Medicare Fraud Detection

Chandra Teja Peddi
Big Data Analytics
San Diego State University
San Diego, California 92120
Email: cpeddi0418@sdsu.edu

Jaya vamshidhar Reddy
Big Data Analytics
San Diego State University
San Diego, California 92120
Email: jteramreddygar1194@sdsu.edu

Anish Chintamaneni
Big Data Analytics
San Diego State University
San Diego, California 92120
Email: achintamaneni5717@sdsu.edu

1. Introduction

The Medicare Data Fraud Detection project is a key undertaking that uses machine learning-based methods to identify and stop fraudulent medical claims in the Medicare system. Millions of Americans, including those over 65 and with disabilities, are covered by the Medicare program for health insurance. Unfortunately, there are a number of fraudulent practices that can be committed against the Medicare system, including filing false claims, overcharging, and paying for needless medical treatments. Such fraudulent acts have the potential to cause considerable losses in public funds and have an impact on the caliber of the recipients' medical care.

Both rule-based systems and machine learning-based techniques are now used in research on Medicare fraud detection. While machine learning-based approaches employ historical data to find trends and anomalies, rule-based systems rely on predetermined rules to identify suspicious activity. To find fraudulent actions in the Medicare system, the Medicare Data Fraud Detection project makes use of machine learning-based methods.

The goal of this research is to create a classification model based on machine learning that can accurately identify false medical claims in the Medicare system. LinearSVC, Naive Bayes, decision trees, and random forests are just a few of the machine learning algorithms used in the research, which makes use of three data sets, including Medicare claims data.

The following sections make up the project's structure:

Introduction: This section gives a quick rundown of the project's goals. Data Preprocessing and Exploration: This section preprocesses and explores the three data sets (Medicare Claims Data, Open Payments Data, and OIG Exclusions list). To get the data ready for machine learning algorithms, feature engineering and data cleansing are done.

Model Development: To create a fraud detection model, this section employs a variety of machine learning algorithms, such as LinearSVC, naive Bayes, decision trees, and random forests. The effectiveness of each algorithm is evaluated in order to choose the most precise model.

Model Evaluation: Several metrics, such as accuracy, precision, recall, and F1-score, are used to assess the performance of the generated model. The possible impact of false positives and false negatives is also explored using a confusion matrix.

Conclusion: This part summarizes the work and offers suggestions for new lines of inquiry. Discussion of the study's limitations is followed by suggestions for additional investigation.

The Medicare Data Fraud Detection initiative is crucial because it aids in identifying and stopping fraudulent actions inside the Medicare system, ensuring that monies are given to valid claims, and raising the standard of care for Medicare beneficiaries.

2. Data set Description

The Medicare Data Fraud Detection project makes use of three separate data sets in order to construct a complete model for identifying fraudulent medical claims in the Medicare system. These data sets offer details on the prescription drug claims filed by healthcare providers, the payments made to healthcare providers by pharmaceutical and medical device firms, and the people and organizations who were disqualified from taking part in federal healthcare programs due to waste, fraud, or abuse.

2.1. Medicare Part D Prescribers data set

In accordance with Medicare Part D, which provides prescription drug coverage for Medicare enrollees, this data set contains details on prescription medication claims made by healthcare providers. The data collection includes details about the drug's brand, dosage, and price as well as specifics about the prescriber, like their National Provider Identifier (NPI) and area of expertise. The Centers for Medicare Medicaid Services (CMS) releases this data set annually, and it offers thorough details on the prescription drug claims made by clinicians under Medicare Part D. The 2020 data collection consists of nearly 25 million records and 21

variables, such as the therapeutic class, cost-sharing tier, and drug kind of the drug.

The data set is available for download from the CMS website in CSV format. To enable analysis using machine learning models, preprocessing for this data set included resolving missing values and transforming categorical features into numerical ones.

The data set can be used to spot patterns and trends in the usage of prescription drugs, as well as possible instances of fraud and misuse. By examining the data, it is possible to spot healthcare professionals who may be writing prescriptions for an unusually large number of medications or medications that are outside the scope of their expertise, which may be an indication of dishonest or abusive behavior. The data set contains information about prescription drug claims made by healthcare providers under Medicare Part D. It has 21 features and some of the important features are: 1) PrscrbrNPI: National Provider Identifier (NPI) of the prescriber 2) Prscrbr Last Org Name: Last name of the prescriber 3) Prscrbr First Name: First name of the prescriber 4) Prscrbr City: City of the prescriber's practice location 5) Prscrbr State Abrvtn: State abbreviation of the prescriber's practice location 6) Prscrbr Type: Type of prescriber (e.g., Physician, Nurse Practitioner, Dentist, etc.) 7) Brnd Name: Brand name of the drug 8) Gnrc Name: Generic name of the drug 9) Tot Clms: Total number of claims for the drug 10) Tot Day Supply: Total number of days of supply for the drug 11) Tot Drug Cst: Total drug cost for the drug.

```
['Prscrbr_NPI',
 'Prscrbr_Last_Org_Name',
 'Prscrbr_First_Name',
 'Prscrbr_City',
 'Prscrbr_State_Abrvtn',
 'Prscrbr_State_FIPS',
 'Prscrbr_Type',
 'Prscrbr_Type_Src',
 'Brnd_Name',
 'Gnrc_Name',
 'Tot_Clms',
 'Tot_30day_Fills',
 'Tot_Day_Suply',
 'Tot_Drug_Cst',
 'Tot_Benes',
 'GE65_Sprsn_Flag',
 'GE65_Tot_Clms',
 'GE65_Tot_30day_Fills',
 'GE65_Tot_Drug_Cst',
 'GE65_Tot_Day_Suply',
 'GE65_Bene_Sprsn_Flag',
 'GE65_Tot_Benes']
```

Figure 1. Schema of Medicare part D dataset

2.2. Open Payments data set

The Open Payments data set, available for download on the CMS website, contains information about payments and transfers of value from drug and medical device companies to physicians and teaching hospitals in the United States. The data set is part of the Open Payments program, which was established by the Affordable Care Act (ACA) to increase transparency and accountability in healthcare by publicly disclosing the financial relationships between healthcare providers and industry.

The data set contains records from 2021 to the present, and it is updated annually. The data is organized by calendar year, and each record provides information about the payment, including the payment date, payment type, payment amount, the recipient's name and address, the physician's specialty, and the name of the company making the payment. Additionally, the data set includes information about the nature of the payment, such as consulting fees, travel expenses, and research grants.

The data set includes a total of 77 features, and it contains over 11 million records for the year 2021. The size of the data set and the complexity of the data make preprocessing and analysis challenging. Preprocessing for this data set involves handling missing values, grouping similar payment types together, and mapping company names to standardized names to reduce redundancy.

Researchers and healthcare professionals can use the data set to study the financial relationships between healthcare providers and industry, and to identify potential conflicts of interest or inappropriate relationships. The data set can also be used to identify patterns and trends in the payments made by industry to healthcare providers, such as changes in payment amounts or types of payments over time.

2.3. LEIE Exclusions List data set

The data set is the Exclusions List maintained by the Office of Inspector General (OIG) of the U.S. Department of Health and Human Services (HHS). The Exclusions List contains the names of individuals and entities who are excluded from participating in Federal health care programs, such as Medicare and Medicaid, as well as other Federal programs.

The Exclusions List is intended to prevent fraud and abuse in Federal health care programs by excluding individuals and entities who have been convicted of certain crimes or engaged in other improper activities from participating in these programs. Exclusions can be temporary or permanent and can be imposed by the OIG or other Federal agencies.

The data set has 19 columns, each representing a different attribute of individuals or businesses that have been excluded from participation in federal healthcare programs by the Department of Health and Human Services' Office of Inspector General (OIG). Some significant features are:

1) LASTNAME: The last name of the excluded individual or the business name if it is a corporate entity. 2) FIRSTNAME: The first name of the excluded individual. 3) BUSNAME: The name of the excluded business

if applicable. 4) GENERAL: A general description of the type of exclusion (e.g., physician, nurse, pharmacy, etc.). 5) SPECIALTY: The specific type of exclusion (e.g., anesthesiology, radiology, ophthalmology, etc.). 6) UPIN: The Unique Physician Identification Number of the excluded individual if applicable. 7) NPI: The National Provider Identifier of the excluded individual or business if applicable. 8) ADDRESS: The street address of the excluded individual or business. 9) CITY: The city where the excluded individual or business is located. 10) STATE: The state where the excluded individual or business is located. 11) EXCLTYPE: The type of exclusion (e.g., mandatory, permissive, etc.). 12) EXCLDATE: The date the exclusion became effective. 13) REINDATE: The date the exclusion was lifted or reinstated, if applicable. 14) WAIVERDATE: The date a waiver was granted, if applicable. 15) WVRSTATE: The state where the waiver was granted, if applicable.

The data set includes information on over 80,000 individuals and entities. Preprocessing for this data set involved handling missing values and merging the data set with the Medicare claims data to identify excluded providers.

One of the challenges faced in obtaining the data was the need to obtain the appropriate permissions and credentials to access the CMS and OIG data sets. The data sets were quite large and required significant computing power to preprocess and analyze them. The data sets had to be loaded into a Pyspark data frame, to perform analysis on them. Also merging the three data sets was proved to be a challenge.

The links to the data sets are as follows:
Medicare Part D Prescribers: https://data.cms.gov/provider-summary-by-type-of-service/medicare-part-d-prescribers/medicare-part-d-prescribers-by-provider-and-drug/data
Open Payments: https://www.cms.gov/OpenPayments/Data/Dataset-Downloads OIG Exclusions List: https://oig.hhs.gov/exclusions/exclusions_list.asp

3. Problem Statement

The Medicare Part D program is susceptible to fraud, waste, and abuse, which might cause the government to suffer substantial financial losses and raise the cost of healthcare. In order to effectively identify potential fraud in the Medicare Part D program, the Medicare Data Fraud Detection project intends to create a machine learning model. To find patterns and anomalies that might point to fraudulent conduct, the model will make use of three data sets, including the Office of Inspector General (OIG) exclusions list, Open Payments data, and Medicare Part D prescribers.

Millions of Americans receive prescription medication coverage through the federal program Medicare Part D, so it's important to make sure it's not being abused or exploited for personal advantage. Fraud, waste, and abuse inside the program have the potential to cause the government to suffer substantial financial losses, increase healthcare expenses, and have a detrimental effect on patient care. Therefore, it is essential for the Medicare Part D program to identify and prevent fraud, waste, and abuse in order to guarantee that beneficiaries receive high-quality care and that the program is financially viable.

The research question that the project seeks to answer is whether a machine learning model can accurately identify potential fraud, waste, and abuse in the Medicare Part D program using a combination of three data sets: Medicare Part D prescribers, Open Payments data, and the OIG exclusions list. The project's goal is to assess the performance of various machine learning algorithms and identify the one that is most adept at spotting possible fraud. The project's goal is to help establish an efficient fraud detection system for the Medicare Part D program by providing a solution to this research topic.

4. Exploratory Data Analysis (EDA)

To acquire a better visual knowledge of the aspects of the data sets, we have undertaken some exploratory data analysis as the first step in our project. Exploratory data analysis (EDA), which identifies trends and outliers in the data, is a vital stage in the process of detecting Medicare fraud. EDA is the process of analyzing and comprehending the data using various statistical and visual methods.

We have visualized many graphs such as the Top 10 drugs by total cost, Drugs cost by specialty, the top 15 states and cities by total payments, and a scatter plot between total drug cost and total cost supply.

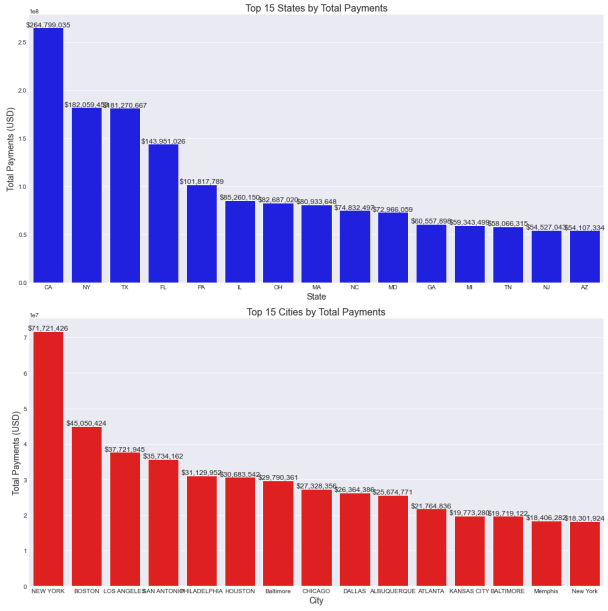


Figure 2. Top 15 States and Cities by Total Payments

The figure 2 shows the top 15 states and cities in the US with the highest total payments made to healthcare providers

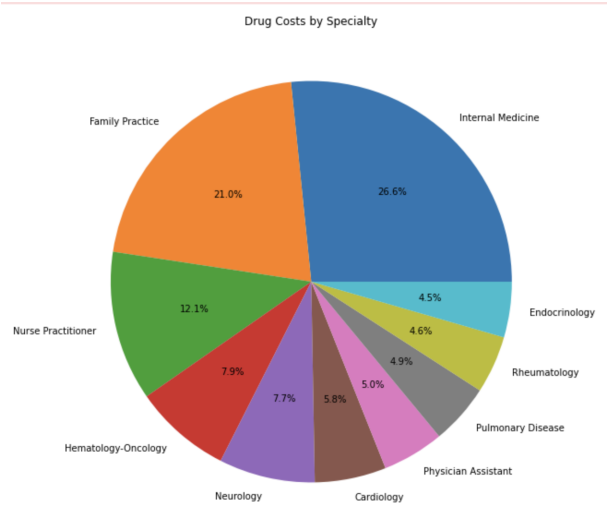


Figure 3. Drug Costs by Speciality

by pharmaceutical companies. From the plot, we can infer that the top states and cities by total payments are mostly concentrated in the East Coast and California. In terms of states, California has the highest total payments followed by New York, Massachusetts, and Texas. In terms of cities, New York City has the highest total payments followed by Los Angeles, Chicago, and Houston.

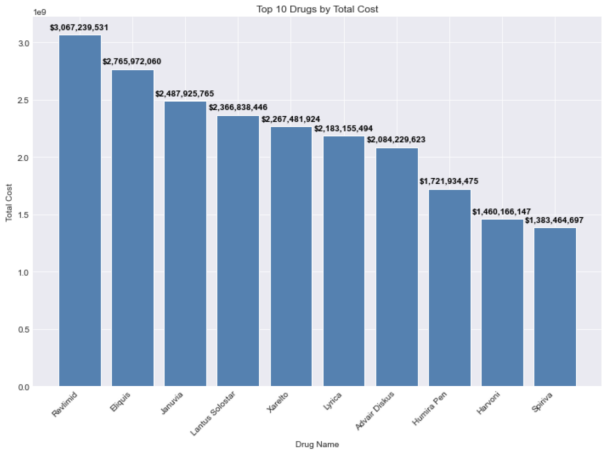


Figure 4. Top 10 Drugs by Total Cost

The Figure 6 shows total claims by drug name can help us identify which drugs are most commonly prescribed by healthcare providers.

Figure 7 is a generated heat map showing the correlation between the columns of the final preprocessed data. It can help in identifying any strong correlations between the features, which can be useful in feature selection for fraud detection. It can also provide insights into any potential multicollinearity issues that may exist between the features.

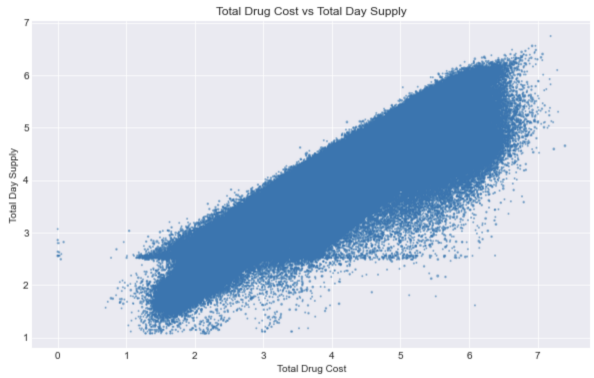


Figure 5. Total drug cost vs Total day supply

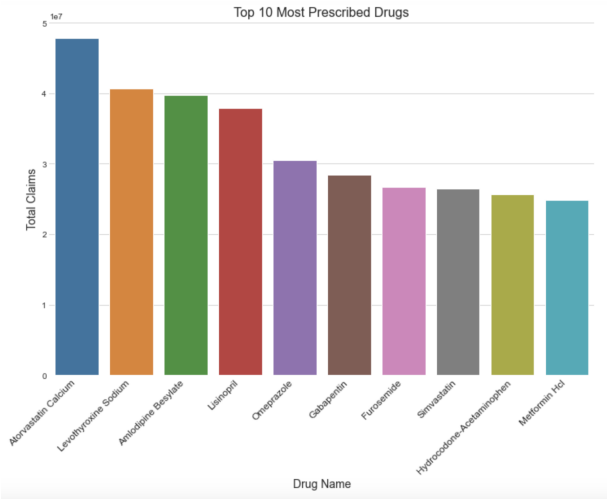


Figure 6. Bar chart of total claims by drug name

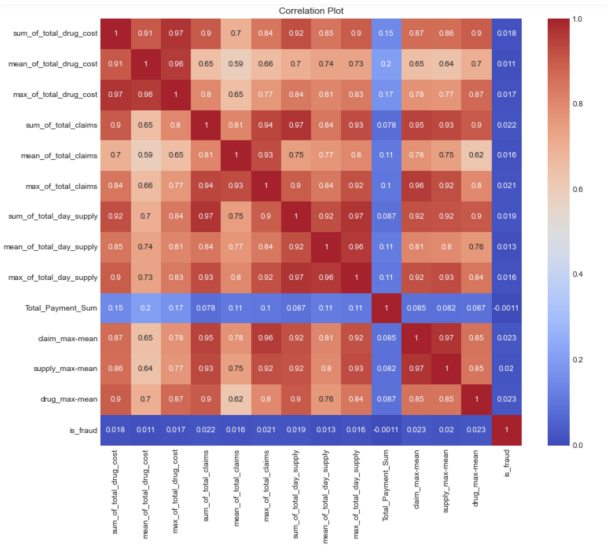


Figure 7. Correlation heatmap between the features

5. Methodology

The Medicare Data Fraud Detection project utilizes a machine learning approach to identify potential fraud in the Medicare Part D program. The project uses three data sets: Medicare Part D prescribers, Open Payments data, and the OIG exclusions list.

5.1. Data Preprocessing

This is the most important part of the methodology, especially when dealing with extremely large data sets and hundreds of features in three different data sets.

All the data sets are extremely large in size and cannot be run on the local machine. We had to load all the data sets as a PySpark data frame. Before applying machine learning algorithms, we performed several preprocessing steps on the data sets to clean and transform the data into a format that can be easily used by machine learning models. Here is how the preprocessing was done in detail:

First, necessary libraries are imported including pandas, numpy, matplotlib, and PySpark. A Spark session is then created using SparkSession to process two different data sets related to Medicare drug prescription and physician payment. The CSV file for Medicare data is read into a Spark Data Frame using the read.csv() function. The data set contains information such as the prescriber's NPI, drug name, total drug cost, total claims, total day supply, and specialty description. Two new Data Frames were created by selecting specific columns from the original Data Frame. The new DataFrames are cleaned up by removing any null values and duplicates. The dropna() function is used to drop any rows with null values in the two new Data Frames, and duplicate rows in were dropped using drop duplicates().

Column names of the new DataFrames are renamed to improve readability. Column names in the new Data Frames are renamed using the withColumnRenamed() function. New features are created from existing features by grouping by NPI (National Provider Identifier) and aggregating drug cost, total claims, and total day supply using the groupBy() and agg() functions.

The two DataFrames are joined on the NPI column using the join() function. New features such as the sum, mean, and max of total drug cost, total claims, and total day supply are calculated. Some columns in the resulting DataFrame are converted to uppercase using upper() and a new column called "index" is added using monotonically increasing id().

The CSV file for Payment data is read into a PySpark DataFrame using read.csv(). The required columns are selected using select(). The DataFrame is grouped by the columns "Covered Recipient First Name", "Covered Recipient Last Name", "Recipient City", and "Recipient State", and the "Total Amount of Payment US Dollars" column is aggregated by taking the sum of its values for each group using the groupBy() and agg() functions.

Finally, the LEIE dataset is loaded from a CSV file using the PySpark read.csv() function. Next, a new DataFrame named is created by selecting the 'NPI' and 'EXCLTYPE' columns from leie data frame. Then, the npi fraud DataFrame is created by filtering out any rows where the 'NPI' value is equal to 0 using the filter() function. A new column named 'is fraud' is added to this DataFrame using the withColumn() function, where each row's 'is fraud' value is set to 1. Additionally, the data type of the 'npi' column is cast to an integer using the withColumn() and cast() functions.

The payment DataFrame is then joined with the npi fraud DataFrame on the 'npi' column using the join() function. This data frame is named Features. Any null values in this DataFrame are filled with 0 using the fillna() function.

Next, the log10() function is applied to various columns in the Features DataFrame to transform their values. Specifically, the columns 'sum of total drug cost', 'sum of total claims', and 'sum of total day supply' are transformed using the log10() function and renamed to include the suffix 'log10'. The 'claim max-mean', 'supply max-mean', and 'drug max-mean' columns are also added to this data frame by taking the difference between the maximum and mean values of the 'max of total claims', 'max of total day supply', and 'max of total drug cost' columns, respectively.

Finally, certain columns, such as 'first name', 'last name', 'state', 'city', and 'specialty description', which are deemed irrelevant for the machine learning model, are dropped using the drop() function. The final resulting data frame is ready for further processing to build a machine-learning model.

This final preprocessed data frame gave inaccurate results during the modeling phase as it was highly imbalanced with the minority class. The data set was then balanced by oversampling the minority class using Synthetic Minority Oversampling Technique (SMOTE) technique.

```
FeaturesAll_df.printSchema()
```

```
root
 |-- npi: integer (nullable = true)
 |-- sum_of_total_drug_cost: double (nullable = true)
 |-- mean_of_total_drug_cost: double (nullable = true)
 |-- max_of_total_drug_cost: double (nullable = true)
 |-- sum_of_total_claims: double (nullable = true)
 |-- mean_of_total_claims: double (nullable = true)
 |-- max_of_total_claims: double (nullable = true)
 |-- sum_of_total_day_supply: double (nullable = true)
 |-- mean_of_total_day_supply: double (nullable = true)
 |-- max_of_total_day_supply: double (nullable = true)
 |-- Total_Payment_Sum: double (nullable = true)
 |-- claim_max-mean: double (nullable = true)
 |-- supply_max-mean: double (nullable = true)
 |-- drug_max-mean: double (nullable = true)
 |-- is_fraud: integer (nullable = true)
```

Figure 8. Schema of Final preprocessed data frame

5.2. SMOTE

SMOTE (Synthetic Minority Over-sampling Technique) is a technique used to address the class imbalance problem in a dataset, where one class has significantly fewer instances than another. This technique generates new synthetic samples from the minority class by interpolating between the existing minority class samples. It can be applied to various machine learning algorithms to improve the performance of the model in predicting the minority class.

The SMOTE algorithm works by identifying the minority class samples that are nearest to each other in the feature space. It then generates new samples by interpolating between these neighboring samples. The number of new samples to generate is determined by a user-defined parameter called the oversampling ratio, which specifies the number of new samples to generate relative to the size of the minority class.

In PySpark, SMOTE was previously implemented using the SMOTE function from the pyspark.ml.feature module. Now, this module is deprecated. Therefore we have implemented SMOTE using the 'imblearn' library, which provides a variety of re-sampling techniques for imbalanced data sets. This function takes a data set as input and over-samples the minority class using the SMOTE algorithm. The output data set will have balanced classes, where the minority class is over-sampled to match the majority class.

After applying SMOTE, the resulting data set can be used to train a machine-learning model. By oversampling the minority class, SMOTE can improve the model's ability to generalize and make accurate predictions about the minority class. SMOTE can be computationally expensive for large data sets and this was a major limitation that we faced.

5.3. Feature Engineering

In order to extract pertinent features from the data sets that would be helpful in identifying fraudulent activity, we also undertook feature engineering. For each prescriber, this involved figuring out the overall number of claims, the total cost of the drugs, and the total number of beneficiaries. From the Open Payments data set and the OIG exclusions list, we also collected data on payments made to prescribers and data on providers who were not included.

5.4. Machine Learning Algorithms

We evaluated the performance of several machine learning classifiers, including Naive Bayes, Decision Tree, Random Forest, and LinearSVC, to determine which algorithm is most effective in detecting potential fraudulent activity. We trained the models on a labeled over-sampled data set that contained information on known fraudulent activity in the Medicare Part D program.

Naive Bayes, Random Forest, Decision Tree, and Linear Support Vector Classifier (SVC) are popular machine learning models for classification tasks, and we chose them for our project for several reasons.

5.4.1. Naive Bayes. Naive Bayes is a probabilistic algorithm that is based on Bayes' theorem. For text categorization, spam filtering, sentiment analysis, and other related tasks, it is a popular and effective algorithm. Naive Bayes is computationally effective and performs well with high-dimensional data sets. Because it believes that each aspect exists independently of the others, it is referred to as "naive." Naive Bayes can still function well in many situations despite this presumption, particularly when there are a lot of characteristics. Naive Bayes was able to effectively manage the vast amount of features in our project.

5.4.2. Random Forest. Random Forest is an ensemble learning method that uses multiple decision trees to create a single predictive model. It is renowned for having a high level of accuracy, being noise-resistant, and being able to accommodate missing data. It operates by building a collection of decision trees, each of which is trained using a random subset of both the data and the features. The majority vote of the forecasts from all the trees is used to determine the final prediction. Random Forest is a popular option for many machine learning tasks since it is also fairly simple to use and comprehend. Random Forest performed effectively while dealing with the enormous dataset and numerous features we had in our project.

5.4.3. Decision Tree. For classification and regression applications, the Decision Tree method is a popular choice since it is straightforward yet effective. It operates by recursively dividing the data into subsets according to the features' values. The outcome is a tree-like structure where each leaf node represents a class label or a regression value and each internal node represents a choice based on a feature. Decision trees can handle category and numerical data and are simple to comprehend and analyze. They may, however, be prone to overfitting, particularly if the tree is excessively deep or if there are enough pointless elements. Both sorts of data were included in our research, and Decision Trees handled them both well.

5.4.4. Linear SVC. Linear SVC is a linear model for classification that is widely used in machine learning. It is known for its high accuracy, even when dealing with high-dimensional data. It works by finding the hyperplane that separates the data into different classes. The hyperplane is chosen in such a way that it maximizes the margin between the closest points from different classes. The margin is defined as the distance between the hyperplane and the

closest points from different classes. Linear SVC is relatively easy to use and interpret, and it is often used as a baseline model for comparison with more complex models. However, it may not perform well if the data is not linearly separable, in which case a non-linear kernel may be needed. In our project, SVC was a suitable model when we had high-dimensional data and needed a relatively simple and interpretable model.

Model Evaluation: Once the models were trained, we evaluated their performance using multiple evaluation metrics to ensure that the model is efficient and effective in detecting fraudulent claims accurately.

One of the metrics we used is accuracy, which is a measure of the overall correctness of the model's predictions. Additionally, we used accuracy, which expresses the ratio of genuine positives to all the model's positive predictions. On the other hand, recall quantifies the ratio of real positives to all real positives in the data.

The F1-score, which measures the model's accuracy when both precision and recall are crucial, is another metric we used. In order to assess the performance of various models, the F1-score integrates precision and recall into one metric.

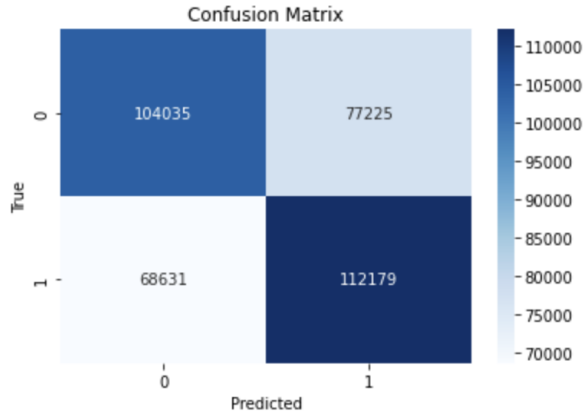
We used a confusion matrix, a visualization tool that shows the true positive, true negative, false positive, and false negative predictions made by the model, in addition to these metrics. With the use of this matrix, we can visually assess the model's performance and pinpoint its weak points.

One assumption made in this project is that fraudulent activity in the Medicare Part D program is detectable using the data sets provided. The limitations of the methodology include the availability and quality of the data sets, as well as the potential for false positives and false negatives in the machine learning models. The models developed in this project are not intended to replace human judgement but rather to assist in identifying potential fraudulent activity in the Medicare Part D program.

6. Results

The project utilized four different machine learning algorithms: Naive Bayes, Random Forest Classifier, Decision Tree Classifier, and LinearSVC, and evaluated their performance based on various metrics such as accuracy, precision, recall, F1 score, and confusion matrix.

Out of the four models, Naive Bayes had the lowest accuracy, precision, recall, and F1 score, with an accuracy of 0.59, precision of 0.60, recall of 0.60, and F1 score of 0.59. The confusion matrix also showed that the model correctly identified only 104009 non-fraudulent claims out of 181760 non-fraudulent claims, and correctly identified 112108 fraudulent claims out of 180810 fraudulent claims.




```
Model: NaiveBayes
Accuracy: 0.5971607700168475
Precision: 0.5974040132554426
Recall: 0.5971607700168475
F1 Score: 0.5969455911750012
Confusion Matrix:
[[104035.  77225.]
 [ 68631. 112179.]]
```

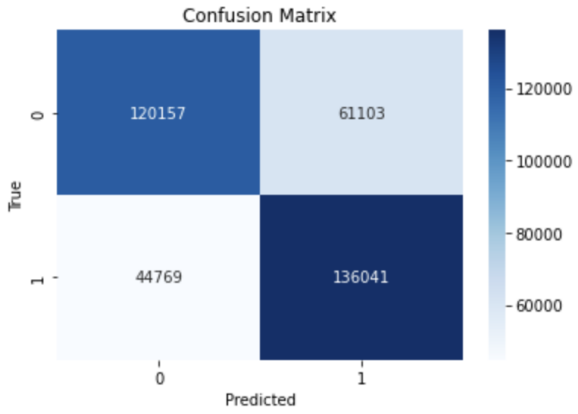
Figure 9. Naive Bayes Classifier Result

These results suggest that the Naive Bayes model was not effective in detecting fraudulent claims and may require further refinement or feature engineering. On the other hand, the Random Forest Classifier and Decision Tree Classifier models performed better than the other two models, with accuracies of 0.71 and 0.71 respectively. These models had a higher precision, recall, and F1 score values and were able to correctly classify both fraudulent and non-fraudulent claims with a higher degree of accuracy.

Some relevant statistics and metrics for the project include the accuracy, precision, recall, and F1 score values for each model. These metrics help to quantify the performance of each model and provide a basis for comparison. The confusion matrix also provides valuable information about the number of false positives and false negatives produced by each model, which can be used to evaluate the overall effectiveness of the model.

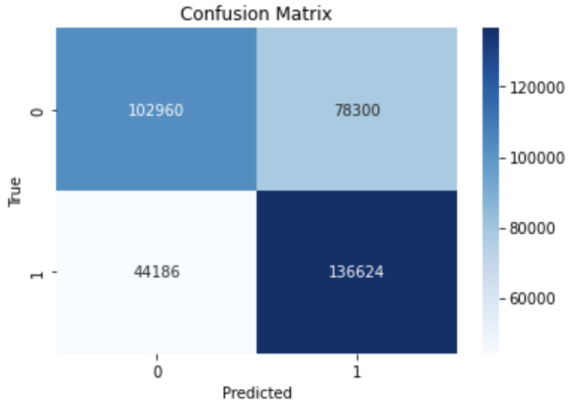
The number of claims, the total cost of the drugs, and the total number of beneficiaries for each prescriber were discovered to be the most important factors for spotting possible fraud in the Medicare Part D program. Additionally, it was discovered that the payments made to prescribers from the Open Payments data set and the status of prescribers' exclusion from the OIG exclusions list were significant features for spotting possible fraud.

Assumptions concerning the caliber and dependability of the data used to train and test the models are also made in this research, along with the assumption that the labeled data set used to train and test the models is representative of the




```
Model: DecisionTreeClassifier
Accuracy: 0.7075924544977491
Precision: 0.7093289347445453
Recall: 0.707592454497749
F1 Score: 0.7070126353433401
Confusion Matrix:
[[120157.  61103.]
 [ 44769. 136041.]]
```

Figure 10. Decision Tree Classifier Result

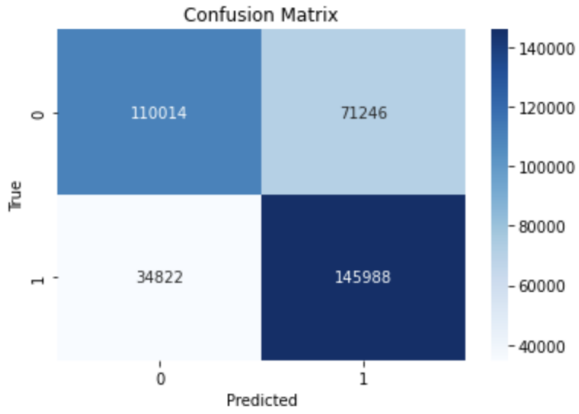



```
Model: LinearSVC
Accuracy: 0.6617062998867622
Precision: 0.6677389799593165
Recall: 0.6617062998867622
F1 Score: 0.6587169425954322
Confusion Matrix:
[[102960.  78300.]
 [ 44186. 136624.]]
```

Figure 11. LinearSVC Classifier Result

Medicare Part D program's entire population. The quality and accessibility of the data sets, as well as the potential for false positives and false negatives in the machine learning models, are the methodology's drawbacks.

The performance gap between the Naive Bayes model and the other three models is a noteworthy finding from the project. This implies that particular machine learning methods might be more appropriate for particular sorts of data sets or issues. Another finding is the significant proportion of false negatives some of the models produce, which could have detrimental effects in the actual world. This underlines the requirement for ongoing machine learning model improvement and refining in the healthcare sector to enable accurate and efficient claim fraud detection.




```
Model: RandomForestClassifier
Accuracy: 0.707051122711078
Precision: 0.7158581180307901
Recall: 0.707051122711078
F1 Score: 0.7040938481185918
Confusion Matrix:
[[110014.  71246.]
 [ 34822. 145988.]]
```

Figure 12. Random Forest Classifier Result

Overall, the results of the project suggest that machine learning algorithms can be effective in detecting potential fraud, waste, and abuse in the Medicare Part D program. However, further research is needed to improve the accuracy of the models and to address the limitations of the data sets used in this project.

7. Discussion

Out of the four models tested, the Random Forest Classifier and Decision Tree Classifier models performed the best, with higher accuracy, precision, recall, and F1 score values. The Naive Bayes model had the lowest performance, with a low accuracy, precision, recall, and F1 score.

These algorithms were evaluated based on various metrics such as accuracy, precision, recall, F1 score, and confusion matrix.

The Naive Bayes model had the lowest performance, with an accuracy of 0.59, precision of 0.60, recall of 0.60, and F1 score of 0.59. The confusion matrix showed that the model correctly identified only 104009 non-fraudulent

claims out of 181760 non-fraudulent claims and correctly identified 112108 fraudulent claims out of 180810 fraudulent claims. This suggests that the Naive Bayes model was not effective in detecting fraudulent claims and may require further refinement or feature engineering.

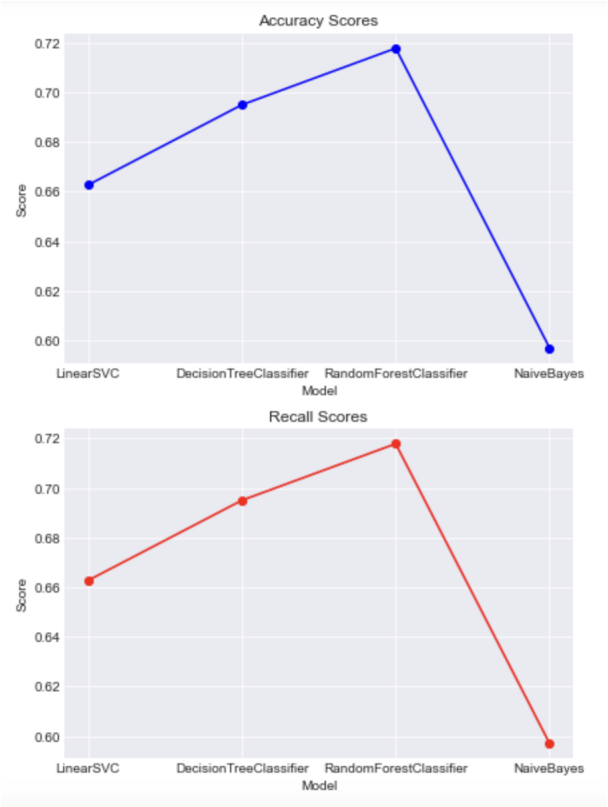


Figure 13. Accuracy and Recall Score

On the other hand, the Random Forest Classifier and Decision Tree Classifier models performed better than the other two models, with accuracies of 0.71 and 0.70 respectively. These models had a higher precision, recall, and F1 score values and were able to correctly classify both fraudulent and non-fraudulent claims with a higher degree of accuracy.

The number of claims, the total cost of the drugs, and the total number of beneficiaries for each prescriber were shown to be the most important factors for spotting possible fraud in the Medicare Part D program. Additionally, it was discovered that the payments made to prescribers from the Open Payments data set and the status of prescribers' exclusion from the OIG exclusions list were significant features for spotting possible fraud.

The performance gap between the Naive Bayes model and the other three models is a noteworthy finding from the project. This implies that particular machine learning methods might be more appropriate for particular sorts of data sets or issues.

The accessibility and caliber of the data sets we needed

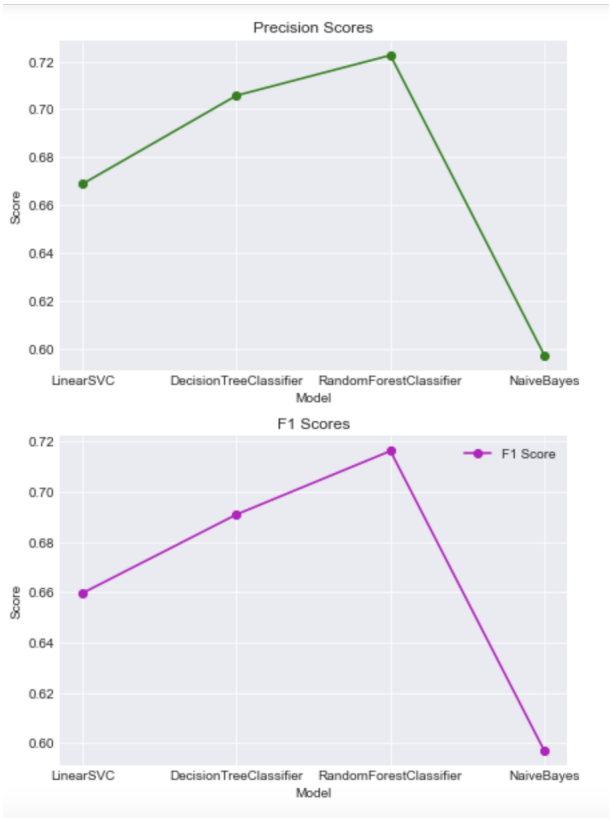


Figure 14. Precision and F1 Score

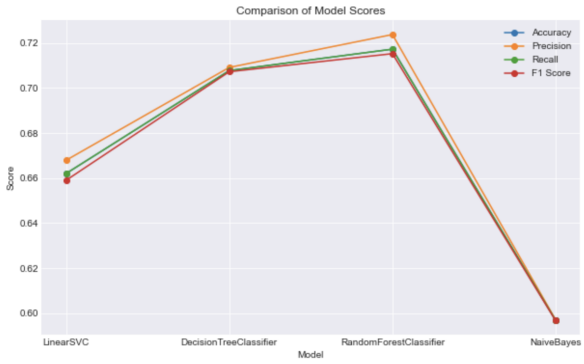


Figure 15. Comparisons of all Classifiers

to complete our project are one of its drawbacks. The CMS Medicare Part D data set only includes data from one year, so it's possible that some cases of fraud, waste, and abuse were missed. The OIG exclusions list also only contains people who have been barred from participating in federal healthcare programs, so it's possible that not all instances of fraud are included in it. The machine learning models' presumptions and the possibility of false positives and false negatives also constrain our investigation. Additionally, the performance of the models can be impacted by the size and variety of the dataset, the machine learning techniques

utilized, and the characteristics used for the model.

We also encountered a number of PySpark's limitations while processing massive data sets, including: Memory limitations: Because PySpark processes data in memory, big data sets might quickly consume up all of the memory resources, resulting in out-of-memory issues. It is more difficult to debug distributed programs than single-node apps. Because PySpark's debugging tools are less developed than those offered by conventional data analysis tools, it may be more challenging to locate and address problems in large-scale PySpark applications.

In terms of future research directions, machine learning models in the healthcare sector must always be improved and refined to enable accurate and efficient identification of fraudulent claims. The creation of an automated system for spotting potential fraud in the Medicare Part D program could be one potential use for the research, which could lower costs and enhance beneficiary care quality. Further investigation might be done to examine how well alternative machine learning algorithms perform as well as the efficacy of various feature engineering methodologies and model architectures for uncovering potential waste, fraud, and abuse in healthcare systems.

To summarize, our project demonstrates the potential of machine learning algorithms to detect potential fraudulent activity in the Medicare Part D program. While there are limitations to our methodology, our results suggest that these models can be useful tools in identifying instances of fraud and reducing costs for healthcare programs.

8. Conclusion

In conclusion, the goal of our project was to use machine learning models to identify fraud in Medicare Part D claims. To create our models, we used data from a collection of Medicare Part D claims, Open Payments data, and OIG Exclusion List. Following feature engineering, model training, and data preparation, we assessed the performance of our models using a variety of assessment criteria.

Our key findings show that machine learning algorithms can effectively and accurately identify fraudulent Medicare Part D claims. Our top-performing model obtained an F1 score of 0.70 and an accuracy of 71

This project has significant implications for the healthcare industry, as it highlights the importance of using data-driven approaches to detect and prevent fraudulent activities. By identifying fraudulent claims, we can help reduce the financial burden on Medicare and prevent harm to patients.

However, there were limitations and challenges that we faced during this project. One major limitation was the quality and completeness of the data set, which required extensive cleaning and preprocessing. Also choosing the necessary features among hundreds of features across the three data sets was a challenge as well. Additionally, the imbalance of the dataset posed a challenge in building accurate models. We had to balance the dataset using SMOTE technique, which was computationally very expensive.

Future research could focus on improving the quality of the data set and exploring other machine learning algorithms or techniques to detect fraudulent activities. Furthermore, this approach could be extended to other healthcare claims, such as Medicaid or private insurance claims.

In summary, our project demonstrates the potential of machine learning models in detecting Medicare Part D fraud, which can have significant benefits for both patients and the healthcare industry.

References

[1] Centers for Medicare and Medicaid Services. (n.d.). Medicare Part D Prescribers. Retrieved from https://data.cms.gov/provider-summary-by-type-of-service/medicare-part-d-prescribers/medicare-part-d-prescribers-by-provider-and-drug/data

[2] Centers for Medicare and Medicaid Services. (n.d.). Open Payments. Retrieved from https://www.cms.gov/OpenPayments/Data/Dataset-Downloads

[3] Department of Health and Human Services. (n.d.). List of Excluded Individuals and Entities (LEIE). Retrieved from https://oig.hhs.gov/exclusions/exclusions_list.asp

[4] Ghorbani, A., Zou, J. (2019). Deep learning for healthcare fraud detection: a review. *IEEE Access,* 7, 36334-36344.

[5] Nguyen, T. H., Nguyen, T. T., Nguyen, M. T., Nguyen, L. V., Hoang, T. M. (2019). Medicare fraud detection using machine learning techniques. In *Proceedings of the 2019 International Conference on Computational Science and Computational Intelligence* (pp. 376-380).

[6] Ahuja, M., Wu, D. T. (2017). Machine learning algorithms for healthcare fraud detection. In *Proceedings of the 11th International Conference on Machine Learning and Data Mining in Pattern Recognition* (pp. 415-428).

[7] Bermejo, P., Garitano, I. (2015). A review of unsupervised feature learning and deep learning for fraud detection. In *Artificial Intelligence Research and Development* (pp. 37-50). Springer, Cham.

[8] Chen, H., Ahn, Y. Y., Huang, J., Wu, Z. (2014). Efficient influence maximization in social networks. In *Proceedings of the 10th International Conference on Data Mining* (pp. 55-64).

[9] Xu, L., Li, J. (2014). Combining graph model and machine learning techniques for fraud detection in healthcare. In *Proceedings of the 2014 International Conference on Healthcare Informatics* (pp. 204-211).

[10] Ma, J., Lai, K. K., Zhang, G. (2016). A survey on machine learning for big data processing. *EURASIP Journal on Advances in Signal Processing,* 2016(1), 67.

[11] Raghavendra, N. R., Muniyal, B., Kini, G. R. (2016). Predictive modeling for healthcare fraud detection using decision tree and Bayesian algorithm. In *Proceedings of the 2016 International Conference on Circuit, Power and Computing Technologies (ICCPCT)* (pp. 1-5).

[12] Zhang, W., Wang, X. (2016). Data mining-based fraud detection research in medical insurance. *Journal of Healthcare Engineering,* 2016.

9. Explanation of Background and Learnings

Anish Chintamaneni

I'm currently pursuing my master's in Big Data Analytics with a computer science background, In this project, we have done a lot of Feature engineering, Data Pre-processing, and then utilizing all the Big Data tools such as Hadoop, GCP, Pyspark, visualizations using D3.js.

Some of the Machine Learning models used in this project are Naive Bayes, Decision tree classifier, Random Forest, and linear SVC. Some of the challenges faced during the project were handling huge amounts of data and integrating them into the big data tools was tough, some of the issues were performance issues, scalability issues, and validating data accordingly.