationships, graph-based methods can identify nuanced patterns of fraud. For example, loans disbursed through the same lender or within the same geographic region often share behavioral correlations, such as repayment anomalies or loan-to-employee ratios.
  + **Scalability**: Spectral truncation techniques optimize graph processing, enabling the model to scale effectively for large datasets, such as the PPP's millions of records.
  + **Label Propagation**: With limited labeled fraud cases, graph-based SSL excels by propagating labels across the graph, generalizing patterns from known fraud instances to unlabeled data points (King et al., 2023).
* **Limitations**
  + **Computational Costs**: Constructing and processing large-scale graphs can be resource-intensive, particularly for high-dimensional datasets.
  + **Sensitivity to Noise**: Graph performance depends on the quality of its edges. Erroneous or noisy connections can lead to misleading patterns and reduced accuracy.
  + **Graph Construction Complexity**: Designing the graph (e.g., choosing similarity metrics or defining edge weights) requires domain expertise and can vary significantly by dataset.

**Pseudo-labeling** is a common semi-supervised learning technique in which the model first labels the unlabeled data based on its predictions from the labeled dataset. These pseudo-labeled data points are then used to retrain the model, gradually improving its accuracy. In the PPP, pseudo-labeling can be applied to flag potentially fraudulent loan applications that share characteristics with known fraudulent cases (Zhao et al., 2024).

1. **Advantages**: Pseudo-labeling allows for continuous improvement of the model’s predictions, even when only a small fraction of the data is labeled. It is particularly useful in fraud detection when the cost of obtaining labeled data is high.
2. **Limitations**: Pseudo-labeling can introduce noise into the model if the initial predictions are incorrect, leading to a self-reinforcing cycle of errors. This can be mitigated through careful validation of the model’s outputs.

**Self-training** is another iterative semi-supervised learning process, where the model is initially trained on the labeled data and then uses its own predictions to label additional data points. With each iteration, the model refines its predictions, becoming more accurate over time (Sarker, 2021).

* **Advantages**: Self-training allows the model to improve with minimal human intervention and is highly scalable, making it suitable for large datasets like the PPP.
* **Limitations**: Similar to pseudo-labeling, self-training runs the risk of propagating errors if the model’s early predictions are inaccurate.

**Active learning** further strengthens SSL by selectively labeling the most informative samples. Zhao et al. (2024) introduced the "Maximizing Expected Model Change" framework, which identifies impactful data points for labeling, reducing computational overhead while increasing model accuracy. These methodologies align with detecting evolving fraud patterns, a significant challenge in government funding oversight.

Finally, pseudo-labeling and self-training techniques can be combined with active learning strategies to achieve iterative improvements, leveraging both structured relationships (e.g., graphs) and prioritized labeling efforts. This hybrid approach offers a robust framework for fraud detection in dynamic environments.

### Evaluation Metrics for Fraud Detection in Supervised Learning Models

Given the imbalanced nature of fraud detection datasets—where fraudulent transactions represent only a small fraction of the total data—it is critical to use appropriate metrics for evaluating the performance of machine learning models. The following metrics are commonly used to assess fraud detection models:

**Accuracy** measures the proportion of correct predictions made by the model. However, in fraud detection, accuracy alone can be misleading due to the imbalance between fraudulent and non-fraudulent cases. For example, a model that predicts all transactions as non-fraudulent may achieve high accuracy but fail to detect actual fraud cases.

**Precision and Recall.** Precision measures the proportion of predicted fraudulent cases that are fraudulent. It is important in scenarios where false positives (incorrectly identifying a transaction as fraudulent) have significant consequences, such as triggering unnecessary investigations. Recall measures the proportion of actual fraudulent cases that the model correctly identifies. It is critical in minimizing false negatives, where the model fails to detect fraud, allowing it to go unchecked (Zhao et al., 2024).

**F1 score** combines precision and recall into a single metric, providing a more balanced evaluation of the model’s performance. It is particularly useful in cases where both false positives and false negatives carry significant costs.

**AUC-ROC curve** evaluates the model’s ability to distinguish between fraudulent and non-fraudulent cases across different decision thresholds. A higher area under the curve (AUC) indicates better model performance. This metric is essential for models that need to be fine-tuned to maximize fraud detection while minimizing false positives.

### Evaluation Metrics for Fraud Detection in Clustering

In unsupervised and semi-supervised learning contexts, evaluating clustering models requires specific metrics due to the absence of labeled ground truth for most data points. Effective clustering evaluation ensures that fraudulent patterns emerge clearly within clusters.

**Internal Validation Metrics**:

* **Silhouette Score**: Assesses the compactness of clusters by comparing intra-cluster similarity with inter-cluster separation. For PPP datasets, the silhouette score can optimize the number of clusters in k-means clustering or verify cluster quality in DBSCAN.
* **Calinski-Harabasz Index**: Quantifies cluster dispersion by analyzing the ratio of inter-cluster separation to intra-cluster cohesion. It is particularly useful for comparing different clustering algorithms in fraud detection, such as hierarchical clustering versus k-means.

**External Validation Metrics**: When a small subset of labeled data is available, external validation metrics provide a benchmark for clustering accuracy:

* **Adjusted Rand Index** evaluates how well clustering results align with labeled data, accounting for chance.
* **Normalized Mutual Information (NMI)** quantifies the agreement between clustering outputs and known labels, ranging from 0 (no agreement) to 1 (perfect agreement).

**Cross-Validation for Clustering Stability**

To ensure clustering robustness, cross-validation techniques are adapted to assess stability:

* **Split-Half Reliability**: Splitting the dataset into halves and applying clustering separately evaluates whether the model produces consistent results across subsets.
* **Bootstrap Sampling**: By creating randomized subsets of the dataset and re-clustering, bootstrap techniques measure consistency in cluster assignments.
* **Cluster Stability Index**: Aggregates cluster consistency metrics across iterations to quantify model reliability, critical for datasets like PPP, where noise or outliers might affect clustering.

**Integration with PPP Fraud Detection**

For the PPP fraud detection:

* **Application**: Metrics like silhouette score and Calinski-Harabasz index guide the iterative refinement of cluster quality, ensuring that fraudulent and legitimate patterns are well-separated before transitioning to binary classification.
* **Scalability**: Cross-validation ensures clustering methods perform reliably on large-scale, high-dimensional PPP datasets, adapting to the program's unique challenges of sparse labeled fraud cases and complex transaction data.

### Conclusion to Machine Learning for Fraud Detection

The combination of supervised, unsupervised, and semi-supervised models offers a comprehensive approach to fraud detection, with each methodology contributing unique strengths for handling high-volume, low-fraud datasets. By focusing on semi-supervised learning, particularly classification through clustering, this study aims to develop an adaptable model that meets the PPP’s unique operational demands while ensuring robust fraud detection.

## Summary

This literature review explored key methodologies, theoretical frameworks, and ethical considerations central to effective fraud detection in the context of the PPP. The review focused on classification through clustering as a primary method for managing the PPP’s imbalanced data structure, where fraudulent cases are vastly outnumbered by legitimate ones. Semi-supervised techniques, particularly those integrating clustering and dimensionality reduction through PCA, provide a scalable approach to identifying subtle anomalies (Gui et al., 2024; López et al., 2012). These methods are bolstered by behavioral frameworks, including the Fraud Triangle and its expanded forms, which allow for the interpretation of anomalies in relation to known fraud motivations, such as opportunity and financial pressure (Bozza, 2024; King et al., 2023).

Ethical and regulatory considerations highlighted the need for transparent, privacy-compliant models, especially in government applications where transparency supports accountability. Adherence to standards like GDPR and CCPA ensures that fraud detection technologies align with public expectations for privacy and ethical AI, underscoring the importance of explainable XAI methods in maintaining compliance (Emilio Ferrara, 2023; Koreff et al., 2023).

### Current State of the Literature

Fraud detection research has primarily focused on supervised and unsupervised learning methods due to their maturity and effectiveness in various domains:

* **Supervised Learning**: Widely applied in financial fraud detection, supervised models such as decision trees and SVMs achieve high precision and recall when sufficient labeled data is available. For instance, Bauder & Khoshgoftaar (2017) examined supervised classifiers for Medicare fraud detection, demonstrating the importance of feature engineering in achieving robust classification results​.
* **Unsupervised Learning**: Techniques like clustering and anomaly detection excel in identifying outliers or unusual patterns without relying on labeled data. Carcillo et al. (2021) demonstrated how clustering methods can effectively detect anomalies within loan application datasets, offering valuable insights in cases where labeled data is unavailable​.
* **Semi-Supervised Learning (SSL)**: SSL, which uses a small set of labeled data alongside a large volume of unlabeled data, is gaining traction in scenarios with limited labeled datasets. Xu et al. (2022) reported the superiority of SSL techniques over traditional supervised methods, particularly in domains where labeled data is scarce​.

### Unmet Needs and Gaps

Despite these advancements, several critical gaps persist in the literature:

* **Limited Research on Government Programs**: The majority of studies focus on private-sector fraud detection (e.g., credit card fraud), with limited exploration of government-specific domains like the PPP. Research on government subsidy fraud, such as the E-Rate program, highlights the need for tailored fraud detection frameworks (USGAO, 2020).
* **Challenges with Imbalanced Datasets**: Fraudulent cases in government programs often constitute a small minority, leading to significant imbalances in datasets. Techniques such as Synthetic Minority Oversampling Technique (SMOTE) have been explored (Benala & Tantati, 2022), but their specific application to PPP datasets remains underdeveloped​​.
* **Adapting to Evolving Fraud Schemes**: Government fraud schemes are dynamic, exploiting new loopholes as regulations change. Traditional supervised methods relying on static labeled datasets struggle to generalize to these novel patterns, as noted by Larson (2020) in discussions of imbalanced learning challenges​.
* **Underexplored Integration of SSL and Imbalance Compensation**: While SSL is increasingly utilized, most studies fail to integrate advanced imbalance compensation techniques such as weighted loss functions or domain-specific oversampling strategies with SSL models. This is particularly crucial for enhancing the detection of minority classes in high-dimensional datasets​​.

### Novelty of Proposed Methodology

This research addresses these gaps through:

1. **Domain-Specific Focus**: By concentrating on PPP fraud detection, this study expands the literature on government subsidy fraud. Prior research has primarily emphasized healthcare (e.g., Medicare) or financial fraud (Bauder & Khoshgoftaar, 2017), leaving government loan programs underrepresented​​.
2. **Semi-Supervised Learning with Imbalance Compensation**: Integrating SSL with imbalance compensation techniques such as SMOTE or cost-sensitive learning is a novel approach, particularly for government datasets. Xu et al. (2022) and Benala & Tantati (2022) highlight the need for this integration to address fraud detection challenges in imbalanced datasets​​.
3. **Dynamic Fraud Pattern Detection**: By leveraging unlabeled data, SSL adapts to emerging fraud patterns, overcoming the limitations of static supervised models. This adaptability is vital for detecting novel PPP fraud schemes​​.

**4. Relevance to PPP Fraud Detection**

The PPP’s unique challenges, including high-dimensional datasets, sparse labeled fraud cases, and evolving fraud schemes, necessitate innovative approaches. SSL, combined with advanced imbalance handling, offers a scalable and adaptable solution, addressing critical gaps in current fraud detection systems while aligning with government program requirements​​.

### Future Directions and Implications for Government Programs

Given these gaps, several future research directions could enhance fraud detection efficacy in government programs:

* **Advanced Clustering Algorithms for Adaptive Fraud Detection**: Developing clustering methods that adapt in real-time to shifts in transactional patterns could significantly improve the model's ability to detect emerging fraud schemes. These adaptive algorithms would be especially beneficial in high-stakes, dynamic contexts like government aid distribution.
* **Expanded Fraud Theoretical Frameworks**: Future research could extend the Fraud Triangle by adding context-specific factors—such as external pressures or changing regulatory landscapes—that are relevant to fraud in emergency relief programs.
* **Hybrid Models to Address Ethical Compliance**: Hybrid models combining rule-based and data-driven techniques may offer a pathway to enhanced interpretability without sacrificing accuracy. This approach could be particularly valuable in public sector applications, where ethical compliance is paramount.

By advancing fraud detection methodologies through these directions, future research can support the development of resilient, transparent, and adaptable systems that protect government resources. For the PPP and similar programs, these improvements promise more secure fraud prevention while maintaining the accessibility and trust that public programs require. The methods and frameworks synthesized in this study lay the groundwork for the semi-supervised approach outlined in Chapter 3, focusing on scalable fraud detection that aligns with the operational and ethical requirements of modern government initiatives.

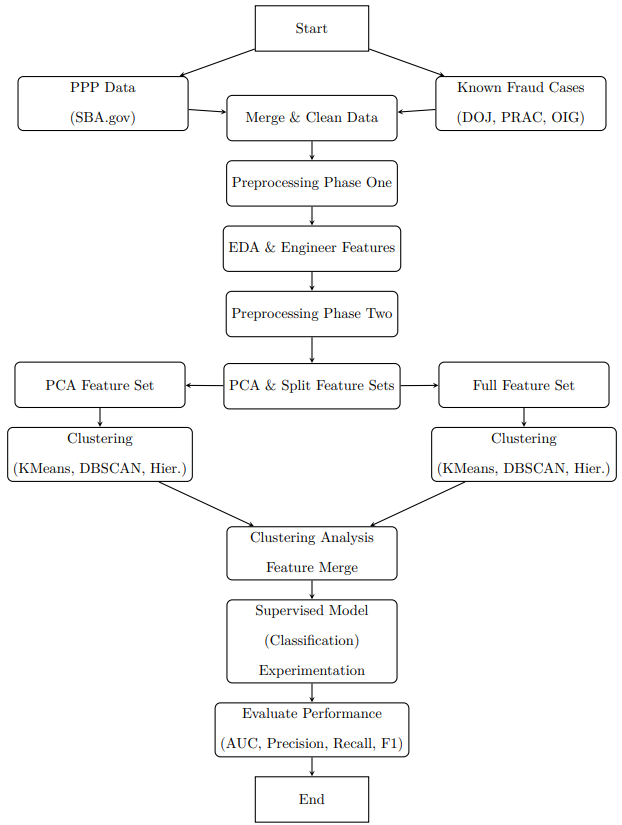
# Chapter 3: Research Method

The problem addressed in this study is the lack of intelligent unsupervised fraud identification in the government domain, specifically in the Paycheck Protection Program (PPP). The purpose of this quantitative experimental study was to identify features and methodologies develop intelligent semi-supervised fraud identification methodologies in the government domain, specifically the PPP. This chapter outlines the research methodology and design of this study. It includes an overview of the population, instrumentation, data collection, and definitions of variables. It then describes the study procedure, the data analytics performed as part pf hypothesis testing, assumptions, limitations, delimitations, and ethical assurances.

## Research Methodology and Design Process Diagram

The methodological workflow used in this study is summarized in Figure 11. It outlines the major phases of the research, beginning with data collection, preprocessing, and exploratory analysis, followed by feature set partitioning and clustering using multiple unsupervised algorithms. Cluster outputs were merged back into the dataset and served as inputs to a series of semi-supervised experiments using various supervised models. Each stage in the process was aligned with the study’s hypotheses, culminating in model evaluation based on precision, recall, F1-score, and AUC-ROC.

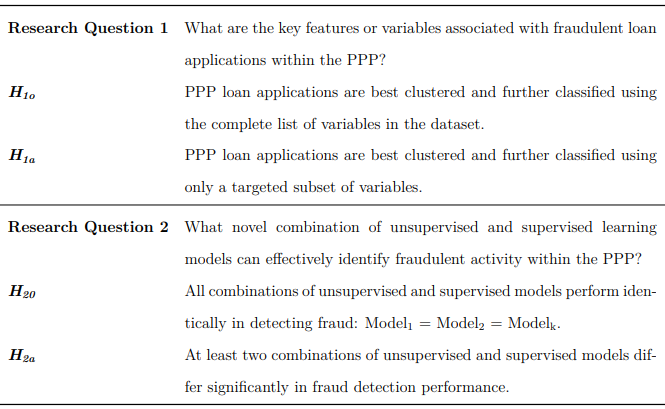
**Figure 11**   
*Research Methodology and Design Process Diagram*



## Research Methodology and Design (Nature of the Study)

This study employed a quantitative experimental design that implemented a Classification through Clustering framework: a hybrid methodology integrating both unsupervised clustering and semi-supervised classification techniques. The design follows a systematic, hypothesis-driven structure, ensuring that each phase directly addressed the research questions and hypotheses shown in the table below:

**Table 1***Research Question and Hypotheses Alignment*



This design was appropriate due to the complexity of the PPP dataset, the limited number of labeled fraud cases, and the need to detect hidden or novel fraud patterns in unlabeled data. Classification through Clustering enabled the discovery of anomalies using unsupervised methods and the refinement of those findings through supervised classification Alternative approaches, such as solely supervised learning, were considered; however, such methods would have required comprehensive labeling, which is challenging in large datasets with limited labeled fraud instances. The chosen multi-phase approach allows for an adaptive, hybrid model that effectively captures anomalies, clusters data, and improves detection accuracy.

### Alternative Methodologies Considered

Three alternative approaches were considered and determined to be less appropriate for the problem context:

**1. Supervised Learning Only:** This approach was dismissed due to the scarcity of labeled fraud cases, the risk of model overfitting, and the difficulty of generalizing predictions across an imbalanced dataset.

**2. Unsupervised Learning Only:** While effective for anomaly detection, unsupervised clustering alone lacked refinement and suffered from high false-positive rates due to the absence of verified fraud labels.

**3. Manual Fraud Detection:** Manual review of loan records was infeasible due to the dataset’s size and the need for scalable, automated methods.

### Justification for the Selected Approach

The Classification through Clustering framework offered the flexibility to process high-dimensional data, incorporate both labeled and unlabeled examples, and adapt to the class imbalance inherent in fraud detection. It also enabled comparative analysis of multiple model pairings, consistent with the study’s hypothesis-driven structure. By combining anomaly detection with confirmed fraud cases, this methodology improved detection precision while preserving scalability and generalizability—critical requirements for analyzing a national-scale financial relief program.

## Population and Sample

The population for this study consisted of all businesses and entities that received loans through the PPP. The publicly available loan-level dataset released by the SBA included more than 11 million records, detailing loan characteristics, borrower location, and business classification.

However, for this study, the scope was limited to PPP loans exceeding $150,000, which represented a subset of 968,525 loans. This threshold was selected based on the elevated financial risk associated with larger loans and the greater likelihood of such loans being subject to federal investigation, which increased the availability of reliable fraud labels. Analyzing this high-value subset supported the study’s objective to identify scalable fraud detection methods capable of addressing the most impactful cases. Additionally, this subset was published by the SBA as a single CSV file, which made the data more accessible and enabled greater reproducibility of the research process across environments and systems with limited local computing power.

**Rationale for Using the Entire Population**

Using the entire population of large PPP loans allowed for greater analytical depth and accuracy in detecting fraud patterns. Larger loans are more likely to reflect complex fraud schemes and provide richer feature sets for modeling. The availability of verifiable fraud cases also improved the quality of supervised learning inputs.

Additionally, the SBA published loans exceeding $150,000 as a single, consolidated CSV file, making this high-value subset easily accessible and highly reproducible for public analysis. In contrast, the smaller loan records, though available, were split across multiple files and would have required processing over 5 GB of raw data. This would have introduced significant computational and storage overhead. Restricting the scope to the over $150K segment allowed for efficient data handling, consistent replication, and alignment with the study’s objective to identify scalable fraud detection methodologies for the most financially significant cases.

## Materials and Instrumentation

This study utilized archived public datasets, a manually labeled fraud vector, custom-developed Python scripts, and open-source machine learning libraries to build and evaluate a hybrid fraud detection pipeline. Instrumentation included both the digital tools used to acquire and process the data and the computational environment that enabled scalable modeling and evaluation.

### ****Data Sources****

**PPP Loans**: The core dataset was obtained from the SBA, which published loan-level details for all approved PPP loans. The subset used in this study consisted of loans exceeding $150,000, comprising 968,525 records and 52 variables, distributed in a single, consolidated CSV file. This format supported straightforward ingestion, consistent replication, and reproducible access to structured loan data including borrower entity name, loan amount, geographic location, NAICS codes, reported jobs, and loan forgiveness status.

**Labeled Fraud Cases**: To create the fraud-labeled subset required for supervised and semi-supervised learning, a custom web scraper was written in Python using BeautifulSoup. This tool collected approximately 2,500 press releases and case summaries from PandemicOversight.gov, which aggregates fraud reports from the DOJ, SBA OIG, and related oversight bodies. Although PRAC maintained a centralized news archive, automated scraping was necessary to extract all cases programmatically. Manual review was then conducted to filter for prosecuted PPP fraud cases, resulting in 301 confirmed fraudulent applications. These cases were matched to SBA loan records using fuzzy logic on borrower name, loan amount, and business location. The final binary label vector was stored and integrated with the broader modeling dataset.

### ****Software and Computational Environment****

**Platform**: All modeling and data engineering tasks were performed in Google Colab Pro, which provided access to both high-RAM CPU instances and A100 GPU accelerators. Exploratory data analysis (EDA) and extract-transform-load (ETL) tasks were conducted in notebook-based workflows, while unsupervised and supervised modeling was scripted in modular Python files to ensure reproducibility. Full programming scripts are provided in a public GitHub repository at https://github.com/sappw1/Dissertation under the MIT License

**Key Libraries and Frameworks**:

* pandas, numpy: For tabular data transformation and feature engineering
* cuml: For GPU-accelerated PCA, K-Means, and DBSCAN implementations
* scikit-learn: For logistic regression, cluster evaluation metrics (e.g., Davies-Bouldin Index), and model validation
* xgboost: For gradient boosting classification on structured data
* keras (with TensorFlow backend): For neural network construction and training
* matplotlib, seaborn: For visualizations and diagnostic plots
* BeautifulSoup: For web scraping PRAC case reports

### Operational Definitions of Variables

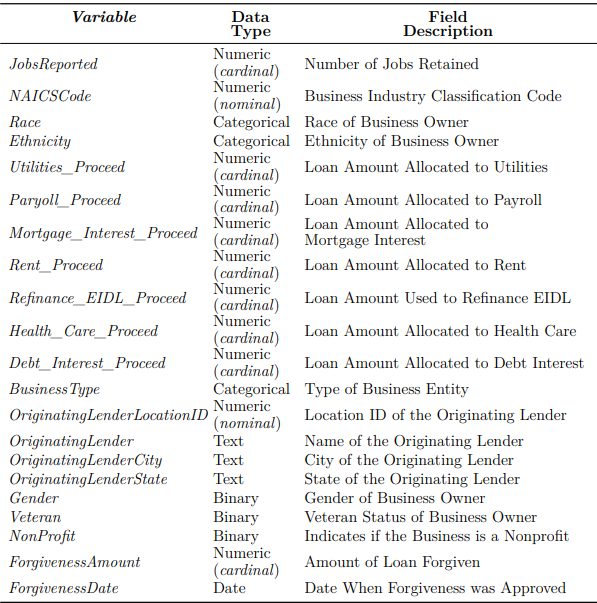
This study used two types of variables: raw features derived from the PPP dataset, which served as inputs to PCA and machine learning models, and experimental variables created to test the study’s hypotheses. The operational definitions below distinguish between these categories and describe how each was used in the modeling framework.

**Original PPP Dataset Features:** The table below lists the primary variables extracted from the SBA’s PPP dataset and used as input features in the modeling process. These include financial indicators, categorical attributes, and derived features engineered to capture anomalous patterns in borrower behavior.

**Table 2***Original Variables for The PPP Dataset*



**Table 2 – continued from the previous page**



Additionally, derived features were created through feature engineering, including:

* *Loan-per-Job* Ratio: Calculated as *LoanAmount* / *JobsReported*, used to detect disproportionately large loan requests.
* *ForgivenessRatio*: Calculated as *ForgivenessAmount* / *CurrentApprovalAmount*, used to evaluate repayment behavior.

**Experimental Variables for Hypotheses Testing:** The table below presents the key variables defined for hypotheses testing within this study. These include experimental configurations such as feature sets and model pairings, as well as engineered features derived from clustering outputs. Each variable played a direct role in evaluating the study’s hypotheses regarding the impact of feature selection H1 and the effectiveness of various clustering-supervised model combinations H2.

**Table 3***Experimental Variables and Hypotheses Alignment*

|  |  |  |
| --- | --- | --- |
| **Variable** | **Description** | **Hypothesis** |
| Feature Configuration: Full Set | All preprocessed PPP fields used in model input | H1, H2 |
| Feature Configuration: PCA Subset | PCA derived subset | H1, H2 |
| Model Configuration | Paring of unsupervised & supervised models (e.g., K-Means + Logistic Regression | H2 |
| Cluster Feature | Encoded label derived from cluster result | H2 |
| *is\_fraudulent* | Binary indicator for fraud | H1, H2 |

## Study Procedures

This section outlines the procedures followed to collect, preprocess, and analyze data for the identification of fraudulent PPP loan applications. The study followed a multi-phase experimental design, incorporating both unsupervised and semi-supervised learning approaches, supported by carefully curated labeled data and scalable cloud-based infrastructure.

### Data Collection

The primary dataset was acquired from the SBA, which publicly released detailed records of approved PPP loans. This study focused on loans exceeding $150,000, comprising a subset of 968,525 records. The decision to restrict the scope to this high-value segment was based on three factors: elevated financial risk associated with larger loans, increased likelihood of investigative scrutiny and prosecution, which facilitated the identification of confirmed fraud cases, and the availability of the full over $150K dataset as a single CSV file, enhancing reproducibility and computational feasibility.

To develop a labeled dataset of fraudulent loans, a custom web scraper was built using Python’s BeautifulSoup library to extract approximately 2,500 COVID-19 relief-related articles and press releases from PandemicOversight.gov, the portal maintained by the PRAC. Each report was manually reviewed to identify instances of confirmed PPP fraud. A total of 301 unique fraudulent loan applications were matched from the primary dataset based on borrower name, loan amount, and geographic location. These cases were encoded in a binary variable *is\_fraudulent*.

### Data Preprocessing and Exploratory Analysis

Preprocessing and exploratory data analysis (EDA) were conducted in two phases to ensure the dataset was privacy-compliant, interpretable, and analytically robust for downstream modeling.

**Initial Preprocessing and EDA:** Prior to formal analysis, the dataset underwent basic transformations necessary for secure handling and visual exploration. These included hashing of PII fields (e.g., *BorrowerName*, *BorrowerAddress*, *FranchiseName*) to preserve record-level uniqueness while PII, as well as preliminary feature engineering. These transformations enabled a structured exploratory analysis. EDA techniques included univariate analysis, boxplots to detect distributional skew or outliers, and correlation matrices to identify feature redundancy and key variables. The analysis from this phase identified the reduced key feature subset which was analyzed alongside the full feature set during PCA.

**Post EDA Preprocessing and PCA:** Following this initial phase, a second round of preprocessing was conducted to prepare the dataset for dimensionality reduction and machine learning. This stage included z-score standardization of continuous variables to support distance-based clustering algorithms, as well as the encoding of categorical variables through a combination of frequency, label, and one-hot encoding methods based on feature cardinality. Missing values were addressed without imputation to avoid masking potential fraud-related patterns; categorical nulls were recoded into a “Missing” category, while missing numeric values were flagged using binary indicators. Temporal fields were further decomposed to extract discrete components such as month and day. PCA was then applied to both the full feature set and the key feature subset, with the full set achieving 95% explained variance using just three components. In contrast, the key feature subset reduced to a single principle component and was therefore discarded as it would likely introduce high levels of oversimplification during clustering. The final dataset for clustering included both PCA-reduced (3 components) and Full feature (52 variable) matrices for model comparison and evaluation.

### Clustering Methodology

Unsupervised learning was conducted to detect structure within the PPP loan dataset prior to fraud classification. Clustering was performed on two versions of the preprocessed feature matrix: a full feature set without dimensionality reduction (*X\_all\_no\_pca*), and a PCA-reduced version (*X\_all\_pca\_3*) that retained 95% of the total variance across three components. Both versions were subjected to structured hyperparameter exploration, with parameter grids explicitly defined in external JSON configuration files. These configurations enabled systematic sweeps across key clustering parameters, such as the number of clusters for K-Means and Hierarchical models, and the ε and min\_samples values for DBSCAN, allowing reproducible experimentation across the full search space. While not conducted using a formal hyperparameter search utility, this programmatic approach functionally mirrored a grid search by executing all defined combinations sequentially in a controlled manner.

Three clustering algorithms were applied to both the non-reduced and PCA-reduced feature spaces: K-Means, Agglomerative Hierarchical Clustering, and DBSCAN. K-Means was evaluated across a range of two to ten clusters to allow for the detection of both broad fraud/non-fraud distinctions and finer substructures, while Hierarchical clustering was tested over a narrower range of two to six clusters to accommodate its higher computational demands and the expectation of fewer natural groupings. DBSCAN was assessed across multiple combinations of epsilon (ε) (0.3 to 1.3) and min\_samples (3 to 15), allowing sensitivity to both compact and diffuse outliers. The final selected DBSCAN configuration (ε = 0.7, min\_samples = 10) was chosen based on qualitative alignment with known fraud cases and favorable cluster validation scores. All algorithms were executed on GPU using RAPIDS cuML to enable scalable experimentation across high-dimensional feature spaces. Each clustering result was saved along with its associated hyperparameters and evaluation metrics to facilitate downstream model integration and reproducibility.

Clustering performance was evaluated using silhouette score and Davies-Bouldin Index (DBI), computed via cuML and Scikit-learn, respectively. Across all configurations, the non-PCA version of the dataset (*X\_all\_no\_pca*) consistently produced higher silhouette and lower DBI scores, indicating better cluster cohesion and separation. Clustering labels and DBSCAN noise indices generated from both the PCA and full feature set were retained for integration into the classification pipeline.

Clustering-derived features were extracted from each cluster configuration result. For example, *K-Means\_full\_n2*, *hier\_full\_n2*, and *dbscan\_pca\_e07\_m10\_noise* indicate which cluster label was assigned to a record or whether the record was captured as noise in their respective cluster configurations. Given 64 distinct cluster model configurations across the three clustering models, this resulted in 128 unique features (including both PCA and full feature set derived cluster features). These variables were appended to the full training dataset for the supervised modelling phase.

### Classification (Supervised) Methodology

Following the unsupervised clustering phase, a supervised learning framework was developed to evaluate the impact of clustering-derived features on fraud detection accuracy. Fraud labels (*n* = 301) obtained from matched cases in PRAC, SBA OIG, and DOJ data were reintegrated into the modeling dataset after clustering. This label set was used to evaluate model performance against the PCA and full feature sets both with and without the inclusion of unsupervised outputs, forming the core of the semi-supervised analysis.

Six classification algorithms were implemented to evaluate fraud detection performance across a range of model complexities and learning paradigms. These included logistic regression, support vector machines (SVM), random forest, XGBoost, Gaussian naïve Bayes, and a feedforward neural network constructed using Keras and TensorFlow. Table 4 summarizes the libraries used and whether GPU acceleration was employed.

**Table 4***Supervised Classification Models and Libraries*

A white text with black text

AI-generated content may be incorrect.

Each model was trained using each of the 128 cluster derived features from both the PCA and full feature-set clustering results, as well as a baseline without cluster-features**.** The core clustering outputs: K-Means cluster assignments, Hierarchical clustering labels, and DBSCAN clustering labels & outlier flags, were injected as additional features in the training data. This factorial design supported H2, which posited that certain combinations of unsupervised and supervised models would outperform others in detecting PPP loan fraud. All experiments were conducted using a stratified 70/30 train-test split, repeated across five random seeds (42, 52, 62, 72, and 82) to ensure robustness. To address class imbalance in the labeled fraud data, SMOTE was applied to the training fold in each iteration. This experimental structure yielded a total of 7,710 semi-supervised classification runs across all model and clustering configurations.

## Data Analysis

The analysis strategy was divided into two stages corresponding to the study’s experimental design: (1) evaluation of clustering configurations and (2) evaluation of classification model performance across augmented feature sets. Both stages were implemented in Python using a modular pipeline that aggregated, analyzed, and exported metrics for hypothesis testing.

In the clustering analysis phase, model outputs, including cluster labels, configuration parameters, and fraud label joins, were compiled for each run of K-Means, Agglomerative Hierarchical Clustering, and DBSCAN. Each configuration was evaluated using internal clustering metrics, including silhouette score and DBI, to assess cohesion and separation. In parallel, fraud prevalence was computed within each cluster using a merged fraud label, allowing the identification of fraud-enriched groupings. For DBSCAN, the density-based noise points were evaluated separately, and those configurations that yielded high fraud enrichment in noise (≥10%) were selected for downstream use. Aggregated results were exported in structured tables and plots to support model selection and interpretation.

In the classification analysis phase, evaluation metrics for each supervised model were collected across all combinations of clustering feature augmentation. For each of the 7,680 classification runs, metrics including AUC-ROC, F1 score, precision, recall, and accuracy were computed and aggregated across five random seeds. Results were grouped by model type and clustering condition and summarized using descriptive statistics and boxplots. This output formed the basis for formal hypothesis testing.

To evaluate H1, which posits that targeted subsets of features enable more effective clustering than the full feature set, clustering was conducted using K-Means, Hierarchical, and DBSCAN algorithms across both full and key feature configurations. Each clustering output was assessed using internal validation metrics, including silhouette score and DBI, to quantify cohesion and separation. To evaluate fraud alignment, fraud label overlays were applied to each cluster configuration, and fraud rates were computed per cluster or noise group. Visual comparisons of fraud density and t-SNE projections were used to assess whether the key feature set yielded clearer or more concentrated fraud separation relative to the full feature set.

To evaluate H2, that not all clustering-model combinations perform equally, two-way ANOVA was conducted to identify statistical significance between model & cluster feature combinations, additionally, one-way ANOVA was conducted within each supervised learning model group to isolate statistically significant variance between cluster feature configurations. Assumptions of normality and homogeneity of variances were checked prior to analysis, and effect sizes were calculated for significant differences.

## Assumptions

Several assumptions underpinned the design, execution, and interpretation of this study. Each assumption is grounded in the structure of the PPP dataset, the modeling approach, and the statistical framework used for evaluating outcomes.

* **Accuracy and Validity of Fraud Labels**: It was assumed that the 301 fraud cases manually labeled from DoJ, SBA OIG, and PRAC sources were accurately reported and correctly matched to PPP loan records. These cases were treated as ground truth for supervised learning and model evaluation.
* **Generalizability of Labeled Fraud Patterns**: The study assumed that known fraud cases share characteristics with yet-unlabeled fraudulent loans in the broader dataset. This assumption is necessary for semi-supervised learning to be effective, as it allows the model to extend learned patterns to unseen examples.
* **Independence of Loan Records**: Each loan was treated as an independent observation. While some entities may have submitted multiple applications, no explicit linking was performed across loans, and the modeling assumed no within-entity dependency.
* **Integrity of Feature Transformations**: Preprocessing steps, including PII hashing, categorical encoding, standardization, and PCA, were assumed to preserve the core structure and variance necessary for effective modeling. This assumption justifies the use of transformed data in both clustering and classification.
* **Validity of Classification Metrics**: AUC-ROC, precision, recall, and F1 score were assumed to provide valid measures of fraud detection performance. These metrics were selected for their sensitivity to class imbalance and their widespread use in evaluating binary classifiers in high-risk domains.
* **Suitability of Statistical Tests**: The hypothesis tests used were assumed appropriate for the structure of the aggregated results. Assumptions of normality and sphericity were tested or otherwise addressed through non-parametric alternatives.
* **Stability of Seeded Model Results**: The study assumed that model performance across five stratified random seeds was sufficiently stable to support averaging and statistical inference. This assumption allowed for generalizable conclusions from repeated train-test splits.

These assumptions were necessary to ensure the experimental pipeline could support the research questions and hypotheses. Where possible, assumptions were validated empirically (e.g., via diagnostic plots or effect size estimation), and their implications are addressed in the Limitations section.

## Limitations

Several limitations affected the scope, execution, and generalizability of this study. These limitations were either inherent to the dataset or introduced through methodological constraints required to operationalize the hybrid machine learning framework.

* **Limited Size and Scope of Labeled Fraud Cases:** Although over 960,000 PPP loans exceeding $150,000 were analyzed, only 301 cases were confidently labeled as fraudulent. These were sourced from public enforcement and oversight reports, meaning undetected or unprosecuted fraud cases were not included. This limited ground truth restricted the diversity of known fraud patterns available for model training and evaluation.
* **Potential Bias in Publicly Reported Fraud**: The labeled fraud cases were drawn exclusively from publicly documented sources (e.g., DOJ, SBA OIG, PRAC). These sources may overrepresent more egregious or easily identifiable fraud, introducing reporting bias. As a result, the semi-supervised model may be less sensitive to more subtle or sophisticated fraud strategies not reflected in the training labels.
* **Feature Limitations Due to Data Privacy**: PII was hashed to preserve anonymity, which prevented linkage across related records (e.g., multiple loans to the same entity). This constrained the model's ability to detect coordinated or serial fraud schemes, potentially underestimating fraud prevalence among repeat applicants.
* **Interpretability Constraints from PCA and Encoding**: Dimensionality reduction (via PCA) and categorical encoding transformed original variables into abstract representations. While these transformations improved computational performance and clustering quality, they reduced interpretability, especially in explaining model predictions or attributing risk to specific borrower characteristics.
* **Class Imbalance and Synthetic Oversampling**: The extreme class imbalance between fraudulent and non-fraudulent loans required the use of SMOTE to synthetically generate minority-class examples during model training. While SMOTE is a widely accepted technique, synthetic examples may not perfectly replicate true fraud behaviors, and their inclusion could affect generalizability.
* **Computational Constraints on Full Dataset Inclusion**: This study was limited to loans exceeding $150,000 due to file size and processing constraints. Including all PPP loans would have required more than 5 GB of raw data and significantly greater computational resources. As a result, the findings may not fully extend to lower-value loans, which constitute a substantial portion of the overall PPP distribution.

These limitations do not undermine the core findings of the study, but they do inform its scope and potential applicability. They are addressed through complementary strategies, including semi-supervised learning, robust metric validation, and transparent reporting of preprocessing and modeling choices.

## Delimitations

This study included several deliberate boundaries that were set to ensure feasibility, focus, and alignment with the research questions. These delimitations reflect methodological choices made by the researcher and define the scope of inference for the findings.

* **Loan Size Threshold**: The study was limited to PPP loans exceeding $150,000. This subset of 968,525 records was chosen due to its greater financial impact and the higher likelihood of enforcement activity, which increased the availability and reliability of fraud labels. Smaller loans were excluded to manage computational load and to ensure the dataset remained reproducible using a single, publicly available CSV file from the SBA.
* **Focus on Publicly Labeled Fraud Cases**: Fraud labels were derived exclusively from publicly documented enforcement actions reported by the DOJ, SBA OIG, and PRAC. No proprietary or internal investigative data were used. This approach ensured transparency and reproducibility but excluded unreported or pending fraud cases, limiting the comprehensiveness of the fraud label set.
* **Use of Publicly Available Features Only**: The feature set was restricted to fields provided in the SBA PPP dataset, supplemented only with engineered features derived from those columns. No external commercial datasets or private borrower information were integrated. This constraint maintained ethical compliance and data availability for replication, but it may have excluded relevant behavioral or financial indicators.
* **Experimental Framing of Unsupervised & Supervised Modeling**: The study was explicitly framed as a comparison of hybrid learning configurations, specifically, clustering outputs used as input features for supervised models. Alternative architectures, such as fully unsupervised anomaly detection pipelines or deep learning-based fraud detection systems, were not explored. This design choice enabled a controlled evaluation of specific hypotheses related to feature and model combinations.
* **Temporal Scope of the Dataset**: Only PPP loans issued during the COVID-19 relief period were included, as defined by the SBA data snapshot. Loans originated outside this timeframe or under different federal aid programs (e.g., EIDL) were not considered. This focus ensured contextual consistency but limits generalization to other funding programs or economic conditions.

These boundaries were necessary to maintain coherence between the study’s purpose, research questions, and available data. They align with the Classification through Clustering framework, ensuring that methodological decisions supported the detection of fraud at scale within a high-impact federal relief program.

## Ethical Assurances

This study received approval from the University’s Institutional Review Board (IRB) prior to data analysis. The IRB review confirmed that the research posed no more than minimal risk, as it was based exclusively on secondary, publicly available data with no direct interaction with human participants. A copy of the IRB approval letter is included in Appendix A.

All data used in the study were derived from the SBA’s publicly released PPP dataset and publicly available government records of prosecuted fraud cases (e.g., PRAC, DOJ, SBA OIG). These sources are fully open-access and contain no PII beyond what is already public record.

To further ensure confidentiality, all PII fields included in the PPP dataset were securely hashed during preprocessing. This transformation ensured that individual records could not be re-identified, while preserving unique identifiers necessary for modeling and analysis. No attempt was made to reverse-engineer or expose personally linked data.

All datasets, model outputs, and documentation were stored securely using encrypted, access-controlled cloud storage (Google Drive and Colab Pro environments), in compliance with institutional data protection standards. No data were transferred to third-party platforms outside the research environment.

The researcher’s role was solely analytical and technical. Although the researcher has professional experience in data science and government analytics, no affiliation exists with the SBA or other agencies involved in the administration or oversight of the PPP. To mitigate potential bias, methodological decisions, such as feature selection, model evaluation, and fraud labeling, were made transparently and documented with reproducible code and data transformations. Labeling of fraudulent loans relied exclusively on verified government sources and was not subject to personal interpretation or inference by the researcher.

These ethical safeguards ensured the study adhered to standards for data security, privacy, and research integrity, while maintaining transparency and reproducibility.

## Summary

This chapter detailed the methodology used to investigate fraud within the PPP dataset through a hybrid machine learning framework. The study employed a quantitative, experimental design leveraging unsupervised clustering and semi-supervised classification to detect anomalous loan patterns indicative of fraud. The methodology was aligned with the research questions and hypotheses, and the study design was justified as appropriate given the scarcity of labeled fraud data and the scale of the PPP dataset.

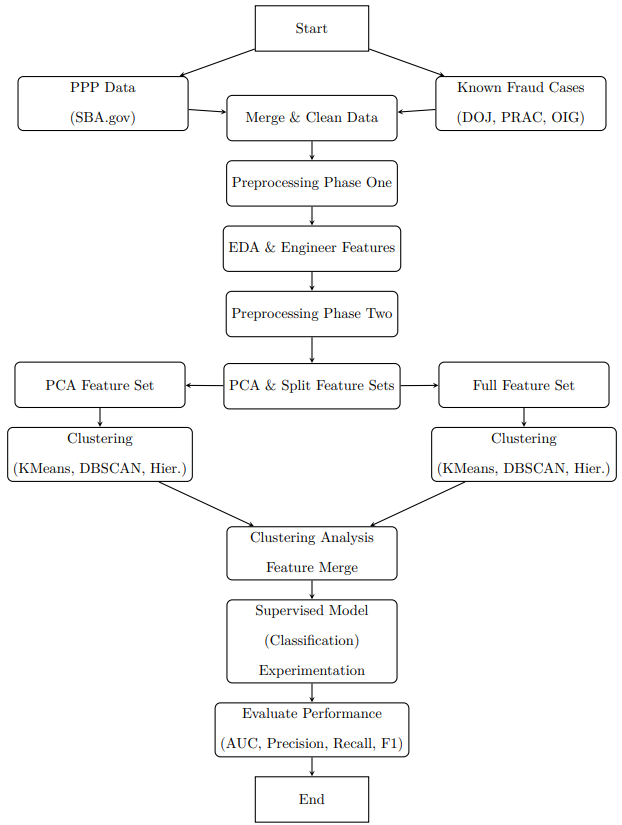
The population included all PPP loans exceeding $150,000, totaling 968,525 records. Fraud labels were curated from verified public sources, and the data underwent extensive preprocessing and exploratory analysis. Two feature configuration, full and reduced, were evaluated alongside features derived from clustering algorithms (K-Means, DBSCAN, Hierarchical) and supervised models (e.g., SVM, Logistic Regression, Random Forest, Neural Network, Naïve Bayes, XGBoost). These configurations served as experimental conditions for evaluating fraud detection performance.

The chapter also described the instrumentation used, data storage procedures, and the structured pipeline for preprocessing, modeling, and hypothesis testing. Assumptions, limitations, and delimitations were acknowledged, and ethical safeguards were outlined, including IRB approval and data privacy protections.

# Chapter 4: Findings

This chapter presents the results of a quantitative experimental study addressing the lack of intelligent unsupervised fraud identification in the government domain, specifically within the Paycheck Protection Program (PPP). The purpose of this study was to identify features and develop intelligent semi-supervised fraud detection methodologies tailored to the PPP. Following the methodology outlined in Chapter 3, this chapter reports the outcomes of each phase of the research process, including data cleaning, feature engineering, unsupervised clustering, and semi-supervised classification. The findings are organized around the study’s two primary research questions and their corresponding hypotheses, and each analytic step is documented with summary metrics, visualizations, and tables as appropriate. A schematic overview of the modeling pipeline is presented in Figure 12, with full programming scripts provided in a public GitHub repository at https://github.com/sappw1/Dissertation under the MIT License.

**Figure 12***Research Methodology and Design Process Diagram (Repeated From Figure 11)*



## Data Preprocessing and Modeling

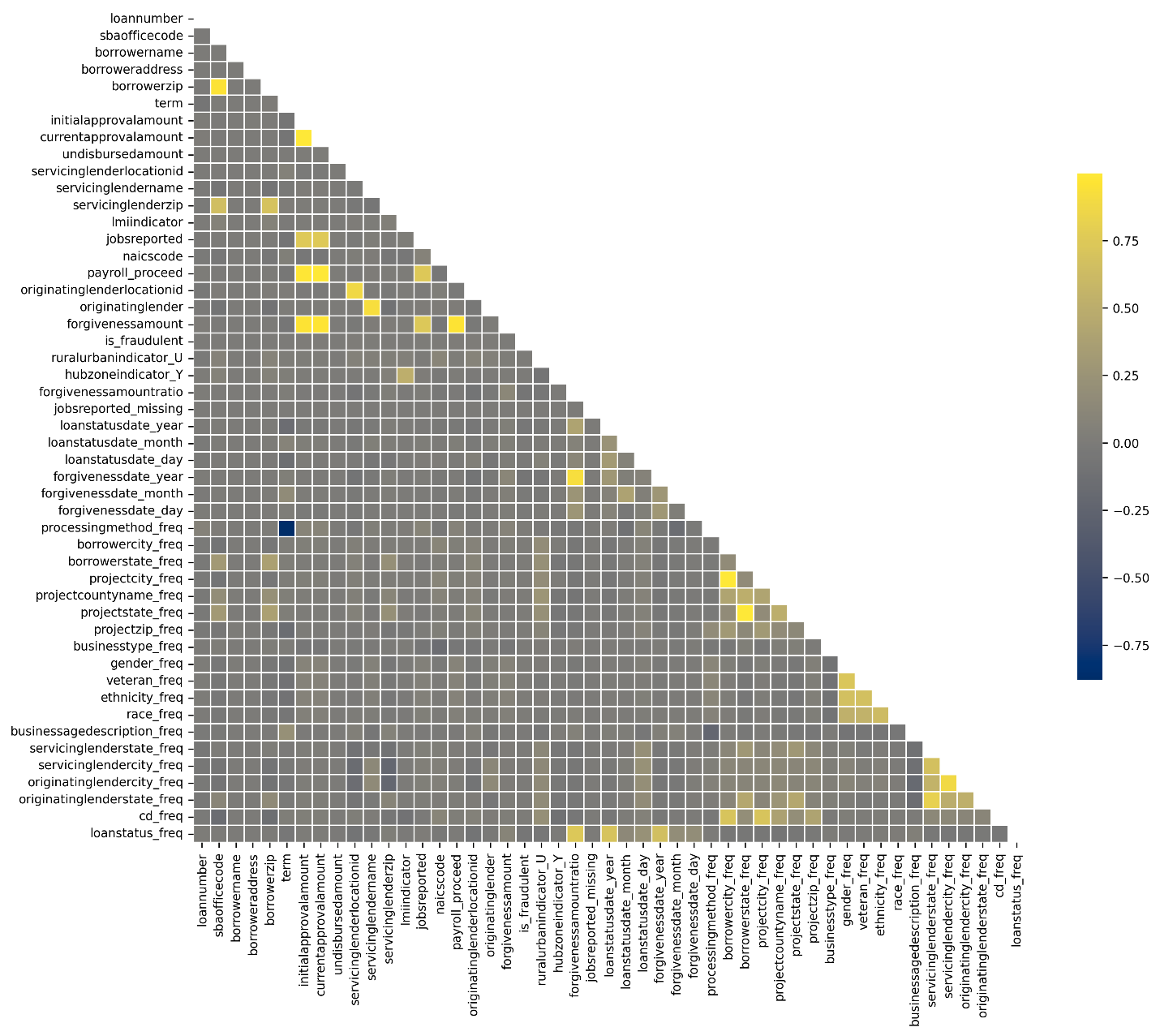
The original dataset obtained from the United States Small Business Administration (SBA) included 968,525 loan records for businesses that received more than $150,000 under the PPP. Each record contained structured loan data including borrower entity name, loan amount, geographic location, NAICS codes, reported jobs, and loan forgiveness status. This raw dataset contained 52 primary columns, of which a subset was retained or engineered for downstream analysis.

### Data Cleaning and Integration

An initial round of preprocessing was performed to support safe handling and exploratory analysis. PII such as *BorrowerName* and *BorrowerAddress* was hashed to ensure record-level uniqueness without exposing sensitive data. After this, fraud labels were integrated by linking prosecuted fraud cases gathered from PRAC, SBA OIG, and DOJ sources to specific loan records using a custom BeautifulSoup-based scraper and manual validation. This process resulted in 301 labeled fraudulent records, which were appended to the full dataset.

### Feature Engineering and Transformation

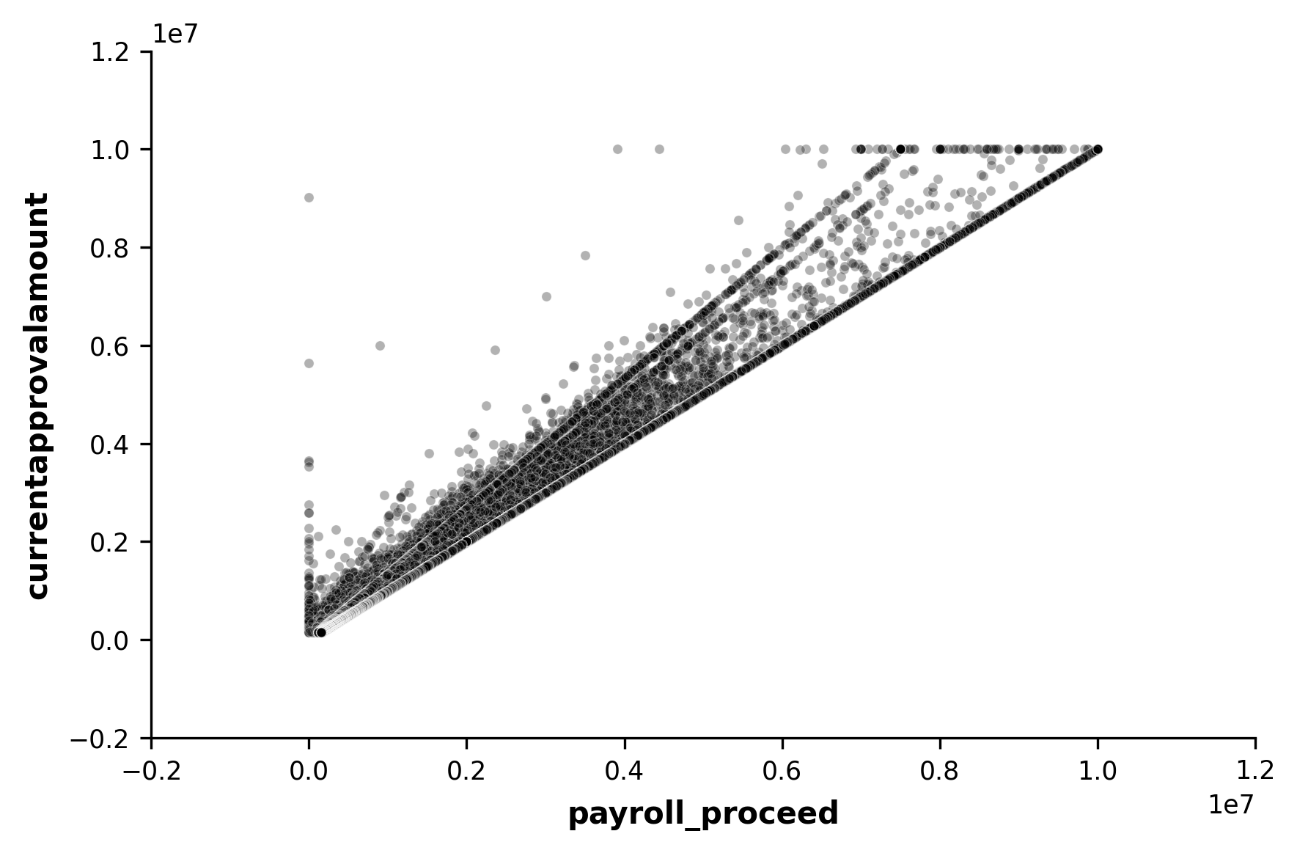
Two stages of transformation followed. In the first phase, initial features were created to capture potential fraud signals, such as the loan-to-employee ratio. Before the second phase, EDA, to include univariate and correlation analysis was conducted to identify the dataset distribution as well as feature redundancy and importance. To assess feature redundancy and evaluate linear relationships among variables, a correlation matrix using Pearson’s correlation coefficient was computed using the full feature set (see Figure 13). To support correlation-based feature analysis and assess redundancy among key variables, individual scatterplots were generated for variable pairs exhibiting the strongest Pearson correlations. These included relationships among loan approval amounts, forgiveness totals, payroll allocations, and reported employee counts. The pronounced linear patterns observed in these visualizations (Figures 14-18) reinforce the presence of multicollinearity within the dataset and visually confirm the patterns suggested by the correlation matrix.

**Figure 13***Correlation Matrix, Full Feature Set*

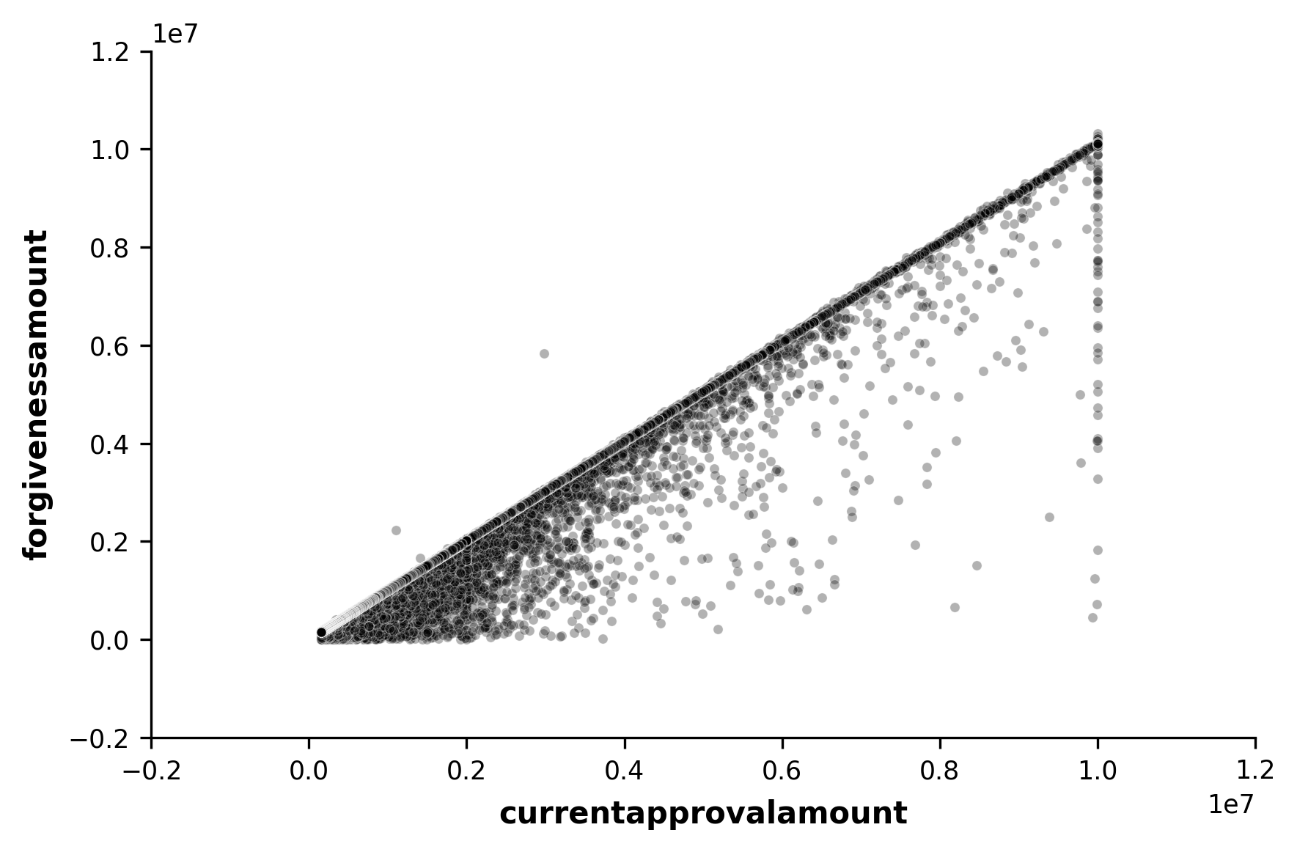
**Figure 14***Scatterplot Showing currentapprovalamount vs initialapprovalamount*A graph of a point

AI-generated content may be incorrect.

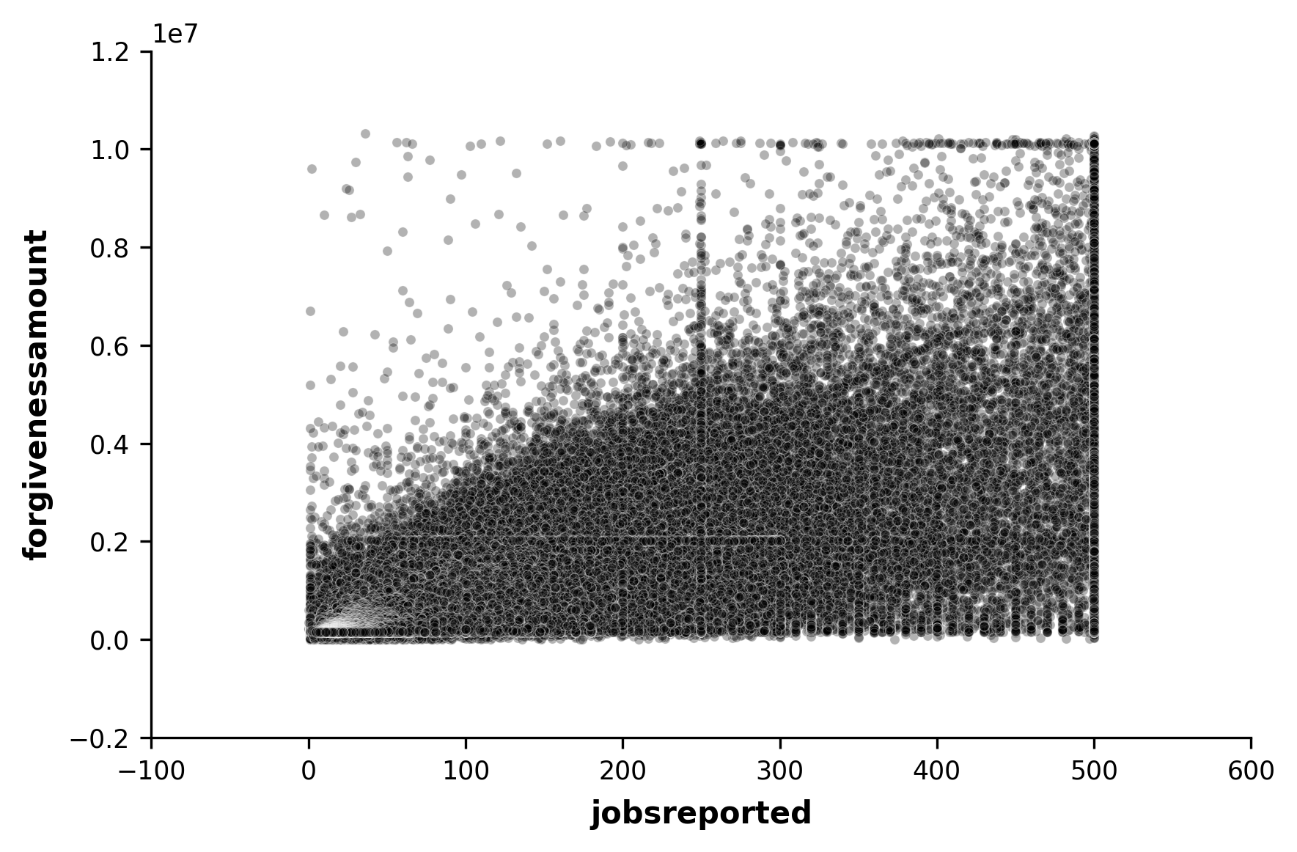
**Figure 15***Scatterplot Showing currentapprovalamount vs payroll\_proceed*



**Figure 16***Scatterplot Showing forgivenessamount vs currentapprovalamount*



**Figure 17***Scatterplot Showing forgivenessamount vs jobsreported*



**Figure 18***Scatterplot Showing payroll\_process vs jobsreported*



Both the full feature set and the reduced feature subset built for dimensionality analysis and clustering reflect the preprocessing strategy outlined in Chapter 3, where missing values were intentionally not imputed in order to preserve potential fraud-related signals. Instead, binary flags such as *jobsreported\_missing* were retained as independent features to capture patterns of incomplete reporting, a known fraud risk indicator. Categorical variables were transformed using frequency or binary encoding depending on their cardinality, resulting in features such as *businesstype\_freq*, *hubzoneindicator\_Y*, and *ruralurbanindicator\_U*, all of which carry potential relevance in the context of PPP oversight.

Continuous variables were filtered using Pearson correlation to identify and remove redundant dimensions; for example, *initialapprovalamount* was dropped due to near-perfect correlation with *currentapprovalamount*. The selection process prioritized features that were both statistically informative and interpretable within a fraud detection framework. The final key subset included ten variables: *currentapprovalamount*, *jobsreported*, *forgivenessamountratio*, *ruralurbanindicator\_U*, *hubzoneindicator\_Y*, *naicscode*, *businesstype\_freq*, *lmiindicator*, *jobsreported\_missing*, and *currentapprovalamount\_missing* (see Figure 19). These were evaluated alongside the full feature set in downstream PCA and clustering analysis with the reduced set serving as a streamlined benchmark for comparison.

**Figure 19**   
*Correlation Matrix, Key Feature Subset*

A graph with yellow squares

AI-generated content may be incorrect.

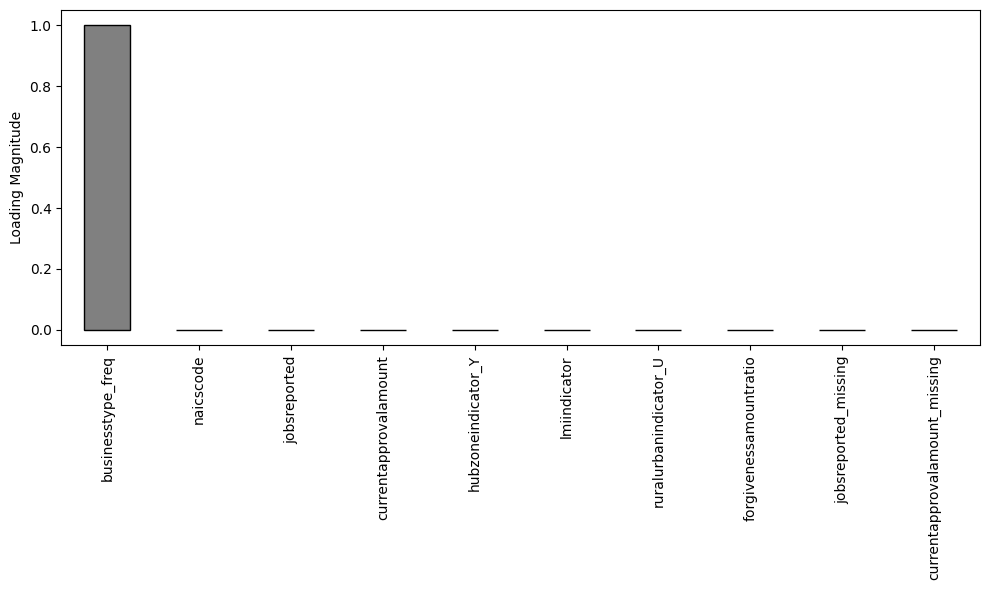
Although PCA is commonly used for feature reduction based on variance retention, this study intentionally did not use PCA as a feature selection tool. Instead, features were filtered using a combination of Pearson correlation analysis and domain-specific interpretability to retain variables most relevant to PPP fraud detection. PCA was subsequently applied to both the full and reduced feature sets as a dimensionality reduction technique to assess structural variance and support clustering. The key feature subset collapsed into a single dominant component explaining 95% of the total variance, with loadings concentrated primarily in a single categorical feature, *businesstype\_freq* (see Figures 20 & 21 below).

**Figure 20***Scree Plot for Key Feature Subset*

A line graph with dots

AI-generated content may be incorrect.

**Figure 21***PCA Loading Magnitude, Key Feature Subset*



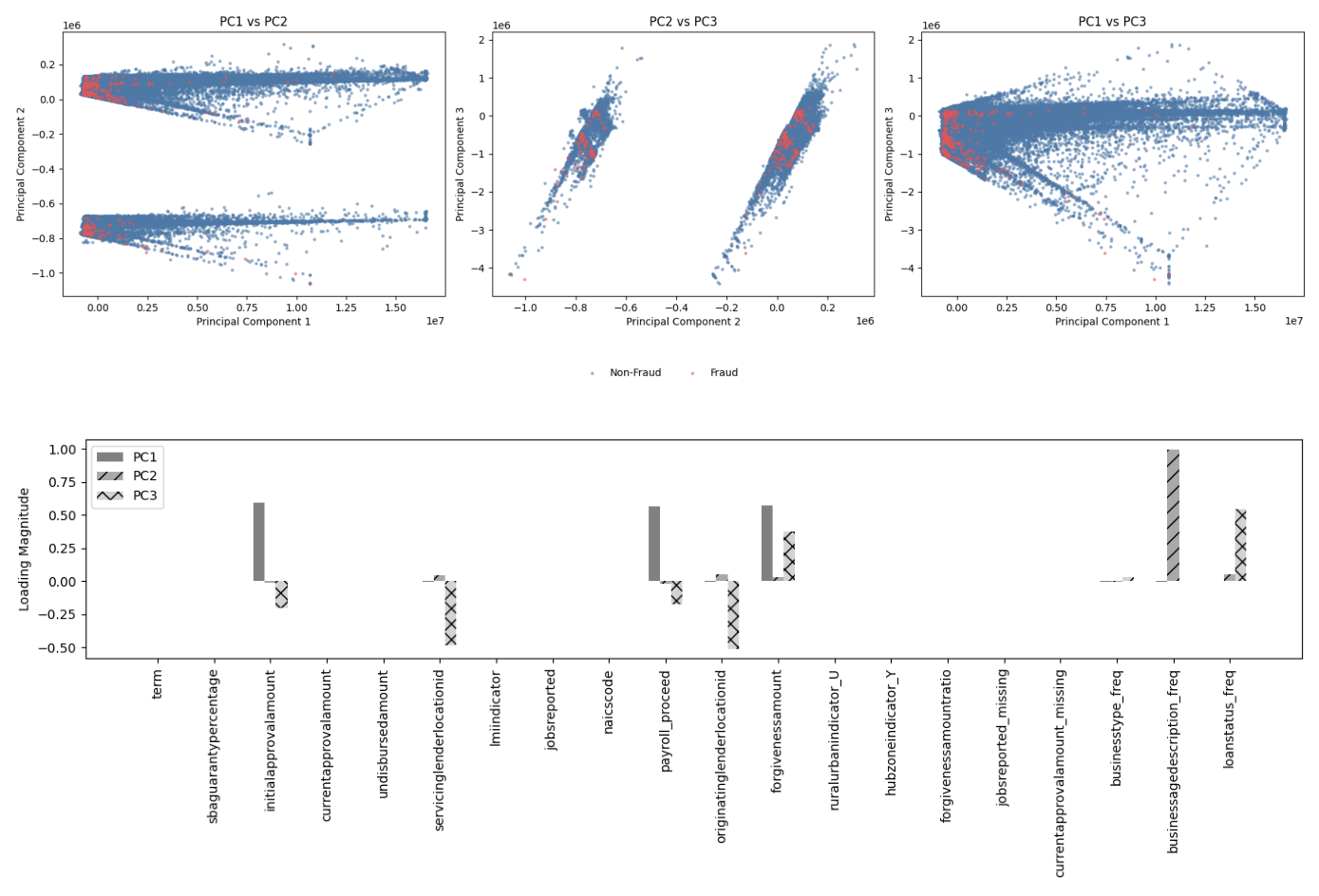
In contrast, the full feature set required three principal components to achieve comparable variance retention, with loadings more evenly distributed across a broader set of features. This richer representation was further visualized through PCA scatter plots, which revealed clearer structure and separation in the full feature space, particularly after known fraud labels were reintroduced (see Figures 22 & 23 below).

**Figure 22***Scree Plot for Full Feature Set*

A line graph with dots

AI-generated content may be incorrect.

**Figure 23***PCA Projection Scatterplot and Feature Loadings (Full Set, 3Components)*



*Note*. The top panel displays pairwise scatter plots of the first three principal components derived from PCA on the full feature set, with red points indicating fraud cases and blue for non-fraud. The bottom panel shows the loading magnitudes of original features on PC1, PC2, and PC3, highlighting their respective contributions. Together, these visualizations demonstrate both structural separation in PCA space and the relative influence of features, supporting the interpretability and effectiveness of dimensionality reduction.

### Unsupervised Learning (Clustering Phase)

The unsupervised phase of the analysis employed three clustering algorithms — K-Means, Agglomerative Hierarchical Clustering, and DBSCAN — to identify structural patterns and potential anomalies within the PPP loan dataset. Clustering was conducted on two versions of the feature matrix: the full uncompressed feature set (*X\_all\_no\_pca*) and a PCA-reduced variant (*X\_all\_pca\_3*), in which the number of components was selected to preserve approximately 95% of the variance in the original features.

To evaluate the influence of feature dimensionality on clustering quality, both versions were subjected to systematic exploration of hyperparameters. Parameters were defined in structured JSON configuration files, functioning as a manual grid search, enabling hyperparameter tuning by systematically passing all combinations into the pipeline for sequential execution. This approach supported reproducibility while allowing flexibility across multiple parameter ranges.

K-Means and Agglomerative Clustering were each evaluated over a series of cluster counts (*k* = 2–10 for K-Means; 2–6 for Hierarchical). DBSCAN was tested using combinations of ε values ranging from 0.3 to 1.3 and min\_samples values from 3 to 15. All clustering operations were executed using GPU-accelerated RAPIDS cuML implementations to ensure efficient processing across the high-dimensional feature spaces. Cluster quality was assessed using two metrics: the silhouette score to evaluate intra-cluster cohesion and inter-cluster separation, and the Davies-Bouldin Index (DBI) to assess average similarity between clusters (see Figures 24-26 in the Results section below). While both hierarchical and DBSCAN models are deterministic, reproducibility and consistency of centroid initialization for K-Means was ensured by setting the random state to 42 for each configuration.

Notably, across all algorithms and configurations, clustering on the non-PCA feature space (*X\_all\_no\_pca*) consistently outperformed the PCA-reduced variant, yielding better cohesion and separation as measured by both metrics. Clustering results from both the full feature set and the PCA feature set were encoded as features to be appended to the PCA and full feature sets as engineered features for evaluation during the supervised classification phase.

### Supervised Classification

Following unsupervised clustering, supervised classification was employed to test the predictive power of hybrid models that combined raw PPP loan features with clustering-derived variables. This approach aligned directly with Research Question 2 and Hypothesis 2, which tested whether model combinations incorporating unsupervised learning improved fraud detection accuracy.

All models were trained on the both the PCA and non-PCA full feature sets. The dataset was augmented with 128 cluster derived features indicating either cluster label or noise index (for DBSCAN). These engineered features captured structural insights uncovered during unsupervised learning and were injected into the supervised learning pipeline as additional predictors. The binary target variable *is\_fraudulent* was appended from the labeled set of 301 known fraud cases identified via DOJ, SBA OIG, and PRAC datasets. Each classification model was evaluated using a 70/30 stratified train-test split, repeated across five random seeds (42, 52, 62, 72, 82) to assess variability and robustness. SMOTE was applied to the training data in each fold to address the severe class imbalance, with known fraud cases representing less than 0.05% of the population. The following classification algorithms were tested:

* Logistic Regression
* Support Vector Machine (SVM)
* Random Forest
* XGBoost
* Neural Network (Keras/TensorFlow)
* Gaussian Naïve Bayes

Each model was paired with one clustering-derived features to form distinct experimental configurations, as well as a baseline configuration without cluster-features. A total of 7,710 model runs were executed, each representing a unique combination of model, clustering-derived feature, and random seed. Model performance was evaluated using five key metrics:

* AUC ROC
* Accuracy
* Precision
* Recall
* F1 Score

These metrics provided a multidimensional perspective on classification quality, particularly useful in the context of rare-event detection like fraud. All metrics were calculated on the test folds and aggregated across the five seeds. Performance variation and statistical comparisons are addressed in the next section.

## Results

This section presents the study’s findings direct alignment with the research questions and hypotheses. The analysis is organized sequentially, beginning with exploratory data analysis and progressing through the results of unsupervised and semi-supervised learning phases. Model performance is reported using standard classification metrics. The source code used in this study is publicly available at https://github.com/sappw1/Dissertation, licensed under the MIT License. This repository includes all scripts required for data preprocessing, clustering, classification, and analysis. Due to the size constraints of GitHub as a hosting platform, the modified (intermediate and final) data files are not included directly in the repository. However, all code to transform the publicly released PPP dataset and labeled fraud cases into the final preprocessed form is provided, ensuring full reproducibility of the experiments.

### Data Modeling Evaluation

In the unsupervised modeling phase three clustering algorithms (K-Means, Agglomerative Hierarchical Clustering, and DBSCAN) were applied to identify structural groupings and anomalous patterns in the PPP dataset. These models were tested across both PCA-transformed and non-transformed feature spaces, with a specific emphasis on detecting configurations that revealed or enriched fraudulent behavior.

**Clustering Configuration Space:** Two primary versions of the feature matrix were constructed for analysis:

* *X\_all\_no\_pca*: The full feature set with no dimensionality reduction applied.
* *X\_all\_pca\_3*: The same feature set reduced to three components using PCA retaining approximately 95% of total variance.

Clustering was conducted over the following configuration ranges:

* K-Means: Cluster numbers *k* ∈ [2,10]
* Agglomerative Hierarchical Clustering: Cluster numbers *k* ∈ [2,6]
* DBSCAN: ε values ranging from 0.3 to 1.3, and min\_samples ranging from 3 to 15

All experiments were executed on NVIDIA A100 GPUs using the RAPIDS cuML framework, allowing for scalable in-memory computation. Outputs were persisted with accompanying configuration metadata to support traceability and reproducibility.

**Cluster Quality Evaluation:** Cluster validation was assessed using two internal metrics:

* Silhouette Score: Measures the cohesion and separation of clusters.
* Davies-Bouldin Index (DBI): Evaluates intra-cluster similarity and inter-cluster distinctiveness; lower values are preferred.

As shown in figures 14-16 below, non-PCA versions of the feature matrix consistently yielded superior silhouette scores across all three algorithms, indicating better-defined groupings in the raw feature space. DBI results mirrored this trend, particularly for K-Means and DBSCAN, where PCA-transformed inputs degraded clustering coherence.

**Figure 24***Cluster Metrics: K-Means*

A graph of different sizes and shapes

AI-generated content may be incorrect.

**Figure 25***Cluster Metrics: Hierarchical*

A graph of different sizes and colors

AI-generated content may be incorrect.

**Figure 26***Cluster Metrics: DBSCAN*

A graph of two people

AI-generated content may be incorrect.

**Cluster Label Utility and Fraud Enrichment:** Post hoc analysis was performed to evaluate the usefulness of clustering outputs in relation to known fraud labels (*n*=301). This included calculating the following for each configuration:

* *fraud\_cluster\_max*: The maximum proportion of fraud in any single cluster.
* *fraud\_cluster\_avg*: The average proportion of fraud across clusters.

For DBSCAN, clusters identified as “noise” (i.e., not assigned to any group) were of particular interest. Several configurations showed substantial enrichment of fraud in the noise set, a counterintuitive but useful result suggesting that fraudulent loans frequently exhibit anomalous behavior that resists clustering. Together, these outputs supported downstream experimentation with hybrid clustering-classification models. Each classification model was tested across all configurations of the three clustering inputs (K-Means, Hierarchical, and DBSCAN) alongside a baseline condition with no clustering features included.

Following the integration of cluster-derived features, six classification models, Logistic Regression, Support Vector Machine, Neural Network, Gaussian Naïve Bayes, Random Forest, and XGBoost, were trained and evaluated using stratified 70/30 train-test splits with SMOTE applied to the training partition. Each classifier was tested under the eight cluster feature configurations. Each experiment was repeated across five random seeds, and average results were computed.

Performance was measured using five metrics: ROC AUC, accuracy, precision, recall, and F1 score. The ranges of metric results per classification model are shown below in Figures 27-31. Complete classification modelling results are included in Appendices B and C.

**Figure 27***ROC AUC Metrics per Classification Model*

A diagram of a graph

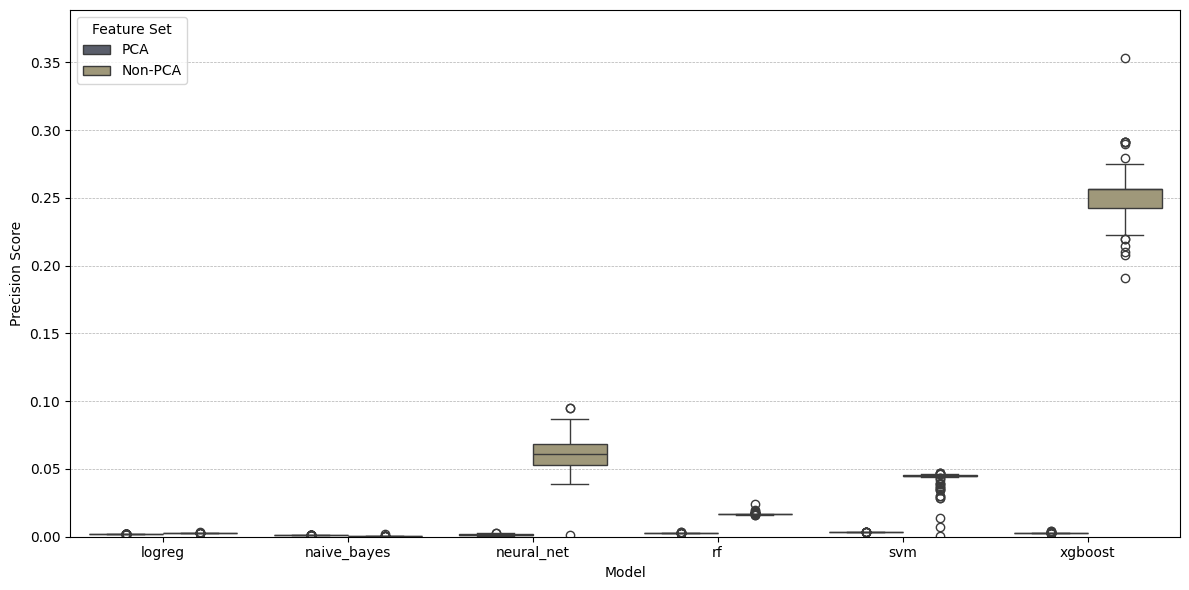
AI-generated content may be incorrect.

**Figure 28***Accuracy Metrics per Classification Model*

A graph with a diagram

AI-generated content may be incorrect.

**Figure 29***Precision Metrics per Classification Model*



**Figure 30***Recall Metrics per Classification Model*

A diagram of a diagram

AI-generated content may be incorrect.

**Figure 31***F1 Score Metrics per Classification Model*

A graph showing a number of objects

AI-generated content may be incorrect.

These results establish the foundation for the formal statistical comparisons discussed in the following section.

### Data Analysis

Statistical analyses were conducted to evaluate both the quality of unsupervised clustering and the performance differences among supervised model configurations, with direct alignment to the study’s hypotheses.

To assess whether clustering quality varied across feature configurations, Silhouette Scores and Davies-Bouldin Indices (DBI) were computed for each algorithm-feature pairing. Additionally, the maximum average fraud ratio across clusters and the amount of fraud captured in DBSCAN as noise was calculated as an external reference metric to capture how well each algorithm surfaced groupings enriched with known fraud cases. These comparative metrics are reported below in tables 5 through 8.

**Table 5***K-Means Clustering Metrics*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Features | Clusters | Silhouette | DBI | *fraud\_cluster\_max* | *fraud\_cluster\_avg* |
| PCA | 2 | 0.843245522 | 0.510316545 | 0.000665705 | 0.000483147 |
| PCA | 3 | 0.718384425 | 0.546506307 | 0.001251324 | 0.000675178 |
| PCA | 4 | 0.668992344 | 0.569960437 | 0.001237317 | 0.000664451 |
| PCA | 5 | 0.55264807 | 0.629113447 | 0.001558603 | 0.000667545 |
| PCA | 6 | 0.526036533 | 0.649623743 | 0.001508043 | 0.000593333 |
| PCA | 7 | 0.589643802 | 0.590588094 | 0.001572602 | 0.000619572 |
| PCA | 8 | 0.588936113 | 0.723486169 | 0.028806584 | 0.004015314 |
| PCA | 9 | 0.422826032 | 0.759276939 | 0.03649635 | 0.004460822 |
| PCA | 10 | 0.422998204 | 0.814352247 | 0.036764706 | 0.00499318 |
| Full | 2 | 0.990042503 | 0.006963791 | 0.000310782 | 0.000155391 |
| Full | 3 | 0.942619304 | 0.247048342 | 0.000310783 | 0.000103594 |
| Full | 4 | 0.942625485 | 0.197440679 | 0.000310783 | 7.77E-05 |
| Full | 5 | 0.237202516 | 0.862692387 | 0.000523362 | 0.000164249 |
| Full | 6 | 0.057900921 | 1.696663268 | 0.000488129 | 0.000162442 |
| Full | 7 | 0.110286393 | 1.353748264 | 0.000518973 | 0.000159807 |
| Full | 8 | 0.115929659 | 1.357979656 | 0.000709649 | 0.000241513 |
| Full | 9 | 0.066596872 | 1.496911041 | 0.000784929 | 0.000217015 |
| Full | 10 | 0.100858788 | 1.41492132 | 0.00567698 | 0.000671921 |

**Table 6***Hierarchical Clustering Metrics*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Features | Clusters | Silhouette | DBI | *fraud\_cluster\_max* | *fraud\_cluster\_avg* |
| PCA | 2 | 0.892468504 | 0.069875188 | 0.000310782 | 0.000155391 |
| PCA | 3 | 0.886305208 | 0.0726508 | 0.000310783 | 0.000103594 |
| PCA | 4 | 0.886239269 | 0.068863185 | 0.000310783 | 7.77E-05 |
| PCA | 5 | 0.88606367 | 0.067111184 | 0.000310783 | 6.22E-05 |
| PCA | 6 | 0.867178223 | 0.073874812 | 0.000310784 | 5.18E-05 |
| Full | 2 | 0.990525439 | 0.006626085 | 0.000310782 | 0.000155391 |
| Full | 3 | 0.990043294 | 0.006850178 | 0.000310783 | 0.000103594 |
| Full | 4 | 0.960269456 | 0.017296308 | 0.000310783 | 7.77E-05 |
| Full | 5 | 0.926243701 | 0.046628992 | 0.000310783 | 6.22E-05 |
| Full | 6 | 0.825266343 | 0.06585075 | 0.000310784 | 5.18E-05 |

**Table 7***DBSCAN Clustering Metrics, PCA Feature Set*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Params | Clusters | Silhouette | DBI | *fraud\_cluster\_max* | *fraud\_cluster\_avg* | *capture\_pct* |
| [0.3,3] | 92 |  |  | 0.714285714 | 0.011267997 | 0.980066445 |
| [0.5,3] | 92 |  |  | 0.714285714 | 0.011267997 | 0.980066445 |
| [0.7,3] | 132 |  |  | 0.714285714 | 0.007879126 | 0.980066445 |
| [1,3] | 132 |  |  | 0.714285714 | 0.007879126 | 0.980066445 |
| [1.3,3] | 167 |  |  | 0.714285714 | 0.006237642 | 0.980066445 |
| [0.3,5] | 12 | -0.53478 | 0.68 | 0.714285714 | 0.054968566 | 0.983388704 |
| [0.5,5] | 12 | -0.53478 | 0.68 | 0.714285714 | 0.054968566 | 0.983388704 |
| [0.7,5] | 22 | -0.66663 | 0.87 | 0.714285714 | 0.03106919 | 0.983388704 |
| [1,5] | 22 | -0.66663 | 0.87 | 0.714285714 | 0.03106919 | 0.983388704 |
| [1.3,5] | 30 | -0.67646 | 0.91 | 0.714285714 | 0.023051335 | 0.983388704 |
| [0.3,7] | 4 | 0.282345 | 0.53 | 0.714285714 | 0.142918269 | 0.983388704 |
| [0.5,7] | 4 | 0.282345 | 0.53 | 0.714285714 | 0.142918269 | 0.983388704 |
| [0.7,7] | 5 | 0.275095 | 0.51 | 0.714285714 | 0.119098558 | 0.983388704 |
| [1,7] | 5 | 0.275095 | 0.51 | 0.714285714 | 0.119098558 | 0.983388704 |
| [1.3,7] | 6 | -0.47355 | 0.68 | 0.714285714 | 0.102084478 | 0.983388704 |
| [0.3,10] | 0 |  |  | 0.000310782 | 0.000310782 | 1 |
| [0.5,10] | 0 |  |  | 0.000310782 | 0.000310782 | 1 |
| [0.7,10] | 0 |  |  | 0.000310782 | 0.000310782 | 1 |
| [1,10] | 0 |  |  | 0.000310782 | 0.000310782 | 1 |
| [1.3,10] | 1 |  |  | 0.000310785 | 0.000155393 | 1 |
| [0.3,15] | 0 |  |  | 0.000310782 | 0.000310782 | 1 |
| [0.5,15] | 0 |  |  | 0.000310782 | 0.000310782 | 1 |
| [0.7,15] | 0 |  |  | 0.000310782 | 0.000310782 | 1 |
| [1,15] | 0 |  |  | 0.000310782 | 0.000310782 | 1 |
| [1.3,15] | 0 |  |  | 0.000310782 | 0.000310782 | 1 |

**Table 8***DBSCAN Clustering Metrics, Full Feature set*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Params | Clusters | Silhouette | DBI | *fraud\_cluster\_max* | *fraud\_cluster\_avg* | *capture\_pct* |
| [0.3,3] | 21 | -0.28654 | 1.13 | 0.000310805 | 1.41E-05 | 1 |
| [0.5,3] | 3 |  |  | 0.000310881 | 3.99E-06 | 1 |
| [0.7,3] | 188 |  |  | 0.00031102 | 1.65E-06 | 1 |
| [1,3] | 610 |  |  | 0.000311537 | 5.10E-07 | 1 |
| [1.3,3] | 2790 |  |  | 0.333333333 | 0.000119544 | 0.996677741 |
| [0.3,5] | 2 | -0.18542 | 1.41 | 0.000310785 | 0.000103595 | 1 |
| [0.5,5] | 10 | -0.24851 | 1.31 | 0.000310803 | 2.83E-05 | 1 |
| [0.7,5] | 21 | -0.29637 | 1.31 | 0.000310834 | 1.41E-05 | 1 |
| [1,5] | 61 |  |  | 0.000310929 | 5.01E-06 | 1 |
| [1.3,5] | 325 |  |  | 0.000311684 | 9.56E-07 | 1 |
| [0.3,7] | 0 |  |  | 0.000310782 | 0.000310782 | 1 |
| [0.5,7] | 3 | -0.19559 | 1.3 | 0.000310789 | 7.77E-05 | 1 |
| [0.7,7] | 8 | -0.24595 | 1.33 | 0.000310808 | 3.45E-05 | 1 |
| [1,7] | 19 | -0.29185 | 1.32 | 0.00031085 | 1.55E-05 | 1 |
| [1.3,7] | 79 |  |  | 0.000311089 | 3.89E-06 | 1 |
| [0.3,10] | 0 |  |  | 0.000310782 | 0.000310782 | 1 |
| [0.5,10] | 0 |  |  | 0.000310782 | 0.000310782 | 1 |
| [0.7,10] | 0 | -0.14015 | 1.16 | 0.000310789 | 0.000103596 | 1 |
| [1,10] | 5 | -0.20029 | 1.26 | 0.000310804 | 5.18E-05 | 1 |
| [1.3,10] | 18 | -0.28556 | 1.3 | 0.000310874 | 1.64E-05 | 1 |
| [0.3,15] | 0 |  |  | 0.000310782 | 0.000310782 | 1 |
| [0.5,15] | 0 |  |  | 0.000310782 | 0.000310782 | 1 |
| [0.7,15] | 0 |  |  | 0.000310782 | 0.000310782 | 1 |
| [1,15] | 0 |  |  | 0.000310782 | 0.000310782 | 1 |
| [1.3,15] | 3 | -0.18656 | 1.47 | 0.000310797 | 7.77E-05 | 1 |

Prior to ANOVA for supervised classification models, all assumptions were assessed. The experimental design ensured independence of observations: each model-cluster feature-seed combination was evaluated independently on non-overlapping data instances. For each evaluation metric, normality of residuals was evaluated using a Q-Q plot (Figures 32-36 below). Levene’s test was then conducted within classification model groups to confirm the assumption of homogeneity of variance. While certain metrics showed significant heterogeneity (e.g., Full Feature-set Naïve Bayes and SVM), most metrics showed homogeneity of variance. Despite the identified instances of violation of homogeneity of variance, the balanced experimental design (i.e., using equal sample sizes across all model and cluster configuration groups) reduces their effect on the robustness of two-way ANOVA (Glass et al., 1972; Lix et al., 1996).

**Figure 32***Q-Q Plot: AUC ROC*

A graph of a line graph

AI-generated content may be incorrect.

**Figure 33**  
*Q-Q Plot: Accuracy*

A graph of a graph of a graph

AI-generated content may be incorrect.

**Figure 34***Q-Q Plot: Precision*

A graph of a graph showing the value of a certain amount of time

AI-generated content may be incorrect.

**Figure 35**   
*Q-Q Plot: Recall*

A graph of a graph showing the value of a number of individuals

AI-generated content may be incorrect.

**Figure 36***Q-Q Plot: F1 Score*

A graph of a graph showing the value of a certain amount of time

AI-generated content may be incorrect.

**Table 9***Levene's Test Results: PCA Feature-set*

|  |  |  |  |
| --- | --- | --- | --- |
| **Model** | **Metric** | **Levene Stat** | ***p* value** |
| XGBoost | auc\_roc | 1.22E-01 | 1 |
| XGBoost | accuracy | 6.92E-02 | 1 |
| XGBoost | precision | 4.55E-01 | 1 |
| XGBoost | recall | 8.50E-02 | 1 |
| XGBoost | f1 | 4.47E-01 | 1 |
| Neural Net | auc\_roc | 9.35E-01 | 0.673408 |
| Neural Net | accuracy | 9.00E-01 | 0.76371 |
| Neural Net | precision | 7.43E-01 | 0.978981 |
| Neural Net | recall | 8.32E-01 | 0.89741 |
| Neural Net | f1 | 7.41E-01 | 0.97973 |
| Random Forest | auc\_roc | 1.02E-01 | 1 |
| Random Forest | accuracy | 1.06E-01 | 1 |
| Random Forest | precision | 2.51E-01 | 1 |
| Random Forest | recall | 5.84E-02 | 1 |
| Random Forest | f1 | 2.48E-01 | 1 |
| Naïve Bayes | auc\_roc | 2.24E-04 | 1 |
| Naïve Bayes | accuracy | 3.73E-04 | 1 |
| Naïve Bayes | precision | 6.23E-03 | 1 |
| Naïve Bayes | recall | 1.24E-03 | 1 |
| Naïve Bayes | f1 | 6.19E-03 | 1 |
| SVM | auc\_roc | 6.87E-07 | 1 |
| SVM | accuracy | 1.85E-05 | 1 |
| SVM | precision | 6.48E-05 | 1 |
| SVM | recall | 3.35E-29 | 1 |
| SVM | f1 | 6.57E-05 | 1 |
| Logistic Regression | auc\_roc | 3.72E-02 | 1 |
| Logistic Regression | accuracy | 6.66E-01 | 0.997073 |
| Logistic Regression | precision | 3.55E-01 | 1 |
| Logistic Regression | recall | 1.21E-02 | 1 |
| Logistic Regression | f1 | 3.54E-01 | 1 |

**Table 10**   
*Levene's Test Results: Full Feature-set*

|  |  |  |  |
| --- | --- | --- | --- |
| **Model** | **Metric** | **Levene Stat** | ***p* value** |
| XGBoost | auc\_roc | 0.272971 | 1.00E+00 |
| XGBoost | accuracy | 0.236078 | 1.00E+00 |
| XGBoost | precision | 0.235039 | 1.00E+00 |
| XGBoost | recall | 0.307153 | 1.00E+00 |
| XGBoost | f1 | 0.242305 | 1.00E+00 |
| Neural Net | auc\_roc | 0.32831 | 1.00E+00 |
| Neural Net | accuracy | 9.716218 | 7.45E-80 |
| Neural Net | precision | 0.729357 | 9.85E-01 |
| Neural Net | recall | 0.740896 | 9.80E-01 |
| Neural Net | f1 | 0.749649 | 9.76E-01 |
| Random Forest | auc\_roc | 0.590469 | 1.00E+00 |
| Random Forest | accuracy | 0.047099 | 1.00E+00 |
| Random Forest | precision | 0.356146 | 1.00E+00 |
| Random Forest | recall | 0.04261 | 1.00E+00 |
| Random Forest | f1 | 0.315974 | 1.00E+00 |
| Naïve Bayes | auc\_roc | 1.068431 | 3.07E-01 |
| Naïve Bayes | accuracy | 1.398422 | 6.09E-03 |
| Naïve Bayes | precision | 2.183373 | 8.24E-10 |
| Naïve Bayes | recall | 1.707515 | 2.50E-05 |
| Naïve Bayes | f1 | 2.180645 | 8.78E-10 |
| SVM | auc\_roc | 0.186243 | 1.00E+00 |
| SVM | accuracy | 12.473753 | 1.63E-98 |
| SVM | precision | 0.167441 | 1.00E+00 |
| SVM | recall | 0.416753 | 1.00E+00 |
| SVM | f1 | 0.132572 | 1.00E+00 |
| Logistic Regression | auc\_roc | 0.030122 | 1.00E+00 |
| Logistic Regression | accuracy | 0.016147 | 1.00E+00 |
| Logistic Regression | precision | 0.126987 | 1.00E+00 |
| Logistic Regression | recall | 0.04356 | 1.00E+00 |
| Logistic Regression | f1 | 0.126535 | 1.00E+00 |

A series of two-way ANOVA were conducted to test statistically significant differences across supervised model types and clustering configurations (Tables 11-20 below). While classification model selection clearly had significant effects on variance across all metrics, cluster condition (the selected cluster feature for each configuration) had mostly non-significant effects, with the notable exceptions of AUC ROC for both the PCA and full feature-set. In the two-way ANOVA, which compares the effects of model selection, cluster condition, and their relationship, the large variance based on model selection is clearly overshadowing, or drowning out, any cluster condition-based variance within each classification model group. Therefore, it was determined that traditional one-way ANOVA should be tested within each classification model group in order to isolate intra-model variance based solely on cluster condition.

**Table 11***Two-Way ANOVA: AUC ROC, PCA Feature-set*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Metric** | **sum\_sq** | **df** | **F** | **PR(>F)** |
| C(model) | 4.694079 | 5 | 1409.028 | 0 |
| C(cluster\_condition) | 0.11313 | 128 | 1.326496 | 0.009322 |
| C(model):C(cluster\_condition) | 0.414001 | 640 | 0.970869 | 0.679506 |
| Residual | 2.062822 | 3096 |  |  |

**Table 12**   
*Two-Way ANOVA: Accuracy, PCA Feature-set*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Metric** | **sum\_sq** | **df** | **F** | **PR(>F)** |
| C(model) | 10.74882 | 5 | 858.5021 | 0 |
| C(cluster\_condition) | 0.280325 | 128 | 0.874586 | 0.838818 |
| C(model):C(cluster\_condition) | 1.390119 | 640 | 0.867406 | 0.988289 |
| Residual | 7.752655 | 3096 |  |  |

**Table 13**   
*Two-Way ANOVA: Precision, PCA Feature-set*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Metric** | **sum\_sq** | **df** | **F** | **PR(>F)** |
| C(model) | 0.001888 | 5 | 1950.842 | 0 |
| C(cluster\_condition) | 2.15E-05 | 128 | 0.868626 | 0.85106 |
| C(model):C(cluster\_condition) | 8.45E-05 | 640 | 0.682118 | 1 |
| Residual | 0.000599 | 3096 |  |  |

**Table 14**   
*Two-Way ANOVA: Recall, PCA Feature-set*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Metric** | **sum\_sq** | **df** | **F** | **PR(>F)** |
| C(model) | 32.00516 | 5 | 1820.247 | 0 |
| C(cluster\_condition) | 0.335559 | 128 | 0.745485 | 0.984711 |
| C(model):C(cluster\_condition) | 1.405907 | 640 | 0.624679 | 1 |
| Residual | 10.88731 | 3096 |  |  |

**Table 15***Two-Way ANOVA: F1 Score, PCA Feature-set*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Metric** | **sum\_sq** | **df** | **F** | **PR(>F)** |
| C(model) | 0.00743 | 5 | 1951.605 | 0 |
| C(cluster\_condition) | 8.41E-05 | 128 | 0.86245 | 0.863131 |
| C(model):C(cluster\_condition) | 0.000331 | 640 | 0.678797 | 1 |
| Residual | 0.002357 | 3096 |  |  |

**Table 16***Two-Way ANOVA: AUC ROC, Full Feature-set*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Metric** | **sum\_sq** | **df** | **F** | **PR(>F)** |
| C(model) | 95.56745 | 5 | 50214.57 | 0 |
| C(cluster\_condition) | 0.616325 | 128 | 12.64997 | 1.55E-198 |
| C(model):C(cluster\_condition) | 2.543885 | 640 | 10.44257 | 0 |
| Residual | 1.17845 | 3096 |  |  |

**Table 17***Two-Way ANOVA: Accuracy, Full Feature-set*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Metric** | **sum\_sq** | **df** | **F** | **PR(>F)** |
| C(model) | 497.7892 | 5 | 535299.2 | 0 |
| C(cluster\_condition) | 0.462209 | 128 | 19.41555 | 8.50E-305 |
| C(model):C(cluster\_condition) | 7.429878 | 640 | 62.41987 | 0 |
| Residual | 0.575811 | 3096 |  |  |

**Table 18**   
*Two-Way ANOVA: Precision, Full Feature-set*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Metric** | **sum\_sq** | **df** | **F** | **PR(>F)** |
| C(model) | 29.81277 | 5 | 4831.404 | 0 |
| C(cluster\_condition) | 0.047437 | 128 | 0.300292 | 1 |
| C(model):C(cluster\_condition) | 0.291194 | 640 | 0.368674 | 1 |
| Residual | 3.820849 | 3096 |  |  |

**Table 19***Two-Way ANOVA: Recall, Full Feature-set*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Metric** | **sum\_sq** | **df** | **F** | **PR(>F)** |
| C(model) | 466.9194 | 5 | 87908.36 | 0 |
| C(cluster\_condition) | 0.416071 | 128 | 3.059963 | 2.96E-26 |
| C(model):C(cluster\_condition) | 4.705378 | 640 | 6.921064 | 4.63E-302 |
| Residual | 3.28884 | 3096 |  |  |

**Table 20**   
*Two-Way ANOVA: F1 Score, Full Feature-set*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Metric** | **sum\_sq** | **df** | **F** | **PR(>F)** |
| C(model) | 6.11216 | 5 | 4041.361 | 0 |
| C(cluster\_condition) | 0.037941 | 128 | 0.979952 | 0.547373 |
| C(model):C(cluster\_condition) | 0.132217 | 640 | 0.682982 | 1 |
| Residual | 0.936479 | 3096 |  |  |

To further examine the effect of cluster feature selection within each classification model, one-way ANOVA was performed per model (Tables 21-32 below). Notably, ANOVA testing within model groups identified significant variance across multiple metrics for both the PCA and full feature-set, indicating cluster-feature configuration does in fact have a significant effect on classification model performance in certain configurations.

**Table 21***ANOVA: XGBoost, PCA Feature-set*

|  |  |  |
| --- | --- | --- |
| **Metric** | **F-Statistic** | ***p value*** |
| auc\_roc | 0.4457 | 1 |
| accuracy | 5.156 | 0 |
| precision | 5.3448 | 0 |
| recall | 0.5193 | 0.999993 |
| f1 | 5.2075 | 0 |

**Table 22**   
*ANOVA: Neural Network, PCA Feature-set*

|  |  |  |
| --- | --- | --- |
| **Metric** | **F-Statistic** | ***p value*** |
| auc\_roc | 1.0526 | 0.345805 |
| accuracy | 0.8532 | 0.862054 |
| precision | 0.753 | 0.974009 |
| recall | 0.8202 | 0.913333 |
| f1 | 0.7525 | 0.97426 |

**Table 23**   
*ANOVA: Random Forest, PCA Feature-set*

|  |  |  |
| --- | --- | --- |
| **Metric** | **F-Statistic** | ***p value*** |
| auc\_roc | 3.164 | 0 |
| accuracy | 5.2452 | 0 |
| precision | 2.0361 | 0 |
| recall | 0.9669 | 0.583804 |
| f1 | 1.9993 | 0 |

**Table 24**   
*ANOVA: Naïve Bayes, PCA Feature-set*

|  |  |  |
| --- | --- | --- |
| **Metric** | **F-Statistic** | ***p value*** |
| auc\_roc | 0.0013 | 1 |
| accuracy | 13.1677 | 0 |
| precision | 2.5917 | 0 |
| recall | 0.0601 | 1 |
| f1 | 2.5829 | 0 |

**Table 25**   
*ANOVA: SVM, PCA Feature-set*

|  |  |  |
| --- | --- | --- |
| **Metric** | **F-Statistic** | ***p value*** |
| auc\_roc | 0.0005 | 1 |
| accuracy | 0 | 1 |
| precision | 0 | 1 |
| recall | 0.0006 | 1 |
| f1 | 0 | 1 |

**Table 26***ANOVA: Logistic Regression, PCA Feature-set*

|  |  |  |
| --- | --- | --- |
| **Metric** | **F-Statistic** | ***p value*** |
| auc\_roc | 0.0911 | 1 |
| accuracy | 0.9254 | 0.699128 |
| precision | 0.397 | 1 |
| recall | 0.033 | 1 |
| f1 | 0.3949 | 1 |

**Table 27**   
*ANOVA: XGBoost, Full Feature-set*

|  |  |  |
| --- | --- | --- |
| **Metric** | **F-Statistic** | ***p* value** |
| auc\_roc | 0.5702 | 0.999916 |
| accuracy | 0.252 | 1 |
| precision | 0.2472 | 1 |
| recall | 0.2713 | 1 |
| f1 | 0.2221 | 1 |

**Table 28**   
*ANOVA: Neural Network, Full Feature-set*

|  |  |  |
| --- | --- | --- |
| **Metric** | **F-Statistic** | ***p* value** |
| auc\_roc | 1.0706 | 0.301626 |
| accuracy | 30.9057 | 0 |
| precision | 1.6887 | 0.000036 |
| recall | 20.0674 | 0 |
| f1 | 1.4889 | 0.001408 |

**Table 29***ANOVA: Random Forest, Full Feature-set*

|  |  |  |
| --- | --- | --- |
| **Metric** | **F-Statistic** | ***p* value** |
| auc\_roc | 0.1349 | 1 |
| accuracy | 0.9369 | 0.668244 |
| precision | 0.471 | 1 |
| recall | 0.2449 | 1 |
| f1 | 0.3582 | 1 |

**Table 30***Welch's ANOVA: Naïve Bayes, Full Feature-set*

|  |  |  |
| --- | --- | --- |
| **Metric** | **F‑Statistic** | ***p* value** |
| auc\_roc | 0.001054 | 1.00E+00 |
| accuracy | 4.52903 | 1.23E-34 |
| precision | 5.613417 | 2.05E-45 |
| recall | 0.067553 | 1.00E+00 |
| f1 | 5.591583 | 3.32E-45 |

*Note.* Welch’s ANOVA was performed on the Naïve Bayes (Full Feature-set) model group due to significant heterogeneity as identified in Levene’s testing.

**Table 31***ANOVA: SVM, Full Feature-set*

|  |  |  |
| --- | --- | --- |
| **Metric** | **F-Statistic** | ***p* value** |
| auc\_roc | 12.2919 | 0 |
| accuracy | 316.6945 | 0 |
| precision | 4.6195 | 0 |
| recall | 17.2136 | 0 |
| f1 | 4.2171 | 0 |

**Table 32***ANOVA: Logistic Regression, Full Feature-set*

|  |  |  |
| --- | --- | --- |
| **Metric** | **F-Statistic** | ***p* value** |
| auc\_roc | 0.0383 | 1 |
| accuracy | 1.4639 | 0.002138 |
| precision | 3.0632 | 0 |
| recall | 0.1624 | 1 |
| f1 | 3.0411 | 0 |

### Research Question # 1

**RQ1**: What are the key features or variables associated with fraudulent loan applications within the PPP?  
**H10**: PPP loan applications are best clustered and further classified given the complete list of values of each feature or variable in the dataset.  
**H1a**: PPP loan applications are best clustered and further classified given the values of specific features or variables in the dataset.

This research question explored the impact of feature selection on the quality of clustering and its utility in downstream fraud detection. Two feature configurations were compared: the full feature-set and a reduced dimensional version derived using PCA. Each configuration was initially evaluated in the clustering phase using three clustering algorithms (K-Means, Agglomerative Hierarchical Clustering, and DBSCAN) with results assessed using Silhouette Score, DBI, and fraud-targeting metrics. Cluster features derived from both the full feature-set and PCA feature-set were later introduced to the full and PCA feature-sets for the supervised learning classification phase. In the classification phase, feature selection impact was explored from both the cluster-derived and supervised learning feature-set perspectives (i.e., performance of the classification models based on the use of the PCA or full feature-sets as well as inclusion of cluster features derived from the PCA or full feature-set clustering).

As shown in the tables in the previous section, the full feature-set version of the dataset consistently outperformed the PCA-reduced set at lower cluster counts using traditional cluster performance metrics. Silhouette Scores were higher and DBI values lower across all clustering algorithms when using the complete set. Further post-hoc testing, which measured the known fraud capture rates for each cluster configuration, indicated that the PCA feature-set generally performed better than the full feature-set at clustering known fraud instances. In contrast, analysis of DBSCAN noise indices showed that the full feature-set provided a better fraud capture rate when considering fraud as noise vs clustered groups.

Classification model performance varied between the PCA and full feature-set when comparing baseline model performance without cluster features included, as shown in Tables 33-34 below. While XGBoost, Random Forest, Logistic Regression, and Naïve Bayes performed better using the full feature-set, Neural Network performance was similar across both feature-sets and SVM notably performed better using the PCA feature-set.

**Table 33**   
*Baseline Classification Model Performance Metrics: Full Feature-set*

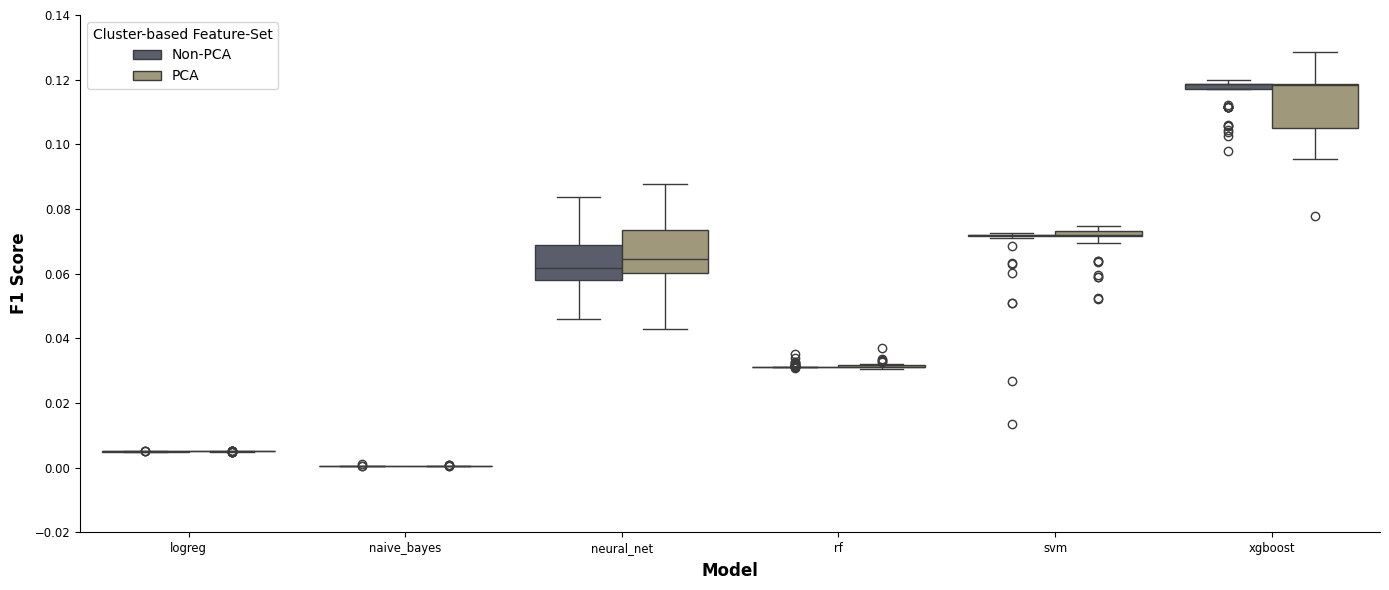
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Model** | **AUC ROC** | **Accuracy** | **Precision** | **Recall** | **F1 Score** |
| XGBoost | 0.901875806 | 0.999675796 | 0.353246606 | 0.057777778 | 0.099146008 |
| Random Forest | 0.901018035 | 0.996986488 | 0.023909098 | 0.213333333 | 0.042903947 |
| Logistic Regresion | 0.873006386 | 0.941156671 | 0.003573336 | 0.673333333 | 0.007108395 |
| Naïve Bayes | 0.85243926 | 0.882892916 | 0.001890581 | 0.711111111 | 0.003771019 |
| Nueral Net | 0.770827353 | 0.781653233 | 0.001213614 | 0.76 | 0.002422881 |
| SVM | 0.524480414 | 0.513412124 | 0.000340104 | 0.535555556 | 0.00067977 |

**Table 34**   
*Baseline Classification Model Performance Metrics: PCA Feature-set*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Model** | **AUC ROC** | **Accuracy** | **Precision** | **Recall** | **F1 Score** |
| SVM | 0.789307821 | 0.938523118 | 0.003303534 | 0.64 | 0.00657233 |
| Random Forest | 0.851268125 | 0.929710419 | 0.002800639 | 0.635555556 | 0.005576644 |
| XGBoost | 0.786013639 | 0.943501125 | 0.002710878 | 0.495555556 | 0.005392133 |
| Logistic Regression | 0.870741505 | 0.871238789 | 0.001686908 | 0.702222222 | 0.003365724 |
| Naïve Bayes | 0.867491829 | 0.840599811 | 0.00142367 | 0.733333333 | 0.00284182 |
| Neural Net | 0.839165819 | 0.804845848 | 0.001207513 | 0.735555556 | 0.0024109 |

In order to determine whether cluster-features derived from the PCA or full feature-set performed better, F1 scores were isolated for each model based on their feature-set derived group (e.g., PCA feature-set derived features vs full featured-set derived features). PCA-derived cluster-features generally performed better when included in the full feature-set while full feature-set derived cluster-features conversely performed generally better when used in the PCA feature-set for classification modelling.

**Figure 37***F1 Scores for PCA and Full Feature-set Derived Cluster Features, Full Feature Dataset*



**Figure 38***F1 Scores for PCA and Full Feature-set Derived Cluster Features, PCA Feature Dataset*

A diagram of a graph

AI-generated content may be incorrect.

**Conclusion**: Traditional cluster quality metrics indicate better clustering performance using the full feature-set in both K-Means and Hierarchical clustering algorithms, while fraud capture rates showed mixed results, with better performance in K-Means using the PCA feature-set and DBSCAN noise indices using the full feature-set. Several classification models performed better at baseline using the full feature-set compared to the PCA feature-set. Finally, cluster-features derived from the PCA feature-set generally performed better in classification models using the full feature-set while cluster-features derived from the PCA feature-set generally performed better when employed with classification models using the PCA feature-set. While these are mixed results across the different phases of the study, there is clear path showing that cluster-features derived from the PCA feature-set ultimately performed better in the model configurations that performed better (i.e., the full-feature-set based classification models). Therefore, the null hypothesis H10 is rejected, thus alternative hypothesis H1a is accepted.

### Research Question # 2

**RQ2**: What novel combination of existing unsupervised and supervised learning models can effectively identify fraudulent activity within the PPP?  
**H20**: All combinations of unsupervised and supervised models perform identically in detecting fraud: Model1 = Model2 = Modelk.  
**H2a**: Not all unsupervised learning models used in conjunction with supervised learning models perform identically in detecting fraudulent activity. At least two model combinations differ.

This research question evaluated the impact of combining cluster-derived features with supervised classifiers for fraud detection. Six classifiers (Logistic Regression, SVM, Random Forest, XGBoost, Neural Network, and Gaussian Naïve Bayes) were tested across 128 cluster feature configurations, plus a no-cluster baseline. Experimentation was conducted using both the PCA and full feature-set.In total, 7,100 experiments were run using stratified 70/30 train-test splits with SMOTE for balancing, repeated across five random seeds. Performance was evaluated using F1 Score, Precision, Recall, AUC-ROC, and Accuracy.

Two-way ANOVA (Tables 11-20 above) indicate that classification model selection had the most significant influence on performance metric variance in all cases (e.g., AUC ROC, accuracy, precision, recall, F1). The extreme variance due to model selection effectively muted any variance caused by cluster-feature selection when comparing inter-model metrics. However, traditional ANOVA performed within each classification model group (Tables 21-32 above) did identify significant variance in performance metrics based on cluster-configuration. Notably, Neural Network, Naïve Bayes, SVM, and Logistic Regression all showed significant variance in F1 scores based on cluster condition using the full feature-set. Similarly, XGBoost, Random Forest, and Naïve Bayes showed significant variance in F1 scores based on cluster condition. Using the full feature-set, XGBoost, SVM, and Neural Network models trained using cluster-featrures had higher F1 scores than their baseline models. Using the PCA feature-set, XGBoost, Random Forest, and Neural Network models trained using cluster-features had higher F1 scores than their baselines (Tables 35-36 below).

**Table 35***Cluster-Feature & Baseline Comparison, Full Feature-set*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Model** | **Cluster Configuration** | **AUC ROC** | **Accuracy** | **Precision** | **Recall** | **F1** |
| XGBoost | dbscan\_pca\_e0.3\_m7 | 0.87869 | 0.99965 | 0.27468 | 0.08444 | 0.12854 |
| XGBoost | Baseline | 0.90188 | 0.99968 | 0.35325 | 0.05778 | 0.09915 |
| SVM | dbscan\_pca\_e0.3\_m7 | 0.59274 | 0.99856 | 0.04678 | 0.18667 | 0.07471 |
| SVM | Baseline | 0.52448 | 0.51341 | 0.00034 | 0.53556 | 0.00068 |
| Neural Net | dbscan\_pca\_e1.0\_m3 | 0.8756 | 0.99946 | 0.09496 | 0.08444 | 0.08763 |
| Neural Net | Baseline | 0.77083 | 0.78165 | 0.00121 | 0.76 | 0.00242 |
| Random Forest | Baseline | 0.90102 | 0.99699 | 0.02391 | 0.21333 | 0.0429 |
| Random Forest | K-Means\_pca\_n10 | 0.90326 | 0.9944 | 0.01955 | 0.34 | 0.03692 |
| Naïve Bayes | Baseline | 0.85244 | 0.88289 | 0.00189 | 0.71111 | 0.00377 |
| Naïve Bayes | dbscan\_full\_e1.3\_m3 | 0.86004 | 0.52767 | 0.00058 | 0.88667 | 0.00116 |
| Logistic Regression | Baseline | 0.87301 | 0.94116 | 0.00357 | 0.67333 | 0.00711 |
| Logistic Regression | K-Means\_full\_n7 | 0.87382 | 0.9105 | 0.00256 | 0.74 | 0.0051 |

**Table 36***Cluster-Feature & Baseline Comparison, PCA Feature-set*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Model** | **Cluster Configuration** | **AUC ROC** | **Accuracy** | **Precision** | **Recall** | **F1** |
| XGBoost | K-Means\_full\_n10 | 0.79033 | 0.9654 | 0.00421 | 0.47111 | 0.00835 |
| XGBoost | Baseline | 0.78601 | 0.9435 | 0.00271 | 0.49556 | 0.00539 |
| SVM | K-Means\_pca\_n5 | 0.78932 | 0.93855 | 0.0033 | 0.64 | 0.00657 |
| SVM | Baseline | 0.78931 | 0.93852 | 0.0033 | 0.64 | 0.00657 |
| Random Forest | K-Means\_full\_n10 | 0.7806 | 0.95664 | 0.00374 | 0.52222 | 0.00744 |
| Random Forest | Baseline | 0.85127 | 0.92971 | 0.0028 | 0.63556 | 0.00558 |
| Neural Net | K-Means\_full\_n5 | 0.83809 | 0.89314 | 0.00247 | 0.66889 | 0.00491 |
| Neural Net | Baseline | 0.83917 | 0.80485 | 0.00121 | 0.73556 | 0.00241 |
| Naïve Bayes | dbscan\_pca\_e1.3\_m3 | 0.86747 | 0.84115 | 0.00143 | 0.73333 | 0.00285 |
| Naïve Bayes | Baseline | 0.86749 | 0.8406 | 0.00142 | 0.73333 | 0.00284 |
| Logistic Regression | K-Means\_full\_n8 | 0.87304 | 0.87343 | 0.00172 | 0.70222 | 0.00343 |
| Logistic Regression | Baseline | 0.87074 | 0.87124 | 0.00169 | 0.70222 | 0.00337 |

**Conclusion:** The observed differences in classification performance across model-cluster configurations clearly supports the alternative hypothesis H2a. While model selection alone had a significant effect on performance, further testing to isolate intra-model performance indicated that cluster-feature configuration does have a significant effect on model performance metric variance. Therefore, the null hypothesis H20 is rejected, thus alternative hypothesis H2a is accepted.

## Evaluation of the Findings

This section evaluates the performance outcomes and empirical observations resulting from the multi-phase classification through clustering experiments. The study's two primary research questions focused on identifying key features associated with fraudulent PPP loan applications and evaluating combinations of unsupervised and supervised machine learning techniques to improve fraud detection efficacy. Model performance was assessed using multiple evaluation metrics across clustering and classification phases, including Silhouette Score, DBI, accuracy, precision, recall, F1-score, and AUC-ROC.

### Impact of Dimensionality Reduction

Contrary to expectations, PCA did not improve traditional clustering performance across any of the tested algorithms. Silhouette and Davies-Bouldin metrics were consistently stronger when clustering was applied to the full, uncompressed feature space. Similarly, baseline classification model performance using both feature-sets showed better performance using the full feature-set. This finding underscores the importance of preserving feature dimensionality in fraud detection models where subtle variance and interaction between features may be critical to capturing anomalous behavior. PCA's tendency to discard lower-variance components appears to have negatively impacted cluster separability and interpretability in this context. However, downstream use of PCA-derived cluster-features outperformed both full feature-set-derived cluster-feature and the baseline model configurations in several instances in classification model configurations using the full feature-set.

Hypothesis H1 posited that certain features or combinations of features would yield better clustering and classification performance than the full feature set. While mixed results do not provide a clear streamline answer that PCA features always perform better in this domain, there is sufficient evidence to support the rejection of H10, that clustering and classification in the context of PPP loan fraud detection performs better using only the full feature-set.

### Evaluation of Clustering Methods

All three clustering algorithms were evaluated using standard unsupervised metrics. Both K-Means and Hierarchical clustering algorithm reported better silhouette and DBI scores when trained on the full feature-set compared to the PCA feature-set. Using both the PCA and full feature-sets, K-Means and Hierarchical traditional performance metrics steadily dropped as the number of clusters were increased. In contrast, fraud capture rates for K-Means and Hierarchical clustering were generally higher when trained using the PCA feature-set, notably so for K-Means in the 3-cluster configuration. These fraud capture results suggest that, while binary clustering is often desired in fraud detection (i.e., fraud vs not fraud), in an oversimplified model, due to either the feature-set (e.g., the PCA set) or fewer clusters in the model configuration, important insights from complex fraud schemes could easily be lost.

In several DBSCAN configurations, the set of loan applications categorized as noise exhibited a higher concentration of known fraudulent applications than any individual cluster. While DBSCAN noise index-based cluster features generally did not perform better than their traditional cluster-label based feature counterparts in downstream classification, this insight suggests further exploration into downstream applications of noise labels in government fraud detection is warranted.

### Model Performance Comparisons

In baseline and cluster-feature configurations, XGBoost outperformed other models when trained using the full feature-set. While there was visible variance in F1 scores for XGBoost, it failed the intra-model ANOVA for significance. Neural Network and SVM, both passing the F1 variance significance test, did also perform higher than other models (though not as well as XGBoost). In each of these three cases cluster-feature model configurations outperformed their baseline. While the identification of a preferred classification model to the novel domain of PPP loan fraud detection is both insightful and important, this study instead focuses on the effects that cluster-feature integration has on downstream classification. Thus, significant variance, as shown in cluster configurations in models such as Neural Network and SVM, is the key insight in model performance comparison.

The second hypothesis addressed the comparative performance of model configurations including baseline and cluster-features. The results clearly refute H20, as not all model pairings performed identically. In this case, both classification model selection and cluster0feature configuration had a significant effect on model performance. This supports H2a: at least two combinations of clustering and classification methods performed significantly differently.

## Limitations

This study was limited in scope to PPP loans exceeding $150,000, which improved reproducibility and analytic efficiency but excluded a large class of small-dollar fraud schemes. Review of PRAC reports revealed that many confirmed fraud cases involved individuals or entities applying for multiple loans under the $150K threshold; a behavior pattern that does not appear in the selected dataset. As a result, the study may underrepresent fraud typologies associated with “loan stacking” or lower value coordinated schemes.

Another key limitation was the availability of labeled fraud cases. All labeled data used for supervised evaluation was manually compiled from publicly available PRAC press releases and DOJ filings, yielding only 301 matched cases from approximately 2,500 reviewed documents. While this reflects a real-world constraint, it also reinforces the necessity of the semi-supervised approach: the study was designed to operate effectively in environments where high-quality labeled data is scarce or fragmented, as is typical in emerging fraud contexts like the PPP.

## Summary

This chapter presented the implementation and results of a semi-supervised machine learning framework for detecting potentially fraudulent PPP loan applications. The study analyzed a subset of 968,525 loans over $150,000 and incorporated 301 manually labeled fraud cases identified from over 2,500 PRAC and DOJ reports. Data preprocessing, exploratory analysis, and feature engineering informed the application of unsupervised clustering techniques, including K-Means, DBSCAN, and Hierarchical Clustering.

Cluster outputs were evaluated using internal validation metrics and used as features in downstream classification models. XGBoost consistently outperformed other classification models, though cluster-feature selection did not have a significant effect on its performance in the better performing full feature-set model configuration. In contrast, while their F1 scores underperformed compared to XGBoost, Neural Network and SVM both had significant variance in F1 score based on cluster-feature configuration. Despite limitations related to dataset scope and the scarcity of labelled fraud cases, this study demonstrated that semi-supervised methods hold potential for enhancing fraud detection in high-volume, sparsely labeled environments such as the PPP.

# Chapter 5: Implications, Recommendations, and Conclusions

This study addressed the lack of intelligent semi-supervised fraud identification in the government domain, with a specific focus on the Paycheck Protection Program (PPP). While existing literature on fraud detection in machine learning has largely focused on supervised methods in credit card and healthcare contexts (Ali et al., 2022; Minastireanu & Mesnita, 2019), there is a notable gap in research related to unsupervised and semi-supervised approaches for detecting fraud in public-sector programs. This gap is particularly significant in the PPP, a federally administered pandemic-era loan program that disbursed over five hundred billion dollars with limited oversight controls (Bailey et al., 2021; (Ma & McKinnon, 2020). Traditional supervised fraud detection methods rely heavily on labeled data to distinguish between legitimate and fraudulent transactions. However, in many real-world domains, especially those involving government programs, confirmed fraud labels are rare, incomplete, or delayed. This limitation introduces bias in model training and increases the risk of high false negative rates, where fraudulent activity goes undetected (Benala & Tantati, 2022). In the case of the PPP, for example, fraud labels rely on official court filings such as guilty pleas or jury convictions. These typically high-profile cases represent only a small subset of actual fraud occurrences, creating an unbalanced view of the full spectrum of fraudulent behavior within the program. Without more effective, scalable fraud detection methods, program misuse may persist unchecked, undermining public trust and jeopardizing future emergency funding initiatives.

The purpose of this study was to develop and evaluate a semi-supervised learning methodology for identifying potentially fraudulent PPP loan records. Unlike previous studies that relied on proprietary financial datasets and institution-specific risk scoring systems, this study employed a Classification Through Clustering approach using only publicly available PPP loan data and limited court filings for labelling. Rather than depending on fully labeled datasets, the research introduced unsupervised clustering to extract structural patterns from the data, which were then incorporated into a supervised learning pipeline. This approach was designed to evaluate whether meaningful fraud detection could still be achieved in contexts where verified fraud labels are limited or incomplete in emerging fraud domains.

To test this approach, a subset of PPP loans exceeding $150,000 (*n =* 968,525) was analyzed, including 301 manually labeled fraud cases identified from federal court records (e.g., PRAC press releases to include guilty pleas and jury convictions). The research process involved extensive data preprocessing, feature engineering, and implementation of three clustering algorithms: K-Means, DBSCAN, and Hierarchical clustering. Clustering outputs were evaluated using internal validation metrics & fraud capture metrics and then encoded as features in multiple classification models. The results indicated that, while the XGBoost model consistently outperformed other supervised models, cluster feature configuration did have a significant effect on model performance in several models, notably Neural Network and SVM.

Although the study was constrained by the scope of the dataset, limited label availability, and reliance on publicly sourced data, the findings demonstrate that semi-supervised learning techniques can potentially support fraud detection in high-volume, low-label environments like the PPP. The remainder of this chapter discusses the study findings’ implications for fraud analytics and methodological research within the PPP and government fraud domain and offers recommendations for practical application and future study.

## Implications

### Research Question # 1

**RQ1**: What are the key features or variables associated with fraudulent loan applications within the PPP?  
**H10**: PPP loan applications are best clustered and further classified given the complete list of values of each feature or variable in the dataset.  
**H1a**: PPP loan applications are best clustered and further classified given the values of specific features or variables in the dataset.

The first research question and its hypotheses were posed to examine whether specific key features or feature-sets in the publicly available PPP loan applications would enhance clustering and fraud classification. For the K-Means and Hierarchical clustering algorithms, the full feature-set outperformed the PCA feature-set in traditional metrics (DBI, Silhouette score) while the PCA feature-set performed better in fraud capture rates. Traditional metrics peaked for each algorithm in low cluster configurations (e.g., 2 clusters) and steadily declined as the number of clusters increased. Conversely, fraud capture rates were markedly higher for K-Means using the PCA feature-set, with peak capture occurring at *k =* 3. DBSCAN, however, achieved consistently high fraud capture rates when considering fraud as noise. While binary clustering or classification is traditionally ideal in fraud detection scenarios, these results imply that oversimplification of cluster space (i.e., only two clusters) reduces fraud capture effectiveness. Oversimplified clustering configurations could potentially be blind to more complex frauds schemes like those found in the government fraud domain.

Downstream classification also examined whether the full or PCA feature-set were more effective at fraud detection. In contrast with the clustering phase findings, the full feature-set consistently outperformed the PCA feature-set in baseline and cluster-feature configurations. However, PCA-derived cluster-features generally outperformed full feature-set-derived cluster features when used with the full feature-set in model configurations. This implies that classification models, when used in the PPP loan fraud domain, generally perform better given complex feature-sets where PCA-derived cluster-features are included as part of those respective complex datasets.

The methodology used in this study is based on the framework presented by López et al. (2012) where the authors proposed their classification through clustering framework to perform binary classification of student pass/fail rates. In their study, the authors advise that cluster configurations be capped at *k* = 2 to match the desired binary classification. In contrast, this study found that complex fraud scenarios require a more complex cluster space, and that oversimplification of the clustering process led to underperformance in downstream classification. These particular insights are crucial to the application of the classification through clustering framework to the government fraud domain.

### Research Question # 2

**RQ2**: What novel combination of existing unsupervised and supervised learning models can effectively identify fraudulent activity within the PPP?  
**H20**: All combinations of unsupervised and supervised models perform identically in detecting fraud: Model1 = Model2 = Modelk.  
**H2a**: Not all unsupervised learning models used in conjunction with supervised learning models perform identically in detecting fraudulent activity. At least two model combinations differ.

The second research question and its hypotheses were posed to examine whether specific combinations of unsupervised and supervised learning models applied in the classification through clustering framework, as outlined by López et al. (2012), were more effective at identifying fraudulent PPP loan applications. While there is extensive literature on both supervised and unsupervised machine learning model performance comparison, research on fraud detection is often limited to supervised machine learning in the credit card or healthcare domains (A. Ali et al., 2022; Dridi, 2022a). This study expands on the existing literature by examining semi supervised learning (classification via clustering) within the government fraud domain.

Using the F1 score as a comparison metric, there was significant variance between supervised learning model performance, in this case the XGBoost model trained using the full feature-set outperformed other models. Since machine learning models perform differently due to their inherent biases, identification of domain specific “preferred-models” (in this case PPP loan fraud) is in and of itself a useful insight (Goldblum et al., 2024; Shalev-Shwartz & Ben-David, 2014). Additional intra-model ANOVA testing, however, also identified significant variance in classification model performance due to introduced cluster-feature configurations. These findings demonstrate that the clustering via classification framework, presented in a novel methodology in this study, can increase fraud detection via improved classification model performance within the PPP loan fraud domain. This is of particular importance as the label space in bourgeoning domains like government fraud detection will continue to be much lower than that of more established domains such as credit card fraud detection.

## Recommendations for Practice

The findings of this study provide several insights that can be used in the practical application of the classification through clustering framework within the government fraud domain. First, this study demonstrated a novel methodology for fraud detection in an underrepresented domain (Ali et al., 2021). The methodology presented in this study supports the use of publicly available data for fraud detection and government oversight by both government and non-governmental entities. For example, public interest groups, journalists, or academic researchers could use similar methodologies to develop transparent fraud detection and oversight tools even in low label domains.

Second, practitioners should consider PCA not only as a preprocessing step but as a tunable component in the clustering phase. This study showed that classification models using the full feature-set consistently outperformed their respective models trained using PCA features, in alignment with the findings in López et al., (2012). However, PCA-derived cluster features generally outperformed their full feature-set counterparts in classification models, though not in every case. Therefore, evaluation of both PCA and full-feature set based cluster-features should be included in future applications of this methodology.

## Recommendations for Future Research

This study expands the research and application of semi-supervised machine learning in the underrepresented government fraud domain, specifically within the PPP. Building on this foundation, this section provides several recommendations for researchers to validate, refine, and extend the findings and methodologies presented in this study. First, further research is needed into cluster feature integration. Researchers should consider expanding the scope of unsupervised clustering methodologies as well as introducing inter cluster-feature combinations (e.g., K-Means derived cluster features + DBSCAN derived cluster features). While beyond the scope of the original experimental design, an expanded model scope will likely enhance the understanding of cluster feature influence on classification within the government fraud domain.

Second, future research should expand the scope of the analysis to include PPP loan applications under the $150,000 level. While the scope of this study was purposefully limited to loans exceeding this threshold in order to simplify reproducibility (the SBA consolidated all loans over $150,000 into a single csv), manual review of PRAC press releases highlighted a common pattern of multiple lower dollar fraudulent loan applications by a single entity or organization. Increasing the scope to include these lower dollar amount applications could increase supervised model performance simply by increasing the amount of labeled training data. However, it could also introduce new fraudulent activity patterns that potentially could be exposed in the clustering phase.

Third, government oversight organizations could employ the methodologies shown in this study leveraging non-public datasets. While this study was designed to rely on only publicly available information, government entities such as Inspectors General have access to non-public data such as complete applications with personally identifiable information (PII) or even IP addresses for loan submissions. The introduction of an expanded feature set could have a significant impact on model performance. However, organizations applying this study’s methodology using an expanded feature set should be careful to compare the performance between the full feature set and the PCA feature set in both the clustering and classification phases. As noted in this study and López et al., (2012), the full feature set unexpectedly performed better than the PCA feature set in the classification through clustering framework.

Next, future researchers should explore methods to automate or streamline fraud labelling. Fraud labelling in this study was conducted by manually reviewing press releases from the PRAC. While a simple web scraper using python enabled quick aggregation of the data, it still needed to be manually reviewed and linked to loan applications. Developing a tool using a natural language processor (NLP) to link press releases to loan applications would make fraud labelling much more efficient, especially in an emerging fraud environment like the PPP.

Finally, researchers should explore the portability of this study’s methodology in other government fraud domains. There are numerous government programs with both public facing and non-public data on applicants. Researchers should explore whether this studies methodology can be adapted to other programs and then compare performance across programs to determine whether it is portable. Researchers should also explore the use of transfer learning, which could further increase fraud detection capabilities in emerging domains that lack labelled datasets.

## Conclusions

The objective of this study was to address the lack of intelligent semi-supervised fraud identification methodologies in the government fraud domain, particularly within the PPP. This study sought to address domain specific gaps in the literature to include underrepresentation of government fraud detection in unsupervised learning and supervised learning, as well as the combination of the two in a semi-supervised classification through clustering framework within the government fraud domain.

This study focused on developing a novel and reproducible methodology that could be used to investigate fraud in emerging domains where labelled data is sparse or entirely unavailable for supervised machine learning model training. Using a classification through clustering framework, unlabeled PPP loan applications were clustered using the DBSCAN, hierarchical, and K-Means algorithms. The cluster labels and noise labels in the case of DBSCAN were integrated into the dataset which then underwent supervised learning training using several models. While the XGBoost model outperformed other models using the full feature-set, cluster-feature configuration significantly impacted classification model performance in several PCA and full feature-set configurations, notable full feature-set trained SVM and Neural Network.

This study is important as it extends the literature and research in the fields of government fraud and semi-supervised learning by presenting a reproducible pipeline that can be adapted across various public-sector programs. By implementing a structured semi-supervised approach, the study demonstrates how latent structure and meaningful features can be extracted from complex datasets for use in downstream classification tasks. As government initiatives continue to scale in both scope and complexity, the need for robust fraud detection methodologies that can operate under label scarcity becomes increasingly critical. This research provides a foundational framework for addressing that need, advancing methodological development in the detection of fraud in high-volume, low-label environments.

# References

*18 USC 1001: Statements or entries generally*. (2004). https://uscode.house.gov/view.xhtml?req=granuleid:USC-prelim-title18-section1001&num=0&edition=prelim

*18 USC 1343: Fraud by wire, radio, or television*. (2008). https://uscode.house.gov/view.xhtml?req=granuleid:USC-prelim-title18-section1343&num=0&edition=prelim

Ali, A., Abd Razak, S., Othman, S. H., Eisa, T. A. E., Al-Dhaqm, A., Nasser, M., Elhassan, T., Elshafie, H., & Saif, A. (2022). Financial Fraud Detection Based on Machine Learning: A Systematic Literature Review. *Applied Sciences*, *12*(19), Article 19. https://doi.org/10.3390/app12199637

Ali, N. A. M., Abu, N. ’Asyiqin, Hussain, W. S., Nordin, E., & Ramlan, N. L. (2021). Critical Success Factors For Financial Fraud Management In Government Agencies. *Turkish Online Journal of Qualitative Inquiry*, *12*(7), 4325–4340.

Ashtiani, M. N., & Raahemi, B. (2022). Intelligent Fraud Detection in Financial Statements Using Machine Learning and Data Mining: A Systematic Literature Review. *IEEE Access*, *10*, 72504–72525. IEEE Access. https://doi.org/10.1109/ACCESS.2021.3096799

Autor, D., Cho, D., Crane, L. D., Goldar, M., Lutz, B., Montes, J., Peterman, W. B., Ratner, D., Villar, D., & Yildirmaz, A. (2022). The $800 Billion Paycheck Protection Program: Where Did the Money Go and Why Did It Go There? *Journal of Economic Perspectives*, *36*(2), 55–80. https://doi.org/10.1257/jep.36.2.55

Awang, N., Hussin, N. S., Razali, F. A., Lyana, S., & Talib, A. (2020). Fraud triangle theory: Calling for new factors. *Board*, *7*, 54–64.

Bailey, C., Brody, R., & Sokolowski, M. (2021). Fraudulent loans and the United States paycheck protection program. *Journal of Financial Crime*, *29*(2), 519–532. https://doi.org/10.1108/JFC-07-2021-0165

Barroga, E., & Matanguihan, G. J. (2022). A Practical Guide to Writing Quantitative and Qualitative Research Questions and Hypotheses in Scholarly Articles. *Journal of Korean Medical Science*, *37*(16), e121. https://doi.org/10.3346/jkms.2022.37.e121

Bauder, R. A., & Khoshgoftaar, T. M. (2017). Medicare Fraud Detection Using Machine Learning Methods. *2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA)*, 858–865. https://doi.org/10.1109/ICMLA.2017.00-48

Benala, T. R., & Tantati, K. (2022). Efficiency of oversampling methods for enhancing software defect prediction by using imbalanced data. *Innovations in Systems and Software Engineering*. https://doi.org/10.1007/s11334-022-00457-3

Bozza, J. G. (2024). Stimulating Fraud: Comparing the Effectiveness of Fraud Recovery Mechanisms Between the United States and the United Kingdom Through the Lens of Public Covid-19 Expenditures. *Georgia Journal of International & Comparative Law*, *52*(2), 462–478.

Bureau of Economic Analysis. (2021). *How does the Paycheck Protection Program impact the national income and product accounts (NIPAs)?* Bureau of Economic Analysis. https://www.bea.gov/help/faq/1408

Carcillo, F., Le Borgne, Y.-A., Caelen, O., Kessaci, Y., Oblé, F., & Bontempi, G. (2021). Combining unsupervised and supervised learning in credit card fraud detection. *Information Sciences*, *557*, 317–331. https://doi.org/10.1016/j.ins.2019.05.042

CDC. (2020, March 28). *COVID Data Tracker*. Centers for Disease Control and Prevention. https://covid.cdc.gov/covid-data-tracker

Cervantes, J., Garcia-Lamont, F., Rodríguez-Mazahua, L., & Lopez, A. (2020). A comprehensive survey on support vector machine classification: Applications, challenges and trends. *Neurocomputing*, *408*, 189–215. https://doi.org/10.1016/j.neucom.2019.10.118

Council of the Inspectors General on Integrity and Efficiency. (2011). *QUALITY STANDARDS FOR INVESTIGATIONS*.

Crowe. (n.d.). *A GUIDE TO Using Analytics to Control PPP Loan Risks*. https://www.crowe.com/-/media/Crowe/LLP/folio-pdf/Using-Analytics-to-Control-PPP-Loan-Risks\_CMC2100-005B.pdf

Debener, J., Heinke, V., & Kriebel, J. (2023). Detecting insurance fraud using supervised and unsupervised machine learning. *Journal of Risk & Insurance*, *90*(3), 743–768. https://doi.org/10.1111/jori.12427

Demko, Z. O., Antar, A. A. R., Blair, P. W., Lambrou, A. S., Yu, T., Brown, D., Walch, S. N., Armstrong, D. T., Mostafa, H. H., Keruly, J. C., Thomas, D. L., Manabe, Y. C., Mehta, S. H., & Ambulatory COVID Study Team. (2021). Clustering of SARS-CoV-2 Infections in Households of Patients Diagnosed in the Outpatient Setting in Baltimore, Maryland. *Open Forum Infectious Diseases*, *8*(4), ofab121. https://doi.org/10.1093/ofid/ofab121

Department Of Treasury. (2024, March 19). *About the CARES Act and the Consolidated Appropriations Act*. U.S. Department of the Treasury. https://home.treasury.gov/policy-issues/coronavirus/about-the-cares-act

Dridi, S. (2022a). *Supervised Learning—A Systematic Literature Review*. OSF Preprints. https://doi.org/10.31219/osf.io/tysr4

Dridi, S. (2022b). *Unsupervised Learning—A Systematic Literature Review*. OSF Preprints. https://doi.org/10.31219/osf.io/kpqr6

Emilio Ferrara. (2023). Fairness and Bias in Artificial Intelligence: A Brief Survey of Sources, Impacts, and Mitigation Strategies. *Sci*, *6*(1), 3–3. https://doi.org/10.3390/sci6010003

Emmons, W. R., & Dahl, D. (2022). *Was the Paycheck Protection Program Effective?* Federal Reserve Bank of St Louis. https://www.stlouisfed.org/publications/regional-economist/2022/jul/was-paycheck-protection-program-effective

Giupponi, G., Landais, C., & Lapeyre, A. (2022). Should We Insure Workers or Jobs During Recessions? *Journal of Economic Perspectives*, *36*(2), 29–54. https://doi.org/10.1257/jep.36.2.29

Glass, G. V., Peckham, P. D., & Sanders, J. R. (1972). Consequences of Failure to Meet Assumptions Underlying the Fixed Effects Analyses of Variance and Covariance. *Review of Educational Research*, *42*(3), 237–288. https://doi.org/10.3102/00346543042003237

Goldblum, M., Finzi, M., Rowan, K., & Wilson, A. G. (2024). *The No Free Lunch Theorem, Kolmogorov Complexity, and the Role of Inductive Biases in Machine Learning* (No. arXiv:2304.05366). arXiv. https://doi.org/10.48550/arXiv.2304.05366

Gorle, V. L. N., & Panigrahi, S. (2024). A semi-supervised Anti-Fraud model based on integrated XGBoost and BiGRU with self-attention network: An application to internet loan fraud detection. *Multimedia Tools and Applications*, *83*(19), 56939–56964. https://doi.org/10.1007/s11042-023-17681-z

Gui, Q., Zhou, H., Guo, N., & Niu, B. (2024). A survey of class-imbalanced semi-supervised learning. *Machine Learning*, *113*(8), 5057–5086. https://doi.org/10.1007/s10994-023-06344-7

Halkidi, M. (2009). Hierarchial Clustering. In L. LIU & M. T. ÖZSU (Eds.), *Encyclopedia of Database Systems* (pp. 1291–1294). Springer US. https://doi.org/10.1007/978-0-387-39940-9\_604

Harrington, B., & Leslie, C. A. (2023). Toward a Multilevel Sociology of Fraud. *Northwestern University Law Review*, *118*(1), 139–166.

Humphries, J. E., Neilson, C. A., & Ulyssea, G. (2020). Information frictions and access to the Paycheck Protection Program. *Journal of Public Economics*, *190*, 104244. https://doi.org/10.1016/j.jpubeco.2020.104244

Itri, B., Mohamed, Y., Mohammed, Q., & Omar, B. (2019). Performance comparative study of machine learning algorithms for automobile insurance fraud detection. *2019 Third International Conference on Intelligent Computing in Data Sciences (ICDS)*, 1–4. https://doi.org/10.1109/ICDS47004.2019.8942277

Kadens, E. (2023). The Persistent Limits of Fraud Prevention in Historical Perspective. *Northwestern University Law Review*, *118*(1), 167–192.

King, D. L., McLanahan, M. L., & Case, C. J. (2023). Frauds Do Not Pause for a Pandemic. *Journal of Business & Educational Leadership*, *13*(1), 67–81.

Koreff, J., Baudot, L., & Sutton, S. G. (2023). Exploring the Impact of Technology Dominance on Audit Professionalism through Data Analytic-Driven Healthcare Audits. *Journal of Information Systems*, *37*(3), 59–80. https://doi.org/10.2308/ISYS-2022-023

Larson, Benjamin James. (2020). *False Positive Reduction in Credit Card Fraud Prediction: An Evaluation of Machine Learning Methodology on Imbalanced Data*. https://www.proquest.com/openview/197642c9490f064f94d434ff0ed8f9e7/1?pq-origsite=gscholar&cbl=18750&diss=y

Li, L., & Strahan, P. (2020). *Who Supplies PPP Loans (And Does it Matter)? Banks, Relationships and the COVID Crisis* (No. w28286; p. w28286). National Bureau of Economic Research. https://doi.org/10.3386/w28286

Lix, L. M., Keselman, J. C., & Keselman, H. J. (1996). Consequences of Assumption Violations Revisited: A Quantitative Review of Alternatives to the One-Way Analysis of Variance “F” Test. *Review of Educational Research*, *66*(4), 579. https://doi.org/10.2307/1170654

López, M. I., Luna, J. M., Romero, C., & Ventura, S. (2012). Classification via clustering for predicting final marks based on student participation in forums. *Proceedings of the 5th International Conference on Educational Data Mining*.

Ma, K. W. F., & McKinnon, T. (2020). *COVID-19 and Cyber Fraud: Emerging Threats During the Pandemic* (SSRN Scholarly Paper No. 3718845). https://doi.org/10.2139/ssrn.3718845

Maalouf, M. (2011). Logistic regression in data analysis: An overview. *International Journal of Data Analysis Techniques and Strategies*, *3*, 281–299. https://doi.org/10.1504/IJDATS.2011.041335

Miller, K. S., & Bertozzi, A. L. (2024). Model Change Active Learning in Graph-Based Semi-supervised Learning. *Communications on Applied Mathematics and Computation*, *6*(2), 1270–1298. https://doi.org/10.1007/s42967-023-00328-z

Minastireanu, E., & Mesnita, G. (2019). *An Analysis of the Most Used Machine Learning Algorithms for Online Fraud Detection*. *Informatica Economica, 23*, 5–16. https://doi.org/10.12948/issn14531305/23.1.2019.01

Nassif, A. B., Talib, M. A., Nasir, Q., & Dakalbab, F. M. (2021). Machine Learning for Anomaly Detection: A Systematic Review. *IEEE Access*, *9*, 78658–78700. IEEE Access. https://doi.org/10.1109/ACCESS.2021.3083060

Rixom, B. A., Rixom, J. M., Pippin, S., & Wong, J. (2021). Contrasting Public Perceptions of Government versus Certified Public Accounting Firm Oversight of Relief Packages. *Accounting & the Public Interest*, *21*(1), 39–63. https://doi.org/10.2308/API-2020-019

Rodriguez, M. Z., Comin, C. H., Casanova, D., Bruno, O. M., Amancio, D. R., Costa, L. da F., & Rodrigues, F. A. (2019). Clustering algorithms: A comparative approach. *PLoS ONE*, *14*(1), e0210236. https://doi.org/10.1371/journal.pone.0210236

Sabasteanski, N., Brooks, J., & Chandler, T. (2021). Saving lives and livelihoods: The Paycheck Protection Program and its efficacy. *EconomiA*, *22*(3), 278–290. https://doi.org/10.1016/j.econ.2021.11.004

Sarker, I. H. (2021). Machine Learning: Algorithms, Real-World Applications and Research Directions. *SN Computer Science*, *2*(3), 160. https://doi.org/10.1007/s42979-021-00592-x

Schonlau, M., & Zou, R. Y. (2020). The random forest algorithm for statistical learning. *The Stata Journal*, *20*(1), 3–29. https://doi.org/10.1177/1536867X20909688

Shalev-Shwartz, S., & Ben-David, S. (2014). *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press. https://doi.org/10.1017/CBO9781107298019

Stoner, J. L., Felix, R., & Stadler Blank, A. (2023). Best practices for implementing experimental research methods. *International Journal of Consumer Studies*, *47*(4), 1579–1595. https://doi.org/10.1111/ijcs.12878

The National Law Review. (2023). *The DOJ is Continuing to Target PPP Recipients for Fraud*. The National Law Review. https://www.natlawreview.com/article/doj-continuing-to-target-ppp-recipients-fraud

USGAO. (2020). *Telecommunications: FCC Should Take Action to Better Manage Persistent Fraud Risks in the Schools and Libraries Program | U.S. GAO*. https://www.gao.gov/products/gao-20-606

USSBA. (2021). *Process Flow Diagram*. US SBA PPP Loan API. https://ussbappp.github.io/usecase/usecase-process-flow.html

USSBA. (2023). *PPP FOIA - U.S. Small Business Administration (SBA) | Open Data*. https://data.sba.gov/dataset/ppp-foia

USSBA OIG. (2023). *COVID-19 Pandemic EIDL and PPP Loan Fraud Landscape* (Nos. 23–09). USSBA. https://www.sba.gov/document/report-23-09-covid-19-pandemic-eidl-ppp-loan-fraud-landscape

West, D. M. (2021). *Using AI and machine learning to reduce government fraud*. https://policycommons.net/artifacts/4142718/using-ai-and-machine-learning-to-reduce-government-fraud/4952011/

Xu, J. J., Chen, D., Chau, M., Li, L., & Zheng, H. (2022). Peer-to-peer loan fraud detection: Constructing features from transaction data. *MIS Quarterly*, *46*(3).

Zhang, X., Shen, X., & Ouyang, T. (2022). Extension of DBSCAN in Online Clustering: An Approach Based on Three-Layer Granular Models. *Applied Sciences*, *12*(19), Article 19. https://doi.org/10.3390/app12199402

Zhao, G., Li, G., Qin, Y., Zhang, J., Chai, Z., Wei, X., Lin, L., & Yu, Y. (2024). Exploration and Exploitation of Unlabeled Data for Open-Set Semi-supervised Learning. *International Journal of Computer Vision*. https://doi.org/10.1007/s11263-024-02155-y

Zhou, N., Zhang, Z., Nair, V. N., Singhal, H., & Chen, J. (2022). Bias, Fairness and Accountability with Artificial Intelligence and Machine Learning Algorithms. *International Statistical Review*, *90*(3), 468–480. https://doi.org/10.1111/insr.12492

# Appendix A IRB Approval Letter

A screenshot of a computer

AI-generated content may be incorrect.

# Appendix B Aggregated Classification Model Results: Full Feature-set

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Model** | **Cluster Condition** | **AUC ROC** | **Accuracy** | **Precision** | **Recall** | **F1** |
| XGBoost | Baseline | 0.901876 | 0.999676 | 0.353247 | 0.05778 | 0.099146 |
| XGBoost | dbscan\_full\_e0.3\_m10\_labels | 0.892987 | 0.999641 | 0.256831 | 0.07778 | 0.118542 |
| XGBoost | dbscan\_full\_e0.3\_m10\_noise | 0.892987 | 0.999641 | 0.256831 | 0.07778 | 0.118542 |
| XGBoost | dbscan\_full\_e0.3\_m15\_labels | 0.892987 | 0.999641 | 0.256831 | 0.07778 | 0.118542 |
| XGBoost | dbscan\_full\_e0.3\_m15\_noise | 0.892987 | 0.999641 | 0.256831 | 0.07778 | 0.118542 |
| XGBoost | dbscan\_full\_e0.3\_m3\_labels | 0.892987 | 0.999641 | 0.256831 | 0.07778 | 0.118542 |
| XGBoost | dbscan\_full\_e0.3\_m3\_noise | 0.892987 | 0.999641 | 0.256831 | 0.07778 | 0.118542 |
| XGBoost | dbscan\_full\_e0.3\_m5\_labels | 0.892987 | 0.999641 | 0.256831 | 0.07778 | 0.118542 |
| XGBoost | dbscan\_full\_e0.3\_m5\_noise | 0.892987 | 0.999641 | 0.256831 | 0.07778 | 0.118542 |
| XGBoost | dbscan\_full\_e0.3\_m7\_labels | 0.892987 | 0.999641 | 0.256831 | 0.07778 | 0.118542 |
| XGBoost | dbscan\_full\_e0.3\_m7\_noise | 0.892987 | 0.999641 | 0.256831 | 0.07778 | 0.118542 |
| XGBoost | dbscan\_full\_e0.5\_m10\_labels | 0.892987 | 0.999641 | 0.256831 | 0.07778 | 0.118542 |
| XGBoost | dbscan\_full\_e0.5\_m10\_noise | 0.892987 | 0.999641 | 0.256831 | 0.07778 | 0.118542 |
| XGBoost | dbscan\_full\_e0.5\_m15\_labels | 0.892987 | 0.999641 | 0.256831 | 0.07778 | 0.118542 |
| XGBoost | dbscan\_full\_e0.5\_m15\_noise | 0.892987 | 0.999641 | 0.256831 | 0.07778 | 0.118542 |
| XGBoost | dbscan\_full\_e0.5\_m3\_labels | 0.892987 | 0.999641 | 0.256831 | 0.07778 | 0.118542 |
| XGBoost | dbscan\_full\_e0.5\_m3\_noise | 0.892987 | 0.999641 | 0.256831 | 0.07778 | 0.118542 |
| XGBoost | dbscan\_full\_e0.5\_m5\_labels | 0.892987 | 0.999641 | 0.256831 | 0.07778 | 0.118542 |
| XGBoost | dbscan\_full\_e0.5\_m5\_noise | 0.892987 | 0.999641 | 0.256831 | 0.07778 | 0.118542 |
| XGBoost | dbscan\_full\_e0.5\_m7\_labels | 0.892987 | 0.999641 | 0.256831 | 0.07778 | 0.118542 |
| XGBoost | dbscan\_full\_e0.5\_m7\_noise | 0.892987 | 0.999641 | 0.256831 | 0.07778 | 0.118542 |
| XGBoost | dbscan\_full\_e0.7\_m10\_labels | 0.892987 | 0.999641 | 0.256831 | 0.07778 | 0.118542 |
| XGBoost | dbscan\_full\_e0.7\_m10\_noise | 0.892987 | 0.999641 | 0.256831 | 0.07778 | 0.118542 |
| XGBoost | dbscan\_full\_e0.7\_m15\_labels | 0.892987 | 0.999641 | 0.256831 | 0.07778 | 0.118542 |
| XGBoost | dbscan\_full\_e0.7\_m15\_noise | 0.892987 | 0.999641 | 0.256831 | 0.07778 | 0.118542 |
| XGBoost | dbscan\_full\_e0.7\_m3\_labels | 0.891028 | 0.999638 | 0.240575 | 0.07333 | 0.111534 |
| XGBoost | dbscan\_full\_e0.7\_m3\_noise | 0.891028 | 0.999638 | 0.240575 | 0.07333 | 0.111534 |
| XGBoost | dbscan\_full\_e0.7\_m5\_labels | 0.891028 | 0.999638 | 0.240575 | 0.07333 | 0.111534 |
| XGBoost | dbscan\_full\_e0.7\_m5\_noise | 0.891028 | 0.999638 | 0.240575 | 0.07333 | 0.111534 |
| XGBoost | dbscan\_full\_e0.7\_m7\_labels | 0.892987 | 0.999641 | 0.256831 | 0.07778 | 0.118542 |
| XGBoost | dbscan\_full\_e0.7\_m7\_noise | 0.892987 | 0.999641 | 0.256831 | 0.07778 | 0.118542 |
| XGBoost | dbscan\_full\_e1.0\_m10\_labels | 0.892987 | 0.999641 | 0.256831 | 0.07778 | 0.118542 |
| XGBoost | dbscan\_full\_e1.0\_m10\_noise | 0.892987 | 0.999641 | 0.256831 | 0.07778 | 0.118542 |
| XGBoost | dbscan\_full\_e1.0\_m15\_labels | 0.892987 | 0.999641 | 0.256831 | 0.07778 | 0.118542 |
| XGBoost | dbscan\_full\_e1.0\_m15\_noise | 0.892987 | 0.999641 | 0.256831 | 0.07778 | 0.118542 |
| XGBoost | dbscan\_full\_e1.0\_m3\_labels | 0.890078 | 0.999639 | 0.234476 | 0.06889 | 0.105817 |
| XGBoost | dbscan\_full\_e1.0\_m3\_noise | 0.890078 | 0.999639 | 0.234476 | 0.06889 | 0.105817 |
| XGBoost | dbscan\_full\_e1.0\_m5\_labels | 0.891028 | 0.999638 | 0.240575 | 0.07333 | 0.111534 |
| XGBoost | dbscan\_full\_e1.0\_m5\_noise | 0.891028 | 0.999638 | 0.240575 | 0.07333 | 0.111534 |
| XGBoost | dbscan\_full\_e1.0\_m7\_labels | 0.892987 | 0.999641 | 0.256831 | 0.07778 | 0.118542 |
| XGBoost | dbscan\_full\_e1.0\_m7\_noise | 0.892987 | 0.999641 | 0.256831 | 0.07778 | 0.118542 |
| XGBoost | dbscan\_full\_e1.3\_m10\_labels | 0.892987 | 0.999641 | 0.256831 | 0.07778 | 0.118542 |
| XGBoost | dbscan\_full\_e1.3\_m10\_noise | 0.892987 | 0.999641 | 0.256831 | 0.07778 | 0.118542 |
| XGBoost | dbscan\_full\_e1.3\_m15\_labels | 0.892987 | 0.999641 | 0.256831 | 0.07778 | 0.118542 |
| XGBoost | dbscan\_full\_e1.3\_m15\_noise | 0.892987 | 0.999641 | 0.256831 | 0.07778 | 0.118542 |
| XGBoost | dbscan\_full\_e1.3\_m3\_labels | 0.893475 | 0.999635 | 0.241201 | 0.08 | 0.119942 |
| XGBoost | dbscan\_full\_e1.3\_m3\_noise | 0.884934 | 0.999641 | 0.2406 | 0.06889 | 0.105929 |
| XGBoost | dbscan\_full\_e1.3\_m5\_labels | 0.891028 | 0.999638 | 0.240575 | 0.07333 | 0.111534 |
| XGBoost | dbscan\_full\_e1.3\_m5\_noise | 0.891028 | 0.999638 | 0.240575 | 0.07333 | 0.111534 |
| XGBoost | dbscan\_full\_e1.3\_m7\_labels | 0.892987 | 0.999641 | 0.256831 | 0.07778 | 0.118542 |
| XGBoost | dbscan\_full\_e1.3\_m7\_noise | 0.892987 | 0.999641 | 0.256831 | 0.07778 | 0.118542 |
| XGBoost | dbscan\_pca\_e0.3\_m10\_labels | 0.892987 | 0.999641 | 0.256831 | 0.07778 | 0.118542 |
| XGBoost | dbscan\_pca\_e0.3\_m10\_noise | 0.892987 | 0.999641 | 0.256831 | 0.07778 | 0.118542 |
| XGBoost | dbscan\_pca\_e0.3\_m15\_labels | 0.892987 | 0.999641 | 0.256831 | 0.07778 | 0.118542 |
| XGBoost | dbscan\_pca\_e0.3\_m15\_noise | 0.892987 | 0.999641 | 0.256831 | 0.07778 | 0.118542 |
| XGBoost | dbscan\_pca\_e0.3\_m3\_labels | 0.890463 | 0.99965 | 0.268 | 0.07556 | 0.117871 |
| XGBoost | dbscan\_pca\_e0.3\_m3\_noise | 0.893399 | 0.999648 | 0.242287 | 0.06444 | 0.101414 |
| XGBoost | dbscan\_pca\_e0.3\_m5\_labels | 0.889009 | 0.999646 | 0.247894 | 0.06889 | 0.107474 |
| XGBoost | dbscan\_pca\_e0.3\_m5\_noise | 0.89321 | 0.999634 | 0.223123 | 0.06667 | 0.101831 |
| XGBoost | dbscan\_pca\_e0.3\_m7\_labels | 0.878693 | 0.999647 | 0.274685 | 0.08444 | 0.128538 |
| XGBoost | dbscan\_pca\_e0.3\_m7\_noise | 0.893026 | 0.999634 | 0.219453 | 0.06667 | 0.101921 |
| XGBoost | dbscan\_pca\_e0.5\_m10\_labels | 0.892987 | 0.999641 | 0.256831 | 0.07778 | 0.118542 |
| XGBoost | dbscan\_pca\_e0.5\_m10\_noise | 0.892987 | 0.999641 | 0.256831 | 0.07778 | 0.118542 |
| XGBoost | dbscan\_pca\_e0.5\_m15\_labels | 0.892987 | 0.999641 | 0.256831 | 0.07778 | 0.118542 |
| XGBoost | dbscan\_pca\_e0.5\_m15\_noise | 0.892987 | 0.999641 | 0.256831 | 0.07778 | 0.118542 |
| XGBoost | dbscan\_pca\_e0.5\_m3\_labels | 0.890463 | 0.99965 | 0.268 | 0.07556 | 0.117871 |
| XGBoost | dbscan\_pca\_e0.5\_m3\_noise | 0.893399 | 0.999648 | 0.242287 | 0.06444 | 0.101414 |
| XGBoost | dbscan\_pca\_e0.5\_m5\_labels | 0.889009 | 0.999646 | 0.247894 | 0.06889 | 0.107474 |
| XGBoost | dbscan\_pca\_e0.5\_m5\_noise | 0.89321 | 0.999634 | 0.223123 | 0.06667 | 0.101831 |
| XGBoost | dbscan\_pca\_e0.5\_m7\_labels | 0.878693 | 0.999647 | 0.274685 | 0.08444 | 0.128538 |
| XGBoost | dbscan\_pca\_e0.5\_m7\_noise | 0.893026 | 0.999634 | 0.219453 | 0.06667 | 0.101921 |
| XGBoost | dbscan\_pca\_e0.7\_m10\_labels | 0.892987 | 0.999641 | 0.256831 | 0.07778 | 0.118542 |
| XGBoost | dbscan\_pca\_e0.7\_m10\_noise | 0.892987 | 0.999641 | 0.256831 | 0.07778 | 0.118542 |
| XGBoost | dbscan\_pca\_e0.7\_m15\_labels | 0.892987 | 0.999641 | 0.256831 | 0.07778 | 0.118542 |
| XGBoost | dbscan\_pca\_e0.7\_m15\_noise | 0.892987 | 0.999641 | 0.256831 | 0.07778 | 0.118542 |
| XGBoost | dbscan\_pca\_e0.7\_m3\_labels | 0.890426 | 0.999649 | 0.242272 | 0.06444 | 0.101615 |
| XGBoost | dbscan\_pca\_e0.7\_m3\_noise | 0.895254 | 0.999651 | 0.263287 | 0.06667 | 0.105911 |
| XGBoost | dbscan\_pca\_e0.7\_m5\_labels | 0.888793 | 0.999654 | 0.290877 | 0.07778 | 0.121977 |
| XGBoost | dbscan\_pca\_e0.7\_m5\_noise | 0.894245 | 0.999634 | 0.222651 | 0.06889 | 0.105055 |
| XGBoost | dbscan\_pca\_e0.7\_m7\_labels | 0.877039 | 0.999646 | 0.256296 | 0.07111 | 0.110719 |
| XGBoost | dbscan\_pca\_e0.7\_m7\_noise | 0.894528 | 0.999636 | 0.224216 | 0.06889 | 0.105078 |
| XGBoost | dbscan\_pca\_e1.0\_m10\_labels | 0.892987 | 0.999641 | 0.256831 | 0.07778 | 0.118542 |
| XGBoost | dbscan\_pca\_e1.0\_m10\_noise | 0.892987 | 0.999641 | 0.256831 | 0.07778 | 0.118542 |
| XGBoost | dbscan\_pca\_e1.0\_m15\_labels | 0.892987 | 0.999641 | 0.256831 | 0.07778 | 0.118542 |
| XGBoost | dbscan\_pca\_e1.0\_m15\_noise | 0.892987 | 0.999641 | 0.256831 | 0.07778 | 0.118542 |
| XGBoost | dbscan\_pca\_e1.0\_m3\_labels | 0.890426 | 0.999649 | 0.242272 | 0.06444 | 0.101615 |
| XGBoost | dbscan\_pca\_e1.0\_m3\_noise | 0.895254 | 0.999651 | 0.263287 | 0.06667 | 0.105911 |
| XGBoost | dbscan\_pca\_e1.0\_m5\_labels | 0.888793 | 0.999654 | 0.290877 | 0.07778 | 0.121977 |
| XGBoost | dbscan\_pca\_e1.0\_m5\_noise | 0.894245 | 0.999634 | 0.222651 | 0.06889 | 0.105055 |
| XGBoost | dbscan\_pca\_e1.0\_m7\_labels | 0.877039 | 0.999646 | 0.256296 | 0.07111 | 0.110719 |
| XGBoost | dbscan\_pca\_e1.0\_m7\_noise | 0.894528 | 0.999636 | 0.224216 | 0.06889 | 0.105078 |
| XGBoost | dbscan\_pca\_e1.3\_m10\_labels | 0.892987 | 0.999641 | 0.256831 | 0.07778 | 0.118542 |
| XGBoost | dbscan\_pca\_e1.3\_m10\_noise | 0.892987 | 0.999641 | 0.256831 | 0.07778 | 0.118542 |
| XGBoost | dbscan\_pca\_e1.3\_m15\_labels | 0.892987 | 0.999641 | 0.256831 | 0.07778 | 0.118542 |
| XGBoost | dbscan\_pca\_e1.3\_m15\_noise | 0.892987 | 0.999641 | 0.256831 | 0.07778 | 0.118542 |
| XGBoost | dbscan\_pca\_e1.3\_m3\_labels | 0.890023 | 0.999636 | 0.207737 | 0.06222 | 0.095354 |
| XGBoost | dbscan\_pca\_e1.3\_m3\_noise | 0.893167 | 0.999648 | 0.270611 | 0.07556 | 0.117856 |
| XGBoost | dbscan\_pca\_e1.3\_m5\_labels | 0.888307 | 0.999643 | 0.234139 | 0.06444 | 0.100584 |
| XGBoost | dbscan\_pca\_e1.3\_m5\_noise | 0.897378 | 0.999634 | 0.226464 | 0.07333 | 0.110633 |
| XGBoost | dbscan\_pca\_e1.3\_m7\_labels | 0.885152 | 0.99965 | 0.290895 | 0.07556 | 0.118893 |
| XGBoost | dbscan\_pca\_e1.3\_m7\_noise | 0.894958 | 0.999635 | 0.214692 | 0.06444 | 0.098629 |
| XGBoost | hier\_full\_n2\_labels | 0.892987 | 0.999641 | 0.256831 | 0.07778 | 0.118542 |
| XGBoost | hier\_full\_n3\_labels | 0.892987 | 0.999641 | 0.256831 | 0.07778 | 0.118542 |
| XGBoost | hier\_full\_n4\_labels | 0.892987 | 0.999641 | 0.256831 | 0.07778 | 0.118542 |
| XGBoost | hier\_full\_n5\_labels | 0.892987 | 0.999641 | 0.256831 | 0.07778 | 0.118542 |
| XGBoost | hier\_full\_n6\_labels | 0.892987 | 0.999641 | 0.256831 | 0.07778 | 0.118542 |
| XGBoost | hier\_pca\_n2\_labels | 0.892987 | 0.999641 | 0.256831 | 0.07778 | 0.118542 |
| XGBoost | hier\_pca\_n3\_labels | 0.892987 | 0.999641 | 0.256831 | 0.07778 | 0.118542 |
| XGBoost | hier\_pca\_n4\_labels | 0.892987 | 0.999641 | 0.256831 | 0.07778 | 0.118542 |
| XGBoost | hier\_pca\_n5\_labels | 0.892987 | 0.999641 | 0.256831 | 0.07778 | 0.118542 |
| XGBoost | hier\_pca\_n6\_labels | 0.892987 | 0.999641 | 0.256831 | 0.07778 | 0.118542 |
| XGBoost | kmeans\_full\_n10\_labels | 0.875666 | 0.999632 | 0.210112 | 0.06444 | 0.098085 |
| XGBoost | kmeans\_full\_n2\_labels | 0.892987 | 0.999641 | 0.256831 | 0.07778 | 0.118542 |
| XGBoost | kmeans\_full\_n3\_labels | 0.892987 | 0.999641 | 0.256831 | 0.07778 | 0.118542 |
| XGBoost | kmeans\_full\_n4\_labels | 0.892987 | 0.999641 | 0.256831 | 0.07778 | 0.118542 |
| XGBoost | kmeans\_full\_n5\_labels | 0.886497 | 0.999641 | 0.242187 | 0.07333 | 0.112233 |
| XGBoost | kmeans\_full\_n6\_labels | 0.874721 | 0.999639 | 0.227116 | 0.06667 | 0.102614 |
| XGBoost | kmeans\_full\_n7\_labels | 0.891457 | 0.999646 | 0.268509 | 0.07778 | 0.119839 |
| XGBoost | kmeans\_full\_n8\_labels | 0.878071 | 0.999643 | 0.239991 | 0.06667 | 0.103929 |
| XGBoost | kmeans\_full\_n9\_labels | 0.890782 | 0.999659 | 0.29116 | 0.06444 | 0.104442 |
| XGBoost | kmeans\_pca\_n10\_labels | 0.878932 | 0.999642 | 0.191081 | 0.04889 | 0.077773 |
| XGBoost | kmeans\_pca\_n2\_labels | 0.887023 | 0.999641 | 0.254256 | 0.07778 | 0.11836 |
| XGBoost | kmeans\_pca\_n3\_labels | 0.876452 | 0.999638 | 0.224617 | 0.06444 | 0.099765 |
| XGBoost | kmeans\_pca\_n4\_labels | 0.89455 | 0.999654 | 0.289463 | 0.07778 | 0.121309 |
| XGBoost | kmeans\_pca\_n5\_labels | 0.881579 | 0.999646 | 0.267312 | 0.07556 | 0.117511 |
| XGBoost | kmeans\_pca\_n6\_labels | 0.88892 | 0.999646 | 0.257777 | 0.07111 | 0.110983 |
| XGBoost | kmeans\_pca\_n7\_labels | 0.879198 | 0.999647 | 0.269476 | 0.07556 | 0.116891 |
| XGBoost | kmeans\_pca\_n8\_labels | 0.888716 | 0.99964 | 0.259683 | 0.08 | 0.121967 |
| XGBoost | kmeans\_pca\_n9\_labels | 0.884668 | 0.999658 | 0.279257 | 0.06667 | 0.107494 |
| SVM | Baseline | 0.52448 | 0.513412 | 0.00034 | 0.53556 | 0.00068 |
| SVM | dbscan\_full\_e0.3\_m10\_labels | 0.589403 | 0.998552 | 0.044773 | 0.18 | 0.071623 |
| SVM | dbscan\_full\_e0.3\_m10\_noise | 0.590508 | 0.998541 | 0.044895 | 0.18222 | 0.07195 |
| SVM | dbscan\_full\_e0.3\_m15\_labels | 0.589403 | 0.998552 | 0.044773 | 0.18 | 0.071623 |
| SVM | dbscan\_full\_e0.3\_m15\_noise | 0.590508 | 0.998541 | 0.044895 | 0.18222 | 0.07195 |
| SVM | dbscan\_full\_e0.3\_m3\_labels | 0.589404 | 0.998555 | 0.044866 | 0.18 | 0.07174 |
| SVM | dbscan\_full\_e0.3\_m3\_noise | 0.590509 | 0.998542 | 0.044919 | 0.18222 | 0.071981 |
| SVM | dbscan\_full\_e0.3\_m5\_labels | 0.589403 | 0.998552 | 0.044773 | 0.18 | 0.071623 |
| SVM | dbscan\_full\_e0.3\_m5\_noise | 0.590509 | 0.998543 | 0.044945 | 0.18222 | 0.072015 |
| SVM | dbscan\_full\_e0.3\_m7\_labels | 0.589403 | 0.998552 | 0.044773 | 0.18 | 0.071623 |
| SVM | dbscan\_full\_e0.3\_m7\_noise | 0.590508 | 0.998541 | 0.044895 | 0.18222 | 0.07195 |
| SVM | dbscan\_full\_e0.5\_m10\_labels | 0.589403 | 0.998552 | 0.044773 | 0.18 | 0.071623 |
| SVM | dbscan\_full\_e0.5\_m10\_noise | 0.590508 | 0.998541 | 0.044895 | 0.18222 | 0.07195 |
| SVM | dbscan\_full\_e0.5\_m15\_labels | 0.589403 | 0.998552 | 0.044773 | 0.18 | 0.071623 |
| SVM | dbscan\_full\_e0.5\_m15\_noise | 0.590508 | 0.998541 | 0.044895 | 0.18222 | 0.07195 |
| SVM | dbscan\_full\_e0.5\_m3\_labels | 0.589399 | 0.998545 | 0.044545 | 0.18 | 0.071329 |
| SVM | dbscan\_full\_e0.5\_m3\_noise | 0.590508 | 0.998541 | 0.044869 | 0.18222 | 0.071916 |
| SVM | dbscan\_full\_e0.5\_m5\_labels | 0.589402 | 0.998551 | 0.044749 | 0.18 | 0.071592 |
| SVM | dbscan\_full\_e0.5\_m5\_noise | 0.590509 | 0.998543 | 0.044945 | 0.18222 | 0.072015 |
| SVM | dbscan\_full\_e0.5\_m7\_labels | 0.589402 | 0.99855 | 0.044724 | 0.18 | 0.071562 |
| SVM | dbscan\_full\_e0.5\_m7\_noise | 0.590509 | 0.998542 | 0.044919 | 0.18222 | 0.071981 |
| SVM | dbscan\_full\_e0.7\_m10\_labels | 0.589403 | 0.998552 | 0.044773 | 0.18 | 0.071623 |
| SVM | dbscan\_full\_e0.7\_m10\_noise | 0.590509 | 0.998542 | 0.044919 | 0.18222 | 0.071981 |
| SVM | dbscan\_full\_e0.7\_m15\_labels | 0.589403 | 0.998552 | 0.044773 | 0.18 | 0.071623 |
| SVM | dbscan\_full\_e0.7\_m15\_noise | 0.590508 | 0.998541 | 0.044895 | 0.18222 | 0.07195 |
| SVM | dbscan\_full\_e0.7\_m3\_labels | 0.592674 | 0.998429 | 0.042065 | 0.18667 | 0.068578 |
| SVM | dbscan\_full\_e0.7\_m3\_noise | 0.590509 | 0.998543 | 0.044943 | 0.18222 | 0.072011 |
| SVM | dbscan\_full\_e0.7\_m5\_labels | 0.589402 | 0.998551 | 0.04475 | 0.18 | 0.071593 |
| SVM | dbscan\_full\_e0.7\_m5\_noise | 0.590508 | 0.998541 | 0.044895 | 0.18222 | 0.07195 |
| SVM | dbscan\_full\_e0.7\_m7\_labels | 0.589404 | 0.998554 | 0.044836 | 0.18 | 0.071705 |
| SVM | dbscan\_full\_e0.7\_m7\_noise | 0.590509 | 0.998543 | 0.044945 | 0.18222 | 0.072015 |
| SVM | dbscan\_full\_e1.0\_m10\_labels | 0.589402 | 0.99855 | 0.044724 | 0.18 | 0.071562 |
| SVM | dbscan\_full\_e1.0\_m10\_noise | 0.590509 | 0.998543 | 0.044945 | 0.18222 | 0.072015 |
| SVM | dbscan\_full\_e1.0\_m15\_labels | 0.589403 | 0.998552 | 0.044773 | 0.18 | 0.071623 |
| SVM | dbscan\_full\_e1.0\_m15\_noise | 0.590508 | 0.998541 | 0.044895 | 0.18222 | 0.07195 |
| SVM | dbscan\_full\_e1.0\_m3\_labels | 0.658656 | 0.992661 | 0.013925 | 0.32444 | 0.026702 |
| SVM | dbscan\_full\_e1.0\_m3\_noise | 0.590508 | 0.998541 | 0.044869 | 0.18222 | 0.071916 |
| SVM | dbscan\_full\_e1.0\_m5\_labels | 0.59051 | 0.998545 | 0.044979 | 0.18222 | 0.072058 |
| SVM | dbscan\_full\_e1.0\_m5\_noise | 0.590509 | 0.998543 | 0.044943 | 0.18222 | 0.07201 |
| SVM | dbscan\_full\_e1.0\_m7\_labels | 0.589404 | 0.998555 | 0.044892 | 0.18 | 0.071778 |
| SVM | dbscan\_full\_e1.0\_m7\_noise | 0.590509 | 0.998543 | 0.044969 | 0.18222 | 0.072044 |
| SVM | dbscan\_full\_e1.3\_m10\_labels | 0.589404 | 0.998554 | 0.044842 | 0.18 | 0.071712 |
| SVM | dbscan\_full\_e1.3\_m10\_noise | 0.590509 | 0.998543 | 0.044937 | 0.18222 | 0.072005 |
| SVM | dbscan\_full\_e1.3\_m15\_labels | 0.589403 | 0.998552 | 0.044774 | 0.18 | 0.071625 |
| SVM | dbscan\_full\_e1.3\_m15\_noise | 0.590509 | 0.998542 | 0.044919 | 0.18222 | 0.071981 |
| SVM | dbscan\_full\_e1.3\_m3\_labels | 0.828512 | 0.968048 | 0.00682 | 0.68889 | 0.013503 |
| SVM | dbscan\_full\_e1.3\_m3\_noise | 0.588295 | 0.998557 | 0.044525 | 0.17778 | 0.071127 |
| SVM | dbscan\_full\_e1.3\_m5\_labels | 0.618724 | 0.997214 | 0.028537 | 0.24 | 0.050978 |
| SVM | dbscan\_full\_e1.3\_m5\_noise | 0.590509 | 0.998543 | 0.044958 | 0.18222 | 0.07203 |
| SVM | dbscan\_full\_e1.3\_m7\_labels | 0.590502 | 0.99853 | 0.044472 | 0.18222 | 0.071409 |
| SVM | dbscan\_full\_e1.3\_m7\_noise | 0.590509 | 0.998542 | 0.044925 | 0.18222 | 0.071986 |
| SVM | dbscan\_pca\_e0.3\_m10\_labels | 0.589403 | 0.998552 | 0.044773 | 0.18 | 0.071623 |
| SVM | dbscan\_pca\_e0.3\_m10\_noise | 0.590508 | 0.998541 | 0.044895 | 0.18222 | 0.07195 |
| SVM | dbscan\_pca\_e0.3\_m15\_labels | 0.589403 | 0.998552 | 0.044773 | 0.18 | 0.071623 |
| SVM | dbscan\_pca\_e0.3\_m15\_noise | 0.590508 | 0.998541 | 0.044895 | 0.18222 | 0.07195 |
| SVM | dbscan\_pca\_e0.3\_m3\_labels | 0.589312 | 0.99837 | 0.039084 | 0.18 | 0.064048 |
| SVM | dbscan\_pca\_e0.3\_m3\_noise | 0.591621 | 0.998546 | 0.045632 | 0.18444 | 0.073076 |
| SVM | dbscan\_pca\_e0.3\_m5\_labels | 0.590521 | 0.998567 | 0.045831 | 0.18222 | 0.07316 |
| SVM | dbscan\_pca\_e0.3\_m5\_noise | 0.591622 | 0.998548 | 0.045685 | 0.18444 | 0.07314 |
| SVM | dbscan\_pca\_e0.3\_m7\_labels | 0.59274 | 0.998562 | 0.046776 | 0.18667 | 0.074712 |
| SVM | dbscan\_pca\_e0.3\_m7\_noise | 0.591623 | 0.998549 | 0.045733 | 0.18444 | 0.073204 |
| SVM | dbscan\_pca\_e0.5\_m10\_labels | 0.589403 | 0.998552 | 0.044773 | 0.18 | 0.071623 |
| SVM | dbscan\_pca\_e0.5\_m10\_noise | 0.590508 | 0.998541 | 0.044895 | 0.18222 | 0.07195 |
| SVM | dbscan\_pca\_e0.5\_m15\_labels | 0.589403 | 0.998552 | 0.044773 | 0.18 | 0.071623 |
| SVM | dbscan\_pca\_e0.5\_m15\_noise | 0.590508 | 0.998541 | 0.044895 | 0.18222 | 0.07195 |
| SVM | dbscan\_pca\_e0.5\_m3\_labels | 0.589312 | 0.99837 | 0.039084 | 0.18 | 0.064048 |
| SVM | dbscan\_pca\_e0.5\_m3\_noise | 0.591621 | 0.998546 | 0.045632 | 0.18444 | 0.073076 |
| SVM | dbscan\_pca\_e0.5\_m5\_labels | 0.590521 | 0.998567 | 0.045831 | 0.18222 | 0.07316 |
| SVM | dbscan\_pca\_e0.5\_m5\_noise | 0.591622 | 0.998548 | 0.045685 | 0.18444 | 0.07314 |
| SVM | dbscan\_pca\_e0.5\_m7\_labels | 0.59274 | 0.998562 | 0.046776 | 0.18667 | 0.074712 |
| SVM | dbscan\_pca\_e0.5\_m7\_noise | 0.591623 | 0.998549 | 0.045733 | 0.18444 | 0.073204 |
| SVM | dbscan\_pca\_e0.7\_m10\_labels | 0.589403 | 0.998552 | 0.044773 | 0.18 | 0.071623 |
| SVM | dbscan\_pca\_e0.7\_m10\_noise | 0.590508 | 0.998541 | 0.044895 | 0.18222 | 0.07195 |
| SVM | dbscan\_pca\_e0.7\_m15\_labels | 0.589403 | 0.998552 | 0.044773 | 0.18 | 0.071623 |
| SVM | dbscan\_pca\_e0.7\_m15\_noise | 0.590508 | 0.998541 | 0.044895 | 0.18222 | 0.07195 |
| SVM | dbscan\_pca\_e0.7\_m3\_labels | 0.600225 | 0.997982 | 0.034655 | 0.20222 | 0.058844 |
| SVM | dbscan\_pca\_e0.7\_m3\_noise | 0.591623 | 0.998548 | 0.045709 | 0.18444 | 0.073176 |
| SVM | dbscan\_pca\_e0.7\_m5\_labels | 0.588293 | 0.998554 | 0.04426 | 0.17778 | 0.070789 |
| SVM | dbscan\_pca\_e0.7\_m5\_noise | 0.591622 | 0.998548 | 0.04568 | 0.18444 | 0.073135 |
| SVM | dbscan\_pca\_e0.7\_m7\_labels | 0.591634 | 0.998571 | 0.046512 | 0.18444 | 0.074205 |
| SVM | dbscan\_pca\_e0.7\_m7\_noise | 0.591623 | 0.998549 | 0.045733 | 0.18444 | 0.073204 |
| SVM | dbscan\_pca\_e1.0\_m10\_labels | 0.589403 | 0.998552 | 0.044773 | 0.18 | 0.071623 |
| SVM | dbscan\_pca\_e1.0\_m10\_noise | 0.590508 | 0.998541 | 0.044895 | 0.18222 | 0.07195 |
| SVM | dbscan\_pca\_e1.0\_m15\_labels | 0.589403 | 0.998552 | 0.044773 | 0.18 | 0.071623 |
| SVM | dbscan\_pca\_e1.0\_m15\_noise | 0.590508 | 0.998541 | 0.044895 | 0.18222 | 0.07195 |
| SVM | dbscan\_pca\_e1.0\_m3\_labels | 0.600225 | 0.997982 | 0.034655 | 0.20222 | 0.058844 |
| SVM | dbscan\_pca\_e1.0\_m3\_noise | 0.591623 | 0.998548 | 0.045709 | 0.18444 | 0.073176 |
| SVM | dbscan\_pca\_e1.0\_m5\_labels | 0.588293 | 0.998554 | 0.04426 | 0.17778 | 0.070789 |
| SVM | dbscan\_pca\_e1.0\_m5\_noise | 0.591622 | 0.998548 | 0.04568 | 0.18444 | 0.073135 |
| SVM | dbscan\_pca\_e1.0\_m7\_labels | 0.591634 | 0.998571 | 0.046512 | 0.18444 | 0.074205 |
| SVM | dbscan\_pca\_e1.0\_m7\_noise | 0.591623 | 0.998549 | 0.045733 | 0.18444 | 0.073204 |
| SVM | dbscan\_pca\_e1.3\_m10\_labels | 0.589403 | 0.998552 | 0.044773 | 0.18 | 0.071623 |
| SVM | dbscan\_pca\_e1.3\_m10\_noise | 0.590508 | 0.998541 | 0.044895 | 0.18222 | 0.07195 |
| SVM | dbscan\_pca\_e1.3\_m15\_labels | 0.589403 | 0.998552 | 0.044773 | 0.18 | 0.071623 |
| SVM | dbscan\_pca\_e1.3\_m15\_noise | 0.590508 | 0.998541 | 0.044895 | 0.18222 | 0.07195 |
| SVM | dbscan\_pca\_e1.3\_m3\_labels | 0.608855 | 0.997468 | 0.029979 | 0.22 | 0.052329 |
| SVM | dbscan\_pca\_e1.3\_m3\_noise | 0.591623 | 0.998548 | 0.045714 | 0.18444 | 0.073179 |
| SVM | dbscan\_pca\_e1.3\_m5\_labels | 0.588278 | 0.998524 | 0.043241 | 0.17778 | 0.069467 |
| SVM | dbscan\_pca\_e1.3\_m5\_noise | 0.591622 | 0.998547 | 0.045662 | 0.18444 | 0.07311 |
| SVM | dbscan\_pca\_e1.3\_m7\_labels | 0.590522 | 0.99857 | 0.045931 | 0.18222 | 0.073281 |
| SVM | dbscan\_pca\_e1.3\_m7\_noise | 0.591624 | 0.99855 | 0.045775 | 0.18444 | 0.073259 |
| SVM | hier\_full\_n2\_labels | 0.589412 | 0.99857 | 0.045446 | 0.18 | 0.072483 |
| SVM | hier\_full\_n3\_labels | 0.589412 | 0.99857 | 0.045446 | 0.18 | 0.072483 |
| SVM | hier\_full\_n4\_labels | 0.589412 | 0.99857 | 0.045432 | 0.18 | 0.072468 |
| SVM | hier\_full\_n5\_labels | 0.589412 | 0.99857 | 0.045407 | 0.18 | 0.072435 |
| SVM | hier\_full\_n6\_labels | 0.589412 | 0.99857 | 0.045432 | 0.18 | 0.072468 |
| SVM | hier\_pca\_n2\_labels | 0.589412 | 0.99857 | 0.045446 | 0.18 | 0.072483 |
| SVM | hier\_pca\_n3\_labels | 0.589412 | 0.99857 | 0.045413 | 0.18 | 0.072442 |
| SVM | hier\_pca\_n4\_labels | 0.589412 | 0.99857 | 0.045413 | 0.18 | 0.072442 |
| SVM | hier\_pca\_n5\_labels | 0.589412 | 0.99857 | 0.045407 | 0.18 | 0.072435 |
| SVM | hier\_pca\_n6\_labels | 0.589412 | 0.998571 | 0.045466 | 0.18 | 0.072509 |
| SVM | kmeans\_full\_n10\_labels | 0.626409 | 0.997032 | 0.028322 | 0.25556 | 0.05097 |
| SVM | kmeans\_full\_n2\_labels | 0.589412 | 0.99857 | 0.045446 | 0.18 | 0.072483 |
| SVM | kmeans\_full\_n3\_labels | 0.589412 | 0.99857 | 0.045432 | 0.18 | 0.072468 |
| SVM | kmeans\_full\_n4\_labels | 0.589412 | 0.99857 | 0.04544 | 0.18 | 0.072476 |
| SVM | kmeans\_full\_n5\_labels | 0.589406 | 0.998559 | 0.045093 | 0.18 | 0.072001 |
| SVM | kmeans\_full\_n6\_labels | 0.594757 | 0.998153 | 0.035812 | 0.19111 | 0.060253 |
| SVM | kmeans\_full\_n7\_labels | 0.605936 | 0.998296 | 0.043467 | 0.21333 | 0.072128 |
| SVM | kmeans\_full\_n8\_labels | 0.604703 | 0.998052 | 0.037115 | 0.21111 | 0.063108 |
| SVM | kmeans\_full\_n9\_labels | 0.611306 | 0.997927 | 0.036829 | 0.22444 | 0.063224 |
| SVM | kmeans\_pca\_n10\_labels | 0.578298 | 0.998559 | 0.039891 | 0.15778 | 0.063549 |
| SVM | kmeans\_pca\_n2\_labels | 0.590509 | 0.998543 | 0.044989 | 0.18222 | 0.072063 |
| SVM | kmeans\_pca\_n3\_labels | 0.588294 | 0.998556 | 0.044361 | 0.17778 | 0.070927 |
| SVM | kmeans\_pca\_n4\_labels | 0.591607 | 0.998517 | 0.044505 | 0.18444 | 0.071622 |
| SVM | kmeans\_pca\_n5\_labels | 0.591636 | 0.998576 | 0.046624 | 0.18444 | 0.074349 |
| SVM | kmeans\_pca\_n6\_labels | 0.586089 | 0.998588 | 0.0444 | 0.17333 | 0.070617 |
| SVM | kmeans\_pca\_n7\_labels | 0.587184 | 0.998557 | 0.04405 | 0.17556 | 0.070348 |
| SVM | kmeans\_pca\_n8\_labels | 0.593645 | 0.998151 | 0.035419 | 0.18889 | 0.059618 |
| SVM | kmeans\_pca\_n9\_labels | 0.594597 | 0.997833 | 0.030253 | 0.19111 | 0.0522 |
| Random Forest | Baseline | 0.901018 | 0.996986 | 0.023909 | 0.21333 | 0.042904 |
| Random Forest | dbscan\_full\_e0.3\_m10\_labels | 0.901102 | 0.993526 | 0.016361 | 0.33556 | 0.031183 |
| Random Forest | dbscan\_full\_e0.3\_m10\_noise | 0.901102 | 0.993526 | 0.016361 | 0.33556 | 0.031183 |
| Random Forest | dbscan\_full\_e0.3\_m15\_labels | 0.901102 | 0.993526 | 0.016361 | 0.33556 | 0.031183 |
| Random Forest | dbscan\_full\_e0.3\_m15\_noise | 0.901102 | 0.993526 | 0.016361 | 0.33556 | 0.031183 |
| Random Forest | dbscan\_full\_e0.3\_m3\_labels | 0.901291 | 0.993533 | 0.016487 | 0.33778 | 0.03142 |
| Random Forest | dbscan\_full\_e0.3\_m3\_noise | 0.901102 | 0.993526 | 0.016361 | 0.33556 | 0.031183 |
| Random Forest | dbscan\_full\_e0.3\_m5\_labels | 0.90113 | 0.993528 | 0.016366 | 0.33556 | 0.031192 |
| Random Forest | dbscan\_full\_e0.3\_m5\_noise | 0.901102 | 0.993526 | 0.016361 | 0.33556 | 0.031183 |
| Random Forest | dbscan\_full\_e0.3\_m7\_labels | 0.901102 | 0.993526 | 0.016361 | 0.33556 | 0.031183 |
| Random Forest | dbscan\_full\_e0.3\_m7\_noise | 0.901102 | 0.993526 | 0.016361 | 0.33556 | 0.031183 |
| Random Forest | dbscan\_full\_e0.5\_m10\_labels | 0.901102 | 0.993526 | 0.016361 | 0.33556 | 0.031183 |
| Random Forest | dbscan\_full\_e0.5\_m10\_noise | 0.901102 | 0.993526 | 0.016361 | 0.33556 | 0.031183 |
| Random Forest | dbscan\_full\_e0.5\_m15\_labels | 0.901102 | 0.993526 | 0.016361 | 0.33556 | 0.031183 |
| Random Forest | dbscan\_full\_e0.5\_m15\_noise | 0.901102 | 0.993526 | 0.016361 | 0.33556 | 0.031183 |
| Random Forest | dbscan\_full\_e0.5\_m3\_labels | 0.902323 | 0.993532 | 0.016375 | 0.33556 | 0.031209 |
| Random Forest | dbscan\_full\_e0.5\_m3\_noise | 0.901102 | 0.993526 | 0.016361 | 0.33556 | 0.031183 |
| Random Forest | dbscan\_full\_e0.5\_m5\_labels | 0.901749 | 0.993532 | 0.016376 | 0.33556 | 0.03121 |
| Random Forest | dbscan\_full\_e0.5\_m5\_noise | 0.901102 | 0.993526 | 0.016361 | 0.33556 | 0.031183 |
| Random Forest | dbscan\_full\_e0.5\_m7\_labels | 0.902019 | 0.993525 | 0.016361 | 0.33556 | 0.031181 |
| Random Forest | dbscan\_full\_e0.5\_m7\_noise | 0.901102 | 0.993526 | 0.016361 | 0.33556 | 0.031183 |
| Random Forest | dbscan\_full\_e0.7\_m10\_labels | 0.901075 | 0.993522 | 0.016347 | 0.33556 | 0.031158 |
| Random Forest | dbscan\_full\_e0.7\_m10\_noise | 0.901102 | 0.993526 | 0.016361 | 0.33556 | 0.031183 |
| Random Forest | dbscan\_full\_e0.7\_m15\_labels | 0.901102 | 0.993526 | 0.016361 | 0.33556 | 0.031183 |
| Random Forest | dbscan\_full\_e0.7\_m15\_noise | 0.901102 | 0.993526 | 0.016361 | 0.33556 | 0.031183 |
| Random Forest | dbscan\_full\_e0.7\_m3\_labels | 0.899484 | 0.993548 | 0.016303 | 0.33333 | 0.031069 |
| Random Forest | dbscan\_full\_e0.7\_m3\_noise | 0.901102 | 0.993526 | 0.016361 | 0.33556 | 0.031183 |
| Random Forest | dbscan\_full\_e0.7\_m5\_labels | 0.90141 | 0.99351 | 0.016323 | 0.33556 | 0.031114 |
| Random Forest | dbscan\_full\_e0.7\_m5\_noise | 0.901102 | 0.993526 | 0.016361 | 0.33556 | 0.031183 |
| Random Forest | dbscan\_full\_e0.7\_m7\_labels | 0.901407 | 0.993521 | 0.016348 | 0.33556 | 0.031158 |
| Random Forest | dbscan\_full\_e0.7\_m7\_noise | 0.901102 | 0.993526 | 0.016361 | 0.33556 | 0.031183 |
| Random Forest | dbscan\_full\_e1.0\_m10\_labels | 0.90141 | 0.993528 | 0.016473 | 0.33778 | 0.031397 |
| Random Forest | dbscan\_full\_e1.0\_m10\_noise | 0.901102 | 0.993526 | 0.016361 | 0.33556 | 0.031183 |
| Random Forest | dbscan\_full\_e1.0\_m15\_labels | 0.901102 | 0.993526 | 0.016361 | 0.33556 | 0.031183 |
| Random Forest | dbscan\_full\_e1.0\_m15\_noise | 0.901102 | 0.993526 | 0.016361 | 0.33556 | 0.031183 |
| Random Forest | dbscan\_full\_e1.0\_m3\_labels | 0.899179 | 0.993474 | 0.016216 | 0.33556 | 0.03092 |
| Random Forest | dbscan\_full\_e1.0\_m3\_noise | 0.901102 | 0.993526 | 0.016361 | 0.33556 | 0.031183 |
| Random Forest | dbscan\_full\_e1.0\_m5\_labels | 0.901624 | 0.993524 | 0.016352 | 0.33556 | 0.031166 |
| Random Forest | dbscan\_full\_e1.0\_m5\_noise | 0.901102 | 0.993526 | 0.016361 | 0.33556 | 0.031183 |
| Random Forest | dbscan\_full\_e1.0\_m7\_labels | 0.90274 | 0.993531 | 0.016585 | 0.34 | 0.031609 |
| Random Forest | dbscan\_full\_e1.0\_m7\_noise | 0.901102 | 0.993526 | 0.016361 | 0.33556 | 0.031183 |
| Random Forest | dbscan\_full\_e1.3\_m10\_labels | 0.901615 | 0.99353 | 0.016482 | 0.33778 | 0.031413 |
| Random Forest | dbscan\_full\_e1.3\_m10\_noise | 0.901102 | 0.993526 | 0.016361 | 0.33556 | 0.031183 |
| Random Forest | dbscan\_full\_e1.3\_m15\_labels | 0.901353 | 0.993536 | 0.016386 | 0.33556 | 0.031228 |
| Random Forest | dbscan\_full\_e1.3\_m15\_noise | 0.901102 | 0.993526 | 0.016361 | 0.33556 | 0.031183 |
| Random Forest | dbscan\_full\_e1.3\_m3\_labels | 0.897933 | 0.993641 | 0.016902 | 0.34 | 0.032185 |
| Random Forest | dbscan\_full\_e1.3\_m3\_noise | 0.908375 | 0.993654 | 0.016548 | 0.33111 | 0.0315 |
| Random Forest | dbscan\_full\_e1.3\_m5\_labels | 0.89963 | 0.993557 | 0.01665 | 0.34 | 0.031728 |
| Random Forest | dbscan\_full\_e1.3\_m5\_noise | 0.901102 | 0.993526 | 0.016361 | 0.33556 | 0.031183 |
| Random Forest | dbscan\_full\_e1.3\_m7\_labels | 0.903878 | 0.99353 | 0.016488 | 0.33778 | 0.031421 |
| Random Forest | dbscan\_full\_e1.3\_m7\_noise | 0.901102 | 0.993526 | 0.016361 | 0.33556 | 0.031183 |
| Random Forest | dbscan\_pca\_e0.3\_m10\_labels | 0.901102 | 0.993526 | 0.016361 | 0.33556 | 0.031183 |
| Random Forest | dbscan\_pca\_e0.3\_m10\_noise | 0.901102 | 0.993526 | 0.016361 | 0.33556 | 0.031183 |
| Random Forest | dbscan\_pca\_e0.3\_m15\_labels | 0.901102 | 0.993526 | 0.016361 | 0.33556 | 0.031183 |
| Random Forest | dbscan\_pca\_e0.3\_m15\_noise | 0.901102 | 0.993526 | 0.016361 | 0.33556 | 0.031183 |
| Random Forest | dbscan\_pca\_e0.3\_m3\_labels | 0.905738 | 0.993491 | 0.016311 | 0.33556 | 0.03109 |
| Random Forest | dbscan\_pca\_e0.3\_m3\_noise | 0.906296 | 0.993555 | 0.016534 | 0.33556 | 0.031495 |
| Random Forest | dbscan\_pca\_e0.3\_m5\_labels | 0.905283 | 0.993444 | 0.0163 | 0.33778 | 0.03108 |
| Random Forest | dbscan\_pca\_e0.3\_m5\_noise | 0.907376 | 0.993546 | 0.016629 | 0.33778 | 0.031678 |
| Random Forest | dbscan\_pca\_e0.3\_m7\_labels | 0.899487 | 0.993612 | 0.016795 | 0.34 | 0.031993 |
| Random Forest | dbscan\_pca\_e0.3\_m7\_noise | 0.907376 | 0.993546 | 0.016626 | 0.33778 | 0.031671 |
| Random Forest | dbscan\_pca\_e0.5\_m10\_labels | 0.901102 | 0.993526 | 0.016361 | 0.33556 | 0.031183 |
| Random Forest | dbscan\_pca\_e0.5\_m10\_noise | 0.901102 | 0.993526 | 0.016361 | 0.33556 | 0.031183 |
| Random Forest | dbscan\_pca\_e0.5\_m15\_labels | 0.901102 | 0.993526 | 0.016361 | 0.33556 | 0.031183 |
| Random Forest | dbscan\_pca\_e0.5\_m15\_noise | 0.901102 | 0.993526 | 0.016361 | 0.33556 | 0.031183 |
| Random Forest | dbscan\_pca\_e0.5\_m3\_labels | 0.905738 | 0.993491 | 0.016311 | 0.33556 | 0.03109 |
| Random Forest | dbscan\_pca\_e0.5\_m3\_noise | 0.906296 | 0.993555 | 0.016534 | 0.33556 | 0.031495 |
| Random Forest | dbscan\_pca\_e0.5\_m5\_labels | 0.905283 | 0.993444 | 0.0163 | 0.33778 | 0.03108 |
| Random Forest | dbscan\_pca\_e0.5\_m5\_noise | 0.907376 | 0.993546 | 0.016629 | 0.33778 | 0.031678 |
| Random Forest | dbscan\_pca\_e0.5\_m7\_labels | 0.899487 | 0.993612 | 0.016795 | 0.34 | 0.031993 |
| Random Forest | dbscan\_pca\_e0.5\_m7\_noise | 0.907376 | 0.993546 | 0.016626 | 0.33778 | 0.031671 |
| Random Forest | dbscan\_pca\_e0.7\_m10\_labels | 0.901102 | 0.993526 | 0.016361 | 0.33556 | 0.031183 |
| Random Forest | dbscan\_pca\_e0.7\_m10\_noise | 0.901102 | 0.993526 | 0.016361 | 0.33556 | 0.031183 |
| Random Forest | dbscan\_pca\_e0.7\_m15\_labels | 0.901102 | 0.993526 | 0.016361 | 0.33556 | 0.031183 |
| Random Forest | dbscan\_pca\_e0.7\_m15\_noise | 0.901102 | 0.993526 | 0.016361 | 0.33556 | 0.031183 |
| Random Forest | dbscan\_pca\_e0.7\_m3\_labels | 0.903167 | 0.993488 | 0.016008 | 0.32889 | 0.030511 |
| Random Forest | dbscan\_pca\_e0.7\_m3\_noise | 0.90589 | 0.993547 | 0.016409 | 0.33333 | 0.031261 |
| Random Forest | dbscan\_pca\_e0.7\_m5\_labels | 0.904206 | 0.993445 | 0.016509 | 0.34222 | 0.031479 |
| Random Forest | dbscan\_pca\_e0.7\_m5\_noise | 0.906539 | 0.993543 | 0.016621 | 0.33778 | 0.031663 |
| Random Forest | dbscan\_pca\_e0.7\_m7\_labels | 0.900453 | 0.993627 | 0.01634 | 0.32889 | 0.031117 |
| Random Forest | dbscan\_pca\_e0.7\_m7\_noise | 0.907356 | 0.99355 | 0.016634 | 0.33778 | 0.031686 |
| Random Forest | dbscan\_pca\_e1.0\_m10\_labels | 0.901102 | 0.993526 | 0.016361 | 0.33556 | 0.031183 |
| Random Forest | dbscan\_pca\_e1.0\_m10\_noise | 0.901102 | 0.993526 | 0.016361 | 0.33556 | 0.031183 |
| Random Forest | dbscan\_pca\_e1.0\_m15\_labels | 0.901102 | 0.993526 | 0.016361 | 0.33556 | 0.031183 |
| Random Forest | dbscan\_pca\_e1.0\_m15\_noise | 0.901102 | 0.993526 | 0.016361 | 0.33556 | 0.031183 |
| Random Forest | dbscan\_pca\_e1.0\_m3\_labels | 0.903167 | 0.993488 | 0.016008 | 0.32889 | 0.030511 |
| Random Forest | dbscan\_pca\_e1.0\_m3\_noise | 0.90589 | 0.993547 | 0.016409 | 0.33333 | 0.031261 |
| Random Forest | dbscan\_pca\_e1.0\_m5\_labels | 0.904206 | 0.993445 | 0.016509 | 0.34222 | 0.031479 |
| Random Forest | dbscan\_pca\_e1.0\_m5\_noise | 0.906539 | 0.993543 | 0.016621 | 0.33778 | 0.031663 |
| Random Forest | dbscan\_pca\_e1.0\_m7\_labels | 0.900453 | 0.993627 | 0.01634 | 0.32889 | 0.031117 |
| Random Forest | dbscan\_pca\_e1.0\_m7\_noise | 0.907356 | 0.99355 | 0.016634 | 0.33778 | 0.031686 |
| Random Forest | dbscan\_pca\_e1.3\_m10\_labels | 0.901096 | 0.993542 | 0.016309 | 0.33333 | 0.031078 |
| Random Forest | dbscan\_pca\_e1.3\_m10\_noise | 0.901102 | 0.993526 | 0.016361 | 0.33556 | 0.031183 |
| Random Forest | dbscan\_pca\_e1.3\_m15\_labels | 0.901102 | 0.993526 | 0.016361 | 0.33556 | 0.031183 |
| Random Forest | dbscan\_pca\_e1.3\_m15\_noise | 0.901102 | 0.993526 | 0.016361 | 0.33556 | 0.031183 |
| Random Forest | dbscan\_pca\_e1.3\_m3\_labels | 0.903942 | 0.993508 | 0.01646 | 0.33778 | 0.031373 |
| Random Forest | dbscan\_pca\_e1.3\_m3\_noise | 0.905343 | 0.993554 | 0.016529 | 0.33556 | 0.031488 |
| Random Forest | dbscan\_pca\_e1.3\_m5\_labels | 0.904358 | 0.993464 | 0.016238 | 0.33556 | 0.030959 |
| Random Forest | dbscan\_pca\_e1.3\_m5\_noise | 0.906539 | 0.993543 | 0.016621 | 0.33778 | 0.031663 |
| Random Forest | dbscan\_pca\_e1.3\_m7\_labels | 0.899387 | 0.993511 | 0.016327 | 0.33333 | 0.031109 |
| Random Forest | dbscan\_pca\_e1.3\_m7\_noise | 0.907357 | 0.993549 | 0.016632 | 0.33778 | 0.031683 |
| Random Forest | hier\_full\_n2\_labels | 0.90107 | 0.993526 | 0.016361 | 0.33556 | 0.031183 |
| Random Forest | hier\_full\_n3\_labels | 0.90107 | 0.993526 | 0.016361 | 0.33556 | 0.031183 |
| Random Forest | hier\_full\_n4\_labels | 0.90107 | 0.993526 | 0.016361 | 0.33556 | 0.031183 |
| Random Forest | hier\_full\_n5\_labels | 0.901071 | 0.993524 | 0.016358 | 0.33556 | 0.031178 |
| Random Forest | hier\_full\_n6\_labels | 0.901071 | 0.993524 | 0.016358 | 0.33556 | 0.031178 |
| Random Forest | hier\_pca\_n2\_labels | 0.901102 | 0.993526 | 0.016361 | 0.33556 | 0.031183 |
| Random Forest | hier\_pca\_n3\_labels | 0.901102 | 0.993526 | 0.016361 | 0.33556 | 0.031183 |
| Random Forest | hier\_pca\_n4\_labels | 0.901087 | 0.993532 | 0.016381 | 0.33556 | 0.031218 |
| Random Forest | hier\_pca\_n5\_labels | 0.901014 | 0.993524 | 0.016357 | 0.33556 | 0.031175 |
| Random Forest | hier\_pca\_n6\_labels | 0.901014 | 0.993524 | 0.016357 | 0.33556 | 0.031175 |
| Random Forest | kmeans\_full\_n10\_labels | 0.90218 | 0.993605 | 0.017074 | 0.34222 | 0.032485 |
| Random Forest | kmeans\_full\_n2\_labels | 0.901102 | 0.993526 | 0.016361 | 0.33556 | 0.031183 |
| Random Forest | kmeans\_full\_n3\_labels | 0.901103 | 0.993524 | 0.016358 | 0.33556 | 0.031178 |
| Random Forest | kmeans\_full\_n4\_labels | 0.901071 | 0.993524 | 0.016358 | 0.33556 | 0.031178 |
| Random Forest | kmeans\_full\_n5\_labels | 0.903702 | 0.993721 | 0.017143 | 0.34 | 0.032618 |
| Random Forest | kmeans\_full\_n6\_labels | 0.90126 | 0.993738 | 0.016741 | 0.33111 | 0.031845 |
| Random Forest | kmeans\_full\_n7\_labels | 0.900699 | 0.994318 | 0.018571 | 0.33111 | 0.035131 |
| Random Forest | kmeans\_full\_n8\_labels | 0.90999 | 0.993833 | 0.017767 | 0.34667 | 0.033782 |
| Random Forest | kmeans\_full\_n9\_labels | 0.901784 | 0.994315 | 0.017274 | 0.30889 | 0.032697 |
| Random Forest | kmeans\_pca\_n10\_labels | 0.903265 | 0.994396 | 0.019545 | 0.34 | 0.036916 |
| Random Forest | kmeans\_pca\_n2\_labels | 0.90814 | 0.993611 | 0.016548 | 0.33333 | 0.03151 |
| Random Forest | kmeans\_pca\_n3\_labels | 0.899946 | 0.993588 | 0.016467 | 0.33333 | 0.031369 |
| Random Forest | kmeans\_pca\_n4\_labels | 0.910376 | 0.9938 | 0.017336 | 0.33778 | 0.032957 |
| Random Forest | kmeans\_pca\_n5\_labels | 0.895929 | 0.99385 | 0.0176 | 0.33778 | 0.033419 |
| Random Forest | kmeans\_pca\_n6\_labels | 0.907947 | 0.99374 | 0.016445 | 0.32444 | 0.031274 |
| Random Forest | kmeans\_pca\_n7\_labels | 0.901392 | 0.994173 | 0.017204 | 0.31556 | 0.032609 |
| Random Forest | kmeans\_pca\_n8\_labels | 0.906269 | 0.993893 | 0.016796 | 0.32222 | 0.031891 |
| Random Forest | kmeans\_pca\_n9\_labels | 0.911507 | 0.994235 | 0.017839 | 0.32 | 0.033757 |
| Neural Net | Baseline | 0.770827 | 0.781653 | 0.001214 | 0.76 | 0.002423 |
| Neural Net | dbscan\_full\_e0.3\_m10\_labels | 0.868528 | 0.999391 | 0.0591 | 0.06444 | 0.061284 |
| Neural Net | dbscan\_full\_e0.3\_m10\_noise | 0.864489 | 0.999411 | 0.057675 | 0.05778 | 0.055405 |
| Neural Net | dbscan\_full\_e0.3\_m15\_labels | 0.879777 | 0.999276 | 0.062243 | 0.09111 | 0.073275 |
| Neural Net | dbscan\_full\_e0.3\_m15\_noise | 0.859366 | 0.999422 | 0.063355 | 0.06222 | 0.061827 |
| Neural Net | dbscan\_full\_e0.3\_m3\_labels | 0.861566 | 0.999319 | 0.050739 | 0.06889 | 0.057672 |
| Neural Net | dbscan\_full\_e0.3\_m3\_noise | 0.871186 | 0.999161 | 0.0484 | 0.09333 | 0.061218 |
| Neural Net | dbscan\_full\_e0.3\_m5\_labels | 0.867785 | 0.999184 | 0.045154 | 0.07778 | 0.05472 |
| Neural Net | dbscan\_full\_e0.3\_m5\_noise | 0.864562 | 0.999307 | 0.043454 | 0.05333 | 0.045915 |
| Neural Net | dbscan\_full\_e0.3\_m7\_labels | 0.873524 | 0.999416 | 0.063228 | 0.06667 | 0.064469 |
| Neural Net | dbscan\_full\_e0.3\_m7\_noise | 0.863356 | 0.999257 | 0.050201 | 0.08 | 0.061573 |
| Neural Net | dbscan\_full\_e0.5\_m10\_labels | 0.868246 | 0.9994 | 0.060872 | 0.06444 | 0.061796 |
| Neural Net | dbscan\_full\_e0.5\_m10\_noise | 0.865283 | 0.999304 | 0.06369 | 0.08444 | 0.070294 |
| Neural Net | dbscan\_full\_e0.5\_m15\_labels | 0.871763 | 0.999063 | 0.045912 | 0.08667 | 0.058486 |
| Neural Net | dbscan\_full\_e0.5\_m15\_noise | 0.864692 | 0.999411 | 0.064295 | 0.06444 | 0.063263 |
| Neural Net | dbscan\_full\_e0.5\_m3\_labels | 0.862917 | 0.999336 | 0.050451 | 0.06 | 0.053783 |
| Neural Net | dbscan\_full\_e0.5\_m3\_noise | 0.866929 | 0.999358 | 0.052638 | 0.06222 | 0.05583 |
| Neural Net | dbscan\_full\_e0.5\_m5\_labels | 0.853042 | 0.999422 | 0.068028 | 0.06 | 0.061604 |
| Neural Net | dbscan\_full\_e0.5\_m5\_noise | 0.866258 | 0.9994 | 0.047398 | 0.04889 | 0.047816 |
| Neural Net | dbscan\_full\_e0.5\_m7\_labels | 0.871621 | 0.999219 | 0.052975 | 0.08 | 0.059986 |
| Neural Net | dbscan\_full\_e0.5\_m7\_noise | 0.861745 | 0.999403 | 0.055808 | 0.05778 | 0.055713 |
| Neural Net | dbscan\_full\_e0.7\_m10\_labels | 0.852282 | 0.999373 | 0.069004 | 0.07556 | 0.071429 |
| Neural Net | dbscan\_full\_e0.7\_m10\_noise | 0.863942 | 0.999229 | 0.056607 | 0.09111 | 0.069075 |
| Neural Net | dbscan\_full\_e0.7\_m15\_labels | 0.880139 | 0.999317 | 0.057862 | 0.07556 | 0.063912 |
| Neural Net | dbscan\_full\_e0.7\_m15\_noise | 0.868585 | 0.999444 | 0.069932 | 0.06667 | 0.066957 |
| Neural Net | dbscan\_full\_e0.7\_m3\_labels | 0.876262 | 0.999408 | 0.070857 | 0.07778 | 0.072478 |
| Neural Net | dbscan\_full\_e0.7\_m3\_noise | 0.866775 | 0.998989 | 0.054385 | 0.09778 | 0.06492 |
| Neural Net | dbscan\_full\_e0.7\_m5\_labels | 0.868201 | 0.999228 | 0.053616 | 0.07333 | 0.058683 |
| Neural Net | dbscan\_full\_e0.7\_m5\_noise | 0.865304 | 0.999423 | 0.071445 | 0.06889 | 0.06875 |
| Neural Net | dbscan\_full\_e0.7\_m7\_labels | 0.863013 | 0.999417 | 0.086563 | 0.08 | 0.079884 |
| Neural Net | dbscan\_full\_e0.7\_m7\_noise | 0.861585 | 0.999341 | 0.053102 | 0.07111 | 0.058464 |
| Neural Net | dbscan\_full\_e1.0\_m10\_labels | 0.86389 | 0.999376 | 0.07506 | 0.08444 | 0.077268 |
| Neural Net | dbscan\_full\_e1.0\_m10\_noise | 0.872728 | 0.999469 | 0.073131 | 0.05111 | 0.058877 |
| Neural Net | dbscan\_full\_e1.0\_m15\_labels | 0.868907 | 0.999197 | 0.052969 | 0.09778 | 0.066733 |
| Neural Net | dbscan\_full\_e1.0\_m15\_noise | 0.872927 | 0.999339 | 0.070821 | 0.08444 | 0.073655 |
| Neural Net | dbscan\_full\_e1.0\_m3\_labels | 0.867516 | 0.999388 | 0.065456 | 0.07333 | 0.069009 |
| Neural Net | dbscan\_full\_e1.0\_m3\_noise | 0.875015 | 0.999206 | 0.045166 | 0.06889 | 0.051602 |
| Neural Net | dbscan\_full\_e1.0\_m5\_labels | 0.868963 | 0.999351 | 0.08189 | 0.09556 | 0.083826 |
| Neural Net | dbscan\_full\_e1.0\_m5\_noise | 0.868157 | 0.999339 | 0.050843 | 0.06222 | 0.053459 |
| Neural Net | dbscan\_full\_e1.0\_m7\_labels | 0.861098 | 0.999334 | 0.063091 | 0.07778 | 0.066183 |
| Neural Net | dbscan\_full\_e1.0\_m7\_noise | 0.856161 | 0.999256 | 0.047072 | 0.06889 | 0.054946 |
| Neural Net | dbscan\_full\_e1.3\_m10\_labels | 0.878426 | 0.999293 | 0.052077 | 0.07111 | 0.058264 |
| Neural Net | dbscan\_full\_e1.3\_m10\_noise | 0.874916 | 0.999204 | 0.045714 | 0.07556 | 0.056493 |
| Neural Net | dbscan\_full\_e1.3\_m15\_labels | 0.862068 | 0.999392 | 0.063347 | 0.06667 | 0.062018 |
| Neural Net | dbscan\_full\_e1.3\_m15\_noise | 0.874348 | 0.999339 | 0.07255 | 0.08889 | 0.078509 |
| Neural Net | dbscan\_full\_e1.3\_m3\_labels | 0.858352 | 0.999303 | 0.046128 | 0.06222 | 0.052614 |
| Neural Net | dbscan\_full\_e1.3\_m3\_noise | 0.878961 | 0.999325 | 0.056265 | 0.07333 | 0.062332 |
| Neural Net | dbscan\_full\_e1.3\_m5\_labels | 0.863665 | 0.999228 | 0.051468 | 0.06444 | 0.053089 |
| Neural Net | dbscan\_full\_e1.3\_m5\_noise | 0.868947 | 0.99925 | 0.048457 | 0.07556 | 0.057842 |
| Neural Net | dbscan\_full\_e1.3\_m7\_labels | 0.860441 | 0.999291 | 0.061858 | 0.08444 | 0.068945 |
| Neural Net | dbscan\_full\_e1.3\_m7\_noise | 0.869264 | 0.999265 | 0.055681 | 0.08222 | 0.064836 |
| Neural Net | dbscan\_pca\_e0.3\_m10\_labels | 0.866686 | 0.999392 | 0.058285 | 0.06222 | 0.059259 |
| Neural Net | dbscan\_pca\_e0.3\_m10\_noise | 0.87256 | 0.999325 | 0.059437 | 0.07778 | 0.065821 |
| Neural Net | dbscan\_pca\_e0.3\_m15\_labels | 0.865049 | 0.999314 | 0.067423 | 0.08889 | 0.073145 |
| Neural Net | dbscan\_pca\_e0.3\_m15\_noise | 0.868998 | 0.999317 | 0.055648 | 0.07333 | 0.058679 |
| Neural Net | dbscan\_pca\_e0.3\_m3\_labels | 0.866583 | 0.999345 | 0.069123 | 0.08889 | 0.075787 |
| Neural Net | dbscan\_pca\_e0.3\_m3\_noise | 0.878069 | 0.998706 | 0.040149 | 0.09778 | 0.054579 |
| Neural Net | dbscan\_pca\_e0.3\_m5\_labels | 0.865693 | 0.999282 | 0.056548 | 0.07556 | 0.061875 |
| Neural Net | dbscan\_pca\_e0.3\_m5\_noise | 0.872785 | 0.999216 | 0.058282 | 0.08667 | 0.063089 |
| Neural Net | dbscan\_pca\_e0.3\_m7\_labels | 0.873958 | 0.999409 | 0.064593 | 0.06667 | 0.064463 |
| Neural Net | dbscan\_pca\_e0.3\_m7\_noise | 0.862799 | 0.999334 | 0.06742 | 0.07778 | 0.068924 |
| Neural Net | dbscan\_pca\_e0.5\_m10\_labels | 0.872313 | 0.999228 | 0.057333 | 0.07778 | 0.063613 |
| Neural Net | dbscan\_pca\_e0.5\_m10\_noise | 0.873238 | 0.999331 | 0.054783 | 0.07111 | 0.061398 |
| Neural Net | dbscan\_pca\_e0.5\_m15\_labels | 0.870495 | 0.999376 | 0.047343 | 0.05333 | 0.049637 |
| Neural Net | dbscan\_pca\_e0.5\_m15\_noise | 0.875865 | 0.999367 | 0.076015 | 0.09333 | 0.083274 |
| Neural Net | dbscan\_pca\_e0.5\_m3\_labels | 0.870934 | 0.999432 | 0.074025 | 0.07111 | 0.071984 |
| Neural Net | dbscan\_pca\_e0.5\_m3\_noise | 0.859 | 0.999257 | 0.070582 | 0.10222 | 0.080673 |
| Neural Net | dbscan\_pca\_e0.5\_m5\_labels | 0.861785 | 0.999312 | 0.07147 | 0.08222 | 0.070193 |
| Neural Net | dbscan\_pca\_e0.5\_m5\_noise | 0.859521 | 0.999475 | 0.07421 | 0.05778 | 0.063978 |
| Neural Net | dbscan\_pca\_e0.5\_m7\_labels | 0.871603 | 0.999259 | 0.045648 | 0.06667 | 0.052517 |
| Neural Net | dbscan\_pca\_e0.5\_m7\_noise | 0.863944 | 0.999356 | 0.059038 | 0.06444 | 0.058614 |
| Neural Net | dbscan\_pca\_e0.7\_m10\_labels | 0.867699 | 0.999315 | 0.063086 | 0.07333 | 0.065304 |
| Neural Net | dbscan\_pca\_e0.7\_m10\_noise | 0.861327 | 0.999255 | 0.055455 | 0.07333 | 0.060315 |
| Neural Net | dbscan\_pca\_e0.7\_m15\_labels | 0.877058 | 0.999416 | 0.077222 | 0.07556 | 0.073496 |
| Neural Net | dbscan\_pca\_e0.7\_m15\_noise | 0.867208 | 0.99931 | 0.05699 | 0.07556 | 0.06412 |
| Neural Net | dbscan\_pca\_e0.7\_m3\_labels | 0.863028 | 0.999405 | 0.056936 | 0.06 | 0.05782 |
| Neural Net | dbscan\_pca\_e0.7\_m3\_noise | 0.864813 | 0.999237 | 0.048532 | 0.07556 | 0.058382 |
| Neural Net | dbscan\_pca\_e0.7\_m5\_labels | 0.88276 | 0.99936 | 0.074109 | 0.08889 | 0.078066 |
| Neural Net | dbscan\_pca\_e0.7\_m5\_noise | 0.869764 | 0.999434 | 0.085805 | 0.07333 | 0.073181 |
| Neural Net | dbscan\_pca\_e0.7\_m7\_labels | 0.858618 | 0.999398 | 0.074195 | 0.07333 | 0.071312 |
| Neural Net | dbscan\_pca\_e0.7\_m7\_noise | 0.873153 | 0.999339 | 0.077147 | 0.09111 | 0.080146 |
| Neural Net | dbscan\_pca\_e1.0\_m10\_labels | 0.863515 | 0.999387 | 0.051453 | 0.05556 | 0.053357 |
| Neural Net | dbscan\_pca\_e1.0\_m10\_noise | 0.871345 | 0.999226 | 0.06278 | 0.09778 | 0.073577 |
| Neural Net | dbscan\_pca\_e1.0\_m15\_labels | 0.872819 | 0.999385 | 0.070125 | 0.08 | 0.073628 |
| Neural Net | dbscan\_pca\_e1.0\_m15\_noise | 0.872594 | 0.999061 | 0.05768 | 0.09556 | 0.063566 |
| Neural Net | dbscan\_pca\_e1.0\_m3\_labels | 0.875601 | 0.99946 | 0.094963 | 0.08444 | 0.087635 |
| Neural Net | dbscan\_pca\_e1.0\_m3\_noise | 0.866443 | 0.999168 | 0.051932 | 0.08889 | 0.062084 |
| Neural Net | dbscan\_pca\_e1.0\_m5\_labels | 0.878087 | 0.999329 | 0.077876 | 0.1 | 0.086057 |
| Neural Net | dbscan\_pca\_e1.0\_m5\_noise | 0.870007 | 0.999355 | 0.053099 | 0.06222 | 0.056907 |
| Neural Net | dbscan\_pca\_e1.0\_m7\_labels | 0.864518 | 0.999252 | 0.062547 | 0.10222 | 0.076041 |
| Neural Net | dbscan\_pca\_e1.0\_m7\_noise | 0.862856 | 0.999418 | 0.065534 | 0.06667 | 0.064395 |
| Neural Net | dbscan\_pca\_e1.3\_m10\_labels | 0.860402 | 0.999401 | 0.063856 | 0.06667 | 0.064727 |
| Neural Net | dbscan\_pca\_e1.3\_m10\_noise | 0.870084 | 0.999422 | 0.073099 | 0.07111 | 0.070558 |
| Neural Net | dbscan\_pca\_e1.3\_m15\_labels | 0.864005 | 0.999487 | 0.064388 | 0.04889 | 0.054945 |
| Neural Net | dbscan\_pca\_e1.3\_m15\_noise | 0.864792 | 0.99942 | 0.058733 | 0.05778 | 0.058048 |
| Neural Net | dbscan\_pca\_e1.3\_m3\_labels | 0.871428 | 0.999368 | 0.068684 | 0.07333 | 0.067095 |
| Neural Net | dbscan\_pca\_e1.3\_m3\_noise | 0.873423 | 0.999326 | 0.069309 | 0.09556 | 0.079201 |
| Neural Net | dbscan\_pca\_e1.3\_m5\_labels | 0.85585 | 0.999252 | 0.066661 | 0.09333 | 0.072924 |
| Neural Net | dbscan\_pca\_e1.3\_m5\_noise | 0.872032 | 0.999356 | 0.061587 | 0.06889 | 0.064093 |
| Neural Net | dbscan\_pca\_e1.3\_m7\_labels | 0.869177 | 0.999482 | 0.095216 | 0.07333 | 0.081518 |
| Neural Net | dbscan\_pca\_e1.3\_m7\_noise | 0.872255 | 0.999368 | 0.075812 | 0.08 | 0.074498 |
| Neural Net | hier\_full\_n2\_labels | 0.870668 | 0.999215 | 0.062664 | 0.09556 | 0.067378 |
| Neural Net | hier\_full\_n3\_labels | 0.878782 | 0.999281 | 0.057424 | 0.07333 | 0.059156 |
| Neural Net | hier\_full\_n4\_labels | 0.875615 | 0.999348 | 0.055342 | 0.07333 | 0.061511 |
| Neural Net | hier\_full\_n5\_labels | 0.87487 | 0.999384 | 0.0802 | 0.08889 | 0.08193 |
| Neural Net | hier\_full\_n6\_labels | 0.863894 | 0.999314 | 0.054757 | 0.07111 | 0.061044 |
| Neural Net | hier\_pca\_n2\_labels | 0.875529 | 0.999127 | 0.06091 | 0.11111 | 0.073334 |
| Neural Net | hier\_pca\_n3\_labels | 0.864524 | 0.999309 | 0.066973 | 0.09333 | 0.077312 |
| Neural Net | hier\_pca\_n4\_labels | 0.869917 | 0.999281 | 0.069028 | 0.10444 | 0.077317 |
| Neural Net | hier\_pca\_n5\_labels | 0.870733 | 0.999305 | 0.052807 | 0.06667 | 0.055578 |
| Neural Net | hier\_pca\_n6\_labels | 0.858722 | 0.999419 | 0.064363 | 0.06222 | 0.062473 |
| Neural Net | kmeans\_full\_n10\_labels | 0.859367 | 0.999312 | 0.049325 | 0.07111 | 0.056965 |
| Neural Net | kmeans\_full\_n2\_labels | 0.871413 | 0.999426 | 0.074829 | 0.07556 | 0.074121 |
| Neural Net | kmeans\_full\_n3\_labels | 0.872493 | 0.999316 | 0.055912 | 0.07111 | 0.059476 |
| Neural Net | kmeans\_full\_n4\_labels | 0.874098 | 0.999336 | 0.058806 | 0.07556 | 0.061756 |
| Neural Net | kmeans\_full\_n5\_labels | 0.868865 | 0.999336 | 0.069896 | 0.08667 | 0.073587 |
| Neural Net | kmeans\_full\_n6\_labels | 0.87407 | 0.999407 | 0.065944 | 0.06889 | 0.066978 |
| Neural Net | kmeans\_full\_n7\_labels | 0.879235 | 0.999259 | 0.062919 | 0.07333 | 0.061843 |
| Neural Net | kmeans\_full\_n8\_labels | 0.876362 | 0.999093 | 0.055661 | 0.11556 | 0.07262 |
| Neural Net | kmeans\_full\_n9\_labels | 0.864343 | 0.999282 | 0.06297 | 0.08667 | 0.068762 |
| Neural Net | kmeans\_pca\_n10\_labels | 0.866677 | 0.999371 | 0.061574 | 0.07556 | 0.062963 |
| Neural Net | kmeans\_pca\_n2\_labels | 0.865366 | 0.999281 | 0.054037 | 0.07333 | 0.060098 |
| Neural Net | kmeans\_pca\_n3\_labels | 0.875032 | 0.999078 | 0.046242 | 0.06222 | 0.049058 |
| Neural Net | kmeans\_pca\_n4\_labels | 0.866141 | 0.999264 | 0.052082 | 0.08222 | 0.062112 |
| Neural Net | kmeans\_pca\_n5\_labels | 0.864153 | 0.999273 | 0.05074 | 0.07778 | 0.060234 |
| Neural Net | kmeans\_pca\_n6\_labels | 0.872416 | 0.999339 | 0.059179 | 0.07333 | 0.064333 |
| Neural Net | kmeans\_pca\_n7\_labels | 0.872971 | 0.999299 | 0.062446 | 0.08444 | 0.068059 |
| Neural Net | kmeans\_pca\_n8\_labels | 0.863321 | 0.999263 | 0.060812 | 0.08444 | 0.066132 |
| Neural Net | kmeans\_pca\_n9\_labels | 0.856904 | 0.999333 | 0.039138 | 0.05111 | 0.042771 |
| Naive Bayes | Baseline | 0.852439 | 0.882893 | 0.001891 | 0.71111 | 0.003771 |
| Naive Bayes | dbscan\_full\_e0.3\_m10\_labels | 0.5 | 0.00031 | 0.00031 | 1 | 0.000619 |
| Naive Bayes | dbscan\_full\_e0.3\_m10\_noise | 0.5 | 0.00031 | 0.00031 | 1 | 0.000619 |
| Naive Bayes | dbscan\_full\_e0.3\_m15\_labels | 0.5 | 0.00031 | 0.00031 | 1 | 0.000619 |
| Naive Bayes | dbscan\_full\_e0.3\_m15\_noise | 0.5 | 0.00031 | 0.00031 | 1 | 0.000619 |
| Naive Bayes | dbscan\_full\_e0.3\_m3\_labels | 0.5 | 0.00031 | 0.00031 | 1 | 0.000619 |
| Naive Bayes | dbscan\_full\_e0.3\_m3\_noise | 0.5 | 0.00031 | 0.00031 | 1 | 0.000619 |
| Naive Bayes | dbscan\_full\_e0.3\_m5\_labels | 0.5 | 0.00031 | 0.00031 | 1 | 0.000619 |
| Naive Bayes | dbscan\_full\_e0.3\_m5\_noise | 0.5 | 0.00031 | 0.00031 | 1 | 0.000619 |
| Naive Bayes | dbscan\_full\_e0.3\_m7\_labels | 0.5 | 0.00031 | 0.00031 | 1 | 0.000619 |
| Naive Bayes | dbscan\_full\_e0.3\_m7\_noise | 0.5 | 0.00031 | 0.00031 | 1 | 0.000619 |
| Naive Bayes | dbscan\_full\_e0.5\_m10\_labels | 0.5 | 0.00031 | 0.00031 | 1 | 0.000619 |
| Naive Bayes | dbscan\_full\_e0.5\_m10\_noise | 0.5 | 0.00031 | 0.00031 | 1 | 0.000619 |
| Naive Bayes | dbscan\_full\_e0.5\_m15\_labels | 0.5 | 0.00031 | 0.00031 | 1 | 0.000619 |
| Naive Bayes | dbscan\_full\_e0.5\_m15\_noise | 0.5 | 0.00031 | 0.00031 | 1 | 0.000619 |
| Naive Bayes | dbscan\_full\_e0.5\_m3\_labels | 0.5 | 0.00031 | 0.00031 | 1 | 0.000619 |
| Naive Bayes | dbscan\_full\_e0.5\_m3\_noise | 0.5 | 0.00031 | 0.00031 | 1 | 0.000619 |
| Naive Bayes | dbscan\_full\_e0.5\_m5\_labels | 0.5 | 0.00031 | 0.00031 | 1 | 0.000619 |
| Naive Bayes | dbscan\_full\_e0.5\_m5\_noise | 0.5 | 0.00031 | 0.00031 | 1 | 0.000619 |
| Naive Bayes | dbscan\_full\_e0.5\_m7\_labels | 0.5 | 0.00031 | 0.00031 | 1 | 0.000619 |
| Naive Bayes | dbscan\_full\_e0.5\_m7\_noise | 0.5 | 0.00031 | 0.00031 | 1 | 0.000619 |
| Naive Bayes | dbscan\_full\_e0.7\_m10\_labels | 0.5 | 0.00031 | 0.00031 | 1 | 0.000619 |
| Naive Bayes | dbscan\_full\_e0.7\_m10\_noise | 0.5 | 0.00031 | 0.00031 | 1 | 0.000619 |
| Naive Bayes | dbscan\_full\_e0.7\_m15\_labels | 0.5 | 0.00031 | 0.00031 | 1 | 0.000619 |
| Naive Bayes | dbscan\_full\_e0.7\_m15\_noise | 0.5 | 0.00031 | 0.00031 | 1 | 0.000619 |
| Naive Bayes | dbscan\_full\_e0.7\_m3\_labels | 0.5 | 0.00031 | 0.00031 | 1 | 0.000619 |
| Naive Bayes | dbscan\_full\_e0.7\_m3\_noise | 0.5 | 0.00031 | 0.00031 | 1 | 0.000619 |
| Naive Bayes | dbscan\_full\_e0.7\_m5\_labels | 0.5 | 0.00031 | 0.00031 | 1 | 0.000619 |
| Naive Bayes | dbscan\_full\_e0.7\_m5\_noise | 0.5 | 0.00031 | 0.00031 | 1 | 0.000619 |
| Naive Bayes | dbscan\_full\_e0.7\_m7\_labels | 0.5 | 0.00031 | 0.00031 | 1 | 0.000619 |
| Naive Bayes | dbscan\_full\_e0.7\_m7\_noise | 0.5 | 0.00031 | 0.00031 | 1 | 0.000619 |
| Naive Bayes | dbscan\_full\_e1.0\_m10\_labels | 0.5 | 0.00031 | 0.00031 | 1 | 0.000619 |
| Naive Bayes | dbscan\_full\_e1.0\_m10\_noise | 0.5 | 0.00031 | 0.00031 | 1 | 0.000619 |
| Naive Bayes | dbscan\_full\_e1.0\_m15\_labels | 0.5 | 0.00031 | 0.00031 | 1 | 0.000619 |
| Naive Bayes | dbscan\_full\_e1.0\_m15\_noise | 0.5 | 0.00031 | 0.00031 | 1 | 0.000619 |
| Naive Bayes | dbscan\_full\_e1.0\_m3\_labels | 0.708583 | 0.08158 | 0.000331 | 0.98222 | 0.000662 |
| Naive Bayes | dbscan\_full\_e1.0\_m3\_noise | 0.5 | 0.00031 | 0.00031 | 1 | 0.000619 |
| Naive Bayes | dbscan\_full\_e1.0\_m5\_labels | 0.5 | 0.00031 | 0.00031 | 1 | 0.000619 |
| Naive Bayes | dbscan\_full\_e1.0\_m5\_noise | 0.5 | 0.00031 | 0.00031 | 1 | 0.000619 |
| Naive Bayes | dbscan\_full\_e1.0\_m7\_labels | 0.5 | 0.00031 | 0.00031 | 1 | 0.000619 |
| Naive Bayes | dbscan\_full\_e1.0\_m7\_noise | 0.5 | 0.00031 | 0.00031 | 1 | 0.000619 |
| Naive Bayes | dbscan\_full\_e1.3\_m10\_labels | 0.5 | 0.00031 | 0.00031 | 1 | 0.000619 |
| Naive Bayes | dbscan\_full\_e1.3\_m10\_noise | 0.5 | 0.00031 | 0.00031 | 1 | 0.000619 |
| Naive Bayes | dbscan\_full\_e1.3\_m15\_labels | 0.5 | 0.00031 | 0.00031 | 1 | 0.000619 |
| Naive Bayes | dbscan\_full\_e1.3\_m15\_noise | 0.5 | 0.00031 | 0.00031 | 1 | 0.000619 |
| Naive Bayes | dbscan\_full\_e1.3\_m3\_labels | 0.860041 | 0.527667 | 0.000583 | 0.88667 | 0.001165 |
| Naive Bayes | dbscan\_full\_e1.3\_m3\_noise | 0.5 | 0.00031 | 0.00031 | 1 | 0.000619 |
| Naive Bayes | dbscan\_full\_e1.3\_m5\_labels | 0.680614 | 0.063993 | 0.000326 | 0.98667 | 0.000653 |
| Naive Bayes | dbscan\_full\_e1.3\_m5\_noise | 0.5 | 0.00031 | 0.00031 | 1 | 0.000619 |
| Naive Bayes | dbscan\_full\_e1.3\_m7\_labels | 0.5 | 0.00031 | 0.00031 | 1 | 0.000619 |
| Naive Bayes | dbscan\_full\_e1.3\_m7\_noise | 0.5 | 0.00031 | 0.00031 | 1 | 0.000619 |
| Naive Bayes | dbscan\_pca\_e0.3\_m10\_labels | 0.5 | 0.00031 | 0.00031 | 1 | 0.000619 |
| Naive Bayes | dbscan\_pca\_e0.3\_m10\_noise | 0.5 | 0.00031 | 0.00031 | 1 | 0.000619 |
| Naive Bayes | dbscan\_pca\_e0.3\_m15\_labels | 0.5 | 0.00031 | 0.00031 | 1 | 0.000619 |
| Naive Bayes | dbscan\_pca\_e0.3\_m15\_noise | 0.5 | 0.00031 | 0.00031 | 1 | 0.000619 |
| Naive Bayes | dbscan\_pca\_e0.3\_m3\_labels | 0.569483 | 0.063259 | 0.000333 | 0.98889 | 0.000666 |
| Naive Bayes | dbscan\_pca\_e0.3\_m3\_noise | 0.5 | 0.00031 | 0.00031 | 1 | 0.000619 |
| Naive Bayes | dbscan\_pca\_e0.3\_m5\_labels | 0.5 | 0.00031 | 0.00031 | 1 | 0.000619 |
| Naive Bayes | dbscan\_pca\_e0.3\_m5\_noise | 0.5 | 0.00031 | 0.00031 | 1 | 0.000619 |
| Naive Bayes | dbscan\_pca\_e0.3\_m7\_labels | 0.5 | 0.00031 | 0.00031 | 1 | 0.000619 |
| Naive Bayes | dbscan\_pca\_e0.3\_m7\_noise | 0.5 | 0.00031 | 0.00031 | 1 | 0.000619 |
| Naive Bayes | dbscan\_pca\_e0.5\_m10\_labels | 0.5 | 0.00031 | 0.00031 | 1 | 0.000619 |
| Naive Bayes | dbscan\_pca\_e0.5\_m10\_noise | 0.5 | 0.00031 | 0.00031 | 1 | 0.000619 |
| Naive Bayes | dbscan\_pca\_e0.5\_m15\_labels | 0.5 | 0.00031 | 0.00031 | 1 | 0.000619 |
| Naive Bayes | dbscan\_pca\_e0.5\_m15\_noise | 0.5 | 0.00031 | 0.00031 | 1 | 0.000619 |
| Naive Bayes | dbscan\_pca\_e0.5\_m3\_labels | 0.569482 | 0.063259 | 0.000333 | 0.98889 | 0.000666 |
| Naive Bayes | dbscan\_pca\_e0.5\_m3\_noise | 0.5 | 0.00031 | 0.00031 | 1 | 0.000619 |
| Naive Bayes | dbscan\_pca\_e0.5\_m5\_labels | 0.5 | 0.00031 | 0.00031 | 1 | 0.000619 |
| Naive Bayes | dbscan\_pca\_e0.5\_m5\_noise | 0.5 | 0.00031 | 0.00031 | 1 | 0.000619 |
| Naive Bayes | dbscan\_pca\_e0.5\_m7\_labels | 0.5 | 0.00031 | 0.00031 | 1 | 0.000619 |
| Naive Bayes | dbscan\_pca\_e0.5\_m7\_noise | 0.5 | 0.00031 | 0.00031 | 1 | 0.000619 |
| Naive Bayes | dbscan\_pca\_e0.7\_m10\_labels | 0.5 | 0.00031 | 0.00031 | 1 | 0.000619 |
| Naive Bayes | dbscan\_pca\_e0.7\_m10\_noise | 0.5 | 0.00031 | 0.00031 | 1 | 0.000619 |
| Naive Bayes | dbscan\_pca\_e0.7\_m15\_labels | 0.5 | 0.00031 | 0.00031 | 1 | 0.000619 |
| Naive Bayes | dbscan\_pca\_e0.7\_m15\_noise | 0.5 | 0.00031 | 0.00031 | 1 | 0.000619 |
| Naive Bayes | dbscan\_pca\_e0.7\_m3\_labels | 0.782038 | 0.291603 | 0.000423 | 0.94 | 0.000846 |
| Naive Bayes | dbscan\_pca\_e0.7\_m3\_noise | 0.5 | 0.00031 | 0.00031 | 1 | 0.000619 |
| Naive Bayes | dbscan\_pca\_e0.7\_m5\_labels | 0.5 | 0.00031 | 0.00031 | 1 | 0.000619 |
| Naive Bayes | dbscan\_pca\_e0.7\_m5\_noise | 0.5 | 0.00031 | 0.00031 | 1 | 0.000619 |
| Naive Bayes | dbscan\_pca\_e0.7\_m7\_labels | 0.5 | 0.00031 | 0.00031 | 1 | 0.000619 |
| Naive Bayes | dbscan\_pca\_e0.7\_m7\_noise | 0.5 | 0.00031 | 0.00031 | 1 | 0.000619 |
| Naive Bayes | dbscan\_pca\_e1.0\_m10\_labels | 0.5 | 0.00031 | 0.00031 | 1 | 0.000619 |
| Naive Bayes | dbscan\_pca\_e1.0\_m10\_noise | 0.5 | 0.00031 | 0.00031 | 1 | 0.000619 |
| Naive Bayes | dbscan\_pca\_e1.0\_m15\_labels | 0.5 | 0.00031 | 0.00031 | 1 | 0.000619 |
| Naive Bayes | dbscan\_pca\_e1.0\_m15\_noise | 0.5 | 0.00031 | 0.00031 | 1 | 0.000619 |
| Naive Bayes | dbscan\_pca\_e1.0\_m3\_labels | 0.782038 | 0.291602 | 0.000423 | 0.94 | 0.000846 |
| Naive Bayes | dbscan\_pca\_e1.0\_m3\_noise | 0.5 | 0.00031 | 0.00031 | 1 | 0.000619 |
| Naive Bayes | dbscan\_pca\_e1.0\_m5\_labels | 0.5 | 0.00031 | 0.00031 | 1 | 0.000619 |
| Naive Bayes | dbscan\_pca\_e1.0\_m5\_noise | 0.5 | 0.00031 | 0.00031 | 1 | 0.000619 |
| Naive Bayes | dbscan\_pca\_e1.0\_m7\_labels | 0.5 | 0.00031 | 0.00031 | 1 | 0.000619 |
| Naive Bayes | dbscan\_pca\_e1.0\_m7\_noise | 0.5 | 0.00031 | 0.00031 | 1 | 0.000619 |
| Naive Bayes | dbscan\_pca\_e1.3\_m10\_labels | 0.5 | 0.00031 | 0.00031 | 1 | 0.000619 |
| Naive Bayes | dbscan\_pca\_e1.3\_m10\_noise | 0.5 | 0.00031 | 0.00031 | 1 | 0.000619 |
| Naive Bayes | dbscan\_pca\_e1.3\_m15\_labels | 0.5 | 0.00031 | 0.00031 | 1 | 0.000619 |
| Naive Bayes | dbscan\_pca\_e1.3\_m15\_noise | 0.5 | 0.00031 | 0.00031 | 1 | 0.000619 |
| Naive Bayes | dbscan\_pca\_e1.3\_m3\_labels | 0.784011 | 0.320423 | 0.00044 | 0.93111 | 0.00088 |
| Naive Bayes | dbscan\_pca\_e1.3\_m3\_noise | 0.5 | 0.00031 | 0.00031 | 1 | 0.000619 |
| Naive Bayes | dbscan\_pca\_e1.3\_m5\_labels | 0.5 | 0.00031 | 0.00031 | 1 | 0.000619 |
| Naive Bayes | dbscan\_pca\_e1.3\_m5\_noise | 0.5 | 0.00031 | 0.00031 | 1 | 0.000619 |
| Naive Bayes | dbscan\_pca\_e1.3\_m7\_labels | 0.5 | 0.00031 | 0.00031 | 1 | 0.000619 |
| Naive Bayes | dbscan\_pca\_e1.3\_m7\_noise | 0.5 | 0.00031 | 0.00031 | 1 | 0.000619 |
| Naive Bayes | hier\_full\_n2\_labels | 0.5 | 0.00031 | 0.00031 | 1 | 0.000619 |
| Naive Bayes | hier\_full\_n3\_labels | 0.5 | 0.00031 | 0.00031 | 1 | 0.000619 |
| Naive Bayes | hier\_full\_n4\_labels | 0.5 | 0.00031 | 0.00031 | 1 | 0.000619 |
| Naive Bayes | hier\_full\_n5\_labels | 0.5 | 0.00031 | 0.00031 | 1 | 0.000619 |
| Naive Bayes | hier\_full\_n6\_labels | 0.5 | 0.00031 | 0.00031 | 1 | 0.000619 |
| Naive Bayes | hier\_pca\_n2\_labels | 0.5 | 0.00031 | 0.00031 | 1 | 0.000619 |
| Naive Bayes | hier\_pca\_n3\_labels | 0.5 | 0.00031 | 0.00031 | 1 | 0.000619 |
| Naive Bayes | hier\_pca\_n4\_labels | 0.5 | 0.00031 | 0.00031 | 1 | 0.000619 |
| Naive Bayes | hier\_pca\_n5\_labels | 0.5 | 0.00031 | 0.00031 | 1 | 0.000619 |
| Naive Bayes | hier\_pca\_n6\_labels | 0.5 | 0.00031 | 0.00031 | 1 | 0.000619 |
| Naive Bayes | kmeans\_full\_n10\_labels | 0.5 | 0.00031 | 0.00031 | 1 | 0.000619 |
| Naive Bayes | kmeans\_full\_n2\_labels | 0.5 | 0.00031 | 0.00031 | 1 | 0.000619 |
| Naive Bayes | kmeans\_full\_n3\_labels | 0.5 | 0.00031 | 0.00031 | 1 | 0.000619 |
| Naive Bayes | kmeans\_full\_n4\_labels | 0.5 | 0.00031 | 0.00031 | 1 | 0.000619 |
| Naive Bayes | kmeans\_full\_n5\_labels | 0.5 | 0.00031 | 0.00031 | 1 | 0.000619 |
| Naive Bayes | kmeans\_full\_n6\_labels | 0.5 | 0.00031 | 0.00031 | 1 | 0.000619 |
| Naive Bayes | kmeans\_full\_n7\_labels | 0.5 | 0.00031 | 0.00031 | 1 | 0.000619 |
| Naive Bayes | kmeans\_full\_n8\_labels | 0.5 | 0.00031 | 0.00031 | 1 | 0.000619 |
| Naive Bayes | kmeans\_full\_n9\_labels | 0.5 | 0.00031 | 0.00031 | 1 | 0.000619 |
| Naive Bayes | kmeans\_pca\_n10\_labels | 0.5 | 0.00031 | 0.00031 | 1 | 0.000619 |
| Naive Bayes | kmeans\_pca\_n2\_labels | 0.5 | 0.00031 | 0.00031 | 1 | 0.000619 |
| Naive Bayes | kmeans\_pca\_n3\_labels | 0.5 | 0.00031 | 0.00031 | 1 | 0.000619 |
| Naive Bayes | kmeans\_pca\_n4\_labels | 0.5 | 0.00031 | 0.00031 | 1 | 0.000619 |
| Naive Bayes | kmeans\_pca\_n5\_labels | 0.5 | 0.00031 | 0.00031 | 1 | 0.000619 |
| Naive Bayes | kmeans\_pca\_n6\_labels | 0.5 | 0.00031 | 0.00031 | 1 | 0.000619 |
| Naive Bayes | kmeans\_pca\_n7\_labels | 0.5 | 0.00031 | 0.00031 | 1 | 0.000619 |
| Naive Bayes | kmeans\_pca\_n8\_labels | 0.5 | 0.00031 | 0.00031 | 1 | 0.000619 |
| Naive Bayes | kmeans\_pca\_n9\_labels | 0.5 | 0.00031 | 0.00031 | 1 | 0.000619 |
| LogReg | Baseline | 0.873006 | 0.941157 | 0.003573 | 0.67333 | 0.007108 |
| LogReg | dbscan\_full\_e0.3\_m10\_labels | 0.874432 | 0.908862 | 0.002515 | 0.74 | 0.005012 |
| LogReg | dbscan\_full\_e0.3\_m10\_noise | 0.874432 | 0.908862 | 0.002515 | 0.74 | 0.005012 |
| LogReg | dbscan\_full\_e0.3\_m15\_labels | 0.874432 | 0.908862 | 0.002515 | 0.74 | 0.005012 |
| LogReg | dbscan\_full\_e0.3\_m15\_noise | 0.874432 | 0.908862 | 0.002515 | 0.74 | 0.005012 |
| LogReg | dbscan\_full\_e0.3\_m3\_labels | 0.87453 | 0.908803 | 0.002505 | 0.73778 | 0.004994 |
| LogReg | dbscan\_full\_e0.3\_m3\_noise | 0.874339 | 0.908855 | 0.002515 | 0.74 | 0.005012 |
| LogReg | dbscan\_full\_e0.3\_m5\_labels | 0.874358 | 0.908846 | 0.002514 | 0.74 | 0.005011 |
| LogReg | dbscan\_full\_e0.3\_m5\_noise | 0.874257 | 0.908882 | 0.002515 | 0.74 | 0.005014 |
| LogReg | dbscan\_full\_e0.3\_m7\_labels | 0.874432 | 0.908862 | 0.002515 | 0.74 | 0.005012 |
| LogReg | dbscan\_full\_e0.3\_m7\_noise | 0.874432 | 0.908862 | 0.002515 | 0.74 | 0.005012 |
| LogReg | dbscan\_full\_e0.5\_m10\_labels | 0.874432 | 0.908862 | 0.002515 | 0.74 | 0.005012 |
| LogReg | dbscan\_full\_e0.5\_m10\_noise | 0.874432 | 0.908862 | 0.002515 | 0.74 | 0.005012 |
| LogReg | dbscan\_full\_e0.5\_m15\_labels | 0.874432 | 0.908862 | 0.002515 | 0.74 | 0.005012 |
| LogReg | dbscan\_full\_e0.5\_m15\_noise | 0.874432 | 0.908862 | 0.002515 | 0.74 | 0.005012 |
| LogReg | dbscan\_full\_e0.5\_m3\_labels | 0.874271 | 0.908839 | 0.002506 | 0.73778 | 0.004995 |
| LogReg | dbscan\_full\_e0.5\_m3\_noise | 0.8743 | 0.908807 | 0.002506 | 0.73778 | 0.004994 |
| LogReg | dbscan\_full\_e0.5\_m5\_labels | 0.874532 | 0.908749 | 0.002512 | 0.74 | 0.005006 |
| LogReg | dbscan\_full\_e0.5\_m5\_noise | 0.874258 | 0.908885 | 0.002515 | 0.74 | 0.005014 |
| LogReg | dbscan\_full\_e0.5\_m7\_labels | 0.874336 | 0.90888 | 0.002515 | 0.74 | 0.005013 |
| LogReg | dbscan\_full\_e0.5\_m7\_noise | 0.874247 | 0.908883 | 0.002515 | 0.74 | 0.005014 |
| LogReg | dbscan\_full\_e0.7\_m10\_labels | 0.874276 | 0.908858 | 0.002515 | 0.74 | 0.005012 |
| LogReg | dbscan\_full\_e0.7\_m10\_noise | 0.87444 | 0.908882 | 0.002515 | 0.74 | 0.005013 |
| LogReg | dbscan\_full\_e0.7\_m15\_labels | 0.874432 | 0.908862 | 0.002515 | 0.74 | 0.005012 |
| LogReg | dbscan\_full\_e0.7\_m15\_noise | 0.874432 | 0.908862 | 0.002515 | 0.74 | 0.005012 |
| LogReg | dbscan\_full\_e0.7\_m3\_labels | 0.874472 | 0.908897 | 0.002516 | 0.74 | 0.005014 |
| LogReg | dbscan\_full\_e0.7\_m3\_noise | 0.874528 | 0.908796 | 0.002505 | 0.73778 | 0.004993 |
| LogReg | dbscan\_full\_e0.7\_m5\_labels | 0.874394 | 0.908917 | 0.002508 | 0.73778 | 0.005 |
| LogReg | dbscan\_full\_e0.7\_m5\_noise | 0.874262 | 0.908831 | 0.002514 | 0.74 | 0.005011 |
| LogReg | dbscan\_full\_e0.7\_m7\_labels | 0.874338 | 0.908756 | 0.002512 | 0.74 | 0.005007 |
| LogReg | dbscan\_full\_e0.7\_m7\_noise | 0.87426 | 0.908896 | 0.002516 | 0.74 | 0.005014 |
| LogReg | dbscan\_full\_e1.0\_m10\_labels | 0.874344 | 0.908765 | 0.002512 | 0.74 | 0.005007 |
| LogReg | dbscan\_full\_e1.0\_m10\_noise | 0.874302 | 0.908866 | 0.002515 | 0.74 | 0.005012 |
| LogReg | dbscan\_full\_e1.0\_m15\_labels | 0.874432 | 0.908862 | 0.002515 | 0.74 | 0.005012 |
| LogReg | dbscan\_full\_e1.0\_m15\_noise | 0.874432 | 0.908862 | 0.002515 | 0.74 | 0.005012 |
| LogReg | dbscan\_full\_e1.0\_m3\_labels | 0.874735 | 0.908841 | 0.002514 | 0.74 | 0.005011 |
| LogReg | dbscan\_full\_e1.0\_m3\_noise | 0.874681 | 0.908827 | 0.002514 | 0.74 | 0.005011 |
| LogReg | dbscan\_full\_e1.0\_m5\_labels | 0.874407 | 0.908824 | 0.002514 | 0.74 | 0.00501 |
| LogReg | dbscan\_full\_e1.0\_m5\_noise | 0.874407 | 0.908799 | 0.002505 | 0.73778 | 0.004994 |
| LogReg | dbscan\_full\_e1.0\_m7\_labels | 0.874359 | 0.908961 | 0.002518 | 0.74 | 0.005018 |
| LogReg | dbscan\_full\_e1.0\_m7\_noise | 0.874341 | 0.908817 | 0.002506 | 0.73778 | 0.004994 |
| LogReg | dbscan\_full\_e1.3\_m10\_labels | 0.874374 | 0.908862 | 0.002507 | 0.73778 | 0.004997 |
| LogReg | dbscan\_full\_e1.3\_m10\_noise | 0.87434 | 0.908848 | 0.002507 | 0.73778 | 0.004996 |
| LogReg | dbscan\_full\_e1.3\_m15\_labels | 0.874331 | 0.908837 | 0.002514 | 0.74 | 0.005011 |
| LogReg | dbscan\_full\_e1.3\_m15\_noise | 0.874363 | 0.908855 | 0.002515 | 0.74 | 0.005012 |
| LogReg | dbscan\_full\_e1.3\_m3\_labels | 0.875413 | 0.908462 | 0.002496 | 0.73778 | 0.004975 |
| LogReg | dbscan\_full\_e1.3\_m3\_noise | 0.874835 | 0.908681 | 0.002502 | 0.73778 | 0.004986 |
| LogReg | dbscan\_full\_e1.3\_m5\_labels | 0.87475 | 0.908826 | 0.002514 | 0.74 | 0.00501 |
| LogReg | dbscan\_full\_e1.3\_m5\_noise | 0.874768 | 0.90872 | 0.002503 | 0.73778 | 0.004989 |
| LogReg | dbscan\_full\_e1.3\_m7\_labels | 0.874533 | 0.908835 | 0.002514 | 0.74 | 0.005011 |
| LogReg | dbscan\_full\_e1.3\_m7\_noise | 0.874459 | 0.908756 | 0.002504 | 0.73778 | 0.004991 |
| LogReg | dbscan\_pca\_e0.3\_m10\_labels | 0.874432 | 0.908862 | 0.002515 | 0.74 | 0.005012 |
| LogReg | dbscan\_pca\_e0.3\_m10\_noise | 0.874432 | 0.908862 | 0.002515 | 0.74 | 0.005012 |
| LogReg | dbscan\_pca\_e0.3\_m15\_labels | 0.874432 | 0.908862 | 0.002515 | 0.74 | 0.005012 |
| LogReg | dbscan\_pca\_e0.3\_m15\_noise | 0.874432 | 0.908862 | 0.002515 | 0.74 | 0.005012 |
| LogReg | dbscan\_pca\_e0.3\_m3\_labels | 0.874063 | 0.908853 | 0.002514 | 0.74 | 0.005012 |
| LogReg | dbscan\_pca\_e0.3\_m3\_noise | 0.874432 | 0.909127 | 0.002522 | 0.74 | 0.005027 |
| LogReg | dbscan\_pca\_e0.3\_m5\_labels | 0.874447 | 0.908906 | 0.002516 | 0.74 | 0.005015 |
| LogReg | dbscan\_pca\_e0.3\_m5\_noise | 0.874466 | 0.909111 | 0.002514 | 0.73778 | 0.005011 |
| LogReg | dbscan\_pca\_e0.3\_m7\_labels | 0.874522 | 0.909095 | 0.002513 | 0.73778 | 0.00501 |
| LogReg | dbscan\_pca\_e0.3\_m7\_noise | 0.874525 | 0.909139 | 0.002515 | 0.73778 | 0.005012 |
| LogReg | dbscan\_pca\_e0.5\_m10\_labels | 0.874432 | 0.908862 | 0.002515 | 0.74 | 0.005012 |
| LogReg | dbscan\_pca\_e0.5\_m10\_noise | 0.874432 | 0.908862 | 0.002515 | 0.74 | 0.005012 |
| LogReg | dbscan\_pca\_e0.5\_m15\_labels | 0.874432 | 0.908862 | 0.002515 | 0.74 | 0.005012 |
| LogReg | dbscan\_pca\_e0.5\_m15\_noise | 0.874432 | 0.908862 | 0.002515 | 0.74 | 0.005012 |
| LogReg | dbscan\_pca\_e0.5\_m3\_labels | 0.874063 | 0.908853 | 0.002514 | 0.74 | 0.005012 |
| LogReg | dbscan\_pca\_e0.5\_m3\_noise | 0.874432 | 0.909127 | 0.002522 | 0.74 | 0.005027 |
| LogReg | dbscan\_pca\_e0.5\_m5\_labels | 0.874447 | 0.908906 | 0.002516 | 0.74 | 0.005015 |
| LogReg | dbscan\_pca\_e0.5\_m5\_noise | 0.874466 | 0.909111 | 0.002514 | 0.73778 | 0.005011 |
| LogReg | dbscan\_pca\_e0.5\_m7\_labels | 0.874522 | 0.909095 | 0.002513 | 0.73778 | 0.00501 |
| LogReg | dbscan\_pca\_e0.5\_m7\_noise | 0.874525 | 0.909139 | 0.002515 | 0.73778 | 0.005012 |
| LogReg | dbscan\_pca\_e0.7\_m10\_labels | 0.874432 | 0.908862 | 0.002515 | 0.74 | 0.005012 |
| LogReg | dbscan\_pca\_e0.7\_m10\_noise | 0.874432 | 0.908862 | 0.002515 | 0.74 | 0.005012 |
| LogReg | dbscan\_pca\_e0.7\_m15\_labels | 0.874432 | 0.908862 | 0.002515 | 0.74 | 0.005012 |
| LogReg | dbscan\_pca\_e0.7\_m15\_noise | 0.874432 | 0.908862 | 0.002515 | 0.74 | 0.005012 |
| LogReg | dbscan\_pca\_e0.7\_m3\_labels | 0.874098 | 0.908871 | 0.002507 | 0.73778 | 0.004998 |
| LogReg | dbscan\_pca\_e0.7\_m3\_noise | 0.874484 | 0.909084 | 0.002521 | 0.74 | 0.005025 |
| LogReg | dbscan\_pca\_e0.7\_m5\_labels | 0.874251 | 0.90899 | 0.00251 | 0.73778 | 0.005004 |
| LogReg | dbscan\_pca\_e0.7\_m5\_noise | 0.874509 | 0.90912 | 0.002522 | 0.74 | 0.005027 |
| LogReg | dbscan\_pca\_e0.7\_m7\_labels | 0.874546 | 0.90906 | 0.00252 | 0.74 | 0.005023 |
| LogReg | dbscan\_pca\_e0.7\_m7\_noise | 0.874453 | 0.909129 | 0.002522 | 0.74 | 0.005027 |
| LogReg | dbscan\_pca\_e1.0\_m10\_labels | 0.874432 | 0.908862 | 0.002515 | 0.74 | 0.005012 |
| LogReg | dbscan\_pca\_e1.0\_m10\_noise | 0.874432 | 0.908862 | 0.002515 | 0.74 | 0.005012 |
| LogReg | dbscan\_pca\_e1.0\_m15\_labels | 0.874432 | 0.908862 | 0.002515 | 0.74 | 0.005012 |
| LogReg | dbscan\_pca\_e1.0\_m15\_noise | 0.874432 | 0.908862 | 0.002515 | 0.74 | 0.005012 |
| LogReg | dbscan\_pca\_e1.0\_m3\_labels | 0.874098 | 0.908871 | 0.002507 | 0.73778 | 0.004998 |
| LogReg | dbscan\_pca\_e1.0\_m3\_noise | 0.874484 | 0.909084 | 0.002521 | 0.74 | 0.005025 |
| LogReg | dbscan\_pca\_e1.0\_m5\_labels | 0.874251 | 0.90899 | 0.00251 | 0.73778 | 0.005004 |
| LogReg | dbscan\_pca\_e1.0\_m5\_noise | 0.874509 | 0.90912 | 0.002522 | 0.74 | 0.005027 |
| LogReg | dbscan\_pca\_e1.0\_m7\_labels | 0.874546 | 0.90906 | 0.00252 | 0.74 | 0.005023 |
| LogReg | dbscan\_pca\_e1.0\_m7\_noise | 0.874453 | 0.909129 | 0.002522 | 0.74 | 0.005027 |
| LogReg | dbscan\_pca\_e1.3\_m10\_labels | 0.874422 | 0.908844 | 0.002514 | 0.74 | 0.005011 |
| LogReg | dbscan\_pca\_e1.3\_m10\_noise | 0.874422 | 0.908844 | 0.002514 | 0.74 | 0.005011 |
| LogReg | dbscan\_pca\_e1.3\_m15\_labels | 0.874432 | 0.908862 | 0.002515 | 0.74 | 0.005012 |
| LogReg | dbscan\_pca\_e1.3\_m15\_noise | 0.874432 | 0.908862 | 0.002515 | 0.74 | 0.005012 |
| LogReg | dbscan\_pca\_e1.3\_m3\_labels | 0.874228 | 0.908829 | 0.002514 | 0.74 | 0.00501 |
| LogReg | dbscan\_pca\_e1.3\_m3\_noise | 0.874605 | 0.909046 | 0.002512 | 0.73778 | 0.005007 |
| LogReg | dbscan\_pca\_e1.3\_m5\_labels | 0.874342 | 0.908999 | 0.002511 | 0.73778 | 0.005004 |
| LogReg | dbscan\_pca\_e1.3\_m5\_noise | 0.874322 | 0.909119 | 0.002514 | 0.73778 | 0.005011 |
| LogReg | dbscan\_pca\_e1.3\_m7\_labels | 0.874608 | 0.90905 | 0.00252 | 0.74 | 0.005023 |
| LogReg | dbscan\_pca\_e1.3\_m7\_noise | 0.874359 | 0.909019 | 0.002511 | 0.73778 | 0.005006 |
| LogReg | hier\_full\_n2\_labels | 0.874421 | 0.908868 | 0.002507 | 0.73778 | 0.004997 |
| LogReg | hier\_full\_n3\_labels | 0.874422 | 0.908873 | 0.002507 | 0.73778 | 0.004998 |
| LogReg | hier\_full\_n4\_labels | 0.874422 | 0.908872 | 0.002507 | 0.73778 | 0.004997 |
| LogReg | hier\_full\_n5\_labels | 0.874455 | 0.908871 | 0.002507 | 0.73778 | 0.004997 |
| LogReg | hier\_full\_n6\_labels | 0.874416 | 0.908864 | 0.002507 | 0.73778 | 0.004997 |
| LogReg | hier\_pca\_n2\_labels | 0.874424 | 0.908883 | 0.002508 | 0.73778 | 0.004998 |
| LogReg | hier\_pca\_n3\_labels | 0.874422 | 0.908878 | 0.002507 | 0.73778 | 0.004998 |
| LogReg | hier\_pca\_n4\_labels | 0.874434 | 0.908884 | 0.002508 | 0.73778 | 0.004998 |
| LogReg | hier\_pca\_n5\_labels | 0.874403 | 0.90888 | 0.002507 | 0.73778 | 0.004998 |
| LogReg | hier\_pca\_n6\_labels | 0.874439 | 0.908881 | 0.002507 | 0.73778 | 0.004998 |
| LogReg | kmeans\_full\_n10\_labels | 0.875503 | 0.909291 | 0.002544 | 0.74444 | 0.00507 |
| LogReg | kmeans\_full\_n2\_labels | 0.874421 | 0.908871 | 0.002507 | 0.73778 | 0.004997 |
| LogReg | kmeans\_full\_n3\_labels | 0.874409 | 0.908868 | 0.002507 | 0.73778 | 0.004997 |
| LogReg | kmeans\_full\_n4\_labels | 0.874435 | 0.908868 | 0.002507 | 0.73778 | 0.004997 |
| LogReg | kmeans\_full\_n5\_labels | 0.872051 | 0.908846 | 0.002508 | 0.73778 | 0.004999 |
| LogReg | kmeans\_full\_n6\_labels | 0.87509 | 0.908232 | 0.002499 | 0.74 | 0.004981 |
| LogReg | kmeans\_full\_n7\_labels | 0.873817 | 0.910496 | 0.002561 | 0.74 | 0.005105 |
| LogReg | kmeans\_full\_n8\_labels | 0.87533 | 0.908824 | 0.002523 | 0.74222 | 0.005029 |
| LogReg | kmeans\_full\_n9\_labels | 0.875749 | 0.909309 | 0.002535 | 0.74222 | 0.005053 |
| LogReg | kmeans\_pca\_n10\_labels | 0.873735 | 0.908023 | 0.002476 | 0.73556 | 0.004934 |
| LogReg | kmeans\_pca\_n2\_labels | 0.874797 | 0.908895 | 0.002516 | 0.74 | 0.005015 |
| LogReg | kmeans\_pca\_n3\_labels | 0.877613 | 0.902913 | 0.002374 | 0.74444 | 0.004733 |
| LogReg | kmeans\_pca\_n4\_labels | 0.881455 | 0.901988 | 0.002378 | 0.75333 | 0.004742 |
| LogReg | kmeans\_pca\_n5\_labels | 0.872631 | 0.90467 | 0.002394 | 0.73778 | 0.004772 |
| LogReg | kmeans\_pca\_n6\_labels | 0.867336 | 0.908567 | 0.002505 | 0.74 | 0.004994 |
| LogReg | kmeans\_pca\_n7\_labels | 0.874059 | 0.909063 | 0.002512 | 0.73778 | 0.005007 |
| LogReg | kmeans\_pca\_n8\_labels | 0.873293 | 0.906401 | 0.002456 | 0.74222 | 0.004895 |
| LogReg | kmeans\_pca\_n9\_labels | 0.874431 | 0.907316 | 0.002464 | 0.73778 | 0.004912 |

# Appendix C Aggregated Classification Model Results: PCA Feature-set

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Model** | **Cluster Condition** | **AUC ROC** | **Accuracy** | **Precision** | **Recall** | **F1** |
| XGBoost | Baseline | 0.786014 | 0.943501 | 0.002711 | 0.495556 | 0.005392 |
| XGBoost | dbscan\_full\_e0.3\_m10\_labels | 0.786014 | 0.943501 | 0.002711 | 0.495556 | 0.005392 |
| XGBoost | dbscan\_full\_e0.3\_m10\_noise | 0.786014 | 0.943501 | 0.002711 | 0.495556 | 0.005392 |
| XGBoost | dbscan\_full\_e0.3\_m15\_labels | 0.786014 | 0.943501 | 0.002711 | 0.495556 | 0.005392 |
| XGBoost | dbscan\_full\_e0.3\_m15\_noise | 0.786014 | 0.943501 | 0.002711 | 0.495556 | 0.005392 |
| XGBoost | dbscan\_full\_e0.3\_m3\_labels | 0.786014 | 0.943501 | 0.002711 | 0.495556 | 0.005392 |
| XGBoost | dbscan\_full\_e0.3\_m3\_noise | 0.785981 | 0.942969 | 0.002673 | 0.493333 | 0.005317 |
| XGBoost | dbscan\_full\_e0.3\_m5\_labels | 0.786014 | 0.943501 | 0.002711 | 0.495556 | 0.005392 |
| XGBoost | dbscan\_full\_e0.3\_m5\_noise | 0.786014 | 0.943501 | 0.002711 | 0.495556 | 0.005392 |
| XGBoost | dbscan\_full\_e0.3\_m7\_labels | 0.786014 | 0.943501 | 0.002711 | 0.495556 | 0.005392 |
| XGBoost | dbscan\_full\_e0.3\_m7\_noise | 0.786014 | 0.943501 | 0.002711 | 0.495556 | 0.005392 |
| XGBoost | dbscan\_full\_e0.5\_m10\_labels | 0.786014 | 0.943501 | 0.002711 | 0.495556 | 0.005392 |
| XGBoost | dbscan\_full\_e0.5\_m10\_noise | 0.786014 | 0.943501 | 0.002711 | 0.495556 | 0.005392 |
| XGBoost | dbscan\_full\_e0.5\_m15\_labels | 0.786014 | 0.943501 | 0.002711 | 0.495556 | 0.005392 |
| XGBoost | dbscan\_full\_e0.5\_m15\_noise | 0.786014 | 0.943501 | 0.002711 | 0.495556 | 0.005392 |
| XGBoost | dbscan\_full\_e0.5\_m3\_labels | 0.777191 | 0.942356 | 0.00271 | 0.504444 | 0.005391 |
| XGBoost | dbscan\_full\_e0.5\_m3\_noise | 0.785757 | 0.942928 | 0.002758 | 0.508889 | 0.005485 |
| XGBoost | dbscan\_full\_e0.5\_m5\_labels | 0.786014 | 0.943501 | 0.002711 | 0.495556 | 0.005392 |
| XGBoost | dbscan\_full\_e0.5\_m5\_noise | 0.785936 | 0.943006 | 0.002727 | 0.502222 | 0.005424 |
| XGBoost | dbscan\_full\_e0.5\_m7\_labels | 0.786014 | 0.943501 | 0.002711 | 0.495556 | 0.005392 |
| XGBoost | dbscan\_full\_e0.5\_m7\_noise | 0.786014 | 0.943501 | 0.002711 | 0.495556 | 0.005392 |
| XGBoost | dbscan\_full\_e0.7\_m10\_labels | 0.786144 | 0.943706 | 0.002721 | 0.495556 | 0.005411 |
| XGBoost | dbscan\_full\_e0.7\_m10\_noise | 0.78547 | 0.943699 | 0.00272 | 0.495556 | 0.005411 |
| XGBoost | dbscan\_full\_e0.7\_m15\_labels | 0.786014 | 0.943501 | 0.002711 | 0.495556 | 0.005392 |
| XGBoost | dbscan\_full\_e0.7\_m15\_noise | 0.786014 | 0.943501 | 0.002711 | 0.495556 | 0.005392 |
| XGBoost | dbscan\_full\_e0.7\_m3\_labels | 0.787207 | 0.943494 | 0.002774 | 0.506667 | 0.005517 |
| XGBoost | dbscan\_full\_e0.7\_m3\_noise | 0.775299 | 0.942389 | 0.00261 | 0.484444 | 0.005191 |
| XGBoost | dbscan\_full\_e0.7\_m5\_labels | 0.78807 | 0.942941 | 0.002722 | 0.502222 | 0.005414 |
| XGBoost | dbscan\_full\_e0.7\_m5\_noise | 0.787658 | 0.943626 | 0.002766 | 0.504444 | 0.005502 |
| XGBoost | dbscan\_full\_e0.7\_m7\_labels | 0.786144 | 0.943706 | 0.002721 | 0.495556 | 0.005411 |
| XGBoost | dbscan\_full\_e0.7\_m7\_noise | 0.787195 | 0.943523 | 0.002764 | 0.504444 | 0.005497 |
| XGBoost | dbscan\_full\_e1.0\_m10\_labels | 0.788237 | 0.942913 | 0.002697 | 0.497778 | 0.005365 |
| XGBoost | dbscan\_full\_e1.0\_m10\_noise | 0.786113 | 0.944195 | 0.002747 | 0.495556 | 0.005463 |
| XGBoost | dbscan\_full\_e1.0\_m15\_labels | 0.786014 | 0.943501 | 0.002711 | 0.495556 | 0.005392 |
| XGBoost | dbscan\_full\_e1.0\_m15\_noise | 0.786014 | 0.943501 | 0.002711 | 0.495556 | 0.005392 |
| XGBoost | dbscan\_full\_e1.0\_m3\_labels | 0.780223 | 0.942889 | 0.002696 | 0.497778 | 0.005364 |
| XGBoost | dbscan\_full\_e1.0\_m3\_noise | 0.788851 | 0.942614 | 0.002709 | 0.502222 | 0.005389 |
| XGBoost | dbscan\_full\_e1.0\_m5\_labels | 0.783755 | 0.942802 | 0.002627 | 0.486667 | 0.005226 |
| XGBoost | dbscan\_full\_e1.0\_m5\_noise | 0.782635 | 0.94352 | 0.00269 | 0.491111 | 0.005352 |
| XGBoost | dbscan\_full\_e1.0\_m7\_labels | 0.782875 | 0.942732 | 0.002625 | 0.486667 | 0.005222 |
| XGBoost | dbscan\_full\_e1.0\_m7\_noise | 0.785225 | 0.943469 | 0.002664 | 0.486667 | 0.005298 |
| XGBoost | dbscan\_full\_e1.3\_m10\_labels | 0.785041 | 0.943008 | 0.002714 | 0.5 | 0.005399 |
| XGBoost | dbscan\_full\_e1.3\_m10\_noise | 0.785363 | 0.942969 | 0.002712 | 0.5 | 0.005396 |
| XGBoost | dbscan\_full\_e1.3\_m15\_labels | 0.786057 | 0.943487 | 0.00271 | 0.495556 | 0.005391 |
| XGBoost | dbscan\_full\_e1.3\_m15\_noise | 0.785674 | 0.9436 | 0.002703 | 0.493333 | 0.005376 |
| XGBoost | dbscan\_full\_e1.3\_m3\_labels | 0.785622 | 0.9428 | 0.002643 | 0.488889 | 0.005258 |
| XGBoost | dbscan\_full\_e1.3\_m3\_noise | 0.780921 | 0.943527 | 0.002631 | 0.48 | 0.005233 |
| XGBoost | dbscan\_full\_e1.3\_m5\_labels | 0.776839 | 0.943298 | 0.002694 | 0.493333 | 0.005359 |
| XGBoost | dbscan\_full\_e1.3\_m5\_noise | 0.782067 | 0.94265 | 0.002652 | 0.491111 | 0.005275 |
| XGBoost | dbscan\_full\_e1.3\_m7\_labels | 0.784555 | 0.94246 | 0.002633 | 0.488889 | 0.005238 |
| XGBoost | dbscan\_full\_e1.3\_m7\_noise | 0.784897 | 0.943773 | 0.002725 | 0.495556 | 0.00542 |
| XGBoost | dbscan\_pca\_e0.3\_m10\_labels | 0.786014 | 0.943501 | 0.002711 | 0.495556 | 0.005392 |
| XGBoost | dbscan\_pca\_e0.3\_m10\_noise | 0.786014 | 0.943501 | 0.002711 | 0.495556 | 0.005392 |
| XGBoost | dbscan\_pca\_e0.3\_m15\_labels | 0.786014 | 0.943501 | 0.002711 | 0.495556 | 0.005392 |
| XGBoost | dbscan\_pca\_e0.3\_m15\_noise | 0.786014 | 0.943501 | 0.002711 | 0.495556 | 0.005392 |
| XGBoost | dbscan\_pca\_e0.3\_m3\_labels | 0.778471 | 0.943958 | 0.002699 | 0.488889 | 0.005367 |
| XGBoost | dbscan\_pca\_e0.3\_m3\_noise | 0.784102 | 0.943751 | 0.002689 | 0.488889 | 0.005349 |
| XGBoost | dbscan\_pca\_e0.3\_m5\_labels | 0.780971 | 0.943372 | 0.002564 | 0.468889 | 0.0051 |
| XGBoost | dbscan\_pca\_e0.3\_m5\_noise | 0.777076 | 0.943832 | 0.00263 | 0.477778 | 0.005231 |
| XGBoost | dbscan\_pca\_e0.3\_m7\_labels | 0.787203 | 0.943646 | 0.002633 | 0.48 | 0.005238 |
| XGBoost | dbscan\_pca\_e0.3\_m7\_noise | 0.780092 | 0.943754 | 0.002628 | 0.477778 | 0.005227 |
| XGBoost | dbscan\_pca\_e0.5\_m10\_labels | 0.786014 | 0.943501 | 0.002711 | 0.495556 | 0.005392 |
| XGBoost | dbscan\_pca\_e0.5\_m10\_noise | 0.786014 | 0.943501 | 0.002711 | 0.495556 | 0.005392 |
| XGBoost | dbscan\_pca\_e0.5\_m15\_labels | 0.786014 | 0.943501 | 0.002711 | 0.495556 | 0.005392 |
| XGBoost | dbscan\_pca\_e0.5\_m15\_noise | 0.786014 | 0.943501 | 0.002711 | 0.495556 | 0.005392 |
| XGBoost | dbscan\_pca\_e0.5\_m3\_labels | 0.778471 | 0.943958 | 0.002699 | 0.488889 | 0.005367 |
| XGBoost | dbscan\_pca\_e0.5\_m3\_noise | 0.784102 | 0.943751 | 0.002689 | 0.488889 | 0.005349 |
| XGBoost | dbscan\_pca\_e0.5\_m5\_labels | 0.780971 | 0.943372 | 0.002564 | 0.468889 | 0.0051 |
| XGBoost | dbscan\_pca\_e0.5\_m5\_noise | 0.777076 | 0.943832 | 0.00263 | 0.477778 | 0.005231 |
| XGBoost | dbscan\_pca\_e0.5\_m7\_labels | 0.787203 | 0.943646 | 0.002633 | 0.48 | 0.005238 |
| XGBoost | dbscan\_pca\_e0.5\_m7\_noise | 0.780092 | 0.943754 | 0.002628 | 0.477778 | 0.005227 |
| XGBoost | dbscan\_pca\_e0.7\_m10\_labels | 0.786014 | 0.943501 | 0.002711 | 0.495556 | 0.005392 |
| XGBoost | dbscan\_pca\_e0.7\_m10\_noise | 0.786014 | 0.943501 | 0.002711 | 0.495556 | 0.005392 |
| XGBoost | dbscan\_pca\_e0.7\_m15\_labels | 0.786014 | 0.943501 | 0.002711 | 0.495556 | 0.005392 |
| XGBoost | dbscan\_pca\_e0.7\_m15\_noise | 0.786014 | 0.943501 | 0.002711 | 0.495556 | 0.005392 |
| XGBoost | dbscan\_pca\_e0.7\_m3\_labels | 0.778197 | 0.943109 | 0.002644 | 0.486667 | 0.00526 |
| XGBoost | dbscan\_pca\_e0.7\_m3\_noise | 0.784847 | 0.943785 | 0.00273 | 0.495556 | 0.00543 |
| XGBoost | dbscan\_pca\_e0.7\_m5\_labels | 0.779981 | 0.943242 | 0.002645 | 0.484444 | 0.005262 |
| XGBoost | dbscan\_pca\_e0.7\_m5\_noise | 0.770696 | 0.943628 | 0.002586 | 0.471111 | 0.005143 |
| XGBoost | dbscan\_pca\_e0.7\_m7\_labels | 0.78696 | 0.942672 | 0.002663 | 0.493333 | 0.005297 |
| XGBoost | dbscan\_pca\_e0.7\_m7\_noise | 0.785017 | 0.94316 | 0.002685 | 0.493333 | 0.00534 |
| XGBoost | dbscan\_pca\_e1.0\_m10\_labels | 0.786014 | 0.943501 | 0.002711 | 0.495556 | 0.005392 |
| XGBoost | dbscan\_pca\_e1.0\_m10\_noise | 0.786014 | 0.943501 | 0.002711 | 0.495556 | 0.005392 |
| XGBoost | dbscan\_pca\_e1.0\_m15\_labels | 0.786014 | 0.943501 | 0.002711 | 0.495556 | 0.005392 |
| XGBoost | dbscan\_pca\_e1.0\_m15\_noise | 0.786014 | 0.943501 | 0.002711 | 0.495556 | 0.005392 |
| XGBoost | dbscan\_pca\_e1.0\_m3\_labels | 0.778197 | 0.943109 | 0.002644 | 0.486667 | 0.00526 |
| XGBoost | dbscan\_pca\_e1.0\_m3\_noise | 0.784847 | 0.943785 | 0.00273 | 0.495556 | 0.00543 |
| XGBoost | dbscan\_pca\_e1.0\_m5\_labels | 0.779981 | 0.943242 | 0.002645 | 0.484444 | 0.005262 |
| XGBoost | dbscan\_pca\_e1.0\_m5\_noise | 0.770696 | 0.943628 | 0.002586 | 0.471111 | 0.005143 |
| XGBoost | dbscan\_pca\_e1.0\_m7\_labels | 0.78696 | 0.942672 | 0.002663 | 0.493333 | 0.005297 |
| XGBoost | dbscan\_pca\_e1.0\_m7\_noise | 0.785017 | 0.94316 | 0.002685 | 0.493333 | 0.00534 |
| XGBoost | dbscan\_pca\_e1.3\_m10\_labels | 0.783364 | 0.943556 | 0.002715 | 0.495556 | 0.0054 |
| XGBoost | dbscan\_pca\_e1.3\_m10\_noise | 0.783364 | 0.943556 | 0.002715 | 0.495556 | 0.0054 |
| XGBoost | dbscan\_pca\_e1.3\_m15\_labels | 0.786014 | 0.943501 | 0.002711 | 0.495556 | 0.005392 |
| XGBoost | dbscan\_pca\_e1.3\_m15\_noise | 0.786014 | 0.943501 | 0.002711 | 0.495556 | 0.005392 |
| XGBoost | dbscan\_pca\_e1.3\_m3\_labels | 0.783186 | 0.943629 | 0.002682 | 0.488889 | 0.005334 |
| XGBoost | dbscan\_pca\_e1.3\_m3\_noise | 0.779035 | 0.94352 | 0.002642 | 0.482222 | 0.005256 |
| XGBoost | dbscan\_pca\_e1.3\_m5\_labels | 0.78834 | 0.94291 | 0.002604 | 0.48 | 0.005179 |
| XGBoost | dbscan\_pca\_e1.3\_m5\_noise | 0.775084 | 0.943754 | 0.002654 | 0.482222 | 0.00528 |
| XGBoost | dbscan\_pca\_e1.3\_m7\_labels | 0.791867 | 0.942098 | 0.002741 | 0.513333 | 0.005454 |
| XGBoost | dbscan\_pca\_e1.3\_m7\_noise | 0.783658 | 0.942518 | 0.002657 | 0.493333 | 0.005285 |
| XGBoost | hier\_full\_n2\_labels | 0.786014 | 0.943501 | 0.002711 | 0.495556 | 0.005392 |
| XGBoost | hier\_full\_n3\_labels | 0.786014 | 0.943501 | 0.002711 | 0.495556 | 0.005392 |
| XGBoost | hier\_full\_n4\_labels | 0.786014 | 0.943501 | 0.002711 | 0.495556 | 0.005392 |
| XGBoost | hier\_full\_n5\_labels | 0.786014 | 0.943501 | 0.002711 | 0.495556 | 0.005392 |
| XGBoost | hier\_full\_n6\_labels | 0.786014 | 0.943501 | 0.002711 | 0.495556 | 0.005392 |
| XGBoost | hier\_pca\_n2\_labels | 0.786014 | 0.943501 | 0.002711 | 0.495556 | 0.005392 |
| XGBoost | hier\_pca\_n3\_labels | 0.786014 | 0.943501 | 0.002711 | 0.495556 | 0.005392 |
| XGBoost | hier\_pca\_n4\_labels | 0.786014 | 0.943501 | 0.002711 | 0.495556 | 0.005392 |
| XGBoost | hier\_pca\_n5\_labels | 0.786014 | 0.943501 | 0.002711 | 0.495556 | 0.005392 |
| XGBoost | hier\_pca\_n6\_labels | 0.786014 | 0.943501 | 0.002711 | 0.495556 | 0.005392 |
| XGBoost | kmeans\_full\_n10\_labels | 0.790331 | 0.965401 | 0.004215 | 0.471111 | 0.008355 |
| XGBoost | kmeans\_full\_n2\_labels | 0.786014 | 0.943501 | 0.002711 | 0.495556 | 0.005392 |
| XGBoost | kmeans\_full\_n3\_labels | 0.786014 | 0.943501 | 0.002711 | 0.495556 | 0.005392 |
| XGBoost | kmeans\_full\_n4\_labels | 0.786014 | 0.943501 | 0.002711 | 0.495556 | 0.005392 |
| XGBoost | kmeans\_full\_n5\_labels | 0.782005 | 0.945255 | 0.002762 | 0.488889 | 0.005493 |
| XGBoost | kmeans\_full\_n6\_labels | 0.78611 | 0.960747 | 0.003187 | 0.402222 | 0.006324 |
| XGBoost | kmeans\_full\_n7\_labels | 0.737057 | 0.955556 | 0.002796 | 0.4 | 0.005553 |
| XGBoost | kmeans\_full\_n8\_labels | 0.772951 | 0.96612 | 0.003778 | 0.413333 | 0.007487 |
| XGBoost | kmeans\_full\_n9\_labels | 0.75921 | 0.966984 | 0.003663 | 0.388889 | 0.007256 |
| XGBoost | kmeans\_pca\_n10\_labels | 0.767825 | 0.948693 | 0.002842 | 0.471111 | 0.005651 |
| XGBoost | kmeans\_pca\_n2\_labels | 0.778601 | 0.942844 | 0.002599 | 0.48 | 0.005169 |
| XGBoost | kmeans\_pca\_n3\_labels | 0.78039 | 0.944554 | 0.002743 | 0.491111 | 0.005456 |
| XGBoost | kmeans\_pca\_n4\_labels | 0.776449 | 0.944648 | 0.002683 | 0.48 | 0.005336 |
| XGBoost | kmeans\_pca\_n5\_labels | 0.773948 | 0.946774 | 0.002619 | 0.451111 | 0.005208 |
| XGBoost | kmeans\_pca\_n6\_labels | 0.767092 | 0.94768 | 0.002706 | 0.457778 | 0.005379 |
| XGBoost | kmeans\_pca\_n7\_labels | 0.774895 | 0.947795 | 0.002788 | 0.471111 | 0.005543 |
| XGBoost | kmeans\_pca\_n8\_labels | 0.769187 | 0.946323 | 0.002785 | 0.482222 | 0.005537 |
| XGBoost | kmeans\_pca\_n9\_labels | 0.771718 | 0.948481 | 0.002819 | 0.468889 | 0.005604 |
| SVM | Baseline | 0.789308 | 0.938523 | 0.003304 | 0.64 | 0.006572 |
| SVM | dbscan\_full\_e0.3\_m10\_labels | 0.789311 | 0.938529 | 0.003304 | 0.64 | 0.006572 |
| SVM | dbscan\_full\_e0.3\_m10\_noise | 0.789306 | 0.93852 | 0.003303 | 0.64 | 0.006571 |
| SVM | dbscan\_full\_e0.3\_m15\_labels | 0.789311 | 0.938529 | 0.003304 | 0.64 | 0.006572 |
| SVM | dbscan\_full\_e0.3\_m15\_noise | 0.789306 | 0.93852 | 0.003303 | 0.64 | 0.006571 |
| SVM | dbscan\_full\_e0.3\_m3\_labels | 0.789311 | 0.938529 | 0.003304 | 0.64 | 0.006572 |
| SVM | dbscan\_full\_e0.3\_m3\_noise | 0.789309 | 0.938526 | 0.003303 | 0.64 | 0.006572 |
| SVM | dbscan\_full\_e0.3\_m5\_labels | 0.789311 | 0.938529 | 0.003304 | 0.64 | 0.006572 |
| SVM | dbscan\_full\_e0.3\_m5\_noise | 0.789306 | 0.93852 | 0.003303 | 0.64 | 0.006571 |
| SVM | dbscan\_full\_e0.3\_m7\_labels | 0.789311 | 0.938529 | 0.003304 | 0.64 | 0.006572 |
| SVM | dbscan\_full\_e0.3\_m7\_noise | 0.789306 | 0.93852 | 0.003303 | 0.64 | 0.006571 |
| SVM | dbscan\_full\_e0.5\_m10\_labels | 0.789311 | 0.938529 | 0.003304 | 0.64 | 0.006572 |
| SVM | dbscan\_full\_e0.5\_m10\_noise | 0.789306 | 0.93852 | 0.003303 | 0.64 | 0.006571 |
| SVM | dbscan\_full\_e0.5\_m15\_labels | 0.789311 | 0.938529 | 0.003304 | 0.64 | 0.006572 |
| SVM | dbscan\_full\_e0.5\_m15\_noise | 0.789306 | 0.93852 | 0.003303 | 0.64 | 0.006571 |
| SVM | dbscan\_full\_e0.5\_m3\_labels | 0.789301 | 0.938509 | 0.003302 | 0.64 | 0.00657 |
| SVM | dbscan\_full\_e0.5\_m3\_noise | 0.789308 | 0.938523 | 0.003303 | 0.64 | 0.006571 |
| SVM | dbscan\_full\_e0.5\_m5\_labels | 0.789311 | 0.938529 | 0.003304 | 0.64 | 0.006572 |
| SVM | dbscan\_full\_e0.5\_m5\_noise | 0.789306 | 0.93852 | 0.003303 | 0.64 | 0.006571 |
| SVM | dbscan\_full\_e0.5\_m7\_labels | 0.789311 | 0.938529 | 0.003304 | 0.64 | 0.006572 |
| SVM | dbscan\_full\_e0.5\_m7\_noise | 0.789306 | 0.93852 | 0.003303 | 0.64 | 0.006571 |
| SVM | dbscan\_full\_e0.7\_m10\_labels | 0.789311 | 0.938529 | 0.003304 | 0.64 | 0.006572 |
| SVM | dbscan\_full\_e0.7\_m10\_noise | 0.789306 | 0.93852 | 0.003303 | 0.64 | 0.006571 |
| SVM | dbscan\_full\_e0.7\_m15\_labels | 0.789311 | 0.938529 | 0.003304 | 0.64 | 0.006572 |
| SVM | dbscan\_full\_e0.7\_m15\_noise | 0.789306 | 0.93852 | 0.003303 | 0.64 | 0.006571 |
| SVM | dbscan\_full\_e0.7\_m3\_labels | 0.789305 | 0.938518 | 0.003303 | 0.64 | 0.006571 |
| SVM | dbscan\_full\_e0.7\_m3\_noise | 0.788192 | 0.938513 | 0.003293 | 0.637778 | 0.006552 |
| SVM | dbscan\_full\_e0.7\_m5\_labels | 0.789311 | 0.938529 | 0.003304 | 0.64 | 0.006572 |
| SVM | dbscan\_full\_e0.7\_m5\_noise | 0.789309 | 0.938526 | 0.003303 | 0.64 | 0.006572 |
| SVM | dbscan\_full\_e0.7\_m7\_labels | 0.789311 | 0.938529 | 0.003304 | 0.64 | 0.006572 |
| SVM | dbscan\_full\_e0.7\_m7\_noise | 0.789306 | 0.93852 | 0.003303 | 0.64 | 0.006571 |
| SVM | dbscan\_full\_e1.0\_m10\_labels | 0.789311 | 0.938529 | 0.003304 | 0.64 | 0.006572 |
| SVM | dbscan\_full\_e1.0\_m10\_noise | 0.789306 | 0.93852 | 0.003303 | 0.64 | 0.006571 |
| SVM | dbscan\_full\_e1.0\_m15\_labels | 0.789311 | 0.938529 | 0.003304 | 0.64 | 0.006572 |
| SVM | dbscan\_full\_e1.0\_m15\_noise | 0.789306 | 0.93852 | 0.003303 | 0.64 | 0.006571 |
| SVM | dbscan\_full\_e1.0\_m3\_labels | 0.789305 | 0.938517 | 0.003303 | 0.64 | 0.006572 |
| SVM | dbscan\_full\_e1.0\_m3\_noise | 0.789293 | 0.938494 | 0.003302 | 0.64 | 0.006568 |
| SVM | dbscan\_full\_e1.0\_m5\_labels | 0.789309 | 0.938525 | 0.003304 | 0.64 | 0.006573 |
| SVM | dbscan\_full\_e1.0\_m5\_noise | 0.788202 | 0.938533 | 0.003295 | 0.637778 | 0.006555 |
| SVM | dbscan\_full\_e1.0\_m7\_labels | 0.789311 | 0.938529 | 0.003304 | 0.64 | 0.006572 |
| SVM | dbscan\_full\_e1.0\_m7\_noise | 0.789309 | 0.938526 | 0.003303 | 0.64 | 0.006572 |
| SVM | dbscan\_full\_e1.3\_m10\_labels | 0.789311 | 0.938529 | 0.003304 | 0.64 | 0.006572 |
| SVM | dbscan\_full\_e1.3\_m10\_noise | 0.789306 | 0.93852 | 0.003303 | 0.64 | 0.006571 |
| SVM | dbscan\_full\_e1.3\_m15\_labels | 0.789311 | 0.938529 | 0.003304 | 0.64 | 0.006572 |
| SVM | dbscan\_full\_e1.3\_m15\_noise | 0.789306 | 0.93852 | 0.003303 | 0.64 | 0.006571 |
| SVM | dbscan\_full\_e1.3\_m3\_labels | 0.78931 | 0.938528 | 0.003303 | 0.64 | 0.006572 |
| SVM | dbscan\_full\_e1.3\_m3\_noise | 0.789303 | 0.938513 | 0.003303 | 0.64 | 0.006571 |
| SVM | dbscan\_full\_e1.3\_m5\_labels | 0.789298 | 0.938503 | 0.003302 | 0.64 | 0.00657 |
| SVM | dbscan\_full\_e1.3\_m5\_noise | 0.788192 | 0.938513 | 0.003293 | 0.637778 | 0.006552 |
| SVM | dbscan\_full\_e1.3\_m7\_labels | 0.789319 | 0.938546 | 0.003305 | 0.64 | 0.006574 |
| SVM | dbscan\_full\_e1.3\_m7\_noise | 0.789309 | 0.938525 | 0.003303 | 0.64 | 0.006572 |
| SVM | dbscan\_pca\_e0.3\_m10\_labels | 0.789311 | 0.938529 | 0.003304 | 0.64 | 0.006572 |
| SVM | dbscan\_pca\_e0.3\_m10\_noise | 0.789306 | 0.93852 | 0.003303 | 0.64 | 0.006571 |
| SVM | dbscan\_pca\_e0.3\_m15\_labels | 0.789311 | 0.938529 | 0.003304 | 0.64 | 0.006572 |
| SVM | dbscan\_pca\_e0.3\_m15\_noise | 0.789306 | 0.93852 | 0.003303 | 0.64 | 0.006571 |
| SVM | dbscan\_pca\_e0.3\_m3\_labels | 0.789306 | 0.938519 | 0.003303 | 0.64 | 0.006571 |
| SVM | dbscan\_pca\_e0.3\_m3\_noise | 0.789291 | 0.938489 | 0.003302 | 0.64 | 0.006568 |
| SVM | dbscan\_pca\_e0.3\_m5\_labels | 0.788199 | 0.938528 | 0.003295 | 0.637778 | 0.006555 |
| SVM | dbscan\_pca\_e0.3\_m5\_noise | 0.789314 | 0.938535 | 0.003304 | 0.64 | 0.006573 |
| SVM | dbscan\_pca\_e0.3\_m7\_labels | 0.789306 | 0.93852 | 0.003303 | 0.64 | 0.006572 |
| SVM | dbscan\_pca\_e0.3\_m7\_noise | 0.789316 | 0.93854 | 0.003304 | 0.64 | 0.006574 |
| SVM | dbscan\_pca\_e0.5\_m10\_labels | 0.789311 | 0.938529 | 0.003304 | 0.64 | 0.006572 |
| SVM | dbscan\_pca\_e0.5\_m10\_noise | 0.789306 | 0.93852 | 0.003303 | 0.64 | 0.006571 |
| SVM | dbscan\_pca\_e0.5\_m15\_labels | 0.789311 | 0.938529 | 0.003304 | 0.64 | 0.006572 |
| SVM | dbscan\_pca\_e0.5\_m15\_noise | 0.789306 | 0.93852 | 0.003303 | 0.64 | 0.006571 |
| SVM | dbscan\_pca\_e0.5\_m3\_labels | 0.789306 | 0.938519 | 0.003303 | 0.64 | 0.006571 |
| SVM | dbscan\_pca\_e0.5\_m3\_noise | 0.789291 | 0.938489 | 0.003302 | 0.64 | 0.006568 |
| SVM | dbscan\_pca\_e0.5\_m5\_labels | 0.788199 | 0.938528 | 0.003295 | 0.637778 | 0.006555 |
| SVM | dbscan\_pca\_e0.5\_m5\_noise | 0.789314 | 0.938535 | 0.003304 | 0.64 | 0.006573 |
| SVM | dbscan\_pca\_e0.5\_m7\_labels | 0.789306 | 0.93852 | 0.003303 | 0.64 | 0.006572 |
| SVM | dbscan\_pca\_e0.5\_m7\_noise | 0.789316 | 0.93854 | 0.003304 | 0.64 | 0.006574 |
| SVM | dbscan\_pca\_e0.7\_m10\_labels | 0.789311 | 0.938529 | 0.003304 | 0.64 | 0.006572 |
| SVM | dbscan\_pca\_e0.7\_m10\_noise | 0.789306 | 0.93852 | 0.003303 | 0.64 | 0.006571 |
| SVM | dbscan\_pca\_e0.7\_m15\_labels | 0.789311 | 0.938529 | 0.003304 | 0.64 | 0.006572 |
| SVM | dbscan\_pca\_e0.7\_m15\_noise | 0.789306 | 0.93852 | 0.003303 | 0.64 | 0.006571 |
| SVM | dbscan\_pca\_e0.7\_m3\_labels | 0.789315 | 0.938537 | 0.003304 | 0.64 | 0.006573 |
| SVM | dbscan\_pca\_e0.7\_m3\_noise | 0.789291 | 0.938489 | 0.003302 | 0.64 | 0.006568 |
| SVM | dbscan\_pca\_e0.7\_m5\_labels | 0.789297 | 0.938502 | 0.003302 | 0.64 | 0.00657 |
| SVM | dbscan\_pca\_e0.7\_m5\_noise | 0.789314 | 0.938535 | 0.003304 | 0.64 | 0.006573 |
| SVM | dbscan\_pca\_e0.7\_m7\_labels | 0.789299 | 0.938505 | 0.003302 | 0.64 | 0.00657 |
| SVM | dbscan\_pca\_e0.7\_m7\_noise | 0.789316 | 0.93854 | 0.003304 | 0.64 | 0.006574 |
| SVM | dbscan\_pca\_e1.0\_m10\_labels | 0.789311 | 0.938529 | 0.003304 | 0.64 | 0.006572 |
| SVM | dbscan\_pca\_e1.0\_m10\_noise | 0.789306 | 0.93852 | 0.003303 | 0.64 | 0.006571 |
| SVM | dbscan\_pca\_e1.0\_m15\_labels | 0.789311 | 0.938529 | 0.003304 | 0.64 | 0.006572 |
| SVM | dbscan\_pca\_e1.0\_m15\_noise | 0.789306 | 0.93852 | 0.003303 | 0.64 | 0.006571 |
| SVM | dbscan\_pca\_e1.0\_m3\_labels | 0.789315 | 0.938537 | 0.003304 | 0.64 | 0.006573 |
| SVM | dbscan\_pca\_e1.0\_m3\_noise | 0.789291 | 0.938489 | 0.003302 | 0.64 | 0.006568 |
| SVM | dbscan\_pca\_e1.0\_m5\_labels | 0.789297 | 0.938502 | 0.003302 | 0.64 | 0.00657 |
| SVM | dbscan\_pca\_e1.0\_m5\_noise | 0.789314 | 0.938535 | 0.003304 | 0.64 | 0.006573 |
| SVM | dbscan\_pca\_e1.0\_m7\_labels | 0.789299 | 0.938505 | 0.003302 | 0.64 | 0.00657 |
| SVM | dbscan\_pca\_e1.0\_m7\_noise | 0.789316 | 0.93854 | 0.003304 | 0.64 | 0.006574 |
| SVM | dbscan\_pca\_e1.3\_m10\_labels | 0.789311 | 0.938529 | 0.003304 | 0.64 | 0.006572 |
| SVM | dbscan\_pca\_e1.3\_m10\_noise | 0.789306 | 0.93852 | 0.003303 | 0.64 | 0.006571 |
| SVM | dbscan\_pca\_e1.3\_m15\_labels | 0.789311 | 0.938529 | 0.003304 | 0.64 | 0.006572 |
| SVM | dbscan\_pca\_e1.3\_m15\_noise | 0.789306 | 0.93852 | 0.003303 | 0.64 | 0.006571 |
| SVM | dbscan\_pca\_e1.3\_m3\_labels | 0.789298 | 0.938503 | 0.003302 | 0.64 | 0.006569 |
| SVM | dbscan\_pca\_e1.3\_m3\_noise | 0.789306 | 0.93852 | 0.003303 | 0.64 | 0.006571 |
| SVM | dbscan\_pca\_e1.3\_m5\_labels | 0.789295 | 0.938498 | 0.003302 | 0.64 | 0.006568 |
| SVM | dbscan\_pca\_e1.3\_m5\_noise | 0.789314 | 0.938535 | 0.003304 | 0.64 | 0.006573 |
| SVM | dbscan\_pca\_e1.3\_m7\_labels | 0.789302 | 0.938512 | 0.003303 | 0.64 | 0.00657 |
| SVM | dbscan\_pca\_e1.3\_m7\_noise | 0.789316 | 0.93854 | 0.003304 | 0.64 | 0.006574 |
| SVM | hier\_full\_n2\_labels | 0.789305 | 0.938517 | 0.003303 | 0.64 | 0.006572 |
| SVM | hier\_full\_n3\_labels | 0.789305 | 0.938517 | 0.003303 | 0.64 | 0.006572 |
| SVM | hier\_full\_n4\_labels | 0.789305 | 0.938517 | 0.003303 | 0.64 | 0.006572 |
| SVM | hier\_full\_n5\_labels | 0.789305 | 0.938517 | 0.003303 | 0.64 | 0.006572 |
| SVM | hier\_full\_n6\_labels | 0.789305 | 0.938517 | 0.003303 | 0.64 | 0.006572 |
| SVM | hier\_pca\_n2\_labels | 0.789305 | 0.938517 | 0.003303 | 0.64 | 0.006572 |
| SVM | hier\_pca\_n3\_labels | 0.789305 | 0.938517 | 0.003303 | 0.64 | 0.006572 |
| SVM | hier\_pca\_n4\_labels | 0.789305 | 0.938517 | 0.003303 | 0.64 | 0.006572 |
| SVM | hier\_pca\_n5\_labels | 0.789305 | 0.938517 | 0.003303 | 0.64 | 0.006572 |
| SVM | hier\_pca\_n6\_labels | 0.789305 | 0.938517 | 0.003303 | 0.64 | 0.006572 |
| SVM | kmeans\_full\_n10\_labels | 0.789315 | 0.938538 | 0.003304 | 0.64 | 0.006572 |
| SVM | kmeans\_full\_n2\_labels | 0.789305 | 0.938517 | 0.003303 | 0.64 | 0.006572 |
| SVM | kmeans\_full\_n3\_labels | 0.789305 | 0.938517 | 0.003303 | 0.64 | 0.006572 |
| SVM | kmeans\_full\_n4\_labels | 0.789305 | 0.938517 | 0.003303 | 0.64 | 0.006572 |
| SVM | kmeans\_full\_n5\_labels | 0.789304 | 0.938516 | 0.003303 | 0.64 | 0.006571 |
| SVM | kmeans\_full\_n6\_labels | 0.789296 | 0.9385 | 0.003302 | 0.64 | 0.006569 |
| SVM | kmeans\_full\_n7\_labels | 0.789304 | 0.938515 | 0.003303 | 0.64 | 0.006571 |
| SVM | kmeans\_full\_n8\_labels | 0.78931 | 0.938528 | 0.003304 | 0.64 | 0.006573 |
| SVM | kmeans\_full\_n9\_labels | 0.789294 | 0.938495 | 0.003302 | 0.64 | 0.006569 |
| SVM | kmeans\_pca\_n10\_labels | 0.789309 | 0.938524 | 0.003303 | 0.64 | 0.006572 |
| SVM | kmeans\_pca\_n2\_labels | 0.7893 | 0.938508 | 0.003303 | 0.64 | 0.006571 |
| SVM | kmeans\_pca\_n3\_labels | 0.789306 | 0.93852 | 0.003303 | 0.64 | 0.006571 |
| SVM | kmeans\_pca\_n4\_labels | 0.789295 | 0.938498 | 0.003302 | 0.64 | 0.00657 |
| SVM | kmeans\_pca\_n5\_labels | 0.789323 | 0.938553 | 0.003305 | 0.64 | 0.006575 |
| SVM | kmeans\_pca\_n6\_labels | 0.789293 | 0.938494 | 0.003302 | 0.64 | 0.006569 |
| SVM | kmeans\_pca\_n7\_labels | 0.789302 | 0.938511 | 0.003303 | 0.64 | 0.006571 |
| SVM | kmeans\_pca\_n8\_labels | 0.789311 | 0.938529 | 0.003304 | 0.64 | 0.006573 |
| SVM | kmeans\_pca\_n9\_labels | 0.789303 | 0.938513 | 0.003302 | 0.64 | 0.00657 |
| Random Forest | Baseline | 0.851268 | 0.92971 | 0.002801 | 0.635556 | 0.005577 |
| Random Forest | dbscan\_full\_e0.3\_m10\_labels | 0.811609 | 0.930966 | 0.002517 | 0.562222 | 0.005012 |
| Random Forest | dbscan\_full\_e0.3\_m10\_noise | 0.811609 | 0.930966 | 0.002517 | 0.562222 | 0.005012 |
| Random Forest | dbscan\_full\_e0.3\_m15\_labels | 0.811609 | 0.930966 | 0.002517 | 0.562222 | 0.005012 |
| Random Forest | dbscan\_full\_e0.3\_m15\_noise | 0.811609 | 0.930966 | 0.002517 | 0.562222 | 0.005012 |
| Random Forest | dbscan\_full\_e0.3\_m3\_labels | 0.812052 | 0.930634 | 0.002518 | 0.564444 | 0.005013 |
| Random Forest | dbscan\_full\_e0.3\_m3\_noise | 0.811609 | 0.930966 | 0.002517 | 0.562222 | 0.005012 |
| Random Forest | dbscan\_full\_e0.3\_m5\_labels | 0.811216 | 0.931362 | 0.002533 | 0.562222 | 0.005044 |
| Random Forest | dbscan\_full\_e0.3\_m5\_noise | 0.811609 | 0.930966 | 0.002517 | 0.562222 | 0.005012 |
| Random Forest | dbscan\_full\_e0.3\_m7\_labels | 0.811609 | 0.930966 | 0.002517 | 0.562222 | 0.005012 |
| Random Forest | dbscan\_full\_e0.3\_m7\_noise | 0.811609 | 0.930966 | 0.002517 | 0.562222 | 0.005012 |
| Random Forest | dbscan\_full\_e0.5\_m10\_labels | 0.811609 | 0.930966 | 0.002517 | 0.562222 | 0.005012 |
| Random Forest | dbscan\_full\_e0.5\_m10\_noise | 0.811609 | 0.930966 | 0.002517 | 0.562222 | 0.005012 |
| Random Forest | dbscan\_full\_e0.5\_m15\_labels | 0.811609 | 0.930966 | 0.002517 | 0.562222 | 0.005012 |
| Random Forest | dbscan\_full\_e0.5\_m15\_noise | 0.811609 | 0.930966 | 0.002517 | 0.562222 | 0.005012 |
| Random Forest | dbscan\_full\_e0.5\_m3\_labels | 0.812763 | 0.930781 | 0.002492 | 0.557778 | 0.004962 |
| Random Forest | dbscan\_full\_e0.5\_m3\_noise | 0.811609 | 0.930966 | 0.002517 | 0.562222 | 0.005012 |
| Random Forest | dbscan\_full\_e0.5\_m5\_labels | 0.812548 | 0.931183 | 0.002542 | 0.564444 | 0.005061 |
| Random Forest | dbscan\_full\_e0.5\_m5\_noise | 0.811609 | 0.930966 | 0.002517 | 0.562222 | 0.005012 |
| Random Forest | dbscan\_full\_e0.5\_m7\_labels | 0.811825 | 0.9313 | 0.002543 | 0.564444 | 0.005063 |
| Random Forest | dbscan\_full\_e0.5\_m7\_noise | 0.811609 | 0.930966 | 0.002517 | 0.562222 | 0.005012 |
| Random Forest | dbscan\_full\_e0.7\_m10\_labels | 0.812059 | 0.931146 | 0.002516 | 0.56 | 0.00501 |
| Random Forest | dbscan\_full\_e0.7\_m10\_noise | 0.811609 | 0.930966 | 0.002517 | 0.562222 | 0.005012 |
| Random Forest | dbscan\_full\_e0.7\_m15\_labels | 0.811609 | 0.930966 | 0.002517 | 0.562222 | 0.005012 |
| Random Forest | dbscan\_full\_e0.7\_m15\_noise | 0.811609 | 0.930966 | 0.002517 | 0.562222 | 0.005012 |
| Random Forest | dbscan\_full\_e0.7\_m3\_labels | 0.811341 | 0.930287 | 0.002508 | 0.564444 | 0.004995 |
| Random Forest | dbscan\_full\_e0.7\_m3\_noise | 0.811609 | 0.930966 | 0.002517 | 0.562222 | 0.005012 |
| Random Forest | dbscan\_full\_e0.7\_m5\_labels | 0.810776 | 0.931256 | 0.002538 | 0.564444 | 0.005053 |
| Random Forest | dbscan\_full\_e0.7\_m5\_noise | 0.811609 | 0.930966 | 0.002517 | 0.562222 | 0.005012 |
| Random Forest | dbscan\_full\_e0.7\_m7\_labels | 0.812947 | 0.931596 | 0.002564 | 0.566667 | 0.005105 |
| Random Forest | dbscan\_full\_e0.7\_m7\_noise | 0.811609 | 0.930966 | 0.002517 | 0.562222 | 0.005012 |
| Random Forest | dbscan\_full\_e1.0\_m10\_labels | 0.812239 | 0.931574 | 0.002543 | 0.562222 | 0.005064 |
| Random Forest | dbscan\_full\_e1.0\_m10\_noise | 0.811609 | 0.930966 | 0.002517 | 0.562222 | 0.005012 |
| Random Forest | dbscan\_full\_e1.0\_m15\_labels | 0.811609 | 0.930966 | 0.002517 | 0.562222 | 0.005012 |
| Random Forest | dbscan\_full\_e1.0\_m15\_noise | 0.811609 | 0.930966 | 0.002517 | 0.562222 | 0.005012 |
| Random Forest | dbscan\_full\_e1.0\_m3\_labels | 0.809711 | 0.928894 | 0.002467 | 0.566667 | 0.004913 |
| Random Forest | dbscan\_full\_e1.0\_m3\_noise | 0.811609 | 0.930966 | 0.002517 | 0.562222 | 0.005012 |
| Random Forest | dbscan\_full\_e1.0\_m5\_labels | 0.812708 | 0.930281 | 0.002496 | 0.562222 | 0.004971 |
| Random Forest | dbscan\_full\_e1.0\_m5\_noise | 0.811609 | 0.930966 | 0.002517 | 0.562222 | 0.005012 |
| Random Forest | dbscan\_full\_e1.0\_m7\_labels | 0.811031 | 0.930794 | 0.002544 | 0.568889 | 0.005065 |
| Random Forest | dbscan\_full\_e1.0\_m7\_noise | 0.811609 | 0.930966 | 0.002517 | 0.562222 | 0.005012 |
| Random Forest | dbscan\_full\_e1.3\_m10\_labels | 0.811589 | 0.931113 | 0.002535 | 0.564444 | 0.005046 |
| Random Forest | dbscan\_full\_e1.3\_m10\_noise | 0.811609 | 0.930966 | 0.002517 | 0.562222 | 0.005012 |
| Random Forest | dbscan\_full\_e1.3\_m15\_labels | 0.812543 | 0.931429 | 0.00254 | 0.562222 | 0.005056 |
| Random Forest | dbscan\_full\_e1.3\_m15\_noise | 0.811609 | 0.930966 | 0.002517 | 0.562222 | 0.005012 |
| Random Forest | dbscan\_full\_e1.3\_m3\_labels | 0.806986 | 0.929765 | 0.002508 | 0.568889 | 0.004994 |
| Random Forest | dbscan\_full\_e1.3\_m3\_noise | 0.807397 | 0.930033 | 0.002486 | 0.562222 | 0.004951 |
| Random Forest | dbscan\_full\_e1.3\_m5\_labels | 0.809873 | 0.929737 | 0.002507 | 0.568889 | 0.004993 |
| Random Forest | dbscan\_full\_e1.3\_m5\_noise | 0.811609 | 0.930966 | 0.002517 | 0.562222 | 0.005012 |
| Random Forest | dbscan\_full\_e1.3\_m7\_labels | 0.812618 | 0.931413 | 0.00258 | 0.571111 | 0.005136 |
| Random Forest | dbscan\_full\_e1.3\_m7\_noise | 0.811609 | 0.930966 | 0.002517 | 0.562222 | 0.005012 |
| Random Forest | dbscan\_pca\_e0.3\_m10\_labels | 0.811609 | 0.930966 | 0.002517 | 0.562222 | 0.005012 |
| Random Forest | dbscan\_pca\_e0.3\_m10\_noise | 0.811609 | 0.930966 | 0.002517 | 0.562222 | 0.005012 |
| Random Forest | dbscan\_pca\_e0.3\_m15\_labels | 0.811609 | 0.930966 | 0.002517 | 0.562222 | 0.005012 |
| Random Forest | dbscan\_pca\_e0.3\_m15\_noise | 0.811609 | 0.930966 | 0.002517 | 0.562222 | 0.005012 |
| Random Forest | dbscan\_pca\_e0.3\_m3\_labels | 0.803822 | 0.931973 | 0.002488 | 0.546667 | 0.004954 |
| Random Forest | dbscan\_pca\_e0.3\_m3\_noise | 0.810008 | 0.931819 | 0.002501 | 0.551111 | 0.004979 |
| Random Forest | dbscan\_pca\_e0.3\_m5\_labels | 0.807145 | 0.931236 | 0.002446 | 0.544444 | 0.00487 |
| Random Forest | dbscan\_pca\_e0.3\_m5\_noise | 0.8099 | 0.931256 | 0.002479 | 0.551111 | 0.004937 |
| Random Forest | dbscan\_pca\_e0.3\_m7\_labels | 0.808321 | 0.931398 | 0.002472 | 0.548889 | 0.004922 |
| Random Forest | dbscan\_pca\_e0.3\_m7\_noise | 0.810545 | 0.931331 | 0.002492 | 0.553333 | 0.004962 |
| Random Forest | dbscan\_pca\_e0.5\_m10\_labels | 0.811609 | 0.930966 | 0.002517 | 0.562222 | 0.005012 |
| Random Forest | dbscan\_pca\_e0.5\_m10\_noise | 0.811609 | 0.930966 | 0.002517 | 0.562222 | 0.005012 |
| Random Forest | dbscan\_pca\_e0.5\_m15\_labels | 0.811609 | 0.930966 | 0.002517 | 0.562222 | 0.005012 |
| Random Forest | dbscan\_pca\_e0.5\_m15\_noise | 0.811609 | 0.930966 | 0.002517 | 0.562222 | 0.005012 |
| Random Forest | dbscan\_pca\_e0.5\_m3\_labels | 0.803822 | 0.931973 | 0.002488 | 0.546667 | 0.004954 |
| Random Forest | dbscan\_pca\_e0.5\_m3\_noise | 0.810008 | 0.931819 | 0.002501 | 0.551111 | 0.004979 |
| Random Forest | dbscan\_pca\_e0.5\_m5\_labels | 0.807145 | 0.931236 | 0.002446 | 0.544444 | 0.00487 |
| Random Forest | dbscan\_pca\_e0.5\_m5\_noise | 0.8099 | 0.931256 | 0.002479 | 0.551111 | 0.004937 |
| Random Forest | dbscan\_pca\_e0.5\_m7\_labels | 0.808321 | 0.931398 | 0.002472 | 0.548889 | 0.004922 |
| Random Forest | dbscan\_pca\_e0.5\_m7\_noise | 0.810545 | 0.931331 | 0.002492 | 0.553333 | 0.004962 |
| Random Forest | dbscan\_pca\_e0.7\_m10\_labels | 0.811609 | 0.930966 | 0.002517 | 0.562222 | 0.005012 |
| Random Forest | dbscan\_pca\_e0.7\_m10\_noise | 0.811609 | 0.930966 | 0.002517 | 0.562222 | 0.005012 |
| Random Forest | dbscan\_pca\_e0.7\_m15\_labels | 0.811609 | 0.930966 | 0.002517 | 0.562222 | 0.005012 |
| Random Forest | dbscan\_pca\_e0.7\_m15\_noise | 0.811609 | 0.930966 | 0.002517 | 0.562222 | 0.005012 |
| Random Forest | dbscan\_pca\_e0.7\_m3\_labels | 0.805106 | 0.93274 | 0.002535 | 0.551111 | 0.005048 |
| Random Forest | dbscan\_pca\_e0.7\_m3\_noise | 0.808933 | 0.932175 | 0.002524 | 0.553333 | 0.005024 |
| Random Forest | dbscan\_pca\_e0.7\_m5\_labels | 0.805042 | 0.931828 | 0.002478 | 0.546667 | 0.004934 |
| Random Forest | dbscan\_pca\_e0.7\_m5\_noise | 0.807039 | 0.931479 | 0.002499 | 0.553333 | 0.004976 |
| Random Forest | dbscan\_pca\_e0.7\_m7\_labels | 0.809359 | 0.931269 | 0.002487 | 0.553333 | 0.004952 |
| Random Forest | dbscan\_pca\_e0.7\_m7\_noise | 0.810986 | 0.931067 | 0.002483 | 0.553333 | 0.004944 |
| Random Forest | dbscan\_pca\_e1.0\_m10\_labels | 0.811609 | 0.930966 | 0.002517 | 0.562222 | 0.005012 |
| Random Forest | dbscan\_pca\_e1.0\_m10\_noise | 0.811609 | 0.930966 | 0.002517 | 0.562222 | 0.005012 |
| Random Forest | dbscan\_pca\_e1.0\_m15\_labels | 0.811609 | 0.930966 | 0.002517 | 0.562222 | 0.005012 |
| Random Forest | dbscan\_pca\_e1.0\_m15\_noise | 0.811609 | 0.930966 | 0.002517 | 0.562222 | 0.005012 |
| Random Forest | dbscan\_pca\_e1.0\_m3\_labels | 0.805106 | 0.93274 | 0.002535 | 0.551111 | 0.005048 |
| Random Forest | dbscan\_pca\_e1.0\_m3\_noise | 0.808933 | 0.932175 | 0.002524 | 0.553333 | 0.005024 |
| Random Forest | dbscan\_pca\_e1.0\_m5\_labels | 0.805042 | 0.931828 | 0.002478 | 0.546667 | 0.004934 |
| Random Forest | dbscan\_pca\_e1.0\_m5\_noise | 0.807039 | 0.931479 | 0.002499 | 0.553333 | 0.004976 |
| Random Forest | dbscan\_pca\_e1.0\_m7\_labels | 0.809359 | 0.931269 | 0.002487 | 0.553333 | 0.004952 |
| Random Forest | dbscan\_pca\_e1.0\_m7\_noise | 0.810986 | 0.931067 | 0.002483 | 0.553333 | 0.004944 |
| Random Forest | dbscan\_pca\_e1.3\_m10\_labels | 0.812972 | 0.930943 | 0.002562 | 0.571111 | 0.005101 |
| Random Forest | dbscan\_pca\_e1.3\_m10\_noise | 0.811609 | 0.930966 | 0.002517 | 0.562222 | 0.005012 |
| Random Forest | dbscan\_pca\_e1.3\_m15\_labels | 0.811609 | 0.930966 | 0.002517 | 0.562222 | 0.005012 |
| Random Forest | dbscan\_pca\_e1.3\_m15\_noise | 0.811609 | 0.930966 | 0.002517 | 0.562222 | 0.005012 |
| Random Forest | dbscan\_pca\_e1.3\_m3\_labels | 0.803971 | 0.931961 | 0.00249 | 0.546667 | 0.004957 |
| Random Forest | dbscan\_pca\_e1.3\_m3\_noise | 0.8083 | 0.93227 | 0.002528 | 0.553333 | 0.005034 |
| Random Forest | dbscan\_pca\_e1.3\_m5\_labels | 0.805964 | 0.931885 | 0.002491 | 0.548889 | 0.00496 |
| Random Forest | dbscan\_pca\_e1.3\_m5\_noise | 0.80631 | 0.931561 | 0.002481 | 0.548889 | 0.004939 |
| Random Forest | dbscan\_pca\_e1.3\_m7\_labels | 0.810106 | 0.931345 | 0.002481 | 0.551111 | 0.004941 |
| Random Forest | dbscan\_pca\_e1.3\_m7\_noise | 0.810812 | 0.930768 | 0.002474 | 0.553333 | 0.004926 |
| Random Forest | hier\_full\_n2\_labels | 0.812094 | 0.931292 | 0.002529 | 0.562222 | 0.005035 |
| Random Forest | hier\_full\_n3\_labels | 0.812138 | 0.931218 | 0.002537 | 0.564444 | 0.005051 |
| Random Forest | hier\_full\_n4\_labels | 0.812804 | 0.931164 | 0.002504 | 0.557778 | 0.004986 |
| Random Forest | hier\_full\_n5\_labels | 0.812982 | 0.931424 | 0.002546 | 0.564444 | 0.00507 |
| Random Forest | hier\_full\_n6\_labels | 0.811931 | 0.931575 | 0.002551 | 0.564444 | 0.005079 |
| Random Forest | hier\_pca\_n2\_labels | 0.811387 | 0.93114 | 0.002555 | 0.568889 | 0.005087 |
| Random Forest | hier\_pca\_n3\_labels | 0.81213 | 0.931103 | 0.002545 | 0.566667 | 0.005067 |
| Random Forest | hier\_pca\_n4\_labels | 0.812015 | 0.931089 | 0.002534 | 0.564444 | 0.005045 |
| Random Forest | hier\_pca\_n5\_labels | 0.811219 | 0.93162 | 0.002563 | 0.566667 | 0.005103 |
| Random Forest | hier\_pca\_n6\_labels | 0.81183 | 0.931406 | 0.002554 | 0.566667 | 0.005085 |
| Random Forest | kmeans\_full\_n10\_labels | 0.780603 | 0.956645 | 0.003745 | 0.522222 | 0.007436 |
| Random Forest | kmeans\_full\_n2\_labels | 0.812353 | 0.931262 | 0.002518 | 0.56 | 0.005013 |
| Random Forest | kmeans\_full\_n3\_labels | 0.813509 | 0.931564 | 0.00252 | 0.557778 | 0.005017 |
| Random Forest | kmeans\_full\_n4\_labels | 0.812982 | 0.931424 | 0.002546 | 0.564444 | 0.00507 |
| Random Forest | kmeans\_full\_n5\_labels | 0.789435 | 0.934598 | 0.002579 | 0.544444 | 0.005134 |
| Random Forest | kmeans\_full\_n6\_labels | 0.761899 | 0.949652 | 0.002541 | 0.413333 | 0.00505 |
| Random Forest | kmeans\_full\_n7\_labels | 0.694685 | 0.946818 | 0.002392 | 0.406667 | 0.004755 |
| Random Forest | kmeans\_full\_n8\_labels | 0.740541 | 0.95778 | 0.003229 | 0.437778 | 0.006411 |
| Random Forest | kmeans\_full\_n9\_labels | 0.731803 | 0.958237 | 0.002891 | 0.388889 | 0.005739 |
| Random Forest | kmeans\_pca\_n10\_labels | 0.766272 | 0.939694 | 0.002691 | 0.522222 | 0.005355 |
| Random Forest | kmeans\_pca\_n2\_labels | 0.798694 | 0.933309 | 0.002576 | 0.555556 | 0.005128 |
| Random Forest | kmeans\_pca\_n3\_labels | 0.77985 | 0.935315 | 0.002661 | 0.555556 | 0.005296 |
| Random Forest | kmeans\_pca\_n4\_labels | 0.777278 | 0.937096 | 0.002668 | 0.542222 | 0.005309 |
| Random Forest | kmeans\_pca\_n5\_labels | 0.770301 | 0.93797 | 0.002689 | 0.537778 | 0.005352 |
| Random Forest | kmeans\_pca\_n6\_labels | 0.770409 | 0.93823 | 0.002573 | 0.513333 | 0.00512 |
| Random Forest | kmeans\_pca\_n7\_labels | 0.773964 | 0.941206 | 0.002798 | 0.531111 | 0.005567 |
| Random Forest | kmeans\_pca\_n8\_labels | 0.776851 | 0.94049 | 0.002833 | 0.544444 | 0.005636 |
| Random Forest | kmeans\_pca\_n9\_labels | 0.754498 | 0.941035 | 0.002622 | 0.497778 | 0.005216 |
| Neural Net | Baseline | 0.839166 | 0.804846 | 0.001208 | 0.735556 | 0.002411 |
| Neural Net | dbscan\_full\_e0.3\_m10\_labels | 0.820875 | 0.781792 | 0.001696 | 0.737778 | 0.003382 |
| Neural Net | dbscan\_full\_e0.3\_m10\_noise | 0.844122 | 0.861149 | 0.001715 | 0.708889 | 0.003421 |
| Neural Net | dbscan\_full\_e0.3\_m15\_labels | 0.792026 | 0.761045 | 0.001011 | 0.737778 | 0.002018 |
| Neural Net | dbscan\_full\_e0.3\_m15\_noise | 0.833305 | 0.851387 | 0.001534 | 0.704444 | 0.003061 |
| Neural Net | dbscan\_full\_e0.3\_m3\_labels | 0.839104 | 0.864391 | 0.001804 | 0.708889 | 0.003597 |
| Neural Net | dbscan\_full\_e0.3\_m3\_noise | 0.824811 | 0.736511 | 0.001334 | 0.748889 | 0.002661 |
| Neural Net | dbscan\_full\_e0.3\_m5\_labels | 0.828908 | 0.849832 | 0.001982 | 0.708889 | 0.003945 |
| Neural Net | dbscan\_full\_e0.3\_m5\_noise | 0.809554 | 0.780019 | 0.001389 | 0.74 | 0.002771 |
| Neural Net | dbscan\_full\_e0.3\_m7\_labels | 0.788097 | 0.793135 | 0.001699 | 0.711111 | 0.003387 |
| Neural Net | dbscan\_full\_e0.3\_m7\_noise | 0.825161 | 0.851088 | 0.001736 | 0.688889 | 0.003462 |
| Neural Net | dbscan\_full\_e0.5\_m10\_labels | 0.825742 | 0.893772 | 0.002003 | 0.637778 | 0.003993 |
| Neural Net | dbscan\_full\_e0.5\_m10\_noise | 0.844586 | 0.872477 | 0.001869 | 0.684444 | 0.003726 |
| Neural Net | dbscan\_full\_e0.5\_m15\_labels | 0.819951 | 0.807795 | 0.001496 | 0.726667 | 0.002983 |
| Neural Net | dbscan\_full\_e0.5\_m15\_noise | 0.829319 | 0.76953 | 0.001286 | 0.768889 | 0.002567 |
| Neural Net | dbscan\_full\_e0.5\_m3\_labels | 0.796823 | 0.806627 | 0.001395 | 0.671111 | 0.002784 |
| Neural Net | dbscan\_full\_e0.5\_m3\_noise | 0.798546 | 0.741604 | 0.001108 | 0.762222 | 0.002212 |
| Neural Net | dbscan\_full\_e0.5\_m5\_labels | 0.833297 | 0.848306 | 0.001749 | 0.731111 | 0.003488 |
| Neural Net | dbscan\_full\_e0.5\_m5\_noise | 0.828814 | 0.845511 | 0.001665 | 0.68 | 0.00332 |
| Neural Net | dbscan\_full\_e0.5\_m7\_labels | 0.838562 | 0.804156 | 0.001291 | 0.755556 | 0.002576 |
| Neural Net | dbscan\_full\_e0.5\_m7\_noise | 0.821236 | 0.83839 | 0.001428 | 0.733333 | 0.00285 |
| Neural Net | dbscan\_full\_e0.7\_m10\_labels | 0.801166 | 0.845316 | 0.001774 | 0.677778 | 0.003535 |
| Neural Net | dbscan\_full\_e0.7\_m10\_noise | 0.803042 | 0.748155 | 0.000996 | 0.768889 | 0.00199 |
| Neural Net | dbscan\_full\_e0.7\_m15\_labels | 0.802354 | 0.830572 | 0.001733 | 0.693333 | 0.003453 |
| Neural Net | dbscan\_full\_e0.7\_m15\_noise | 0.79537 | 0.728723 | 0.001455 | 0.744444 | 0.0029 |
| Neural Net | dbscan\_full\_e0.7\_m3\_labels | 0.844892 | 0.820814 | 0.001311 | 0.751111 | 0.002618 |
| Neural Net | dbscan\_full\_e0.7\_m3\_noise | 0.812675 | 0.855234 | 0.00163 | 0.671111 | 0.003251 |
| Neural Net | dbscan\_full\_e0.7\_m5\_labels | 0.819366 | 0.766496 | 0.001283 | 0.804444 | 0.002561 |
| Neural Net | dbscan\_full\_e0.7\_m5\_noise | 0.824409 | 0.770486 | 0.001161 | 0.768889 | 0.002318 |
| Neural Net | dbscan\_full\_e0.7\_m7\_labels | 0.83436 | 0.869292 | 0.002169 | 0.702222 | 0.004318 |
| Neural Net | dbscan\_full\_e0.7\_m7\_noise | 0.795915 | 0.90334 | 0.002019 | 0.566667 | 0.004021 |
| Neural Net | dbscan\_full\_e1.0\_m10\_labels | 0.790789 | 0.823069 | 0.00167 | 0.686667 | 0.00333 |
| Neural Net | dbscan\_full\_e1.0\_m10\_noise | 0.800217 | 0.734151 | 0.001009 | 0.766667 | 0.002016 |
| Neural Net | dbscan\_full\_e1.0\_m15\_labels | 0.836016 | 0.820216 | 0.002191 | 0.711111 | 0.004361 |
| Neural Net | dbscan\_full\_e1.0\_m15\_noise | 0.826226 | 0.854781 | 0.001672 | 0.731111 | 0.003336 |
| Neural Net | dbscan\_full\_e1.0\_m3\_labels | 0.820565 | 0.854659 | 0.001589 | 0.704444 | 0.00317 |
| Neural Net | dbscan\_full\_e1.0\_m3\_noise | 0.80624 | 0.846268 | 0.001796 | 0.664444 | 0.003579 |
| Neural Net | dbscan\_full\_e1.0\_m5\_labels | 0.81788 | 0.715395 | 0.001151 | 0.766667 | 0.002298 |
| Neural Net | dbscan\_full\_e1.0\_m5\_noise | 0.814383 | 0.833898 | 0.001463 | 0.706667 | 0.00292 |
| Neural Net | dbscan\_full\_e1.0\_m7\_labels | 0.841338 | 0.812015 | 0.001359 | 0.742222 | 0.002713 |
| Neural Net | dbscan\_full\_e1.0\_m7\_noise | 0.828759 | 0.856015 | 0.001665 | 0.688889 | 0.003321 |
| Neural Net | dbscan\_full\_e1.3\_m10\_labels | 0.810849 | 0.801278 | 0.001191 | 0.731111 | 0.002378 |
| Neural Net | dbscan\_full\_e1.3\_m10\_noise | 0.837569 | 0.801403 | 0.001361 | 0.748889 | 0.002716 |
| Neural Net | dbscan\_full\_e1.3\_m15\_labels | 0.819685 | 0.84826 | 0.0015 | 0.713333 | 0.002993 |
| Neural Net | dbscan\_full\_e1.3\_m15\_noise | 0.816735 | 0.848581 | 0.00149 | 0.675556 | 0.002972 |
| Neural Net | dbscan\_full\_e1.3\_m3\_labels | 0.820071 | 0.791452 | 0.001452 | 0.744444 | 0.002894 |
| Neural Net | dbscan\_full\_e1.3\_m3\_noise | 0.844341 | 0.84803 | 0.001581 | 0.737778 | 0.003154 |
| Neural Net | dbscan\_full\_e1.3\_m5\_labels | 0.817638 | 0.768894 | 0.001459 | 0.72 | 0.002909 |
| Neural Net | dbscan\_full\_e1.3\_m5\_noise | 0.778977 | 0.761092 | 0.001276 | 0.742222 | 0.002547 |
| Neural Net | dbscan\_full\_e1.3\_m7\_labels | 0.817143 | 0.771083 | 0.001528 | 0.742222 | 0.003046 |
| Neural Net | dbscan\_full\_e1.3\_m7\_noise | 0.826844 | 0.803908 | 0.001221 | 0.742222 | 0.002438 |
| Neural Net | dbscan\_pca\_e0.3\_m10\_labels | 0.799863 | 0.765157 | 0.00134 | 0.74 | 0.002673 |
| Neural Net | dbscan\_pca\_e0.3\_m10\_noise | 0.828027 | 0.792869 | 0.001527 | 0.746667 | 0.003045 |
| Neural Net | dbscan\_pca\_e0.3\_m15\_labels | 0.803836 | 0.737657 | 0.00114 | 0.768889 | 0.002276 |
| Neural Net | dbscan\_pca\_e0.3\_m15\_noise | 0.825736 | 0.816021 | 0.001637 | 0.751111 | 0.003265 |
| Neural Net | dbscan\_pca\_e0.3\_m3\_labels | 0.805189 | 0.817209 | 0.002132 | 0.68 | 0.004246 |
| Neural Net | dbscan\_pca\_e0.3\_m3\_noise | 0.817625 | 0.851243 | 0.001703 | 0.713333 | 0.003396 |
| Neural Net | dbscan\_pca\_e0.3\_m5\_labels | 0.779646 | 0.769774 | 0.001139 | 0.746667 | 0.002274 |
| Neural Net | dbscan\_pca\_e0.3\_m5\_noise | 0.74494 | 0.665589 | 0.001667 | 0.78 | 0.003317 |
| Neural Net | dbscan\_pca\_e0.3\_m7\_labels | 0.826437 | 0.822711 | 0.001465 | 0.742222 | 0.002924 |
| Neural Net | dbscan\_pca\_e0.3\_m7\_noise | 0.798212 | 0.810851 | 0.001173 | 0.684444 | 0.002341 |
| Neural Net | dbscan\_pca\_e0.5\_m10\_labels | 0.830855 | 0.884578 | 0.002348 | 0.655556 | 0.004674 |
| Neural Net | dbscan\_pca\_e0.5\_m10\_noise | 0.818977 | 0.783391 | 0.00127 | 0.773333 | 0.002536 |
| Neural Net | dbscan\_pca\_e0.5\_m15\_labels | 0.823544 | 0.863986 | 0.00174 | 0.68 | 0.003471 |
| Neural Net | dbscan\_pca\_e0.5\_m15\_noise | 0.803885 | 0.831796 | 0.001751 | 0.655556 | 0.003475 |
| Neural Net | dbscan\_pca\_e0.5\_m3\_labels | 0.831017 | 0.799886 | 0.001397 | 0.735556 | 0.002787 |
| Neural Net | dbscan\_pca\_e0.5\_m3\_noise | 0.8241 | 0.858061 | 0.00235 | 0.675556 | 0.00466 |
| Neural Net | dbscan\_pca\_e0.5\_m5\_labels | 0.843864 | 0.858723 | 0.001618 | 0.711111 | 0.003228 |
| Neural Net | dbscan\_pca\_e0.5\_m5\_noise | 0.787455 | 0.760635 | 0.001749 | 0.724444 | 0.003482 |
| Neural Net | dbscan\_pca\_e0.5\_m7\_labels | 0.833835 | 0.848211 | 0.001548 | 0.711111 | 0.003089 |
| Neural Net | dbscan\_pca\_e0.5\_m7\_noise | 0.815797 | 0.849154 | 0.001612 | 0.708889 | 0.003215 |
| Neural Net | dbscan\_pca\_e0.7\_m10\_labels | 0.837348 | 0.874537 | 0.00184 | 0.653333 | 0.003668 |
| Neural Net | dbscan\_pca\_e0.7\_m10\_noise | 0.817954 | 0.715706 | 0.001181 | 0.755556 | 0.002357 |
| Neural Net | dbscan\_pca\_e0.7\_m15\_labels | 0.849874 | 0.883894 | 0.00223 | 0.697778 | 0.004443 |
| Neural Net | dbscan\_pca\_e0.7\_m15\_noise | 0.830315 | 0.801121 | 0.001235 | 0.78 | 0.002466 |
| Neural Net | dbscan\_pca\_e0.7\_m3\_labels | 0.749154 | 0.705535 | 0.001175 | 0.702222 | 0.002344 |
| Neural Net | dbscan\_pca\_e0.7\_m3\_noise | 0.784673 | 0.666754 | 0.000797 | 0.773333 | 0.001593 |
| Neural Net | dbscan\_pca\_e0.7\_m5\_labels | 0.835539 | 0.863041 | 0.00164 | 0.722222 | 0.003272 |
| Neural Net | dbscan\_pca\_e0.7\_m5\_noise | 0.811525 | 0.819686 | 0.002022 | 0.733333 | 0.004027 |
| Neural Net | dbscan\_pca\_e0.7\_m7\_labels | 0.838563 | 0.808842 | 0.001711 | 0.744444 | 0.003407 |
| Neural Net | dbscan\_pca\_e0.7\_m7\_noise | 0.827571 | 0.863684 | 0.001975 | 0.677778 | 0.003935 |
| Neural Net | dbscan\_pca\_e1.0\_m10\_labels | 0.79009 | 0.805344 | 0.002376 | 0.644444 | 0.004701 |
| Neural Net | dbscan\_pca\_e1.0\_m10\_noise | 0.783325 | 0.802 | 0.001286 | 0.697778 | 0.002567 |
| Neural Net | dbscan\_pca\_e1.0\_m15\_labels | 0.798294 | 0.80587 | 0.001354 | 0.711111 | 0.002702 |
| Neural Net | dbscan\_pca\_e1.0\_m15\_noise | 0.793502 | 0.757196 | 0.0018 | 0.708889 | 0.00358 |
| Neural Net | dbscan\_pca\_e1.0\_m3\_labels | 0.732462 | 0.670586 | 0.001185 | 0.717778 | 0.002364 |
| Neural Net | dbscan\_pca\_e1.0\_m3\_noise | 0.834847 | 0.906892 | 0.002315 | 0.66 | 0.004613 |
| Neural Net | dbscan\_pca\_e1.0\_m5\_labels | 0.803717 | 0.755101 | 0.001668 | 0.713333 | 0.003321 |
| Neural Net | dbscan\_pca\_e1.0\_m5\_noise | 0.809682 | 0.756897 | 0.00139 | 0.753333 | 0.002772 |
| Neural Net | dbscan\_pca\_e1.0\_m7\_labels | 0.82462 | 0.802529 | 0.001204 | 0.726667 | 0.002404 |
| Neural Net | dbscan\_pca\_e1.0\_m7\_noise | 0.785853 | 0.783816 | 0.001079 | 0.713333 | 0.002154 |
| Neural Net | dbscan\_pca\_e1.3\_m10\_labels | 0.811467 | 0.750727 | 0.001217 | 0.76 | 0.002429 |
| Neural Net | dbscan\_pca\_e1.3\_m10\_noise | 0.817856 | 0.840271 | 0.001494 | 0.724444 | 0.002982 |
| Neural Net | dbscan\_pca\_e1.3\_m15\_labels | 0.76546 | 0.815271 | 0.00203 | 0.655556 | 0.004039 |
| Neural Net | dbscan\_pca\_e1.3\_m15\_noise | 0.822098 | 0.835841 | 0.001475 | 0.724444 | 0.002943 |
| Neural Net | dbscan\_pca\_e1.3\_m3\_labels | 0.841612 | 0.832277 | 0.001488 | 0.755556 | 0.002969 |
| Neural Net | dbscan\_pca\_e1.3\_m3\_noise | 0.846506 | 0.835265 | 0.001364 | 0.702222 | 0.002723 |
| Neural Net | dbscan\_pca\_e1.3\_m5\_labels | 0.803975 | 0.771008 | 0.001293 | 0.731111 | 0.002581 |
| Neural Net | dbscan\_pca\_e1.3\_m5\_noise | 0.815235 | 0.858149 | 0.001849 | 0.651111 | 0.003683 |
| Neural Net | dbscan\_pca\_e1.3\_m7\_labels | 0.775724 | 0.779834 | 0.001178 | 0.691111 | 0.002352 |
| Neural Net | dbscan\_pca\_e1.3\_m7\_noise | 0.797318 | 0.823844 | 0.001221 | 0.668889 | 0.002438 |
| Neural Net | hier\_full\_n2\_labels | 0.810548 | 0.870645 | 0.001709 | 0.642222 | 0.003408 |
| Neural Net | hier\_full\_n3\_labels | 0.816494 | 0.785241 | 0.001302 | 0.762222 | 0.002597 |
| Neural Net | hier\_full\_n4\_labels | 0.794817 | 0.753283 | 0.001241 | 0.764444 | 0.002476 |
| Neural Net | hier\_full\_n5\_labels | 0.8398 | 0.857416 | 0.001666 | 0.726667 | 0.003324 |
| Neural Net | hier\_full\_n6\_labels | 0.842099 | 0.827203 | 0.001548 | 0.766667 | 0.003088 |
| Neural Net | hier\_pca\_n2\_labels | 0.845802 | 0.854561 | 0.001695 | 0.717778 | 0.003382 |
| Neural Net | hier\_pca\_n3\_labels | 0.841621 | 0.800211 | 0.001194 | 0.766667 | 0.002384 |
| Neural Net | hier\_pca\_n4\_labels | 0.828115 | 0.854197 | 0.001618 | 0.675556 | 0.003225 |
| Neural Net | hier\_pca\_n5\_labels | 0.825681 | 0.812139 | 0.001638 | 0.746667 | 0.003264 |
| Neural Net | hier\_pca\_n6\_labels | 0.795073 | 0.83945 | 0.001905 | 0.66 | 0.003795 |
| Neural Net | kmeans\_full\_n10\_labels | 0.824275 | 0.844357 | 0.001704 | 0.737778 | 0.0034 |
| Neural Net | kmeans\_full\_n2\_labels | 0.842569 | 0.825783 | 0.00177 | 0.72 | 0.003522 |
| Neural Net | kmeans\_full\_n3\_labels | 0.807505 | 0.776725 | 0.001486 | 0.737778 | 0.002963 |
| Neural Net | kmeans\_full\_n4\_labels | 0.800508 | 0.750289 | 0.001275 | 0.751111 | 0.002544 |
| Neural Net | kmeans\_full\_n5\_labels | 0.838091 | 0.893141 | 0.002468 | 0.668889 | 0.004912 |
| Neural Net | kmeans\_full\_n6\_labels | 0.790385 | 0.753868 | 0.001618 | 0.702222 | 0.003217 |
| Neural Net | kmeans\_full\_n7\_labels | 0.822424 | 0.843379 | 0.001755 | 0.717778 | 0.003497 |
| Neural Net | kmeans\_full\_n8\_labels | 0.813921 | 0.755811 | 0.001136 | 0.742222 | 0.002269 |
| Neural Net | kmeans\_full\_n9\_labels | 0.84889 | 0.821735 | 0.00144 | 0.768889 | 0.002873 |
| Neural Net | kmeans\_pca\_n10\_labels | 0.817456 | 0.783888 | 0.001247 | 0.762222 | 0.00249 |
| Neural Net | kmeans\_pca\_n2\_labels | 0.820439 | 0.807769 | 0.001667 | 0.735556 | 0.003324 |
| Neural Net | kmeans\_pca\_n3\_labels | 0.836157 | 0.783899 | 0.001162 | 0.775556 | 0.00232 |
| Neural Net | kmeans\_pca\_n4\_labels | 0.811525 | 0.759424 | 0.001251 | 0.768889 | 0.002496 |
| Neural Net | kmeans\_pca\_n5\_labels | 0.798701 | 0.688572 | 0.001418 | 0.804444 | 0.002829 |
| Neural Net | kmeans\_pca\_n6\_labels | 0.828253 | 0.829173 | 0.001628 | 0.744444 | 0.003248 |
| Neural Net | kmeans\_pca\_n7\_labels | 0.829227 | 0.858165 | 0.001592 | 0.7 | 0.003176 |
| Neural Net | kmeans\_pca\_n8\_labels | 0.80712 | 0.825677 | 0.001582 | 0.666667 | 0.003152 |
| Neural Net | kmeans\_pca\_n9\_labels | 0.811581 | 0.736667 | 0.001002 | 0.753333 | 0.002002 |
| Naive Bayes | Baseline | 0.867492 | 0.8406 | 0.001424 | 0.733333 | 0.002842 |
| Naive Bayes | dbscan\_full\_e0.3\_m10\_labels | 0.86748 | 0.8406 | 0.001424 | 0.733333 | 0.002842 |
| Naive Bayes | dbscan\_full\_e0.3\_m10\_noise | 0.86748 | 0.840599 | 0.001424 | 0.733333 | 0.002842 |
| Naive Bayes | dbscan\_full\_e0.3\_m15\_labels | 0.86748 | 0.8406 | 0.001424 | 0.733333 | 0.002842 |
| Naive Bayes | dbscan\_full\_e0.3\_m15\_noise | 0.86748 | 0.8406 | 0.001424 | 0.733333 | 0.002842 |
| Naive Bayes | dbscan\_full\_e0.3\_m3\_labels | 0.86748 | 0.8406 | 0.001424 | 0.733333 | 0.002842 |
| Naive Bayes | dbscan\_full\_e0.3\_m3\_noise | 0.86748 | 0.840599 | 0.001424 | 0.733333 | 0.002842 |
| Naive Bayes | dbscan\_full\_e0.3\_m5\_labels | 0.86748 | 0.8406 | 0.001424 | 0.733333 | 0.002842 |
| Naive Bayes | dbscan\_full\_e0.3\_m5\_noise | 0.86748 | 0.8406 | 0.001424 | 0.733333 | 0.002842 |
| Naive Bayes | dbscan\_full\_e0.3\_m7\_labels | 0.86748 | 0.8406 | 0.001424 | 0.733333 | 0.002842 |
| Naive Bayes | dbscan\_full\_e0.3\_m7\_noise | 0.867481 | 0.8406 | 0.001424 | 0.733333 | 0.002842 |
| Naive Bayes | dbscan\_full\_e0.5\_m10\_labels | 0.86748 | 0.8406 | 0.001424 | 0.733333 | 0.002842 |
| Naive Bayes | dbscan\_full\_e0.5\_m10\_noise | 0.86748 | 0.8406 | 0.001424 | 0.733333 | 0.002842 |
| Naive Bayes | dbscan\_full\_e0.5\_m15\_labels | 0.86748 | 0.8406 | 0.001424 | 0.733333 | 0.002842 |
| Naive Bayes | dbscan\_full\_e0.5\_m15\_noise | 0.86748 | 0.8406 | 0.001424 | 0.733333 | 0.002842 |
| Naive Bayes | dbscan\_full\_e0.5\_m3\_labels | 0.86748 | 0.840595 | 0.001424 | 0.733333 | 0.002842 |
| Naive Bayes | dbscan\_full\_e0.5\_m3\_noise | 0.86748 | 0.8406 | 0.001424 | 0.733333 | 0.002842 |
| Naive Bayes | dbscan\_full\_e0.5\_m5\_labels | 0.86748 | 0.8406 | 0.001424 | 0.733333 | 0.002842 |
| Naive Bayes | dbscan\_full\_e0.5\_m5\_noise | 0.86748 | 0.8406 | 0.001424 | 0.733333 | 0.002842 |
| Naive Bayes | dbscan\_full\_e0.5\_m7\_labels | 0.86748 | 0.8406 | 0.001424 | 0.733333 | 0.002842 |
| Naive Bayes | dbscan\_full\_e0.5\_m7\_noise | 0.867481 | 0.8406 | 0.001424 | 0.733333 | 0.002842 |
| Naive Bayes | dbscan\_full\_e0.7\_m10\_labels | 0.86748 | 0.8406 | 0.001424 | 0.733333 | 0.002842 |
| Naive Bayes | dbscan\_full\_e0.7\_m10\_noise | 0.86748 | 0.8406 | 0.001424 | 0.733333 | 0.002842 |
| Naive Bayes | dbscan\_full\_e0.7\_m15\_labels | 0.86748 | 0.8406 | 0.001424 | 0.733333 | 0.002842 |
| Naive Bayes | dbscan\_full\_e0.7\_m15\_noise | 0.867481 | 0.8406 | 0.001424 | 0.733333 | 0.002842 |
| Naive Bayes | dbscan\_full\_e0.7\_m3\_labels | 0.867488 | 0.840549 | 0.001423 | 0.733333 | 0.002841 |
| Naive Bayes | dbscan\_full\_e0.7\_m3\_noise | 0.86748 | 0.8406 | 0.001424 | 0.733333 | 0.002842 |
| Naive Bayes | dbscan\_full\_e0.7\_m5\_labels | 0.86748 | 0.840599 | 0.001424 | 0.733333 | 0.002842 |
| Naive Bayes | dbscan\_full\_e0.7\_m5\_noise | 0.86748 | 0.8406 | 0.001424 | 0.733333 | 0.002842 |
| Naive Bayes | dbscan\_full\_e0.7\_m7\_labels | 0.86748 | 0.8406 | 0.001424 | 0.733333 | 0.002842 |
| Naive Bayes | dbscan\_full\_e0.7\_m7\_noise | 0.86748 | 0.8406 | 0.001424 | 0.733333 | 0.002842 |
| Naive Bayes | dbscan\_full\_e1.0\_m10\_labels | 0.86748 | 0.8406 | 0.001424 | 0.733333 | 0.002842 |
| Naive Bayes | dbscan\_full\_e1.0\_m10\_noise | 0.86748 | 0.8406 | 0.001424 | 0.733333 | 0.002842 |
| Naive Bayes | dbscan\_full\_e1.0\_m15\_labels | 0.86748 | 0.8406 | 0.001424 | 0.733333 | 0.002842 |
| Naive Bayes | dbscan\_full\_e1.0\_m15\_noise | 0.86748 | 0.8406 | 0.001424 | 0.733333 | 0.002842 |
| Naive Bayes | dbscan\_full\_e1.0\_m3\_labels | 0.867641 | 0.839117 | 0.001415 | 0.735556 | 0.002825 |
| Naive Bayes | dbscan\_full\_e1.0\_m3\_noise | 0.86748 | 0.8406 | 0.001424 | 0.733333 | 0.002842 |
| Naive Bayes | dbscan\_full\_e1.0\_m5\_labels | 0.86748 | 0.840594 | 0.001424 | 0.733333 | 0.002842 |
| Naive Bayes | dbscan\_full\_e1.0\_m5\_noise | 0.86748 | 0.8406 | 0.001424 | 0.733333 | 0.002842 |
| Naive Bayes | dbscan\_full\_e1.0\_m7\_labels | 0.86748 | 0.8406 | 0.001424 | 0.733333 | 0.002842 |
| Naive Bayes | dbscan\_full\_e1.0\_m7\_noise | 0.86748 | 0.8406 | 0.001424 | 0.733333 | 0.002842 |
| Naive Bayes | dbscan\_full\_e1.3\_m10\_labels | 0.86748 | 0.8406 | 0.001424 | 0.733333 | 0.002842 |
| Naive Bayes | dbscan\_full\_e1.3\_m10\_noise | 0.86748 | 0.8406 | 0.001424 | 0.733333 | 0.002842 |
| Naive Bayes | dbscan\_full\_e1.3\_m15\_labels | 0.86748 | 0.8406 | 0.001424 | 0.733333 | 0.002842 |
| Naive Bayes | dbscan\_full\_e1.3\_m15\_noise | 0.86748 | 0.8406 | 0.001424 | 0.733333 | 0.002842 |
| Naive Bayes | dbscan\_full\_e1.3\_m3\_labels | 0.868776 | 0.771649 | 0.001045 | 0.771111 | 0.002088 |
| Naive Bayes | dbscan\_full\_e1.3\_m3\_noise | 0.86748 | 0.8406 | 0.001424 | 0.733333 | 0.002842 |
| Naive Bayes | dbscan\_full\_e1.3\_m5\_labels | 0.867561 | 0.840198 | 0.00142 | 0.733333 | 0.002835 |
| Naive Bayes | dbscan\_full\_e1.3\_m5\_noise | 0.86748 | 0.8406 | 0.001424 | 0.733333 | 0.002842 |
| Naive Bayes | dbscan\_full\_e1.3\_m7\_labels | 0.867481 | 0.840585 | 0.001424 | 0.733333 | 0.002842 |
| Naive Bayes | dbscan\_full\_e1.3\_m7\_noise | 0.867481 | 0.8406 | 0.001424 | 0.733333 | 0.002842 |
| Naive Bayes | dbscan\_pca\_e0.3\_m10\_labels | 0.86748 | 0.8406 | 0.001424 | 0.733333 | 0.002842 |
| Naive Bayes | dbscan\_pca\_e0.3\_m10\_noise | 0.86748 | 0.8406 | 0.001424 | 0.733333 | 0.002842 |
| Naive Bayes | dbscan\_pca\_e0.3\_m15\_labels | 0.86748 | 0.8406 | 0.001424 | 0.733333 | 0.002842 |
| Naive Bayes | dbscan\_pca\_e0.3\_m15\_noise | 0.86748 | 0.8406 | 0.001424 | 0.733333 | 0.002842 |
| Naive Bayes | dbscan\_pca\_e0.3\_m3\_labels | 0.867481 | 0.840714 | 0.001425 | 0.733333 | 0.002844 |
| Naive Bayes | dbscan\_pca\_e0.3\_m3\_noise | 0.86748 | 0.8406 | 0.001424 | 0.733333 | 0.002842 |
| Naive Bayes | dbscan\_pca\_e0.3\_m5\_labels | 0.86748 | 0.840604 | 0.001424 | 0.733333 | 0.002842 |
| Naive Bayes | dbscan\_pca\_e0.3\_m5\_noise | 0.86748 | 0.8406 | 0.001424 | 0.733333 | 0.002842 |
| Naive Bayes | dbscan\_pca\_e0.3\_m7\_labels | 0.86748 | 0.8406 | 0.001424 | 0.733333 | 0.002842 |
| Naive Bayes | dbscan\_pca\_e0.3\_m7\_noise | 0.86748 | 0.8406 | 0.001424 | 0.733333 | 0.002842 |
| Naive Bayes | dbscan\_pca\_e0.5\_m10\_labels | 0.86748 | 0.8406 | 0.001424 | 0.733333 | 0.002842 |
| Naive Bayes | dbscan\_pca\_e0.5\_m10\_noise | 0.86748 | 0.8406 | 0.001424 | 0.733333 | 0.002842 |
| Naive Bayes | dbscan\_pca\_e0.5\_m15\_labels | 0.86748 | 0.8406 | 0.001424 | 0.733333 | 0.002842 |
| Naive Bayes | dbscan\_pca\_e0.5\_m15\_noise | 0.86748 | 0.8406 | 0.001424 | 0.733333 | 0.002842 |
| Naive Bayes | dbscan\_pca\_e0.5\_m3\_labels | 0.86748 | 0.840714 | 0.001425 | 0.733333 | 0.002844 |
| Naive Bayes | dbscan\_pca\_e0.5\_m3\_noise | 0.86748 | 0.8406 | 0.001424 | 0.733333 | 0.002842 |
| Naive Bayes | dbscan\_pca\_e0.5\_m5\_labels | 0.86748 | 0.840604 | 0.001424 | 0.733333 | 0.002842 |
| Naive Bayes | dbscan\_pca\_e0.5\_m5\_noise | 0.86748 | 0.8406 | 0.001424 | 0.733333 | 0.002842 |
| Naive Bayes | dbscan\_pca\_e0.5\_m7\_labels | 0.867481 | 0.8406 | 0.001424 | 0.733333 | 0.002842 |
| Naive Bayes | dbscan\_pca\_e0.5\_m7\_noise | 0.86748 | 0.8406 | 0.001424 | 0.733333 | 0.002842 |
| Naive Bayes | dbscan\_pca\_e0.7\_m10\_labels | 0.86748 | 0.8406 | 0.001424 | 0.733333 | 0.002842 |
| Naive Bayes | dbscan\_pca\_e0.7\_m10\_noise | 0.86748 | 0.8406 | 0.001424 | 0.733333 | 0.002842 |
| Naive Bayes | dbscan\_pca\_e0.7\_m15\_labels | 0.86748 | 0.8406 | 0.001424 | 0.733333 | 0.002842 |
| Naive Bayes | dbscan\_pca\_e0.7\_m15\_noise | 0.86748 | 0.8406 | 0.001424 | 0.733333 | 0.002842 |
| Naive Bayes | dbscan\_pca\_e0.7\_m3\_labels | 0.867478 | 0.840912 | 0.001426 | 0.733333 | 0.002847 |
| Naive Bayes | dbscan\_pca\_e0.7\_m3\_noise | 0.86748 | 0.8406 | 0.001424 | 0.733333 | 0.002842 |
| Naive Bayes | dbscan\_pca\_e0.7\_m5\_labels | 0.86748 | 0.84061 | 0.001424 | 0.733333 | 0.002842 |
| Naive Bayes | dbscan\_pca\_e0.7\_m5\_noise | 0.86748 | 0.8406 | 0.001424 | 0.733333 | 0.002842 |
| Naive Bayes | dbscan\_pca\_e0.7\_m7\_labels | 0.86748 | 0.8406 | 0.001424 | 0.733333 | 0.002842 |
| Naive Bayes | dbscan\_pca\_e0.7\_m7\_noise | 0.86748 | 0.8406 | 0.001424 | 0.733333 | 0.002842 |
| Naive Bayes | dbscan\_pca\_e1.0\_m10\_labels | 0.86748 | 0.8406 | 0.001424 | 0.733333 | 0.002842 |
| Naive Bayes | dbscan\_pca\_e1.0\_m10\_noise | 0.86748 | 0.840599 | 0.001424 | 0.733333 | 0.002842 |
| Naive Bayes | dbscan\_pca\_e1.0\_m15\_labels | 0.86748 | 0.840599 | 0.001424 | 0.733333 | 0.002842 |
| Naive Bayes | dbscan\_pca\_e1.0\_m15\_noise | 0.86748 | 0.8406 | 0.001424 | 0.733333 | 0.002842 |
| Naive Bayes | dbscan\_pca\_e1.0\_m3\_labels | 0.867479 | 0.840912 | 0.001426 | 0.733333 | 0.002847 |
| Naive Bayes | dbscan\_pca\_e1.0\_m3\_noise | 0.86748 | 0.8406 | 0.001424 | 0.733333 | 0.002842 |
| Naive Bayes | dbscan\_pca\_e1.0\_m5\_labels | 0.86748 | 0.84061 | 0.001424 | 0.733333 | 0.002842 |
| Naive Bayes | dbscan\_pca\_e1.0\_m5\_noise | 0.86748 | 0.8406 | 0.001424 | 0.733333 | 0.002842 |
| Naive Bayes | dbscan\_pca\_e1.0\_m7\_labels | 0.86748 | 0.8406 | 0.001424 | 0.733333 | 0.002842 |
| Naive Bayes | dbscan\_pca\_e1.0\_m7\_noise | 0.86748 | 0.8406 | 0.001424 | 0.733333 | 0.002842 |
| Naive Bayes | dbscan\_pca\_e1.3\_m10\_labels | 0.86748 | 0.8406 | 0.001424 | 0.733333 | 0.002842 |
| Naive Bayes | dbscan\_pca\_e1.3\_m10\_noise | 0.86748 | 0.8406 | 0.001424 | 0.733333 | 0.002842 |
| Naive Bayes | dbscan\_pca\_e1.3\_m15\_labels | 0.86748 | 0.8406 | 0.001424 | 0.733333 | 0.002842 |
| Naive Bayes | dbscan\_pca\_e1.3\_m15\_noise | 0.86748 | 0.8406 | 0.001424 | 0.733333 | 0.002842 |
| Naive Bayes | dbscan\_pca\_e1.3\_m3\_labels | 0.867473 | 0.841147 | 0.001429 | 0.733333 | 0.002851 |
| Naive Bayes | dbscan\_pca\_e1.3\_m3\_noise | 0.86748 | 0.8406 | 0.001424 | 0.733333 | 0.002842 |
| Naive Bayes | dbscan\_pca\_e1.3\_m5\_labels | 0.86748 | 0.840625 | 0.001424 | 0.733333 | 0.002842 |
| Naive Bayes | dbscan\_pca\_e1.3\_m5\_noise | 0.867481 | 0.8406 | 0.001424 | 0.733333 | 0.002842 |
| Naive Bayes | dbscan\_pca\_e1.3\_m7\_labels | 0.86748 | 0.8406 | 0.001424 | 0.733333 | 0.002842 |
| Naive Bayes | dbscan\_pca\_e1.3\_m7\_noise | 0.867481 | 0.8406 | 0.001424 | 0.733333 | 0.002842 |
| Naive Bayes | hier\_full\_n2\_labels | 0.86748 | 0.8406 | 0.001424 | 0.733333 | 0.002842 |
| Naive Bayes | hier\_full\_n3\_labels | 0.86748 | 0.8406 | 0.001424 | 0.733333 | 0.002842 |
| Naive Bayes | hier\_full\_n4\_labels | 0.86748 | 0.8406 | 0.001424 | 0.733333 | 0.002842 |
| Naive Bayes | hier\_full\_n5\_labels | 0.86748 | 0.8406 | 0.001424 | 0.733333 | 0.002842 |
| Naive Bayes | hier\_full\_n6\_labels | 0.86748 | 0.8406 | 0.001424 | 0.733333 | 0.002842 |
| Naive Bayes | hier\_pca\_n2\_labels | 0.86748 | 0.8406 | 0.001424 | 0.733333 | 0.002842 |
| Naive Bayes | hier\_pca\_n3\_labels | 0.86748 | 0.8406 | 0.001424 | 0.733333 | 0.002842 |
| Naive Bayes | hier\_pca\_n4\_labels | 0.86748 | 0.8406 | 0.001424 | 0.733333 | 0.002842 |
| Naive Bayes | hier\_pca\_n5\_labels | 0.86748 | 0.8406 | 0.001424 | 0.733333 | 0.002842 |
| Naive Bayes | hier\_pca\_n6\_labels | 0.86748 | 0.8406 | 0.001424 | 0.733333 | 0.002842 |
| Naive Bayes | kmeans\_full\_n10\_labels | 0.867313 | 0.840581 | 0.001423 | 0.733333 | 0.002841 |
| Naive Bayes | kmeans\_full\_n2\_labels | 0.86748 | 0.8406 | 0.001424 | 0.733333 | 0.002842 |
| Naive Bayes | kmeans\_full\_n3\_labels | 0.867481 | 0.8406 | 0.001424 | 0.733333 | 0.002842 |
| Naive Bayes | kmeans\_full\_n4\_labels | 0.86748 | 0.8406 | 0.001424 | 0.733333 | 0.002842 |
| Naive Bayes | kmeans\_full\_n5\_labels | 0.867479 | 0.840601 | 0.001424 | 0.733333 | 0.002842 |
| Naive Bayes | kmeans\_full\_n6\_labels | 0.867527 | 0.840581 | 0.001424 | 0.733333 | 0.002841 |
| Naive Bayes | kmeans\_full\_n7\_labels | 0.867371 | 0.840596 | 0.001424 | 0.733333 | 0.002842 |
| Naive Bayes | kmeans\_full\_n8\_labels | 0.868235 | 0.84059 | 0.001424 | 0.733333 | 0.002842 |
| Naive Bayes | kmeans\_full\_n9\_labels | 0.867948 | 0.840592 | 0.001424 | 0.733333 | 0.002842 |
| Naive Bayes | kmeans\_pca\_n10\_labels | 0.867476 | 0.84059 | 0.001424 | 0.733333 | 0.002842 |
| Naive Bayes | kmeans\_pca\_n2\_labels | 0.86748 | 0.8406 | 0.001424 | 0.733333 | 0.002842 |
| Naive Bayes | kmeans\_pca\_n3\_labels | 0.86748 | 0.8406 | 0.001424 | 0.733333 | 0.002842 |
| Naive Bayes | kmeans\_pca\_n4\_labels | 0.86748 | 0.840603 | 0.001424 | 0.733333 | 0.002842 |
| Naive Bayes | kmeans\_pca\_n5\_labels | 0.867474 | 0.840603 | 0.001424 | 0.733333 | 0.002842 |
| Naive Bayes | kmeans\_pca\_n6\_labels | 0.867482 | 0.840601 | 0.001424 | 0.733333 | 0.002842 |
| Naive Bayes | kmeans\_pca\_n7\_labels | 0.867383 | 0.840616 | 0.001424 | 0.733333 | 0.002842 |
| Naive Bayes | kmeans\_pca\_n8\_labels | 0.867425 | 0.840619 | 0.001424 | 0.733333 | 0.002842 |
| Naive Bayes | kmeans\_pca\_n9\_labels | 0.867472 | 0.840582 | 0.001424 | 0.733333 | 0.002842 |
| LogReg | Baseline | 0.870742 | 0.871239 | 0.001687 | 0.702222 | 0.003366 |
| LogReg | dbscan\_full\_e0.3\_m10\_labels | 0.870741 | 0.871237 | 0.001687 | 0.702222 | 0.003366 |
| LogReg | dbscan\_full\_e0.3\_m10\_noise | 0.870741 | 0.871237 | 0.001687 | 0.702222 | 0.003366 |
| LogReg | dbscan\_full\_e0.3\_m15\_labels | 0.870741 | 0.871237 | 0.001687 | 0.702222 | 0.003366 |
| LogReg | dbscan\_full\_e0.3\_m15\_noise | 0.870741 | 0.871237 | 0.001687 | 0.702222 | 0.003366 |
| LogReg | dbscan\_full\_e0.3\_m3\_labels | 0.870704 | 0.87113 | 0.001685 | 0.702222 | 0.003363 |
| LogReg | dbscan\_full\_e0.3\_m3\_noise | 0.870735 | 0.871231 | 0.001687 | 0.702222 | 0.003366 |
| LogReg | dbscan\_full\_e0.3\_m5\_labels | 0.870741 | 0.871231 | 0.001687 | 0.702222 | 0.003366 |
| LogReg | dbscan\_full\_e0.3\_m5\_noise | 0.870741 | 0.871237 | 0.001687 | 0.702222 | 0.003366 |
| LogReg | dbscan\_full\_e0.3\_m7\_labels | 0.870741 | 0.871237 | 0.001687 | 0.702222 | 0.003366 |
| LogReg | dbscan\_full\_e0.3\_m7\_noise | 0.870741 | 0.871237 | 0.001687 | 0.702222 | 0.003366 |
| LogReg | dbscan\_full\_e0.5\_m10\_labels | 0.870741 | 0.871237 | 0.001687 | 0.702222 | 0.003366 |
| LogReg | dbscan\_full\_e0.5\_m10\_noise | 0.870741 | 0.871237 | 0.001687 | 0.702222 | 0.003366 |
| LogReg | dbscan\_full\_e0.5\_m15\_labels | 0.870741 | 0.871237 | 0.001687 | 0.702222 | 0.003366 |
| LogReg | dbscan\_full\_e0.5\_m15\_noise | 0.870741 | 0.871237 | 0.001687 | 0.702222 | 0.003366 |
| LogReg | dbscan\_full\_e0.5\_m3\_labels | 0.870466 | 0.870012 | 0.001676 | 0.704444 | 0.003344 |
| LogReg | dbscan\_full\_e0.5\_m3\_noise | 0.870709 | 0.87122 | 0.001687 | 0.702222 | 0.003365 |
| LogReg | dbscan\_full\_e0.5\_m5\_labels | 0.870719 | 0.871159 | 0.001686 | 0.702222 | 0.003364 |
| LogReg | dbscan\_full\_e0.5\_m5\_noise | 0.870735 | 0.871235 | 0.001687 | 0.702222 | 0.003366 |
| LogReg | dbscan\_full\_e0.5\_m7\_labels | 0.870738 | 0.871228 | 0.001687 | 0.702222 | 0.003365 |
| LogReg | dbscan\_full\_e0.5\_m7\_noise | 0.870739 | 0.871237 | 0.001687 | 0.702222 | 0.003366 |
| LogReg | dbscan\_full\_e0.7\_m10\_labels | 0.870739 | 0.871227 | 0.001687 | 0.702222 | 0.003365 |
| LogReg | dbscan\_full\_e0.7\_m10\_noise | 0.87074 | 0.871231 | 0.001687 | 0.702222 | 0.003366 |
| LogReg | dbscan\_full\_e0.7\_m15\_labels | 0.870741 | 0.871237 | 0.001687 | 0.702222 | 0.003366 |
| LogReg | dbscan\_full\_e0.7\_m15\_noise | 0.870741 | 0.871237 | 0.001687 | 0.702222 | 0.003366 |
| LogReg | dbscan\_full\_e0.7\_m3\_labels | 0.870813 | 0.871304 | 0.001688 | 0.702222 | 0.003367 |
| LogReg | dbscan\_full\_e0.7\_m3\_noise | 0.870667 | 0.871193 | 0.001686 | 0.702222 | 0.003365 |
| LogReg | dbscan\_full\_e0.7\_m5\_labels | 0.870641 | 0.870971 | 0.001683 | 0.702222 | 0.003359 |
| LogReg | dbscan\_full\_e0.7\_m5\_noise | 0.870724 | 0.871225 | 0.001687 | 0.702222 | 0.003365 |
| LogReg | dbscan\_full\_e0.7\_m7\_labels | 0.870715 | 0.871144 | 0.001686 | 0.702222 | 0.003363 |
| LogReg | dbscan\_full\_e0.7\_m7\_noise | 0.870733 | 0.87123 | 0.001687 | 0.702222 | 0.003365 |
| LogReg | dbscan\_full\_e1.0\_m10\_labels | 0.870728 | 0.871191 | 0.001686 | 0.702222 | 0.003364 |
| LogReg | dbscan\_full\_e1.0\_m10\_noise | 0.870735 | 0.871229 | 0.001687 | 0.702222 | 0.003365 |
| LogReg | dbscan\_full\_e1.0\_m15\_labels | 0.870741 | 0.871237 | 0.001687 | 0.702222 | 0.003366 |
| LogReg | dbscan\_full\_e1.0\_m15\_noise | 0.870741 | 0.871237 | 0.001687 | 0.702222 | 0.003366 |
| LogReg | dbscan\_full\_e1.0\_m3\_labels | 0.87098 | 0.871379 | 0.001689 | 0.702222 | 0.003369 |
| LogReg | dbscan\_full\_e1.0\_m3\_noise | 0.870512 | 0.871134 | 0.001686 | 0.702222 | 0.003363 |
| LogReg | dbscan\_full\_e1.0\_m5\_labels | 0.870443 | 0.870028 | 0.001676 | 0.704444 | 0.003344 |
| LogReg | dbscan\_full\_e1.0\_m5\_noise | 0.870697 | 0.871213 | 0.001687 | 0.702222 | 0.003365 |
| LogReg | dbscan\_full\_e1.0\_m7\_labels | 0.870615 | 0.870921 | 0.001683 | 0.702222 | 0.003357 |
| LogReg | dbscan\_full\_e1.0\_m7\_noise | 0.870719 | 0.871221 | 0.001687 | 0.702222 | 0.003365 |
| LogReg | dbscan\_full\_e1.3\_m10\_labels | 0.870612 | 0.870864 | 0.001682 | 0.702222 | 0.003356 |
| LogReg | dbscan\_full\_e1.3\_m10\_noise | 0.870716 | 0.871226 | 0.001687 | 0.702222 | 0.003365 |
| LogReg | dbscan\_full\_e1.3\_m15\_labels | 0.870732 | 0.871215 | 0.001687 | 0.702222 | 0.003365 |
| LogReg | dbscan\_full\_e1.3\_m15\_noise | 0.870735 | 0.871236 | 0.001687 | 0.702222 | 0.003366 |
| LogReg | dbscan\_full\_e1.3\_m3\_labels | 0.871679 | 0.871313 | 0.001693 | 0.704444 | 0.003378 |
| LogReg | dbscan\_full\_e1.3\_m3\_noise | 0.869542 | 0.871023 | 0.001684 | 0.702222 | 0.00336 |
| LogReg | dbscan\_full\_e1.3\_m5\_labels | 0.871023 | 0.871317 | 0.001688 | 0.702222 | 0.003368 |
| LogReg | dbscan\_full\_e1.3\_m5\_noise | 0.87048 | 0.871157 | 0.001686 | 0.702222 | 0.003364 |
| LogReg | dbscan\_full\_e1.3\_m7\_labels | 0.870306 | 0.868602 | 0.001659 | 0.704444 | 0.003311 |
| LogReg | dbscan\_full\_e1.3\_m7\_noise | 0.870654 | 0.871209 | 0.001687 | 0.702222 | 0.003365 |
| LogReg | dbscan\_pca\_e0.3\_m10\_labels | 0.870741 | 0.871237 | 0.001687 | 0.702222 | 0.003366 |
| LogReg | dbscan\_pca\_e0.3\_m10\_noise | 0.870741 | 0.871237 | 0.001687 | 0.702222 | 0.003366 |
| LogReg | dbscan\_pca\_e0.3\_m15\_labels | 0.870741 | 0.871237 | 0.001687 | 0.702222 | 0.003366 |
| LogReg | dbscan\_pca\_e0.3\_m15\_noise | 0.870741 | 0.871237 | 0.001687 | 0.702222 | 0.003366 |
| LogReg | dbscan\_pca\_e0.3\_m3\_labels | 0.870698 | 0.870163 | 0.001673 | 0.702222 | 0.003338 |
| LogReg | dbscan\_pca\_e0.3\_m3\_noise | 0.870777 | 0.871315 | 0.001688 | 0.702222 | 0.003368 |
| LogReg | dbscan\_pca\_e0.3\_m5\_labels | 0.870832 | 0.871353 | 0.001688 | 0.702222 | 0.003369 |
| LogReg | dbscan\_pca\_e0.3\_m5\_noise | 0.870787 | 0.871325 | 0.001688 | 0.702222 | 0.003368 |
| LogReg | dbscan\_pca\_e0.3\_m7\_labels | 0.870806 | 0.871334 | 0.001688 | 0.702222 | 0.003368 |
| LogReg | dbscan\_pca\_e0.3\_m7\_noise | 0.87079 | 0.871328 | 0.001688 | 0.702222 | 0.003368 |
| LogReg | dbscan\_pca\_e0.5\_m10\_labels | 0.870741 | 0.871237 | 0.001687 | 0.702222 | 0.003366 |
| LogReg | dbscan\_pca\_e0.5\_m10\_noise | 0.870741 | 0.871237 | 0.001687 | 0.702222 | 0.003366 |
| LogReg | dbscan\_pca\_e0.5\_m15\_labels | 0.870741 | 0.871237 | 0.001687 | 0.702222 | 0.003366 |
| LogReg | dbscan\_pca\_e0.5\_m15\_noise | 0.870741 | 0.871237 | 0.001687 | 0.702222 | 0.003366 |
| LogReg | dbscan\_pca\_e0.5\_m3\_labels | 0.870698 | 0.870163 | 0.001673 | 0.702222 | 0.003338 |
| LogReg | dbscan\_pca\_e0.5\_m3\_noise | 0.870777 | 0.871315 | 0.001688 | 0.702222 | 0.003368 |
| LogReg | dbscan\_pca\_e0.5\_m5\_labels | 0.870832 | 0.871353 | 0.001688 | 0.702222 | 0.003369 |
| LogReg | dbscan\_pca\_e0.5\_m5\_noise | 0.870787 | 0.871325 | 0.001688 | 0.702222 | 0.003368 |
| LogReg | dbscan\_pca\_e0.5\_m7\_labels | 0.870806 | 0.871334 | 0.001688 | 0.702222 | 0.003368 |
| LogReg | dbscan\_pca\_e0.5\_m7\_noise | 0.87079 | 0.871328 | 0.001688 | 0.702222 | 0.003368 |
| LogReg | dbscan\_pca\_e0.7\_m10\_labels | 0.870741 | 0.871237 | 0.001687 | 0.702222 | 0.003366 |
| LogReg | dbscan\_pca\_e0.7\_m10\_noise | 0.870741 | 0.871237 | 0.001687 | 0.702222 | 0.003366 |
| LogReg | dbscan\_pca\_e0.7\_m15\_labels | 0.870741 | 0.871237 | 0.001687 | 0.702222 | 0.003366 |
| LogReg | dbscan\_pca\_e0.7\_m15\_noise | 0.870741 | 0.871237 | 0.001687 | 0.702222 | 0.003366 |
| LogReg | dbscan\_pca\_e0.7\_m3\_labels | 0.870599 | 0.868874 | 0.001667 | 0.706667 | 0.003326 |
| LogReg | dbscan\_pca\_e0.7\_m3\_noise | 0.870763 | 0.87131 | 0.001688 | 0.702222 | 0.003368 |
| LogReg | dbscan\_pca\_e0.7\_m5\_labels | 0.870815 | 0.871262 | 0.001687 | 0.702222 | 0.003366 |
| LogReg | dbscan\_pca\_e0.7\_m5\_noise | 0.870782 | 0.871323 | 0.001688 | 0.702222 | 0.003368 |
| LogReg | dbscan\_pca\_e0.7\_m7\_labels | 0.870818 | 0.871351 | 0.001688 | 0.702222 | 0.003369 |
| LogReg | dbscan\_pca\_e0.7\_m7\_noise | 0.87079 | 0.871327 | 0.001688 | 0.702222 | 0.003368 |
| LogReg | dbscan\_pca\_e1.0\_m10\_labels | 0.870741 | 0.871237 | 0.001687 | 0.702222 | 0.003366 |
| LogReg | dbscan\_pca\_e1.0\_m10\_noise | 0.870741 | 0.871237 | 0.001687 | 0.702222 | 0.003366 |
| LogReg | dbscan\_pca\_e1.0\_m15\_labels | 0.870741 | 0.871237 | 0.001687 | 0.702222 | 0.003366 |
| LogReg | dbscan\_pca\_e1.0\_m15\_noise | 0.870741 | 0.871237 | 0.001687 | 0.702222 | 0.003366 |
| LogReg | dbscan\_pca\_e1.0\_m3\_labels | 0.870599 | 0.868874 | 0.001667 | 0.706667 | 0.003326 |
| LogReg | dbscan\_pca\_e1.0\_m3\_noise | 0.870763 | 0.87131 | 0.001688 | 0.702222 | 0.003368 |
| LogReg | dbscan\_pca\_e1.0\_m5\_labels | 0.870815 | 0.871262 | 0.001687 | 0.702222 | 0.003366 |
| LogReg | dbscan\_pca\_e1.0\_m5\_noise | 0.870782 | 0.871323 | 0.001688 | 0.702222 | 0.003368 |
| LogReg | dbscan\_pca\_e1.0\_m7\_labels | 0.870818 | 0.871351 | 0.001688 | 0.702222 | 0.003369 |
| LogReg | dbscan\_pca\_e1.0\_m7\_noise | 0.87079 | 0.871327 | 0.001688 | 0.702222 | 0.003368 |
| LogReg | dbscan\_pca\_e1.3\_m10\_labels | 0.870741 | 0.871237 | 0.001687 | 0.702222 | 0.003366 |
| LogReg | dbscan\_pca\_e1.3\_m10\_noise | 0.870741 | 0.871237 | 0.001687 | 0.702222 | 0.003366 |
| LogReg | dbscan\_pca\_e1.3\_m15\_labels | 0.870741 | 0.871237 | 0.001687 | 0.702222 | 0.003366 |
| LogReg | dbscan\_pca\_e1.3\_m15\_noise | 0.870741 | 0.871237 | 0.001687 | 0.702222 | 0.003366 |
| LogReg | dbscan\_pca\_e1.3\_m3\_labels | 0.870687 | 0.871237 | 0.001687 | 0.702222 | 0.003366 |
| LogReg | dbscan\_pca\_e1.3\_m3\_noise | 0.870751 | 0.871306 | 0.001688 | 0.702222 | 0.003367 |
| LogReg | dbscan\_pca\_e1.3\_m5\_labels | 0.870779 | 0.87115 | 0.001686 | 0.702222 | 0.003363 |
| LogReg | dbscan\_pca\_e1.3\_m5\_noise | 0.870777 | 0.871321 | 0.001688 | 0.702222 | 0.003368 |
| LogReg | dbscan\_pca\_e1.3\_m7\_labels | 0.870825 | 0.871364 | 0.001689 | 0.702222 | 0.003369 |
| LogReg | dbscan\_pca\_e1.3\_m7\_noise | 0.870789 | 0.871326 | 0.001688 | 0.702222 | 0.003368 |
| LogReg | hier\_full\_n2\_labels | 0.870742 | 0.871239 | 0.001687 | 0.702222 | 0.003366 |
| LogReg | hier\_full\_n3\_labels | 0.870742 | 0.871239 | 0.001687 | 0.702222 | 0.003366 |
| LogReg | hier\_full\_n4\_labels | 0.870742 | 0.871239 | 0.001687 | 0.702222 | 0.003366 |
| LogReg | hier\_full\_n5\_labels | 0.870742 | 0.871239 | 0.001687 | 0.702222 | 0.003366 |
| LogReg | hier\_full\_n6\_labels | 0.870742 | 0.871239 | 0.001687 | 0.702222 | 0.003366 |
| LogReg | hier\_pca\_n2\_labels | 0.870742 | 0.871239 | 0.001687 | 0.702222 | 0.003366 |
| LogReg | hier\_pca\_n3\_labels | 0.870742 | 0.871239 | 0.001687 | 0.702222 | 0.003366 |
| LogReg | hier\_pca\_n4\_labels | 0.870742 | 0.871239 | 0.001687 | 0.702222 | 0.003366 |
| LogReg | hier\_pca\_n5\_labels | 0.870742 | 0.871239 | 0.001687 | 0.702222 | 0.003366 |
| LogReg | hier\_pca\_n6\_labels | 0.870742 | 0.871239 | 0.001687 | 0.702222 | 0.003366 |
| LogReg | kmeans\_full\_n10\_labels | 0.8681 | 0.863485 | 0.001625 | 0.713333 | 0.003242 |
| LogReg | kmeans\_full\_n2\_labels | 0.870742 | 0.871239 | 0.001687 | 0.702222 | 0.003366 |
| LogReg | kmeans\_full\_n3\_labels | 0.870742 | 0.871239 | 0.001687 | 0.702222 | 0.003366 |
| LogReg | kmeans\_full\_n4\_labels | 0.870742 | 0.871239 | 0.001687 | 0.702222 | 0.003366 |
| LogReg | kmeans\_full\_n5\_labels | 0.858907 | 0.870946 | 0.001684 | 0.702222 | 0.003359 |
| LogReg | kmeans\_full\_n6\_labels | 0.853944 | 0.856957 | 0.001549 | 0.713333 | 0.003092 |
| LogReg | kmeans\_full\_n7\_labels | 0.8631 | 0.858842 | 0.001589 | 0.708889 | 0.003171 |
| LogReg | kmeans\_full\_n8\_labels | 0.87304 | 0.873435 | 0.001718 | 0.702222 | 0.003428 |
| LogReg | kmeans\_full\_n9\_labels | 0.867948 | 0.86118 | 0.00163 | 0.731111 | 0.003253 |
| LogReg | kmeans\_pca\_n10\_labels | 0.868308 | 0.870855 | 0.00166 | 0.693333 | 0.003313 |
| LogReg | kmeans\_pca\_n2\_labels | 0.869809 | 0.869923 | 0.00167 | 0.702222 | 0.003332 |
| LogReg | kmeans\_pca\_n3\_labels | 0.866467 | 0.869161 | 0.001646 | 0.695556 | 0.003284 |
| LogReg | kmeans\_pca\_n4\_labels | 0.86423 | 0.864026 | 0.001592 | 0.7 | 0.003177 |
| LogReg | kmeans\_pca\_n5\_labels | 0.85929 | 0.865276 | 0.001614 | 0.7 | 0.003221 |
| LogReg | kmeans\_pca\_n6\_labels | 0.866973 | 0.866727 | 0.001636 | 0.702222 | 0.003264 |
| LogReg | kmeans\_pca\_n7\_labels | 0.869647 | 0.872196 | 0.001679 | 0.693333 | 0.00335 |
| LogReg | kmeans\_pca\_n8\_labels | 0.861874 | 0.86704 | 0.00166 | 0.713333 | 0.003312 |
| LogReg | kmeans\_pca\_n9\_labels | 0.863537 | 0.865014 | 0.001625 | 0.704444 | 0.003242 |