Dimensionality Reduction in Urban Analytics

Niren Patel[1]

MSc Data Science and Machine Learning

Supervisor: Prof Philip Treleaven

Supervisor: Dr Zeynep Engin

Supervisor: Dr Michal Galas

Department of Computer Science

University College London (UCL)

September 2018

[1] This report is submitted as part requirement for the MSc Data Science and Machine Learning at University College London. It is substantially the result of my own work except where explicitly indicated in the text. The report may be freely copied and distributed provided the source is explicitly acknowledged.

I, Niren Patel, confirm that the work presented in this thesis is my own. Where information has been derived from other sources, I confirm that this has been indicated in the thesis.

Code to accompany this project:

https://github.com/Niren52/UDL-UCL-NP

Abstract

This thesis explores the importance and effectiveness of different dimensionality reduction techniques in the field of Urban Analytics, with specific focus on solving four common challenges associated with high-dimensional public data:

- Quicker and more efficient ways of training data without significant information loss
- Creating more accurate prediction models by reducing overfitting
- Making complex data easier to understand

The amount of data being created every minute is increasing exponentially in both sample size and dimensionality. This thesis is important because it aims to discuss and provide foundations for the development of tools that will be useful for users of the Urban Dynamics Lab (UDL) platform.

The experiments in this thesis will investigate the effectiveness of three different approaches to dimensionality reduction:

1. Feature Selection performance in dimensionality reduction
2. Feature Extraction performance in dimensionality reduction
3. Effective Visualisation in dimensionality reduction.

Experiment 1: Feature Selection performance in dimensionality reduction

The first experiment applies a select few Feature Selection techniques; ‘Univariate’, ‘Model Based’ and ‘Recursive Feature Elimination (RFE)’. These techniques work in different ways to remove features that are either redundant (highly correlated with other important features) or irrelevant (provide very little information) and keep those which can explain the data the most.

The performance of these algorithms is tested against a benchmark of the original data in two different scenarios; high-sample data with a high number of dimensions and low-sample data with a high number of dimensions. The datasets to be reduced, include high-quality census data from the Office for National Statistics (ONS). Both scenarios will be tested with four different metrics: time taken to train models, F-1 score, the confusion matrix and Area under Receiver Operating Characteristic (ROC) Curve. There is also a discussion on the potential insight generation that can be obtained by further exploring the selected features.

Experiment 2: Feature Extraction performance in dimensionality reduction

The second experiment provides an alternative way to reduce high dimensional data by focusing on weighted feature-based methods, which are Principal Component Analysis (PCA) and Neural Networks. These techniques provide a lower dimensional representation of features that are a weighted combination of all the original features. The performance of both methods is also compared against the benchmark in two different scenarios; high-sample data with a high number of dimensions and low-sample data with a high number of dimensions.

Experiment 3: Visualisation Experimentation in dimensionality reduction.

The final experiment explores visualisation techniques that focus on discovering non-linear, non-local relationships in the data, that are not obvious in the original feature space. A deterministic algorithm, PCA Visualisation, and a stochastic model, t-Stochastic Neighbour Embedding (t-SNE), are run on both high-sample data with a high number of dimensions and low-sample data with a high number of dimensions. The performance of these are subsequently discussed.

Contributions to Science and the Urban Dynamics Lab

The major contribution of this thesis is the benefits of using dimensionality reduction in an Urban Analytics setting. This problem has become increasingly challenging as more public entities have become liberal in making data more openly available. The results of this thesis, will be used to influence the research of creating tools that can be incorporated into the UDL platform for use by the end users, namely project partners and governmental organisations.

Acknowledgements

This dissertation would not be possible without the guidance and expertise of Prof. Philip Treleaven, Dr. Michal Galas and Dr. Zeynep Engin throughout the project.

This Page is Left Intentionally Blank

Contents

List of Figures

List of Tables

Chapter 1: Introduction

This chapter presents an overview of the thesis. Research motivation is initially discussed along with the research topic. This is then followed by research objectives to be explored and data to be used. The final section of this chapter presents an overview of the thesis structure.

1.1 Research Motivation

The increasing prevalence of Machine Learning and Data Science in recent years has led to a surge in efforts to make data public and available to open source analysis. Open data has helped many industries such as Big Business, Sustainable Development and Renewable Energy gain invaluable insights. (1) (2)

In 2013, it was estimated that 2.5 quintillion (2.5 x 10^{18}) bytes of data were being created each day. (3) This number is likely to have risen substantially since then. Data has also been increasing in the number of dimensions, often far more rapidly than the number of instances, which leads to high sparsity. (4) This has resulted in the need for more expensive computing facilities and increased time to train accurate models. (5)

Another inevitable characteristic of high-dimensional data, is the increased likelihood of correlation between variables due to the number of variable-to-variable interactions. This has led to many redundant and irrelevant features being continued to be used in the creation of prediction models. High dimensionality also increases the risk of overfitting to training data when attempting to create models for prediction, particularly in low sample size scenarios. (6) (7) This is highly problematic when data is being assessed on a macro-level, e.g. by council or county levels.

The UDL was created to bring together expertise from numerous research departments from UCL and create a platform that would help their respective project partners, as well as both central and local governments with applied research and exploring avenues for policy impact. (8)

The nature of policy making is often long-term and expensive. Therefore, it is important to make decisions based on sound statistical science, that will make the most impact. The inability to do this often costs organisations, governments and large institutions billions of pounds. (9) Being able to identify the most significant features affecting target variables, can provide the opportunity to gain greater insights and focus efforts on decisions that are likely to provide the best results.

Users of the UDL are likely to want to present some of their justifications to key stakeholders when creating policy. It can be very difficult to show the relationship between data in a high-dimensional space. Hence, visualising the data onto a lower dimensional space can make it easier to understand and provide better intuitions. (10)

1.2 Research Objective

The main objective of the project is to figure out the most useful dimensionality reduction techniques for use in the field of Urban Analytics.
The main hypothesis of this research states that:

- By applying dimensionality reduction techniques, better insights can be acquired, Machine Learning models can be trained quicker and predictions made by models can be made more accurate.
- To validate this hypothesis, there are three main tasks for this research:

1. **Analyse the performance of Feature Selection methods in dimensionality reduction.**

   The first task is to prove that training times of models can be dramatically reduced and still provide credible results. This is because most features provide unnecessary noise to the data and not provide any additional information. The results will be compared against benchmark test results to make comparisons. There is also the additional task of obtaining better insights by viewing the features that will be ultimately selected.

2. **Analyse the performance of Feature Extraction methods in dimensionality reduction.**

   The second task is to prove that training times of models can be dramatically reduced and still provide credible results because the importance of some variables is less likely to be as significant as others. The results will also be compared against benchmark test results to make comparisons.

3. **Investigating Visualisation techniques for acquiring better insights.**

   The third task will focus on projecting the data onto lower dimensional spaces, to increase the efforts of gaining better insights into the data.

1.3 Data used in this Research

The two datasets to be used for the research is the 2011 census data sourced from Nomis, a service provided by the ONS. (11) The reason for using census data, is because it is generally of higher quality, due to the fact it is collected in person via an interview. Although, it costs more for data to be collected in person, the data very rarely has any missing data and is highly accurate. (12)

The datasets will come in a grouped form, by questions answered on the census. For example, one dataset will be of ‘Age Structure’ and will contain age brackets as the different features. Since ONS doesn’t offer unsegmented datasets, each dataset will have to be downloaded separately. Subsequently, they will need to be combined into one large dataset, so that interactions between features from different datasets can form non-linear interactions. The datasets are likely to be untidy after they have been imported into Python. This will require proper pre-processing, which will ensure the values in the cell match the correct features and instances of data.

One of the combined datasets to be formed will be on a microscopic level and contain data at the Postcode Sector level, being of the form “XX# #”, taking the all the beginning alphabetic letters and the first two digits following the letters. i.e. ‘BS9 2’ or ‘B62 8’

The combined dimensions being 8035 data points and 704 features, totalling to 5,905,725 cells. The dataset is likely to be highly sparse, especially for responses to questions containing obscure answers. E.g. questions asking about ethnicity.

The other combined dataset will be of a macroscopic nature and contain data at the Postcode Area level, taking the form of the first one or two alphabetic characters before numbers are introduced in a postcode. i.e. ‘WC – London West Central’ or ‘B-Birmingham’

There will be 105 data points and 852 features, combining to give 89,460 cells. This dataset has been used for the experiments because there can be many situations in which there are a high number of dimensions, but low sample sizes, these situations are particularly common when authorities and governments are looking to gain insights on a more regional level. Postcode Area datasets will have less sparsity because they are an aggregation of Postcode Sector datasets.

A detailed list of all features in each table can be found on the Nomis website. (13)

1.4 Structure of this dissertation

The Structure of this thesis is as follows:

- **Chapter 2** reviews key concepts associated to my project on Machine Learning and Big Data in Urban Analytics

- **Chapter 3** is used as a control experiment without the use of any dimensionality reduction techniques. The aim is to determine a suitable problem and aim form a benchmark for which different dimensionality reduction techniques can be compared against.

- **Chapter 4** assesses the performance of Feature Selection for dimensionality reduction using Univariate Statistics, Model based and RFE Feature Selection

- **Chapter 5** assesses the performance of Feature Extraction for dimensionality reduction using PCA and Neural Networks

- **Chapter 6** Investigates the performance of Visualisation techniques

- **Chapter 7** Summarises this research project and outlines possibilities for future work

Chapter 2: Background

This chapter explores some of the themes centred around this project to provide sufficient context to the reader. The themes to be discussed are Open Data, Big Data, Urban Analytics, Census Data and the Resurgence of Machine Learning

2.1 Open Data

The benefits of open and easily accessible data have benefitted many communities to solve high-impact problems. As a case study, take the destruction caused by Hurricane Katrina; many buildings in New Orleans were decimated beyond recognition. Obtaining information on the status of properties made decision-making around rebuilding attempts difficult. The creation of an open data-powered web application allowed both citizens and building inspectors to access map-based data, which enabled anyone to see the status of any property in a clear format, this allowed better decisions to be made on where to focus aid and effort. (14)

For open data to reap benefits like that above, the following key principles should to be followed (15):

- **Open by Default:** At this moment in time, it is common to query officials for specific information we require. Instead, it should be assumed that all data should be published, unless there is a valid reason to not do this. By having to ask for data, potential users may be deterred.
- **Timely and Comprehensive:** Open data should be provided in the original and unmodified form. It should also be published rapidly to ensure the actual data is still relevant.
- **Accessible and Usable:** Open data should be accessible to all, regardless of position or wealth and should be machine readable. This will encourage more users to use the data.
- **Comparable and Interoperable:** By providing numerous datasets that follow a standardised format, greater links can be formed between datasets.
- **For Improved Governance & Citizen Engagement:** Data should be provided with the intentions of improving public services and clearly understanding the motives of government officials.
- **For Inclusive Development and Innovation:** Data should also be provided with the expectations of individuals using it to solve problems not associated with the government. This can lead to vast improvements in the overall economy, as entrepreneurship and innovation increases.

The principals are not currently being implemented, however many countries are becoming open-minded due to the benefits. According to the Global Open Data Index, currently more than 27 countries share more than half their public data. Taiwan is the pioneer of making data open, with 90% of all its public data being shared. This is followed by Australia and the United Kingdom, both at 79%. (16)

2.2 Big Data

Big Data’s prevalence in recent years has allowed the industry to surpass worldwide revenues of $42 Billion in 2018, with the industry expected to grow to $103 Billion by 2027. (17) The benefits of harnessing Big Data could lead to an increased adoption of Internet of Things (IoT) and to the creation of many other high impact industries. Before this can be achieved, some of the problems associated with Big Data must first be addressed. (18)

The biggest problem facing organisations today is getting value from data. Only a quarter of executives describing Big Data initiatives successful. Getting insights out of "huge lumps" of data is demanding and requires methodological approaches to be taken. (19) This problem has been further compounded from the number of features increasing, which has led to the rise in sparse data structures that can take up unnecessary space and lead to poor performance in training Machine Learning algorithms. (18)

2.3 Urban Analytics

Cities have become as complex, becoming a breeding ground for real-estate, finance, culture and politics. Understanding how this complexity intertwines is therefore of major importance. (20)

Urban Analytics involves the practice of using data and high-level computation to gain insights into urban processes. (21) Cities can benefit greatly if big open data is exploited to provide better services and drive policy.

Urban Analytics should ideally (22):

- **Facilitate the execution of a city's day to day business**
- **Drive quantitative analysis of city operations**
- **Promote transparency between people and organisations**

If these goals are achieved, then the benefits can have improvements on the quality of life and lead to better decision making, because the allocation of resources will be targeted efficiently.

2.4 Census Data

Census data probably has the most detailed data of the population than any other source. It requires the in-person interview of people and the statistics collected can help display how the nation lives. The census data is used by public entities to underpin funding allocation for public services. (23) The only problem with census data is that it is collected infrequently every ten years due to its cost and there are new ways of collecting data being discussed.

The ONS, who provides census data for the United Kingdom, follows many of the open data principles discussed in Section 2.1. Data is open and accessible to anyone through their website. Data is useable since it can be downloaded in many different formats that might suit the user. Datasets are of a standardised form, which makes pre-processing and combining datasets interoperable for easy comparison. The ONS allows for increased citizen engagement and inclusive development by providing case studies on how data has been used by Students, Emergency Services, Governments and Business. Users are also encouraged to share how they used the data. (24)

2.5 Resurgence of Machine Learning

Over the last decade there has been resurgence in techniques related to Machine Learning. Artificial Intelligence (AI) has become more pervasive, as the applications for this technology are wide ranging in numerous situations such as self-driving vehicles. Research attempts from universities and technology companies such as Google, Facebook and Amazon in this field has led to deep learning become more widespread. (25)

2.5.1 Moore's Law

One reason for the resurgence has been because of the increase in computational power. Intel co-founder, Gordon Moore, predicted that the number of transistors on a microchip would double every two years while the costs halved (this has now become 18 months). (26) The increase in transistors has led to powerful compute capabilities. Deep Learning, which did not seem feasible in the mid-1980s, regained popularity within the Machine Learning community around 2010. (27) Figure 2.1 provides a graphical representation of Moore's Law.

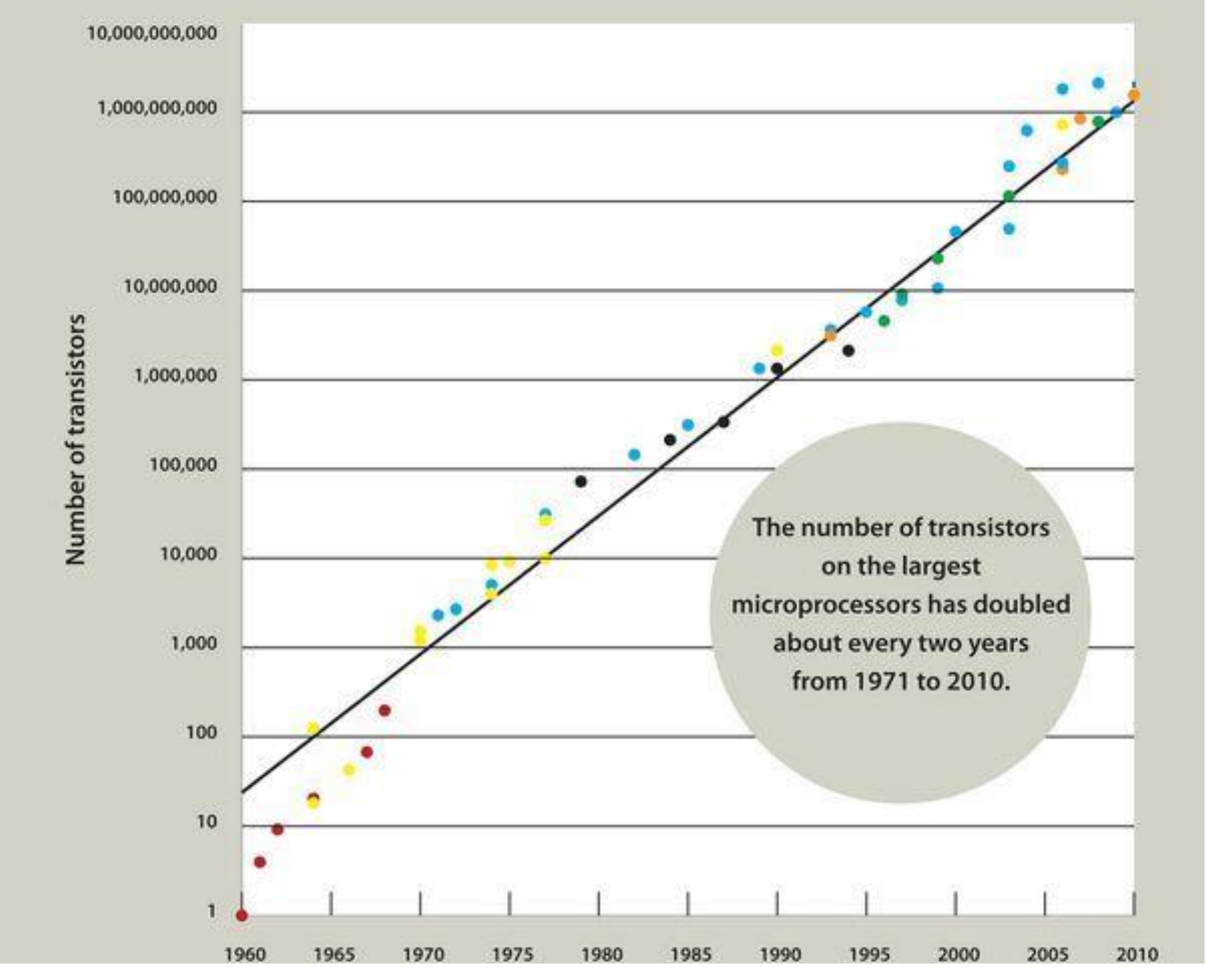

Figure 2.1 – Graph showing the number of transistors on the largest microprocessors has doubled every 2 years (28)

2.5.2 Better Data

Another reason for the resurgence is due to the increase in the number of devices connected to the internet. Technology companies have found it easier to collect data on users of their products with minimal effort. For example, Facebook has been able to collect high quality labelled data from users with the tagging feature on photos. This has given Facebook the largest sample of faces for training data, with each person having a variety of different angles and expressions. Facebook's facial recognition software surpassed that of the FBI's in 2014. (29) This influx of data has also coincided with better technologies in image processing software and hardware, leading to images that are sharper and have higher resolutions. Having access to better data has therefore increased the accuracy of these models.

2.5.3 Increased Research into the subject

The increasing affordability and access to large computing power is making it easier for people to carry out research in this field. More people can use cloud service platforms such as Amazon Web Services (AWS) to analyse larger datasets that may have been intractable on a consumer PC. (30) Algorithms for Machine Learning processes have now become a major research area. There is also more awareness of Computer Science in classrooms, which is allowing children to have access to computers from an early age. This is translating to more students considering careers in Computer Science and subsequently Machine Learning Research since it can be highly lucrative.

Overall, the increase in access to higher quality data, combined with better compute power and superior Machine Learning technologies, has led to major improvements for people living in urban areas. This has occurred directly with the adoption of these technologies from people working in the field but has also transpired because of private companies creating products and services that create a ripple effect which can benefit people in the field. This also includes the number of start-ups created in Urban areas that have created numerous jobs.

Chapter 3: Benchmark Testing and Problem Formulation

This chapter discusses a control experiment on the data without the use of any dimensionality reduction techniques. The purpose of this experiment is to set a benchmark and allow for comparisons in later stages of the thesis. This chapter will focus on problem formulation and will pick suitable Machine Learning algorithms, as well as robust evaluation metrics. The hyperparameters to be tuned for Machine Learning algorithms will also be discussed. In this chapter, I will also formulate a Machine Learning problem and decide on whether there will be a classification or regression problem. Different evaluation metrics will also be discussed along with the justification of which ones are to be used.

3.1 Research Challenge

To be able to fairly assess different dimensionality reduction techniques, it is imperative to look at stable and well tested Machine Learning algorithms and evaluation metrics. Using lesser known algorithms are not likely to have good support from the developer community and be prone to errors. There will need to be an emphasis on replicating real world test conditions as much as possible.

3.2 Choice of Algorithms and Metrics

3.2.1 Problem Formulation

3.2.1.1 Problem Identification

In this section I create a scenario where an institution such as a central or local government is looking to address the problem of there being a deficiency in the proportion of highly skilled workers in certain regions of the United Kingdom. The institution in question may be looking to figure out which factors are causing this and decide which parts of their expenditure they may wish to increase their efforts on.

To formulate this problem, unique geographical datasets of the ONS census data were compiled, which include features such as: Population Density, Male to Female Ratio, Ethnicity, Nationality and General Health Situation. In total, 38 datasets were combined for data at the postcode sector level and 37 at the postcode area level.

The ‘Highest Qualification Achieved’ dataset was turned into proportional form. The next stage involved removing all features contained in ‘Highest Qualification Achieved’, apart from the feature “Highest level of qualification: Level 4 qualifications and above”. Level 4 was chosen as the threshold because it includes qualifications such as a Certificate of Higher Education and the Higher National Certificate. Therefore, this makes it is a good indicator of a skilled workforce. (31)

This will now serve as the target variable for the remainder of the thesis.

3.2.1.2 Linear Regression vs. Classification

The next stage of the problem formulation was to decide whether this would be a linear regression or classification-based problem. Although the problem naturally lends itself to a linear regression solving, this was turned into a classification problem with the use of thresholds, since classification tasks can have more varied algorithms of solving problems. (32)

The creation of a classification problem was achieved by finding the 70% quantile of the target variable and assigning a 0 value to any value that fell below this quantile. Similarly, values of the target variable above 0.7 was set to 1. For the Postcode Sector data this threshold was at 33.2% proportion of students achieving level 4 education or higher, whilst for Postcode Area, this was set at 29.5%.

The reason behind creating this imbalance at a level of 0.7 was to replicate a real-world problem, where most data in any industry will often possess this trait, and standalone accuracy metrics without this information will not provide a good assessment of a Machine Learning algorithm. (33) (34) Figure 3.1 and Figure 3.2 show the class imbalance created

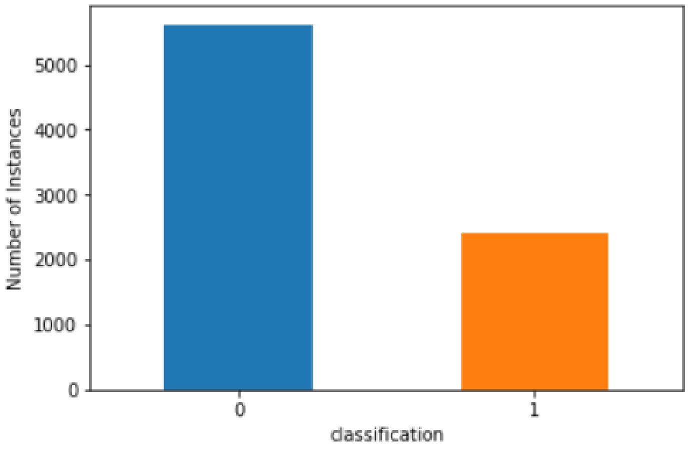

Figure 3.1 - Bar plot showing class imbalance of Postcode Sector data (high-sample data)

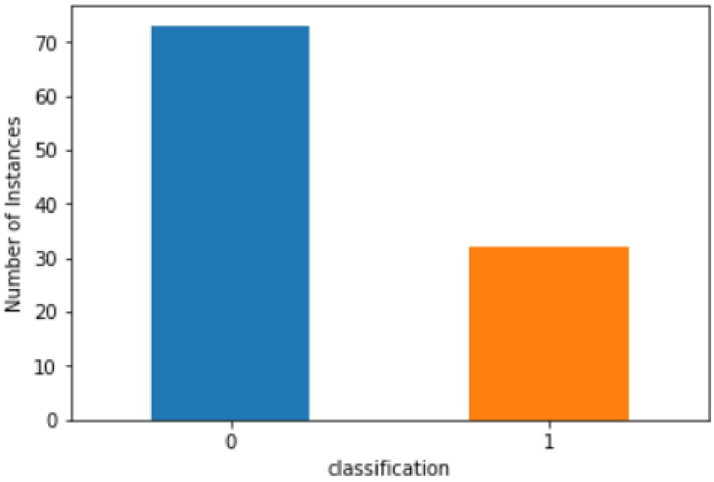

Figure 3.2 - Bar plot showing class imbalance of Postcode Area data (low-sample data)

At this stage, both datasets were now ready for Machine Learning to perform predictions. As a benchmark, if the classifiers were to make predictions as 0, the predictor would receive a 70% accuracy.

3.2.2 Consideration of Different Machine Learning Algorithms

This section focuses on finding suitable Machine Learning methods for the experimentation, the focus will be on robust algorithms that are able to handle data that has high sparsity and contains continuous variables. Testing on classifiers that come from different families of classifiers will allow for more diverse results.

3.2.2.1 Naive Bayes Classifier

The Naive Bayes classifier is a probabilistic supervised learning algorithm based on applying the Bayes' theorem with the "naive assumption" of independence between every pair of features. (35) Although the independence is generally a poor assumption, Naive Bayes can compete well against more sophisticated classifiers that can recognise non-linear interactions between features. (36)

From Bayes Rule, where y represents the class and each x represents the dependent feature vector x_i, where i is a value from 1 to n citation: (35)

$$P(y|x_1, \dots, x_n) = \frac{P(y)P(x_1, \dots, x_n|y)}{P(x_1, \dots, x_n)} \qquad 3.1$$

A naive independence assumption can be used to form the following equation: (35)

$$P(y|x_1, \dots, x_n) = \frac{P(y)\prod_{i=1}^{n} P(x_i|y)}{P(x_1, \dots, x_n)} \qquad 3.2$$

This is proportional to the product of the likelihood and the prior: (35)

$$P(y|x_1, \dots, x_n) \propto P(y)\prod_{i=1}^{n} P(x_i|y) \qquad 3.3$$

Estimation of y is now made by using the Maximum A Posteriori (MAP): (35)

$$\hat{y} = arg\,\max_{y} P(y)\prod_{i=1}^{n} P(x_i|y) \qquad 3.4$$

The Naive Bayes classifier can be a very powerful classification tool and requires moderate to large training samples to achieve reasonable results. It is particularly good at handling noisy data and discrete variables, particularly sparse data. (37) (38) Where the classifier falls short, is in situations where the majority of data is continuous, where a lot of information can ultimately be lost. (39) The classifier also doesn't lend itself to imbalanced classes, often performing poorly. (40)

In the case of the experiment, where data is mostly continuous and imbalanced, algorithm instability could arise with the use of a Naïve Bayes classifier.

3.2.2.2 Random Forest Classifier

The Random Forest Classifier is an ensemble learning method that works by creating multiple decision trees and taking a vote between these trees. It uses the bagging method of combining different Machine Learning models to improve the overall result. By subsampling features and using feature importance from the subset, additional randomness can be created in the model. (41) (42) As can be seen from Figure 3.3, Random forest works by constructing n-trees and allowing each instance of the data create a subset of random features.

The Random Forest Classifier will be a good classifier to test against, since it is robust and can deal effectively with sparse data. One of the potential disadvantages of the Random Forest, overfitting, will be important for the testing phase and seeing how effective dimensionality reduction techniques can be. (43) Another interesting disadvantage of the classifier is the amount of time it could take to train a model, time taken to train a model, generally increases linearly with time. (41) This will also help with seeing how effective dimensionality reduction techniques can be the objective of reducing training time.

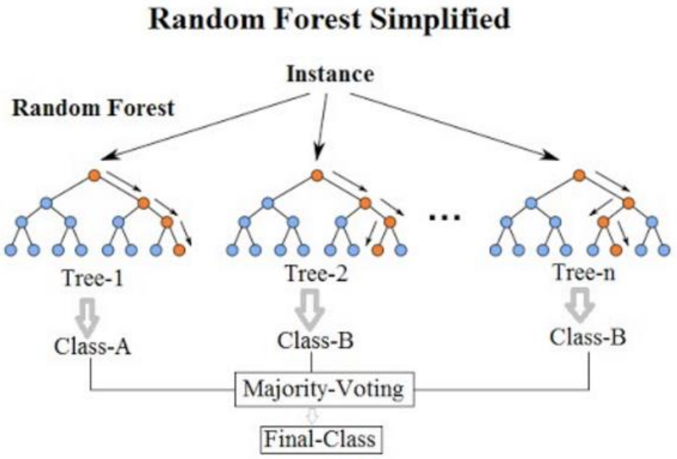

Figure 3.3 – Diagram showing the general way in which a Random Forest performs a prediction (44)

3.2.2.3 Logistic Regression

Logistic Regression is one of the most versatile and well-known classification algorithms. It is a linear form of classification that takes a weighted combination of independent or predictor variables. (45)

Logistic Regression uses the Sigmoid function as follows to return a probability value between 1 and 0:

$$p(x) = \frac{1}{1 + e^{-(z)}} \qquad 3.5$$

Where z is a linear weighted combination of the features:

$$z = Weight_0 + Weight_1 Feature_1 + \cdots + Weight_n Feature_n \qquad 3.6$$

Being one of the most stable algorithms and having many fail-safes built-in, such as numerous regularisation features, Logistic Regression will be one of the simpler algorithms to conduct analysis with. (46)

3.2.2.4 XGBoost

Since being created in 2014, XGBoost has become very influential in the Machine Learning community, winning numerous awards and Kaggle competitions. (47) (48)

XGBoost is the quick and high-performance algorithm that combines the information of weak learners of the data to perform stronger learners. It is an ensemble technique where new trees are added to correct for errors made by existing trees. (49) Theoretically, XGBoost should have the best overall performance throughout the testing phases.

3.2.2.5 Justification of Machine Learning Algorithms to be used

Overall, three classifiers should be enough to understand the performance of the metrics. The algorithms to be used for the rest of the thesis will be Logistic Regression, Random Forest and XGBoost; Naive Bayes will be discarded.

3.2.3 Evaluation metrics

This section focuses on choosing the best evaluation metrics for a classification problem.

3.2.3.1 Classification Accuracy

The classification accuracy metric is the number of correct predictions made, as a ratio of all predictions made. However, this is generally one of the worst forms of prediction since it doesn’t consider class imbalance. (50)

3.2.3.2 Logarithmic Loss

Logarithmic loss provides a measure of confidence for predictions by an algorithm, but this doesn’t provide enough insight into the actual prediction being correctly predicted. (50)

3.2.3.3 Area under ROC Curve

Shows Specificity v sensitivity and provides an instance by instance update of correct predictions from both classes. The graph usually has a diagonal, predictions that form a curve greater than the area under the diagonal, represents performance better than randomness. (51)

3.2.3.4 Confusion matrix

A confusion matrix gives a full breakdown of correct and incorrect predictions and allows for more insightful comparisons. Where more values fall on the diagonal, it is a better indication of a more accurate model.

Table 3.1 gives a general example of what a confusion matrix looks like. (50)

Table 3.1 – Table showing example of confusion matrix

	Predicted: 0	**Predicted: 1**
Actual: 0	True Negative	False Positive
Actual: 1	False Negative	True Positive

3.2.3.5 F1-Score

The F1-Score is the harmonic mean of the precision and the recall and is defined by (52):

$$F1\ Score\ = \frac{2 * Precision * Recall}{Precision + Recall} \tag{3.7}$$

Where:

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Negative} \tag{3.8}$$

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative} \tag{3.9}$$

The values required to solve for Precision and Recall can be found in

Table 3.1. The F1-Score needs to be calculated for both label 0 and label 1 and can then be averaged by the weight of total samples.

Overall the F1-Score is a good metric of performance since the use of the score is to find the best balance between Precision and Recall. (52)

3.2.3.6 Final choice of Evaluation Metrics

The Evaluation metrics to be kept for testing purposes will be the Area Under the ROC Curve, F1-Score and Confusion Matrix. Classification Accuracy will not be suitable for this problem due to the class imbalance, whilst the Logarithmic loss would have been preferable in a scenario where finding a minimum was to be part of the tests.

3.3 Experiment

3.3.1 Experimental Design

This experiment investigates the performance of three different types of Machine Learning algorithms without any dimensionality reduction techniques. By using a Jupyter Notebook, it was possible to clearly and logically display working code for users who are not familiar with the code. The individual datasets were combined and then processed to remove duplicated features with identical values, The ONS data also contained cells which possessed notes on the data but did not have any values. The data was split into training, validation and test sets using the Sci-kit learn framework.

The training and validation sets would allow for optimal hyperparameter tuning, whilst the test set would allow for unbiased testing on unseen data, to confirm there was no overfitting. Numerous functions were manually created and the most relevant hyperparameters to be tuned were set to a feasible range and spaced moderately. By using ‘Grid Search’ and a nested loop, the Machine Learning models can try an exhaustive combination of all the hyperparameters in the specified range and automate the tuning process.

The hyperparameters with the best results on the validation set can be used to create another model and perform unbiased predictions on the test set. The evaluation metrics obtained from performing these predictions on the test set are now ready for analysis.

The hyperparameters to be tuned were decided, based on general heuristics:

For logistic regression, the inverse of the regularisation strength was chosen to better generalise the model for testing on unseen data. (53)The other hyperparameter, maximum number of iterations taken for the solver to converge, was chosen to prevent convergence failure. (54)

For Random Forest hyperparameters to be tuned are; the number of estimators, maximum number of features and the maximum depth were chosen. The most important being the estimators, which represents the number of trees. The latter two hyperparameters are there to help prevent overfitting by setting a limit on the complexity of the model. (55)

For XGBoost, the learning rate hyperparameter controls how quickly the objective function reaches a minimal; a smaller value may lead to less overfitting but take longer to train. The maximum depth controls the complexity of the model for each tree and reduces the chance of overfit. Once again, the number of estimators, indicates the number of trees in the model. (56)

To ensure fair testing was achieved, all simulations were run on the same laptop as a standalone task. The simulations were run separately, preventing computations interfering with each other.

3.3.2 Results

3.3.2.1 Results on high sample dataset

As can be seen from Table 3.2, the Logistic Regression and XGBoost models take in the order of 10^3 to train, at 498.63 and 725.64 seconds respectively. As expected, the Random classifier took longer to train and was in the order of 10^4, with a time of 1239.39 seconds. Also, from Table 3.2, F1-Scores average in the 90s; with the F1-Score of class 0 being slightly higher than that of class1. From the confusion matrix, the high performance is obvious, with the total amount of incorrect predictions for all predictors being below 100 in a total of 1607 samples.

Table 3.2 – Classification performance of different Machine Learning algorithms with no dimensionality reduction for high sample dataset

Machine Learning Models	**Time taken to train (s)**	**Optimal Parameters**	**F1-Score on test set**	**Confusion Matrix for test set**		
Logistic Regression	498.63	Maximum Iterations = 50	0 = 0.97		**Predicted: 0**	**Predicted: 1**
			1 = 0.94	**Actual: 0**	1060	36
		Inverse Regularisation strength (C) = 0.001	Ave = 0.96	**Actual: 1**	30	481
Random Forest	1239.39	Estimators = 100	0 = 0.95		Predicted: 0	Predicted: 1
		Max Features = None	1 = 0.89	**Actual: 0**	1025	71
		Max Depth = 10	Ave = 0.93	**Actual: 1**	43	468
XGBoost	725.64	Estimators = 100	0 = 0.96		Predicted: 0	Predicted: 1
		Learning Rate = 0.1	1 = 0.90	**Actual: 0**	1045	51
		Max Depth = 3	Ave = 0.94	**Actual: 1**	47	464

The ROC curve depicted in Figure 3.4, shows the of true positive prediction rate against that of the false positive prediction rate for logistic regression. The model has been very successful in making predictions, compared to what would have been achieved with random guesses.

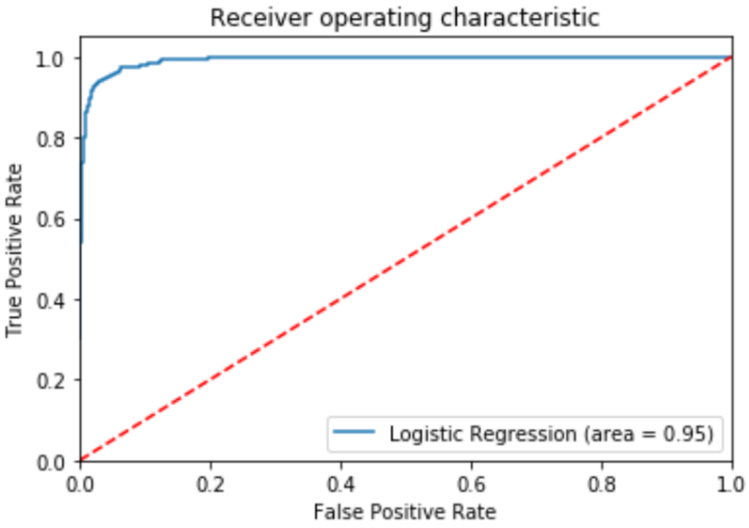

Figure 3.4 - Best ROC Curve with no dimensionality reduction in high sample data (Logistic Regression)

3.3.2.2 Results on low sample dataset

As expected, time taken to train the same models, with the same number of tuning hyperparameters on a lower sample dataset of just 63 was to be expected. In all three cases, time taken to train the model has been reduced by a factor in the order of 10^2.

Both the Random Forest and XGBoost algorithms have performed marginally better than the those from Section 3.3.2.1. With both scoring an average F1-Score of 0.95. Logistic Regression performs poorly in comparison the complementary in Section 3.3.2.1.

Table 3.3 - Classification performance of different Machine Learning algorithms with no dimensionality reduction for low sample dataset

Machine Learning Models	Time taken to train (s)	Optimal Parameters	F1-Score on test set	Confusion Matrix for test set		
Logistic Regression	7.04	Maximum Iterations = 300 Inverse Regularisation strength (C) = 0.5	0 = 0.87 1 = 0.60 Ave = 0.82		**Predicted: 0**	**Predicted: 1**
				Actual: 0	14	3
				Actual: 1	1	3
Random Forest	24.03	Estimators = 50 Max Features = None Max Depth = 10	0 = 0.97 1 = 0.89 Ave = 0.95		**Predicted: 0**	**Predicted: 1**
				Actual: 0	16	1
				Actual: 1	0	4
XGBoost	18.17	Estimators = 200 Learning Rate = 0.01 Max Depth = 2	0 = 0.97 1 = 0.86 Ave = 0.95		**Predicted: 0**	**Predicted: 1**
				Actual: 0	17	0
				Actual: 1	1	3

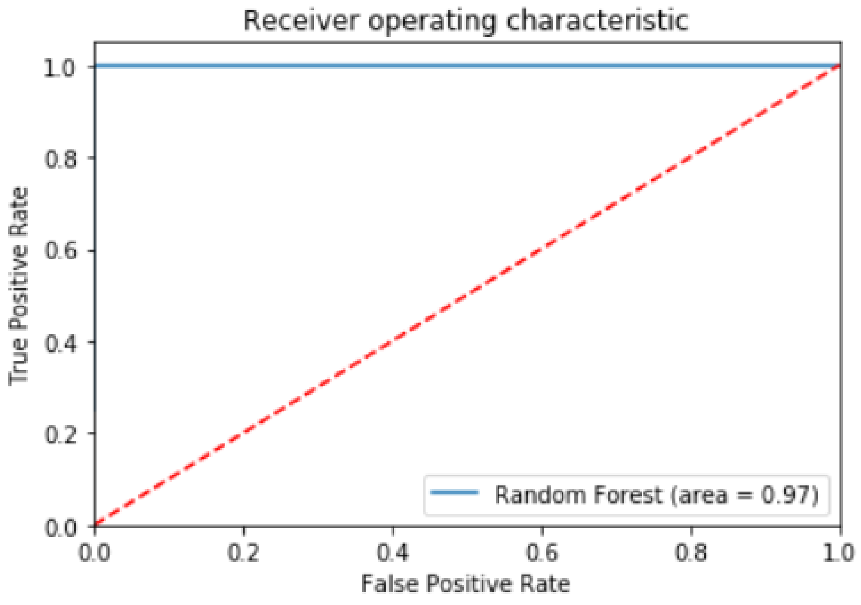

Figure 3.5 - Best ROC Curve with no dimensionality reduction in low sample data (Random Forest)

3.4 Assessment

In this chapter, I have formulated a plausible scenario in the form of a classification problem. I have also recommended appropriate Machine Learning and evaluation metrics that will be used in subsequent chapters. Baseline results have also been set to compare against.

Overall, the three Machine Learning algorithms perform well without any techniques being imposed on the data. The most surprising results come from the improved performance of the Random Forest and XGBoost algorithms on a lower sample dataset (Postcode Area) that is about 80 times smaller than that of the higher sample dataset (Postcode Sector).

Chapter 4: Feature Selection

In this chapter, I discuss the motivation of Feature Selection and investigate different Feature Selection techniques to deal with dimensionality reduction of the data. The results section compares the performance of different techniques against those of the control test. I also do a comparison between the Feature Selection algorithms.

4.1 Background

4.1.1 Motivation

Even with the modern advancements in the speed of computation, Feature Selection remains an important topic in the Machine Learning community. (5) Open data is generally of a high dimensional form. (57) These dimensional datasets tend to be quite noisy and unstructured due to sparsity; and it becomes very clear that most of the information, pattern and general trends can be explained by a very small subset of features. (5)

Being able to gain valuable insights from the noisy data in question could save a lot of time and resources, as users of the UDL platform can obtain key insights on data.

The benefits of exploring Feature Selection as a potential tool mathematical analysis include the following (58):

- Gaining a better understanding of the processes that generated the data
- The removal of redundant and irrelevant information
- Increased accuracies in scenarios of low sample high dimensional data
- Reduced training time for model creation

The benefits of achieving just a few of these benefits could vastly improve the quality and work and analysis of many organisations.

4.1.2 Related Work

An informative introduction to the general topic can be found in (58) and (59) gives some insights into the different algorithms.

A more advanced mode of Feature Selection is given in (60), where a Particle Swarm Optimisation method is discussed.

There are many applications of Feature Selection where the number of features is expected to be vast. This is particularly true in the domains of Neural Language Processing (61), Healthcare (62),Bioinformatics (63) and Biochemistry (64).

The use of Feature Selection in the field of Urban Analytics, includes use for the classification of satellite imagery (65) and Urban Structure Types (66).

4.2 Research Challenge

Being methodical about the choice of Feature Selection techniques will be important to obtaining reliable results. The selection of algorithms to be tested, must approach the problem in different ways to achieve diverse results. The problem remains the same as the one from Section 3.2.1.

4.3 Choice of Feature Selection Algorithms

4.3.1 Univariate Statistics Feature Selection

Univariate Feature Selection works by selecting the best features based on univariate statistical tests. (67) The statistical test determines if there is a statistically significant relationship between the output and each feature. This is known as Analysis of Variance (ANOVA). Each feature is considered in isolation and the highest scoring variables are chosen. (68) This method therefore deals with the problem of irrelevant features, but not redundant features.

This method was chosen because it is a filter type Feature Selection method and, although likely to give poorer results, would give a good baseline result to compare the other Feature Selection methods.

4.3.2 Model Based Feature Selection

Model based Feature Selection uses a supervised model to determine the importance of each feature and keeps the best features. A tree-based model, such as the Random Forest model, has built in feature importance attributes contained within the algorithm, that allow for the successful creation of a model. (69) The time taken to select the features may negate the effects of the subsequently improved training time. However, the features to be selected will likely be more thorough than the selection technique discussed in Section 4.3.1, providing better insights. (70)

Model Based Feature Selection was chosen because it uses a tree-based Machine Learning model, which should provide more sophisticated results.

4.3.3 Recursive Feature Elimination (RFE)

Given an output value, features are selected recursively, by considering smaller and smaller subsets of data. Initially, the estimator is trained on all the original features, and feature importance is computed from each feature. The least important features are pruned at each stage. This process is repeated on the pruned set, until the desired number of features are selected. (71) The time taken for RFE can be extensive, even more than Model Based Feature Selection. (72)

RFE was chosen because it is a wrapper type of Feature Selection, and therefore optimises itself for the model.

4.4 Experiment

4.4.1 Experimental Design

The experiment follows the same procedure of that in Section 3.2.1, however, before the features are passed through the Machine Learning algorithms, they are reduced to a lower dimensional space of 10% of the original features.

4.4.2 Results

4.4.2.1 Results for high sample dataset:

Results continue next page…

Univariate Statistics Feature Selection:

As shown in Table 4.1, The reduction in training time is considerable to that in Table 3.2. Logistic Regression training time decreases 28-fold, The Random Forest decreases by a factor of 7 and 8.5 for XGBoost.

For Logistic Regression, Performance dropped marginally from an average F1-Score of 0.96 to 0.95. However, with feature reduction, XGBoost and Random Forests scores remain the same. The confusion matrix, shows an increase of correct predictions by 1 for the Random Forest algorithm and a decrease of correct predictions by 3 for XGBoost. However, these changes might be explained by a sub-optimal choice of hyperparameters for prediction on the test set or some inherit randomness in computational calculations.

The ROC Curve for Univariate Feature Selection, depicted in Figure 4.1, shows very similar performance to that for no dimensionality reduction, shown in Figure 3.4.

Table 4.1 – Classification performance of different Machine Learning algorithms with Univariate Statistics Feature Selection for high sample dataset

Machine Learning Models	Time taken to train (s)	Optimal Parameters	F1-Score on test set	Confusion Matrix for test set		
					Predicted: 0	Predicted: 1
Logistic Regression	17.80	Maximum Iterations = 50 Inverse Regularisation strength (C) = 0.001	0 = 0.96 1 = 0.92 Ave = 0.95	Actual: 0	1029	67
				Actual: 1	21	490
					Predicted: 0	Predicted: 1
Random Forest	180.89	Estimators = 100 Max Features = None Max Depth = 10	0 = 0.95 1 = 0.89 Ave = 0.93	Actual: 0	1029	67
				Actual: 1	46	465
					Predicted: 0	Predicted: 1
XGBoost	85.02	Estimators = 200 Learning Rate = 0.1 Max Depth = 3	0 = 0.95 1 = 0.90 Ave = 0.94	Actual: 0	1049	47
				Actual: 1	54	457

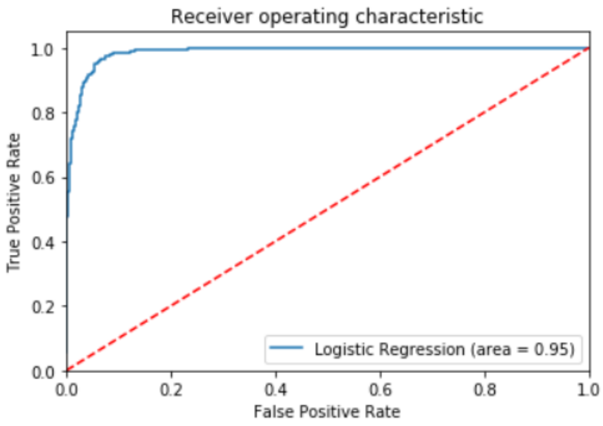

Figure 4.1 Best ROC Curve with Univariate Feature Selection in high sample data (Logistic Regression)

Model Based Random Forest Feature Selection:

As shown in Table 4.2, Training times for the Model Based selection is like that of Univariate Statistics in Table 4.1. However, the Model Based Feature Selection performs better than both Univariate Statistics and the Control Test, which is shown in Table 3.2, for both the Random Forest and XGBoost models. This may be negated by the long time it took to perform the Feature Selection, which was 41.39s. The results are marginal in terms of performance comparison and therefore, Univariate Statistics Feature Selection may be preferred in this case. The Area under the ROC curve shows the same values as both the Control Test and Univariate Feature Selection.

Table 4.2 - Classification performance of different Machine Learning algorithms with Model Based Random Forest Feature Selection for high sample dataset

Machine Learning Models	Time taken to train (s)	Optimal Parameters	F1-Score on test set	Confusion Matrix for test set		
Logistic Regression	18.82	Maximum Iterations = 200 Inverse Regularisation strength (C) = 0.005	0 = 0.97 1 = 0.93 Ave = 0.95		**Predicted: 0**	**Predicted: 1**
				Actual: 0	1042	54
				Actual: 1	21	490
Random Forest	186.36	Estimators = 100 Max Features = None Max Depth = 10	0 = 0.95 1 = 0.90 Ave = 0.94		**Predicted: 0**	**Predicted: 1**
				Actual: 0	1038	58
				Actual: 1	46	465
XGBoost	88.19	Estimators = 200 Learning Rate = 0.1 Max Depth = 3	0 = 0.96 1 = 0.91 Ave = 0.94		**Predicted: 0**	**Predicted: 1**
				Actual: 0	1046	50
				Actual: 1	46	465

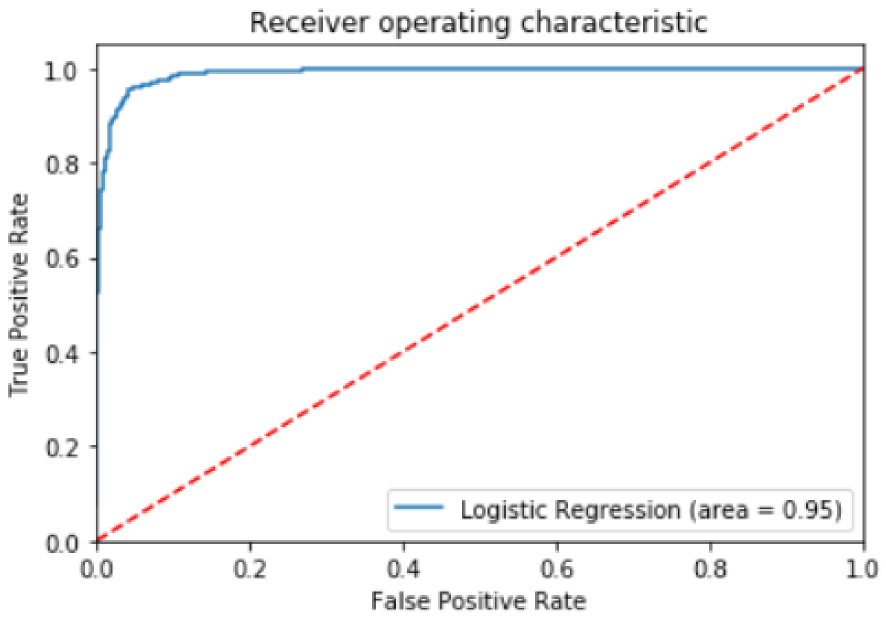

Figure 4.2 - Best ROC Curve with Model Based Feature Selection in high sample data (Logistic Regression)

Recursive Feature Elimination Feature Selection:

The Recursive Feature Elimination for Feature Selection under no circumstances performed better than the Test Control, it did however marginally perform better for Logistic Regression than both Model Based and Univariate Feature Selection modes. Results are very similar, however, just for this dataset RFE took 2324.01s, suggesting that overfitting has occurred. It is likely that parameters, can be tuned for the RFE Feature Selection process, however this just further adds to the time taken, as optimal parameters would need to be retrained for every new dataset. Once again, the ROC Curve Area Under Curve is 0.95, suggesting all the techniques are quite comparable for results.

Table 4.3 - Classification performance of different Machine Learning algorithms with Recursive Feature Elimination Feature Selection for high sample dataset

Machine Learning Models	Time taken to train (s)	Optimal Parameters	F1-Score on test set	Confusion Matrix for test set		
					Predicted: 0	**Predicted: 1**
Logistic Regression	20.01	Maximum Iterations = 300 Inverse Regularisation strength (C) = 0.001	0 = 0.96 1 = 0.93 Ave = 0.95	**Actual: 0**	1046	50
				Actual: 1	26	485
					Predicted: 0	**Predicted: 1**
Random Forest	163.10	Estimators = 100 Max Features = None Max Depth = None	0 = 0.94 1 = 0.87 Ave = 0.92	**Actual: 0**	1039	57
				Actual: 1	76	435
					Predicted: 0	**Predicted: 1**
XGBoost	80.37	Estimators = 200 Learning Rate = 0.1 Max Depth = 3	0 = 0.95 1 = 0.89 Ave = 0.93	**Actual: 0**	1045	51
				Actual: 1	56	455

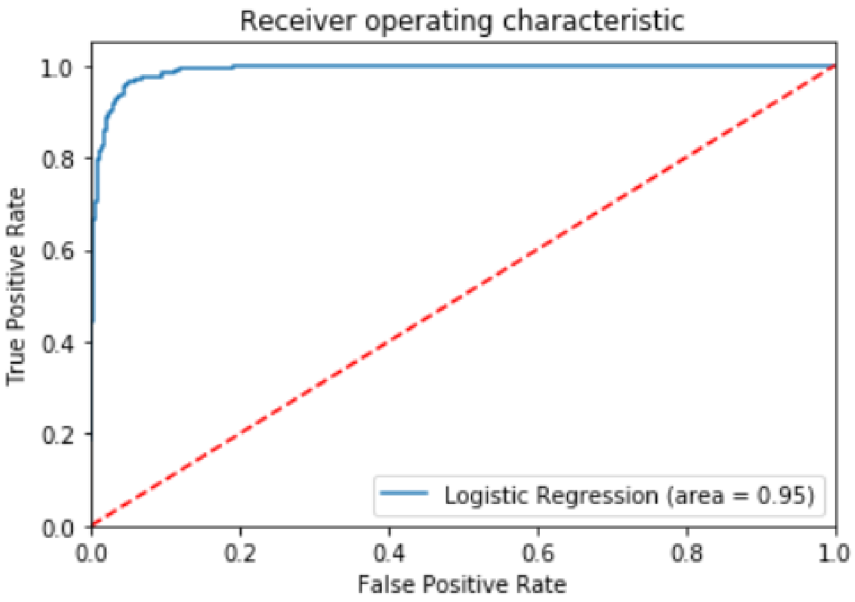

Figure 4.3 Best ROC Curve with RFE Feature Selection in high sample data (Logistic Regression)

Comparison Between Feature Selection Methods:

Figure 4.4, Figure 4.5 and Figure 4.6 represent the features selected for Univariate, Model Based and RFE modes of Feature Selection respectively. The 704 features are displayed in the spectrum graph below, with 60 yellow representing if a feature was selected and purple if it wasn't. As you can see, clusters are formed in Figure 4.4, This suggests correlation and may be due to a correlation statistic being the metric for Univariate Feature Selection. Because features in the combined dataset were formed from combining smaller datasets, they are likely to be more correlated.

The Model Based Feature Selection from Figure 4.5 also shows some small clusters, but not to the extent of Univariate Feature Selection.

RFE selection, shown in Figure 4.6, has a more spread out distribution of features. It also had the worst performance of the algorithms in the task, suggesting that features from the same subset will explain the data better.

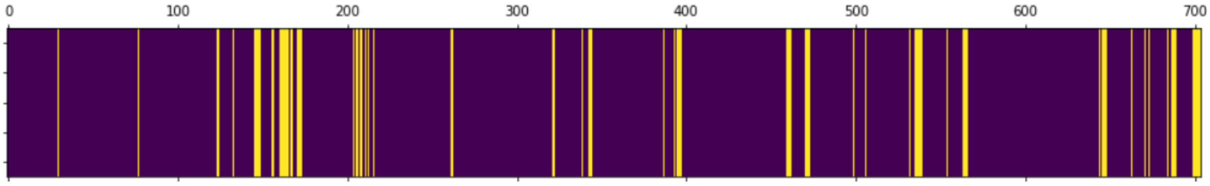

Figure 4.4 – Spectrum showing which variables were selected from the original, high sample feature set with Univariate Feature Selection (marked by lines of yellow)

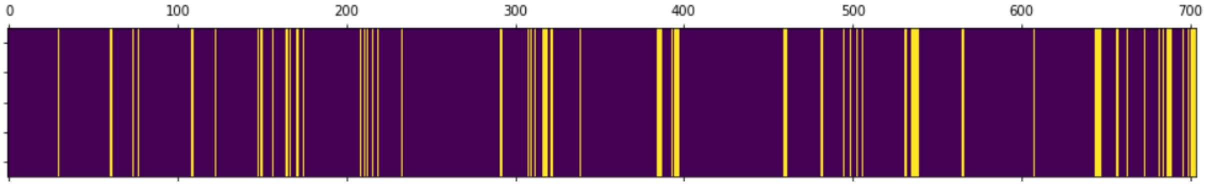

Figure 4.5 – Spectrum showing which variables were selected from the original, high sample feature set with Model Based Feature Selection (marked by lines of yellow)

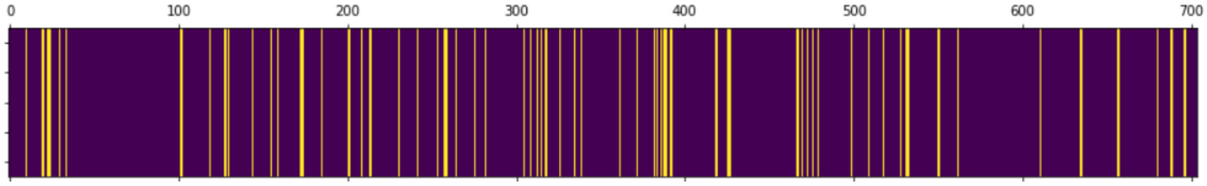

Figure 4.6 – Spectrum showing which variables were selected from the original, high sample feature set with RFE Feature Selection (marked by lines of yellow).

Table 4.4, Table 4.5 and Table 4.6 show the features selected by the models after the analysis for Univariate, Model Based and RFE Selection respectively.

For Univariate, features from the table 'Occupation Type' and 'Industry' were chosen to be the most important.

For Model Based Feature Selection, there are only 4 features selected that in the Top 15 that were from an Occupation or Industry type, and in this case, a variety of features affect whether someone has a Higher Quality Education.

For RFE, the features selected are based more on nationality and ethnicity.

The Feature Selection algorithms performed in a similar fashion to the Test Control in the classification. It appears that Ethnicity and Occupation are the biggest indicators of whether someone has a Higher Education qualification. Some occupations are going to need certain qualifications as a minimum requirement to work and this explains why it is an important feature.

Knowing these are the most important factors can allow a policy maker to investigate these further.

Table 4.4 – Top 15 Features selected by using Univariate Feature Selection on high sample data

1) Occupation: Process, Plant and Machine Operatives	2) Passport Held: No Passport Held	3) Occupation: Skilled Trades Occupation
4) Mode of travel to work: Passenger in a Car or Van	5) Occupation: Elementary Occupation	6) Occupation: Professional, Scientific and Technical Activities
7) Mental Health: Day to day activities limited a lot	8) Industry of Occupation: Manufacturing	9) General Health: Bad General Health
10) Industry of Occupation: Construction	11) Economically Inactive: Long term Sickness or Disability	12) Living Arrangements: One family only: Cohabiting Couple: All children non-dependent
13) Nationality: The Americas and the Caribbean: North America: Other North America	14) Occupation: Water Supply, sewerage, waste management and remediation activities	15) Living Arrangements: One family only: Cohabiting couple: children dependent

Table 4.5 - Top 15 Features selected by using Model Based Feature Selection on high sample data

1) Occupation: Process, Plant and Machine Operatives	2) Passport Held: No Passport Held	3) Occupation: Elementary Occupation
4) Number of bedrooms in the house: 3 bedrooms in house	5) Industry of Occupation: Manufacturing	6) General Health: Bad General Health
7) Industry of Occupation: Construction	8) Nationality: Other Western European	9) Nationality: Japanese
10) Economically Inactive: Long term Sickness or Disability	11) Hours worked: Full-time: 49 or more hours worked	12) Nationality: The Americas and the Caribbean: North America: Other North America
13) Nationality: Australian/New Zealander	14) Living Arrangements: One family only: Cohabiting couple: children dependent	15) Industry of work: Distribution, hotels and restaurants occupation Industry

Table 4.6 - Top 15 Features selected by using RFE Feature Selection on high sample data

1) Occupation: Associate professional and technical occupations	2) Nationality: Australian/New Zealander	3) Relationship Status: Female lone parent: In part-time employment
4) Industry of Occupation: Construction	5) Ethnicity: African	6) Nationality: The Americas and the Caribbean: The Caribbean: Total
7) Nationality: Europe: United Kingdom: Northern Ireland	8) Age of arrival in the UK: 25 to 29	9) Main Household Language: Spanish
10) Resident in UK: 5 years or more but less than 10 years	11) Nationality: Country of Birth: Middle East and Asia: Eastern Asia: Hong Kong (Special Administrative Region of China)	12) Religion: Other religion: Jain
13) Occupation: Managers, directors and senior officials	14) Ethnicity: Arab	15) Number of People in the household: 6 people in the household

All Feature Selection graphs, best F1-Scores were taken as I changed the proportion of features selected. As you can see from Figure 4.7, even a reduction to 1.5% (11 features out of 704), can still achieve a F1-Score that is in the 90s. Furthermore, the rise of the graph at early stages, further solidifies the argument that most data can be explained by a few variables

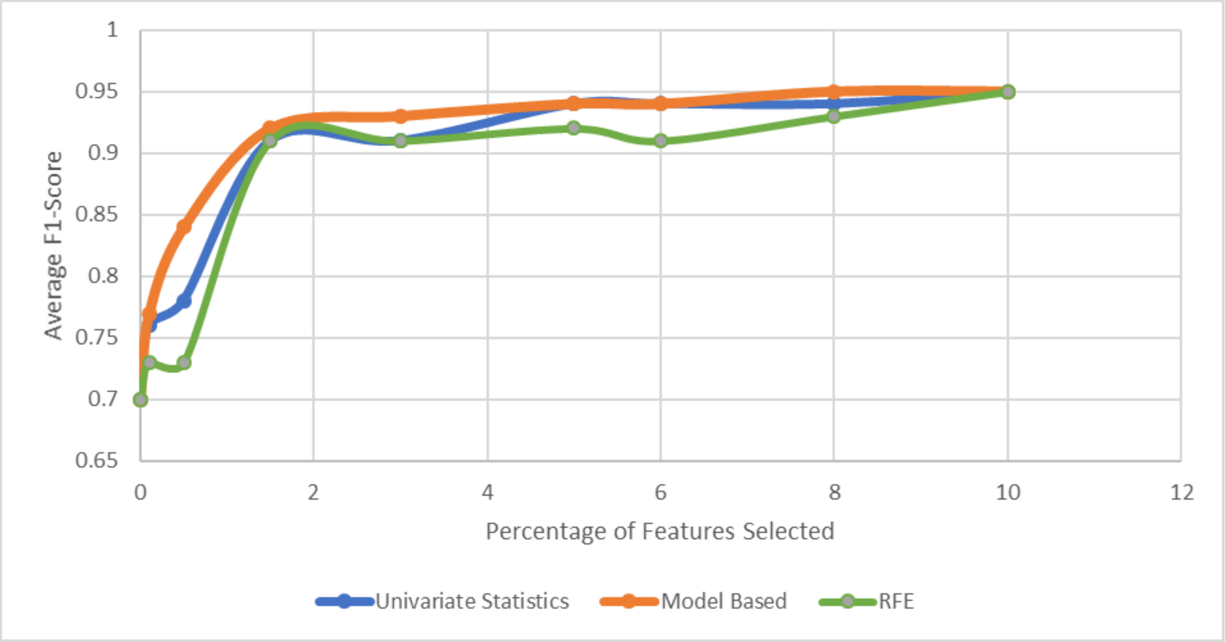

Figure 4.7 - Graph showing How Average F1 Score changes with the number of features selected

4.4.2.2 Results on low sample dataset

Univariate Statistics Feature Selection:

As shown in Table 3.3 and Table 4.7, for the low sample cases, there are not much significant time savings to be had by reducing features. The Univariate Feature Selection performs marginally poorly, in Logistic Regression and XGBoost, however, for the Random Forest, the same exact results are achieved. The Area under the ROC Curve shows the same area for both the Test Control, as shown in Figure 3.5 and Figure 4.8.

Table 4.7 - Classification performance of different Machine Learning algorithms with Univariate Statistics Feature Selection for low sample dataset

<table>
<tr><th>Machine Learning Models</th><th>Time taken to train (s)</th><th>Optimal Parameters</th><th>F1-Score on test set</th><th>Confusion Matrix for test set</th></tr>
<tr><td><b>Logistic Regression</b></td><td>6.01</td><td>Maximum Iterations = 300
Inverse Regularisation strength (C) = 0.5</td><td>0 = 0.74
1 = 0.53
Ave = 0.70</td><td><table><tr><th></th><th>Predicted: 0</th><th>Predicted: 1</th></tr><tr><td><b>Actual: 0</b></td><td>10</td><td>7</td></tr><tr><td><b>Actual: 1</b></td><td>0</td><td>4</td></tr></table></td></tr>
<tr><td><b>Random Forest</b></td><td>17.90</td><td>Estimators = 100
Max Features = None
Max Depth = 10</td><td>0 = 0.97
1 = 0.89
Ave = 0.95</td><td><table><tr><th></th><th>Predicted: 0</th><th>Predicted: 1</th></tr><tr><td><b>Actual: 0</b></td><td>16</td><td>1</td></tr><tr><td><b>Actual: 1</b></td><td>0</td><td>4</td></tr></table></td></tr>
<tr><td><b>XGBoost</b></td><td>6.73</td><td>Estimators = 200
Learning Rate = 0.05
Max Depth = 5</td><td>0 = 0.91
1 = 0.67
Ave = 0.86</td><td><table><tr><th></th><th>Predicted: 0</th><th>Predicted: 1</th></tr><tr><td><b>Actual: 0</b></td><td>15</td><td>2</td></tr><tr><td><b>Actual: 1</b></td><td>1</td><td>3</td></tr></table></td></tr>
</table>

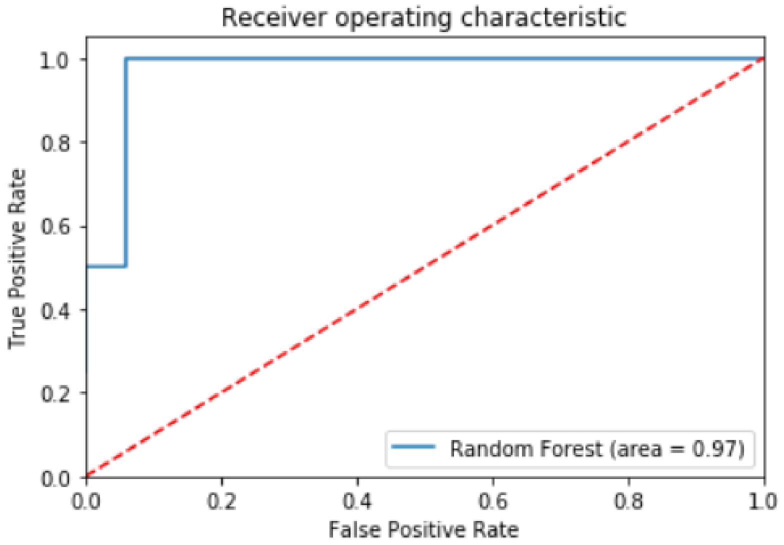

Figure 4.8 - Best ROC Curve with Univariate Statistics Feature Selection in low sample data (Random Forest)

Model Based Selection:

The Model Based Feature Selection, shown in Table 4.8, performs on par with the Control Test, and is marginally better than Univariate Statistics. The reduction in training times is minute and therefore negligible. The time taken to decide on features is also very short and takes only 4.71s.

The ROC Curve, shown in Figure 4.9, is smaller in area, and indicates initially poor False positive predictions.

Table 4.8 - Classification performance of different Machine Learning algorithms with Model Based Random Forest Feature Selection for low sample dataset

Machine Learning Models	**Time taken to train (s)**	**Optimal Parameters**	**F1-Score on test set**	**Confusion Matrix for test set**		
Logistic Regression	5.21	Maximum Iterations = 300 Inverse Regularisation strength (C) = 0.01	0 = 0.83 1 = 0.62 Ave = 0.79		**Predicted: 0**	**Predicted: 1**
				Actual: 0	12	5
				Actual: 1	0	4
Random Forest	17.90	Estimators = 100 Max Features = None Max Depth = 10	0 = 0.97 1 = 0.86 Ave = 0.95		**Predicted: 0**	**Predicted: 1**
				Actual: 0	17	0
				Actual: 1	1	3
XGBoost	6.65	Estimators = 200 Learning Rate = 0.1 Max Depth = 3	0 = 0.97 1 = 0.86 Ave = 0.95		**Predicted: 0**	**Predicted: 1**
				Actual: 0	17	0
				Actual: 1	1	3

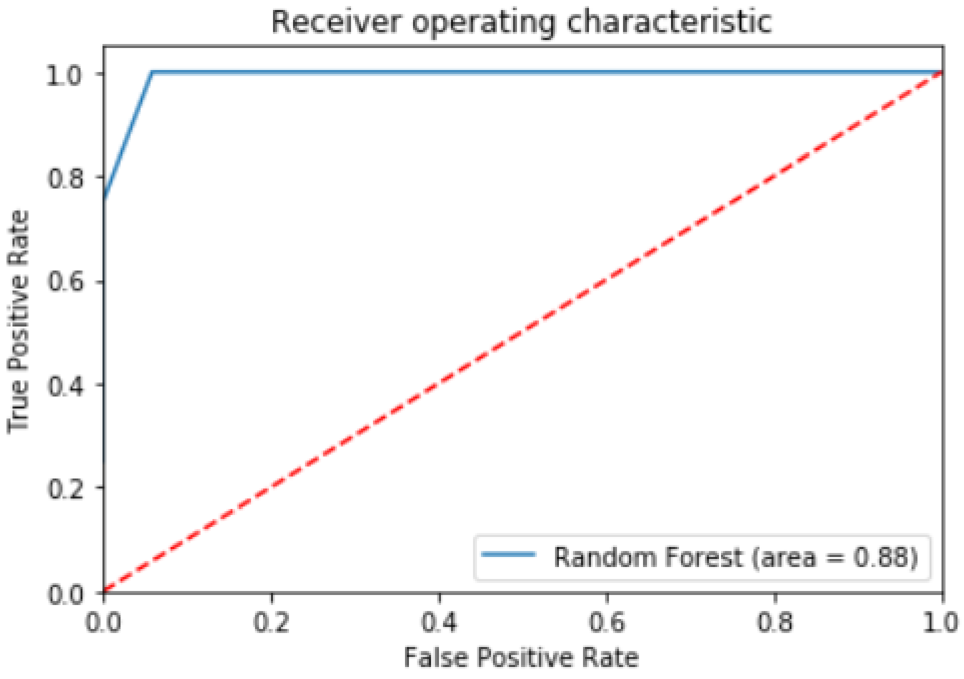

Figure 4.9 - Best ROC Curve with Model Based Feature Selection in low sample data (Random Forest)

Recursive Feature Elimination Feature Selection:

Overall, Recursive Feature Elimination performs better than both Univariate and Model Based Feature Selection, it eclipses the control test for both Logistic Regression and Random Forest. Achieving 100% accuracy in all predictions. Furthermore, the time taken to find features is only 6.33s, which is 0.27% of the 2324.01s taken to compute discussed in Section 4.4.2.1

Table 4.9 - Classification performance of different Machine Learning algorithms with Recursive Feature Elimination Feature Selection for low sample dataset

Machine Learning Models	Time taken to train (s)	Optimal Parameters	F1-Score on test set	Confusion Matrix for test set		
					Predicted: 0	**Predicted: 1**
Logistic Regression	5.14	Maximum Iterations = 300 Inverse Regularisation strength (C) = 0.5	0 = 0.94 1 = 0.80 Ave = 0.91	**Actual: 0**	15	2
				Actual: 1	0	4
					Predicted: 0	**Predicted: 1**
Random Forest	18.35	Estimators = 10 Max Features = None Max Depth = 1	0 = 1.00 1 = 1.00 Ave = 1.00	**Actual: 0**	17	0
				Actual: 1	0	4
					Predicted: 0	**Predicted: 1**
XGBoost	6.95	Estimators = 150 Learning Rate = 0.1 Max Depth = 1	0 = 0.94 1 = 0.67 Ave = 0.89	**Actual: 0**	17	0
				Actual: 1	2	2

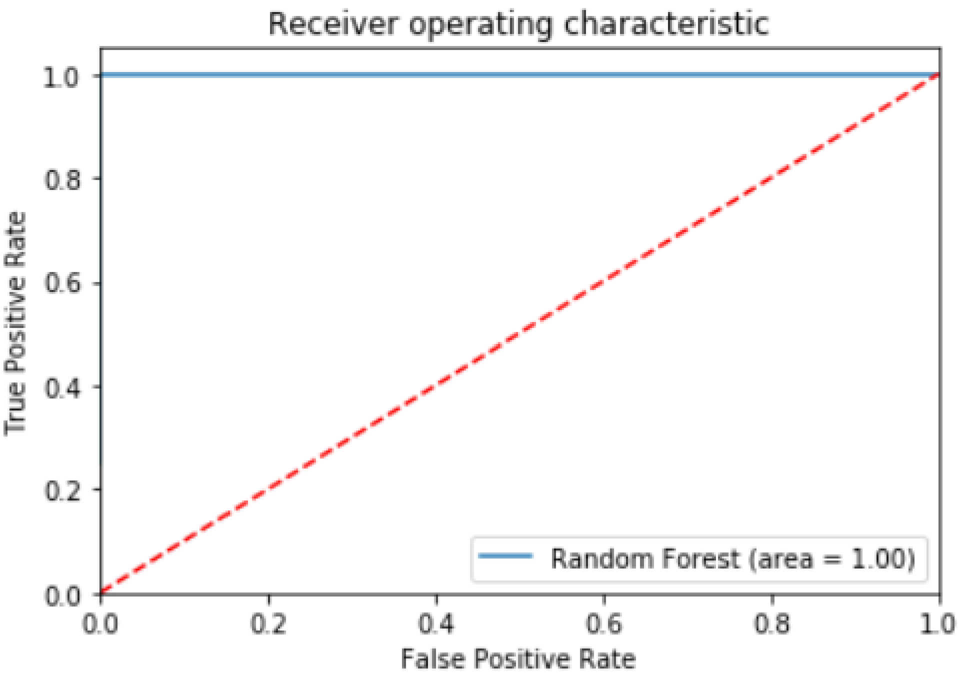

Figure 4.10 - Best ROC Curve with RFE Feature Selection in low sample data (Random Forest)

Comparison Between Feature Selection Methods:

Figure 4.11, Figure 4.4Figure 4.12 and Figure 4.13 represent the features selected for Univariate, Model Based and RFE modes of Feature Selection respectively. The 704 features are displayed in the spectrum graph below, with 60 yellow representing if a feature was selected and purple if it wasn't.
Figure 4.11 and Figure 4.12, show a bit more randomness in comparison the complementary graphs in Section 4.4.2.1, this is especially apparent in Figure 4.12.

However, Figure 4.13 has more clusters than its complimentary in Section 4.4.2.1 and had the best performance in the prediction task. From this and the previous section, it seems that the features to be selected with the best performance are likely to come from the same subset of features.

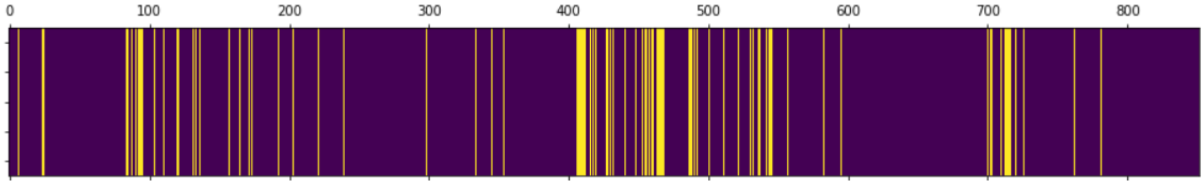

Figure 4.11 – Spectrum showing which variables were selected from the original, low sample feature set with Univariate Feature Selection (marked by lines of yellow)

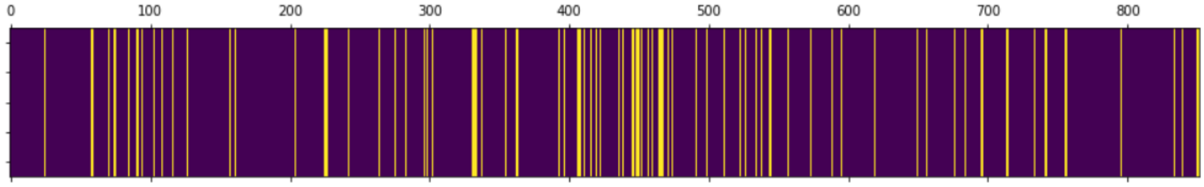

Figure 4.12 - Spectrum showing which variables were selected from the original, low sample feature set with Model Based Feature Selection (marked by lines of yellow)

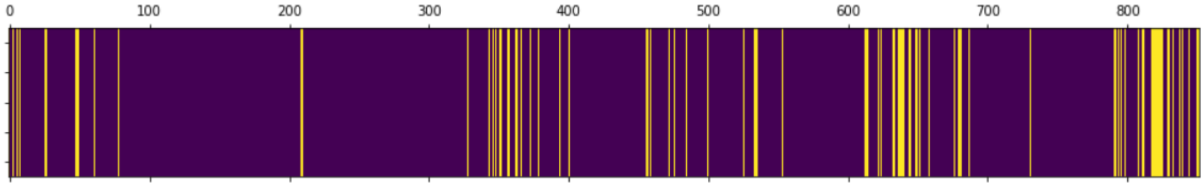

Figure 4.13 – Spectrum showing which variables were selected from the original, low sample feature set with RFE Feature Selection (marked by lines of yellow)

Table 4.10, Table 4.11 and Table 4.12 show the features selected by the models after the analysis for Univariate, Model Based and RFE respectively.

The features selected for Univariate Feature Selection are dominated by Ethnicity and Nationality Statistics, suggesting a similar notion to that from the previous section. The same can also be argued for Model Based Feature Selection.

A trend that has changed significantly from the previous section, is the lack of features associated with Industry and Occupation, this might be the result of the data being aggregated at this macroscopic level, reducing the sparseness considerably for the combined dataset.

The RFE's top 15 features selected show great diversity and seems to pick features from outside the clusters of subset features, only if they are significantly important.

Table 4.10 – Top 15 Features selected by using Univariate Feature Selection on high sample data

1) Nationality: Africa: South and Eastern Africa: South Africa	2) Nationality: Europe: Other Europe: EU countries: Member countries in March 2001: Germany	3) Nationality: The Americas and the Caribbean: North America: Canada
4) Population Statistic: Density (Number of people per hectare)	5) Main Household Language: Finnish	6) Main Household Language: Danish
7) Nationality: Europe: Other Europe: EU countries: Member countries in March 2001: Spain	8) Nationality: Middle East and Asia: South-East Asia: Singapore	9) Ethnicity: White Venezuelan
10) Main Household Language: African Language: Afrikaans	11) Ethnicity: White: Argentinian	12) Ethnicity: White: White African
13) Nationality: Europe: Other Europe: Rest of Europe: Other Europe	14) Nationality: Europe: Other Europe: EU countries: Member countries in March 2001: Total	15) Nationality: Europe: Other Europe: EU countries: Member countries in March 2001: Italy

Table 4.11 – Top 15 Features selected by using Model Based Feature Selection on high sample data

1) Occupation: Elementary occupations	2) Main Household Language: Other European Language: Swedish	3) Passport Held: No passport held
4) Ethnicity: White: Australian/New Zealander	5) Relationship Status: Female lone parent: In part-time employment	6) Main Household Language: Other European Language: Danish
7) Industry: Public administration, education and health	8) Ethnicity: Other ethnic group: Japanese	9) Main Household Language: South Asian Language: Hindi
10) Religion: Other religion: Occult	11) Resident in UK: Less than 2 years	12) Ethnicity: White: Croatian
13) Nationality: Europe: Other Europe: EU countries: Accession countries April 2001 to March 2011: Romania	14) Country of Birth: Antarctica and Oceania: Total	15) Main household Language: Other UK language: Scots

Table 4.12 - *Top 15 Features selected by using RFE Feature Selection on high sample data*

1) Occupation: Professional, scientific and technical activities	2) Number of bedrooms in household: 2 Bedrooms	3) Occupation: Professional Occupations
4) Type of Dwelling: Unshared dwelling: Whole house or bungalow: Terraced (including end-terrace)	5) Type of Dwelling: Unshared dwelling: Flat, maisonette or apartment: Total	6) Ethnicity: Asian/Asian British: Pakistani or British Pakistani
7) Economic Activity: In Employment	8) Type of Central Heating System:	9) 2nd Address: No Second Address
10) Mode of Transportation to work: Underground, metro, light rail, tram	11) Sum of all cars or vans in the area	12) Distance travelled to work: 2km to less than 5km
13) Nationality: Europe: Total	14) Type of Dwelling: Unshared dwelling: Whole house or bungalow: Terraced (including end-terrace)	15) Number of bedrooms in household: 1 Bedrooms

4.5 Assessment

In this chapter I have assessed the performance Univariate Statistics, Model Based and RFE Feature Selection by using metrics such as, training times, F1-Scores, Confusion Matrices and Area under ROC Curve.

It has been proven that, Features in a dataset can be reduced to at least 10% of the data, without significant loss in information. Even a reduction to 1.5% still performs reasonably well in a Machine Learning Scenario. It has also been shown that training times can be reduced significantly in the creation of models for larger datasets, without the loss of significant information. Furthermore, in scenarios of low sample datasets with a high number of features, accuracy can be improved. In addition to this, it seems that Feature Selection has the best performance when features are selected from the same subset of variables.

Chapter 5: Feature Extraction

In this chapter, I first discuss numerous aspects of Feature Extraction, then some background knowledge regarding this topic is examined. This is followed by what I will be trying to investigate, and then an examination of a select choice of handpicked algorithms. Detailed results are also given, with assessment of results, towards the end of the chapter.

5.1 Background

5.1.1 Motivation

As with the motivation provided for Feature Selection in Section 4.1.1, most datasets tend to be noisy and unstructured, due to sparsity. Feature Extraction offers many of the same benefits as Feature Selection, by instead creating a combination of weighted variables, rather than discarding them altogether.

Unlike Feature Selection, Feature Extraction is unable to provide names of the most important features, however, because all features are considered to some degree, this approach may appeal to more technical users, looking to build more powerful prediction models. (73)

The benefits of Feature Extraction include the following (74):

- Reduced training time in the creation of models.
- The suppression of less relevant information.
- Increased accuracies in scenarios of low sample high dimensional data

The benefits of this work are meant for technical users of UDL, that are looking to build faster and more accurate prediction models.

5.1.2 Related Work

A thorough and extensive introduction to Feature Extraction can be found in (73), (74) and (75). More advanced techniques are discussed in (76) and (77).

Feature Extraction has had many applications in technologies such as Facial Recognition (78) (79), Image Processing (80) and Text Categorisation (81). Industries to also make use of Feature Extraction include: Healthcare (82) and Biology (83) and Telecommunications (84).

There is a wider use of Feature Extraction in Urban Analytics than there is for Feature Selection, particularly in the fields of Geographic Information System (GIS) (85) and Classification. (86)

5.2 Research Challenge

Choosing well-known techniques with reliable architectures will be important to gaining high quality results, particularly when all variables will be used. The selection of algorithms must offer different ways of approaching the same problem. The problem is once again the same as the one from Section 3.2.1.

5.3 Choice of Feature Extraction Algorithms

5.3.1 Principal Component Analysis (PCA) Feature Extraction

PCA works by transforming features into a new set of features, which are a linear combination of the original variables. These new variables, known as the Principal Components, are ordered in such a way, that the first few, retain most of the variation present in all the original variables. (87) The eigenvalue is the measure that explains which components are the most important, and which are the most important principle components. In total, the same number of principal components as there are features are calculated, with all associated eigenvectors being orthogonal to each other. (87)

PCA was chosen because it is known to give robust results with the exact same results on repeated tests. It is also very common in this field.

5.3.2 Neural Network Approach

The neural network approach can solve some of the problems associated with dimensionality reduction. This method, does not need to be passed through another Machine Learning algorithm and will directly output classification results, by performing Feature Extraction within the model.

The neural networks, contain nodes called neurons, grouped as layers. Every neuron in each layer is connected to each neuron in the preceding and succeeding layers by weighted links called synapses. The input layer, which is the first layer, must contain the same number of neurons as there are original features. This is followed by a specified number of hidden layers, that may be connected via activation functions, to ensure values for neurons do not become computationally intractable. (88) After the data passes through the final hidden layer, it will reach an output layer, which will output the prediction. Figure 5.1 gives an example of a simple neural network architecture. The aim of the neural network is to iteratively train the model, so the weights between the synapses correspond to the output for the data, whilst trying to minimise the error as much as possible. (88)

The hidden layers within the neural networks are generally difficult to interpret because after a few iterations, the complexity of the computations between the weights become exponentially complex. In general, people associate the hidden layers to a black box. (88)

Neural network was chosen because most other modes of Feature Extraction are variants on the PCA algorithm and offer similar results.

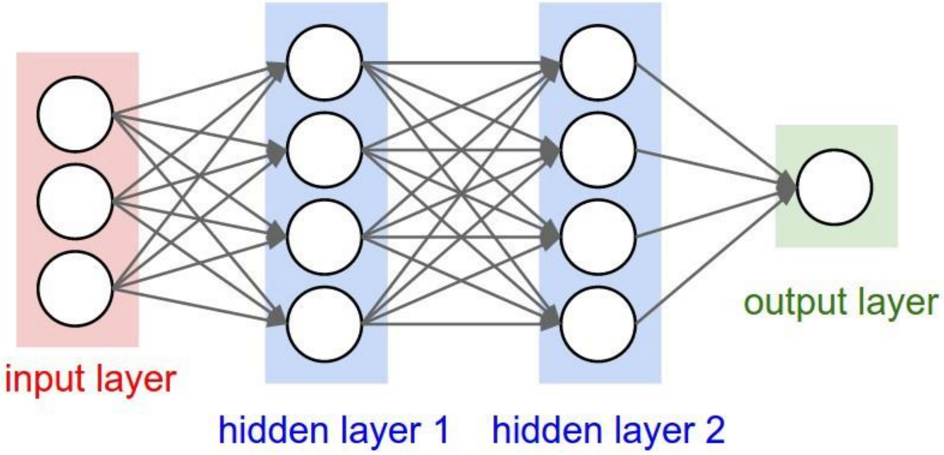

Figure 5.1 - A neural network with three inputs, two hidden layers of 4 neurons each and one output layer (89)

5.4 Experiment

5.4.1 Experimental Design

The experiment for Principal Component Analysis follows the same procedure of that in Section 3.2.1 and Feature Selection. However, before the features are passed through the Machine Learning algorithms, they are reduced to a lower dimensional space of 10% of the original features.

For the neural network approach, the data is taken with its original features and passed through a neural network that has the same architecture of that in Figure 5.1. The number of neurons for the input layer is the same as the number of features in the original dataset. The two hidden layers, will be the hyperparameters to tune, and both layers will contain numbers that are lower in size than original features, so dimensionality reduction occurs. (90) There is only one output, the prediction made by the algorithm.

One other hyperparameter to be tuned, will be the epoch, which represents how many passes all the training data will make through the network.

5.4.2 Results

Continued next page…

5.4.2.1 Results on high sample dataset

PCA Feature Extraction:

From Table 5.1, the time taken to train models using PCA, is significant compared to the test control, showing a reduction by a factor of 53, to 9.34s. This is around double the speed to that of the Feature Selection. However, for Random Forest and XGBoost, training times are from 50-100s slower. This is a surprising result, that was recurrent in repeat experiments.

All the results are marginally poorer than that of the test control, this includes the Area under the ROC Curve metric, which is shown in Figure 5.1

Table 5.1 – Classification performance of different Machine Learning algorithms with PCA Feature Extraction for high sample dataset

Machine Learning Models	Time taken to train (s)	Optimal Parameters	F1-Score on test set	Confusion Matrix for test set		
Logistic Regression	9.34	Maximum Iterations = 150 Inverse Regularisation strength (C) = 0.5	0 = 0.93 1 = 0.87 Ave = 0.91		**Predicted: 0**	**Predicted: 1**
				Actual: 0	988	108
				Actual: 1	31	480
Random Forest	268.87	Estimators = 50 Max Features = Auto Max Depth = 10	0 = 0.93 1 = 0.84 Ave = 0.90		**Predicted: 0**	**Predicted: 1**
				Actual: 0	1030	66
				Actual: 1	92	419
XGBoost	135.81	Estimators = 200 Learning Rate = 0.1 Max Depth = 3	0 = 0.94 1 = 0.87 Ave = 0.92		**Predicted: 0**	**Predicted: 1**
				Actual: 0	1048	48
				Actual: 1	80	431

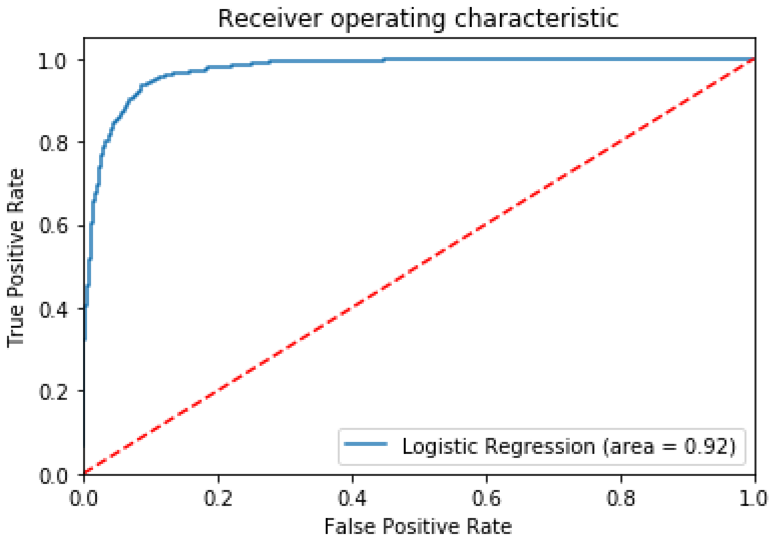

Figure 5.2 - Best ROC Curve with PCA Feature Extraction in high sample data (Logistic Regression)

Deep Learning: Neural Net architecture:

From Table 5.2, the time taken to train the model is large compared to other reduction techniques and is in the order of the original feature space, this is likely to be the result of two factors. Different selection of hyperparameters needing to be tuned, meaning methods are not comparable. The other reason, is because the Feature Extraction takes place within the model, meaning it may be on par with RFE, which was discussed in Section 3.3.2.1. Therefore, the time taken to train the model, is not a valid metric.

In terms of accuracy, the model performs exceptionally, with two hidden layers that have 500 nodes each. The results are once again hard to compare because none of three Machine Learning algorithms were used. The performance of the algorithm is exceptional in general, with results higher than most other techniques, an area under the ROC score of 0.98 is a clear example of this.

Table 5.2 - Classification performance of Neural Network based classification for high sample dataset

<table>
<tr><th>Machine Learning Models</th><th>Time taken to train (s)</th><th>Optimal Parameters</th><th>F1-Score on test set</th><th>Confusion Matrix for test set</th></tr>
<tr><td rowspan="3"><b>Neural Network</b></td><td rowspan="3">753.76</td><td>Hidden Layer 1 = 500</td><td>0 = 0.95</td><td rowspan="3"><table><tr><th></th><th>Predicted: 0</th><th>Predicted: 1</th></tr><tr><td><b>Actual: 0</b></td><td>1034</td><td>62</td></tr><tr><td><b>Actual: 1</b></td><td>43</td><td>468</td></tr></table></td></tr>
<tr><td>Hidden Layer 2 = 500</td><td>1 = 0.90</td></tr>
<tr><td>Epochs = 25</td><td>Ave = 0.93</td></tr>
</table>

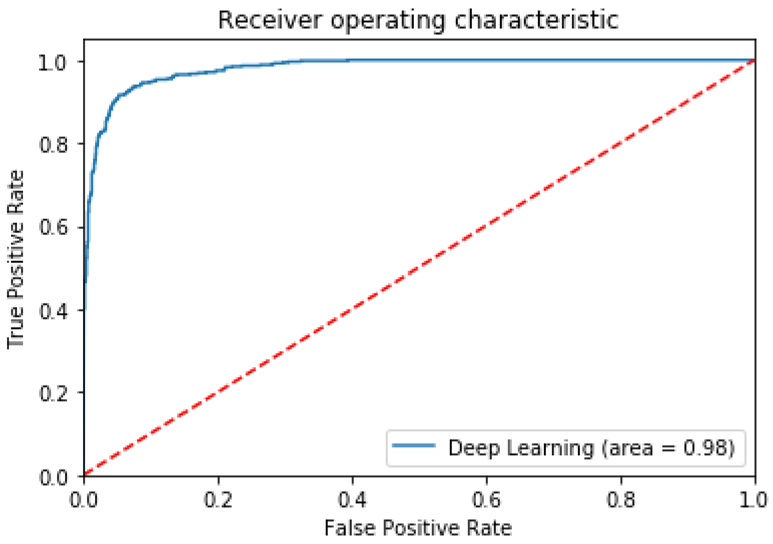

Figure 5.3 – Best ROC Curve with Neural Network in high sample data

5.4.2.2 Results on low sample dataset

PCA Feature Extraction:

From Table 3.1, the time taken to train models, offers surprising results once again. Logistic Regression and Random Forest show improvements in training times. However, the XGBoost classifier repeatedly produced training times that were greater in length than the Control Test. I think the reason for this is down to Principal components containing values that are up to 9 decimal places, whereas the original data consisted of at most, 5 digits in total, with no decimal places. The calculations must be taking longer, due to XGBoost calculating values to a higher accuracy. Random Forest and Logistic Regression are part of the Scikit-learn framework and may contain attributes which allow them to reduce values to a more manageable size.

The results for Logistic Regression are the best in comparison to the Control and Feature Selection tests. However, an average F1-Score of 0.74 and 0.83 indicates a generally poor dimensionality reduction technique.

Table 5.3 – Classification performance of different Machine Learning algorithms with PCA Feature Extraction for low sample dataset

Machine Learning Models	**Time taken to train (s)**	**Optimal Parameters**	**F1-Score on test set**	**Confusion Matrix for test set**		
					Predicted: 0	**Predicted: 1**
Logistic Regression	5.24	Maximum Iterations = 100 Inverse Regularisation strength (C) = 0.01	0 = 0.94 1 = 0.80 Ave = 0.91	**Actual: 0**	15	2
				Actual: 1	0	4
					Predicted: 0	**Predicted: 1**
Random Forest	17.41	Estimators = 50 Max Features = None Max Depth = 1	0 = 0.79 1 = 0.57 Ave = 0.74	**Actual: 0**	11	6
				Actual: 1	0	4
					Predicted: 0	**Predicted: 1**
XGBoost	135.81	Estimators = 150 Learning Rate = 0.05 Max Depth = 1	0 = 0.87 1 = 0.67 Ave = 0.83	**Actual: 0**	13	4
				Actual: 1	0	4

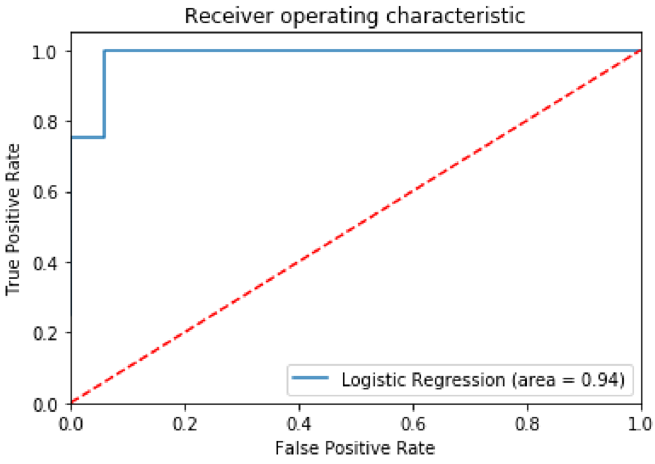

Figure 5.4 - Best ROC Curve with PCA Feature Extraction in high sample data (Logistic Regression)

Deep Learning: Neural Net architecture:

Once again time is not something that will be compared, due to issues mentioned in Section 5.4.2.1. From Table 5.4, the Architectures most optimal parameters are now 500 for layer 1 and 70 for layer 2. The results are once again robust, achieving scores in the 90s for all F1-Scores. There are only two wrong predictions. The Area Under the ROC Curve from Figure 5.5 is also exceptionally high.

Table 5.4 - Classification performance of Neural Network based classification for low sample dataset

<table>
<tr><th>Machine Learning Models</th><th>Time taken to train (s)</th><th>Optimal Parameters</th><th>F1-Score on test set</th><th>Confusion Matrix for test set</th></tr>
<tr><td><b>Neural Network Architecture</b></td><td>80.60</td><td>Hidden Layer 1 = 500
Hidden Layer 2 = 70
Epochs = 10</td><td>0 = 0.95
1 = 0.90
Ave = 0.93</td><td><table><tr><th></th><th>Predicted: 0</th><th>Predicted: 1</th></tr><tr><td><b>Actual: 0</b></td><td>15</td><td>2</td></tr><tr><td><b>Actual: 1</b></td><td>0</td><td>4</td></tr></table></td></tr>
</table>

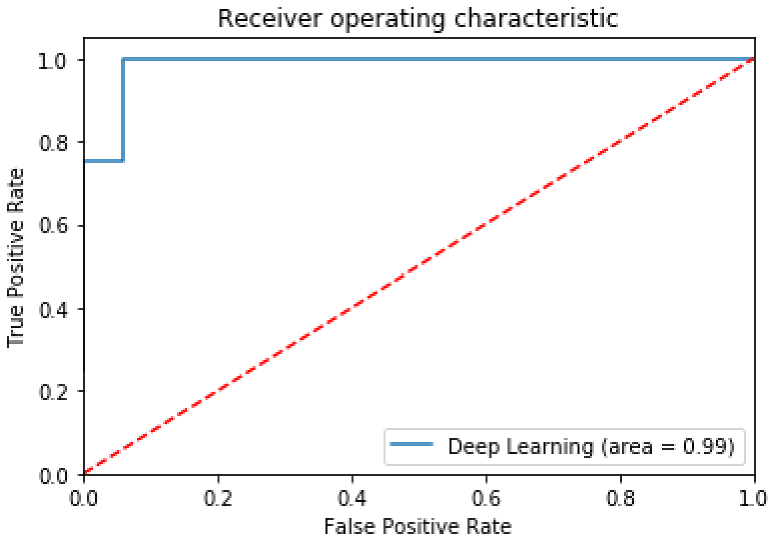

Figure 5.5 - Best ROC Curve for Neural Network in low sample data

5.5 Assessment

In this chapter, I have assessed the benefits of PCA Feature Extraction and Neural Networks by using appropriate evaluation metrics.

The results of the algorithms for the tasks is inconclusive. It is hard to gauge how effective Neural Networks were for the task due to being limited to only its architecture. However, the robustness and potential of Neural Networks are clear to see. PCA, which has numerous applications the Machine Learning industry, is too inconsistent, which makes it a poor choice for dimensionality reduction and the improvement of prediction model performance.

Chapter 6: Effective Visualisation

In this chapter, I investigate Visualisation methods and outline the benefits of effective visualisation. I then go into some detail of the algorithms I will be using for this task, with the potential benefits and drawbacks also discussed. Results will be compared and assessed towards the end of the chapter.

6.1 Background

6.1.1 Motivation

As it is hard for the human vision to comprehend anything greater than three dimensions, garnering insights from high dimensional datasets can become very problematic, especially when data is plentiful in both instances and features. (91) Therefore, being able to compress the data into two or three dimensions without losing information becomes a challenging but extremely important problem.

If done well, visualisation can draw the attention of people, this could be beneficial to users of the UDL platform, who's aim will be to explain key ideas to stakeholders and justify their decisions. (92) This is because numerical analysis doesn't always provide the insight that all stakeholders require. The work may also allow technical users to understand the type of data they're working with, and what they could do to further improve the data for use in building prediction models.

6.1.2 Related Work

There has been plentiful research undertaken in the field of visualisation. The reason for visualising, alongside how to implement modern techniques on computer packages, are discussed in (91). Background to data visualisation, with references to important work in the field is presented in (93).

Results for the advanced technique, Self-Organising Map-based data visualisation, is discussed extensively in (94) and (95).

Advanced forms of visualisation have been used numerous times in Healthcare (96) (97), Machine Damage Detection (98) and Policy Making (99).

In the field of Urban Analytics, interesting pieces of work have been created, which includes an overlay of New York Taxi demand patterns over the City's map. (100) There has also been use of visualisation in the representation of spatial content for Urban Conservation Processes. (101)

6.2 Research Challenge

Visualising the classes created in Section 3.2.1, will help with gaining a better insight into the data. It will be important to test one Deterministic model (there is only one solution), and one that is stochastic (no definitive solution). Different plots will then be compared.

6.3 Choice of Visualisation Methods

6.3.1 Principal Component Analysis (PCA) Visualisation

PCA Visualisation takes either the one, two or three principal components to create either 1-D, 2-D or 3-D plots. (102)The principal components can be computed similarly to that from Section 5.3.1. There is no parameter to tune for this model, the labels of the data can be represented by different colours, allowing for easy comparison in qualitative measures of how similar labels could be. If there are significant clusters being formed by certain colours, then that is a sign of there being significant correlation between features and labels. (103)

The justification for choosing this as a method of Visualisation is because of its deterministic attribute which gives one answer, without the need for the tuning hyperparameters.

6.3.2 t-Distributed Stochastic Neighbour Embedding (t-SNE)

t-SNE works by forming a low dimensional conditional distribution, q, from the original features conditional distribution of p. This is accomplished by minimising the KL divergence between each conditional distribution.

The equation for the KL divergence is (104):

$$E = KL(p|q) = \sum_i \sum_j p_{j|i} log \frac{p_{j|i}}{q_{j|i}} \qquad 6.1$$

The minimisation (and objective function) for this is therefore (104):

$$\frac{\partial E}{\partial \boldsymbol{y_i}} = 2 \sum_j (p_{j|i} - q_{j|i} + p_{i|j} - p_{j|i})(\boldsymbol{y_i} - \boldsymbol{y_j}) \qquad 6.2$$

Where $\mathbf{y_i}$ and $\mathbf{y_j}$ represent neighbours in the lower dimensional form.

t-SNE also uses the student t-distribution to prevent points that would be deemed as extreme values between distribution p based on the Gaussian Normal.

In contrast to PCA, t-SNE has a non-convex objective function and therefore needs to be solved using some iterative mode such as gradient descent. (105)

6.4 Experiment

6.4.1 Experimental Design

This method follows the same experiment from Section 5.4.1, however, instead of reducing the number of features by 10%, they are instead reduced to 2 features in total. These values are then plotted against each other and labelled based on their class.

As for t-SNE, because the number of features is high, the documentation suggests PCA reduction up to the level of 50 features and then perform t-SNE, to suppress noise. (106)

For this experiment, an exhaustive tuning of hyperparameters via grid search for the t-SNE model will be carried out. The learning rate will need to be tuned to ensure the objective function is efficiently minimised, the number of iterations will need to be tuned carefully to balance accuracy and time taken. Perplexity will also be tuned and determined the number of neighbours used for each iteration.

6.4.2 Results

6.4.2.1 Results on high sample dataset

Results start on next page…

Principal Component Analysis Visualisation:

The results from the PCA Visualisation are deterministic, and therefore can only have one answer. From Figure 6.1, there is little information that shows classification of 1 mainly tends to the right, whereas a classification of 0 is mainly to the left. Figure 6.2, which is a 2D projection of the two principal components, shows slight patterns and a clear tendency for values with the label 1, to cluster to the top right. The calculation and plotting of PCA was almost instant, taking a combined total time of 8s.

The results are more of a measure, showing the similarity of any two points in relation to each other based on their features and attributes. Therefore, the applications of this to the UDL Platform could be that an interactive map, that lets a user of the platform hover over data points and see how similar one location is to another.

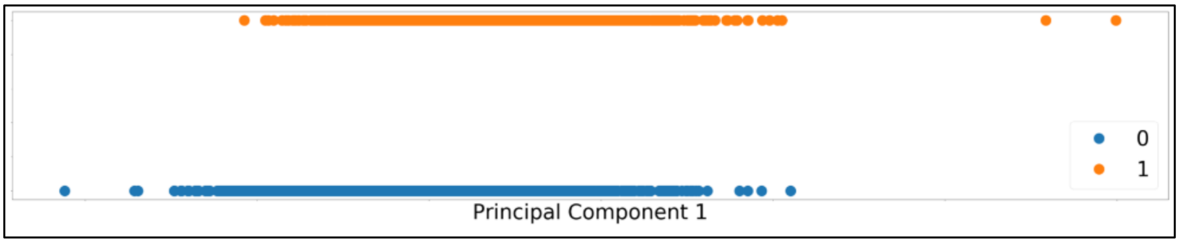

Figure 6.1 - PCA representation of the large sample data in 1-Dimension

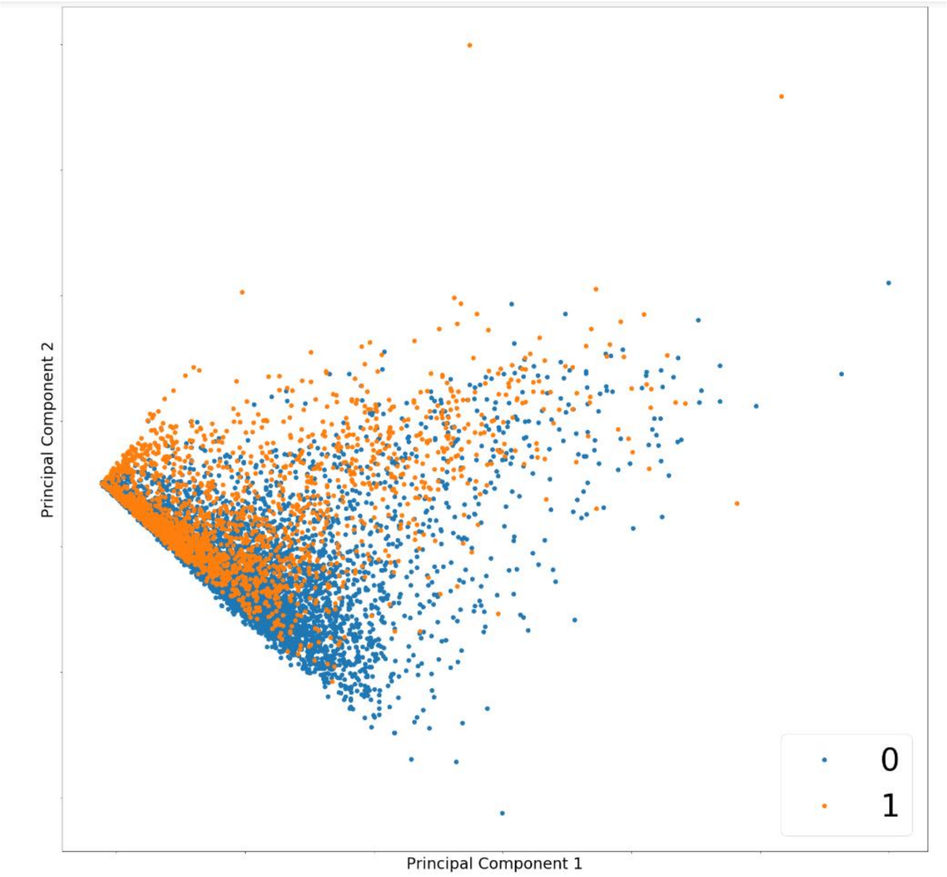

Figure 6.2 - PCA representation of the large sample data in 2-Dimensions

t-SNE:

Because the KL divergence objective function is not symmetric, t-SNE focuses on retaining local structure over global structure, which leads to results presented in the form a lot of small clusters, as shown in Figure 6.3Figure 6.3. When an attempt is made to include global structure, like in Figure 6.4, results are very poor. Having produced a total of 32 t-SNE graphs, I was unable to find any with obvious patterns. The algorithm would not be suitable for use as a visualisation tool on the UDL platform because it is likely to require a lot of hyperparameter tuning. The time taken to run the iterations of all hyperparameters, took 3366.38s, which is an average of 105.19s for each representation. T-SNE can create more visually appealing structures than PCA, requires too much input and compute power to be worthwhile.

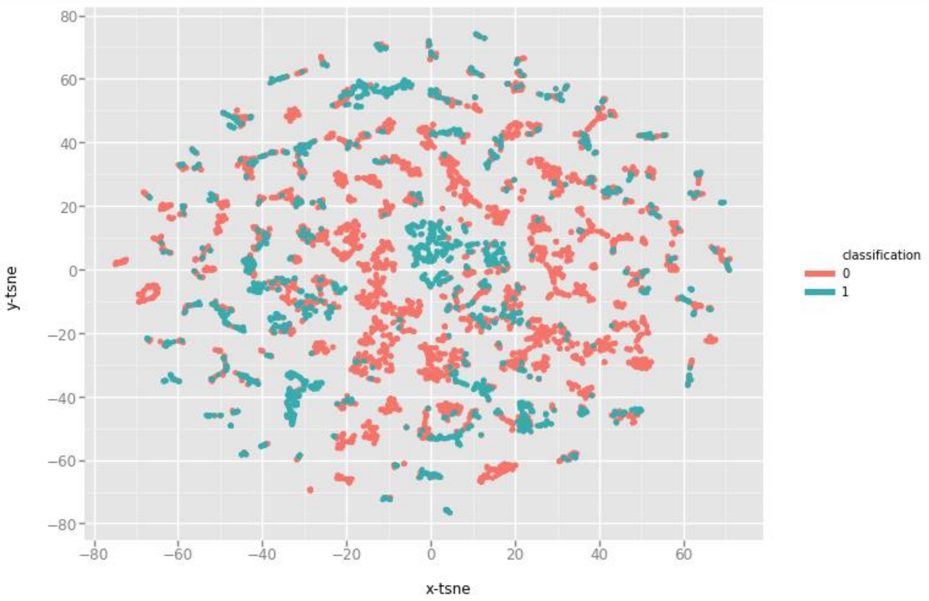

Figure 6.3 - t-SNE high sample representation: Perplexity = 15, No. of Iteration = 500, Learning Rate of 750

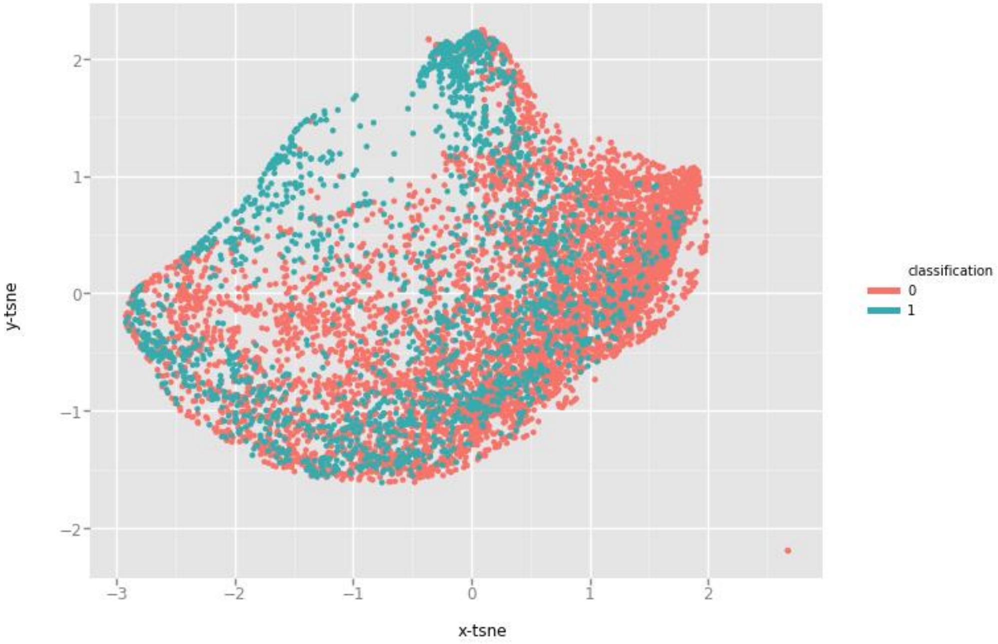

Figure 6.4 - t-SNE high sample representation: Perplexity = 15, No. of Iteration = 250, Learning Rate of 100

6.4.2.2 Results on low sample dataset

Principal Component Analysis Visualisation:

Once again, similar results are achieved for 1D visualisation (Figure 6.5) to that from Section Results on high sample dataset6.4.2.1, with similar representations. Figure 6.6 also shows similar a similar representation to its counterpart from Section 6.4.2.1. However, the relationship is not obvious with a low sampled dataset.

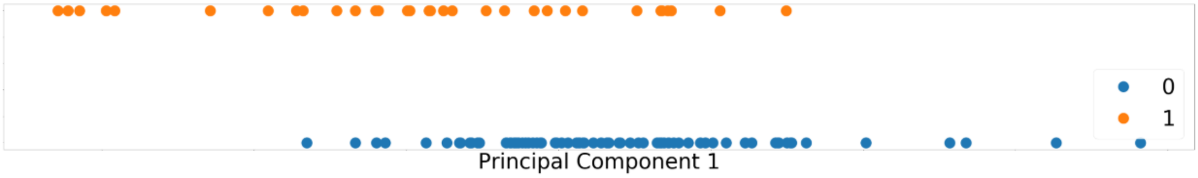

Figure 6.5 - PCA representation of the low sample data in 1-Dimension

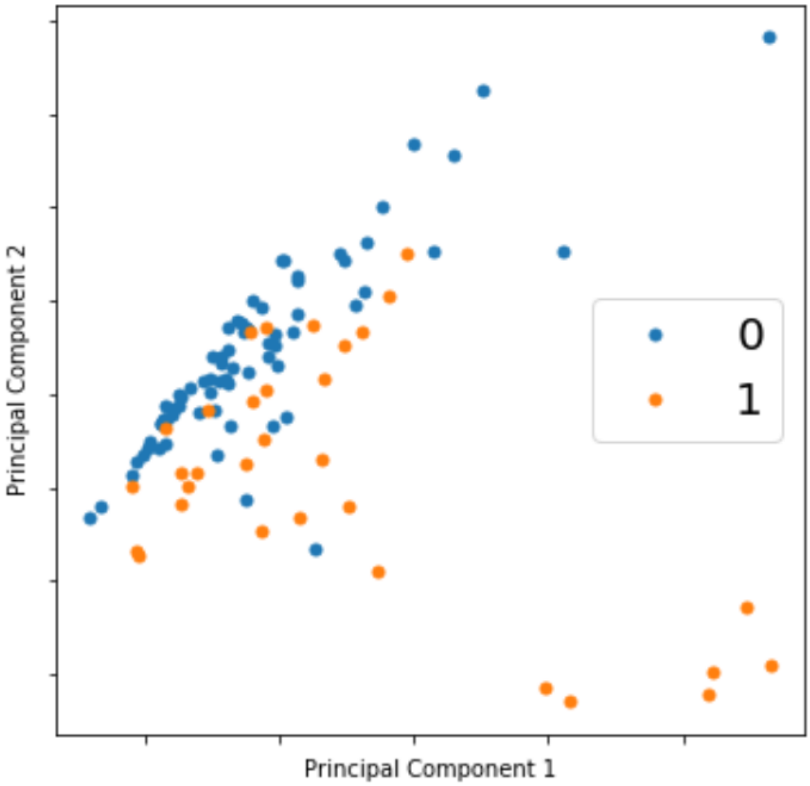

Figure 6.6 - PCA representation of the low sample data in 2-Dimensions

t-SNE:

The t-SNE representations shown in are once again poor, the stochastic nature, once again makes it difficult to gain any insight. The total time taken to train, for all 32 t-SNE representations was 32.75s. Even with the Student-t distribution, t-SNE can still be sensitive to outliers, as shown in Figure 6.8.

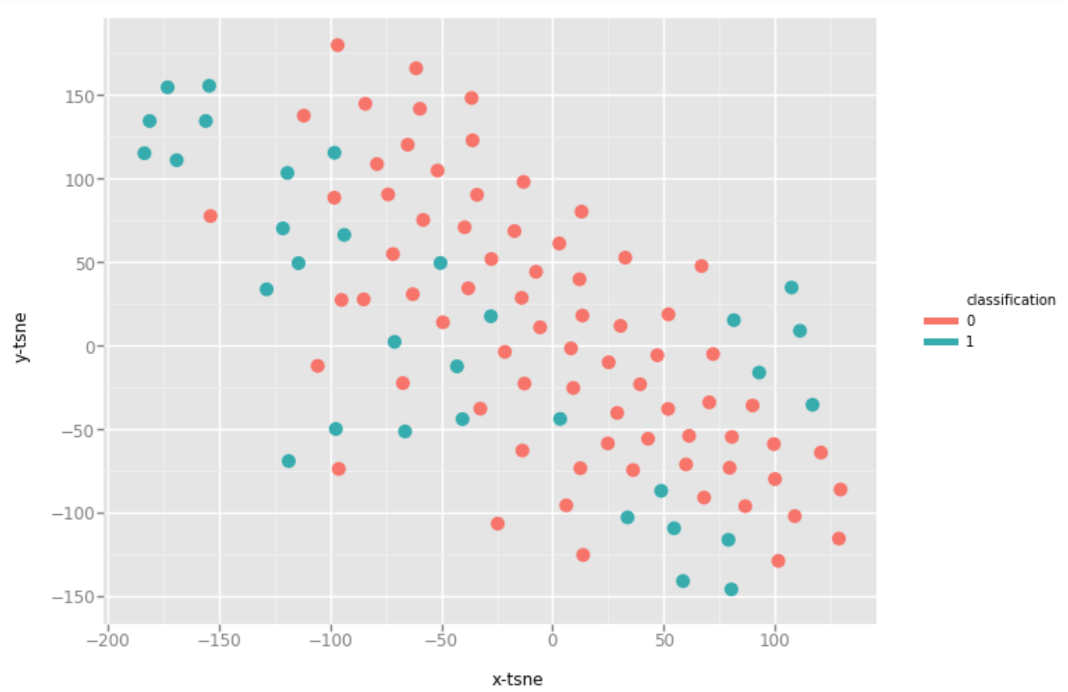

Figure 6.7 - t-SNE high sample representation: Perplexity = 45, No. of Iteration = 750, Learning Rate of 750

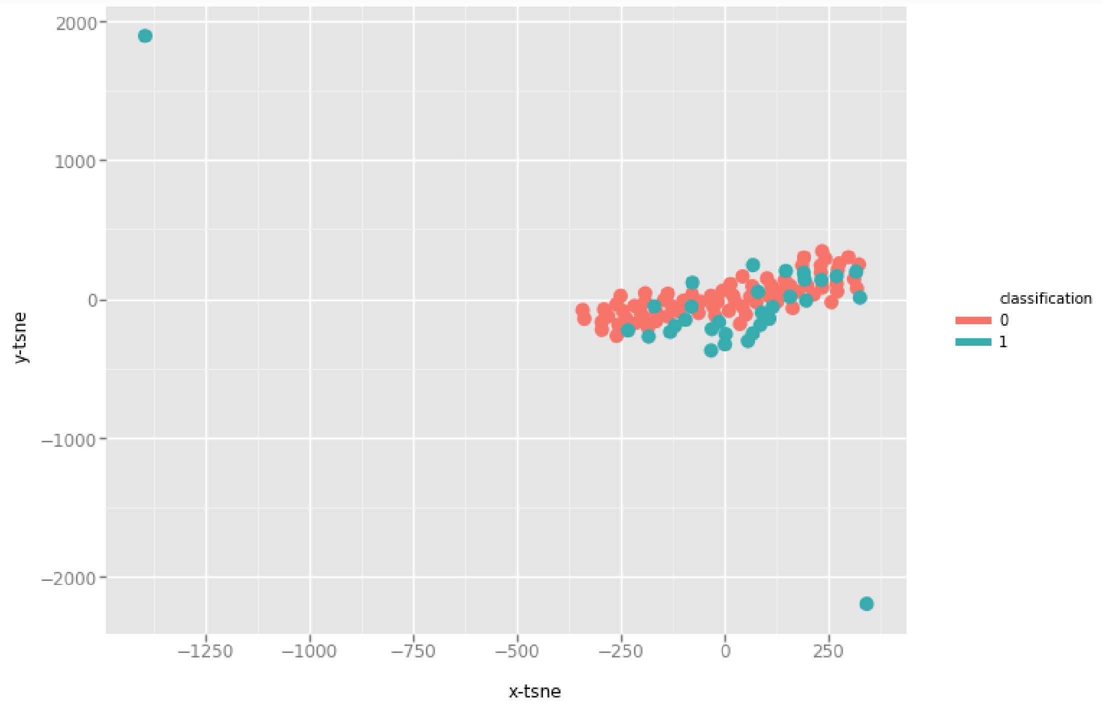

Figure 6.8 - t-SNE high sample representation: Perplexity = 45, No. of Iteration = 100, Learning Rate of 750

6.5 Assessment

This chapter presented an overview of two popular types of visualisation methods in the machine learning community: PCA and t-SNE. I have discussed the reasons why they were or were not successful.

Overall, visualisations can be difficult to produce, particularly when there is a small sample of data. PCA can create reliable visualisations, especially with large data, for use in comparing how closely related two individual data points may be based on their features. With the t-SNE approach it is hard to get good local and global structure because of the asymmetrical KL Divergence Feature.

Chapter 7: Conclusion and Future Work

7.1 Summary of Contributions

This project has provided analysis for dimensionality reduction techniques in three different forms: Feature Selection, Feature Extraction and Data Visualisation, that may be helpful in the creation of automated tools for the UDL platform.

Potential tools include an automatic PCA visualiser that plots into two dimensions with the upload of a dataset in a standardised form. Another possible tool includes an automated Feature Selection tool, which lists the most important features or returns a dataset for further analysis.

The performance of the dimensionality techniques has been measured by using ONS census data and forming a classification problem that aims to predict and target where there are shortages of skilled workers. To solve this problem, three Machine Learning algorithms and suitable evaluation metrics.

By undertaking this research project, I have been able to show, 1) it is possible to find quicker and more efficient ways of training data; 2) it is possible to create more accurate prediction models; 3) Data can be presented in a way that is easier to understand and gives some insight.

> **Chapter 2** introduces some general themes centered around this project and provides some background behind Machine Learning and Neural Networks. In this chapter, I delve into the topics of Open Data, Big Data, Urban Analytics, Census Data and the Resurgence of Machine Learning and discuss the potential of these fields, as well as what is holding them back.
> **Chapter 3** is the formulation of a classification problem, in a scenario where an institution is looking to deal with the problem of there being a shortage of skilled workers. There is also the choice and justification of Machine Learning algorithms and Evaluation Metrics.
> **Chapter 4** provides a thorough evaluation of Feature selection to solve the three objectives of the project, this is done by using three different techniques to reduce the data into more representative features.
> **Chapter 5** explores the use of Feature Extraction, to attain the first two objectives of the project, in this section PCA Feature Extraction and Neural Networks are once again evaluated based on similar metrics as those in Chapter 4.
> **Chapter 6** discusses the importance of effective visualisation and attempts to use t-SNE and PCA Visualisation on the ONS dataset to assess and evaluate the algorithms.

In conclusion, this thesis contributes to the UDL and the field of Urban Analytics in general, in a few ways.
Firstly, it provides ways to increase the speed of creating models, without a significant loss in accuracy through the evaluation of several Feature Extraction and Feature Selection. techniques. Secondly, the research investigates ways to prevent overfitting and increase model accuracy by use of Feature Extraction and Feature Selection. Finally, the thesis presents Visualisation and Feature Selection as ways to get better insights from the data.

7.2 Future Work

The research has pursued its objectives within the pre-defined scope successfully. There are many techniques such as Autoencoders, Non-negative Matrix Factorisation and Linear Discriminant Analysis that could be further explored. Research into other visualisation methods, that closely align to the UDL, such as layering over GIS Mapping may be examined, as well as Feature Extraction for High Spatial images. Future work may also involve the investigation of how live data pipelines can be reduced from Comprehensive Knowledge Archive Network (CKAN) for efficient processing of code.

Chapter 8: References

1. **Manley, Laura and Gurin, Joel.** *Open Data for Sustainable Development.* s.l. : World Bank Group, 2015.

2. **World Bank.** *World Bank Support for Open Data.* s.l. : World Bank, 2017.

3. *The Dawn of Big Data.* **IBM.** London : IBM, 2013.

4. *Feature Selection for High-Dimensional Data: A Fast Correlation-Based Filter Solution.* **Yu, Lei and Liu, Huan.** Tempe, AZ : Association for the Advancement of Artificial Intelligence, 2003.

5. *Information Analysis of High-Dimensional Data and Applications.* **Article ID: 126740, London : Hindawi, 2015, Vol. 2015.**

6. *Overfitting and undercomputing in machine learning.* Dietterich and Tom. New York, NY : ACM Computing Surveys (CSUR), 1995.

7. *Reducing Vector Space Dimensionality in Automatic Classification for Authorship Attribution.* Rico-Sulayes, Antonio. 3, Puebla, Mexico : Electronics, Automation and Communications Engineering Magazine , 2017, Vol. 38.

8. UCL. UCL UDL. *Urban Dynamics Lab: About Us.* [Online] UDL UCL. [Cited: 30 June 2018.] https://www.ucl.ac.uk/urban-dynamics-lab/about-us.

9. Likierman, Andrew. *Planning and Controlling UK Public Expenditure on a Resource Basis, Public Money & Management.* London : s.n., 2010.

10. Sarkar, Dipanjan (DJ). The Art of Effective Visualization of Multi-dimensional Data. *Medium.* [Online] Medium, 15 January 2018. [Cited: 3 July 2018.] https://towardsdatascience.com/the-art-of-effective-visualization-of-multi-dimensional-data-6c7202990c57.

11. Nomis. Nomis Official Labour Market Statistics. *Nomisweb.* [Online] Office for National Statistics. [Cited: 15 June 2018.] https://www.nomisweb.co.uk/census/2011.

12. *The 2010 Census: How It's Done and Why It Matters* . Smith, Stanley K. Austin, TX : Association for University Business and Economic Research, Austin, TX, 2009.

13. Nomis. List tables by Series or Release. *Nomis Web.* [Online] [Cited: 25 June 2018.] https://www.nomisweb.co.uk/census/2011/detailed_characteristics.

14. Deloitte. The story is mightier than the spreadsheet. *Deloitte.* [Online] [Cited: 2 July 2018.] https://www2.deloitte.com/insights/us/en/industry/public-sector/chief-data-officer-government-playbook/open-data-success-stories.html.

15. Principles. *Open Data Charter.* [Online] [Cited: 4 July 2018.] https://opendatacharter.net/principles/.

16. Global Open Data Index. Place Overview. *Global Open Data Index.* [Online] [Cited: 5 July 2018.] https://index.okfn.org/place/.

17. Bartels, Andrew. *Midyear Global Tech Market Outlook For 2017 To 2018.* s.l. : Forrester, 2017.

18. *IFAC-PapersOnLine: The Interplay Between Big Data and Sparsity in Systems Identification: Some Lessons from Machine Learning.* Cheng, Yongfang, et al. 28, 2018 : Elsevier, Vol. 48.

19. piesync. Top 5 Problems with Big Data (and how to solve them). *Piesync.* [Online] [Cited: 5 July 2018.] https://www.piesync.com/blog/top-5-problems-with-big-data-and-how-to-solve-them/.

20. Alan Turing Instititute. Urban analytics. *Alan Turing Instititute.* [Online] [Cited: 5 July 2018.] https://www.turing.ac.uk/research/interest-groups/urban-analytics.

21. Murray, Peter. Urban Analytics: A 21st Century Introduction to Spatial Analytics. *Carto.* [Online] [Cited: 6 July 2018.] https://carto.com/blog/urban-analytics-introduction-spatial-analytics/.

22. Mashariki, Amen Ra. THE EVOLUTION OF URBAN ANALYTICS. *Government Loop.* [Online] [Cited: 6 July 2018.] https://www.govloop.com/community/blog/evolution-urban-analytics/.

23. ONS. ONS CENSUS. *ONS.* [Online] [Cited: 9 July 2018.] https://www.ons.gov.uk/census/2011census.

24. —. Census Benefits. *ONS.* [Online] [Cited: 10 July 2018.] https://www.ons.gov.uk/census/2011census/2011censusbenefits.

25. Levy, Steven. INSIDE AMAZON'S ARTIFICIAL INTELLIGENCE FLYWHEEL. *Wired.* [Online] [Cited: 7 July 2018.] https://www.wired.com/story/amazon-artificial-intelligence-flywheel/.

26. Schaller, Robert R. *MOORE'S LAW: Past, Present and Future.* s.l. : IEEE Spectrum, 1997. 0018-9235.

27. Parloff, Roger. Why Deep Learning Is Suddenly Changing Your Life. *Fortune.* [Online] [Cited: 8 July 2018.] http://fortune.com/ai-artificial-intelligence-deep-machine-learning/.

28. Computer History Museum. Moore's Law. *Computer History Museum.* [Online] [Cited: 12 July 2018.] http://www.computerhistory.org/revolution/digital-logic/12/267.

29. Brandom, Russell. Why Facebook is beating the FBI at facial recognition. *The Verge.* [Online] [Cited: 8 July 2018.] https://www.theverge.com/2014/7/7/5878069/why-facebook-is-beating-the-fbi-at-facial-recognition.

30. Jackson, Keith R., Lavanya Ramakrishnan, Krishna Muriki, Shane Canon, Shreyas Cholia, John Shalf, Harvey J. Wasserman, and Nicholas J. Wright. *2010 IEEE Second International Conference on Cloud Computing Technology and Science (2010): Performance Analysis of High Performance Computing Applications on the Amazon Web Services Cloud.* Indianpolis, IN : IEEE, 2010. 978-0-7695-4302-4.

31. NI Direct Government Services. Qualifications: what the different levels mean. *NI Direct Government Services.* [Online] [Cited: 5 July 2018.] https://www.nidirect.gov.uk/articles/qualifications-what-different-levels-mean.

32. *Supervised Machine Learning: of Classification Techniques.* Kotsiantis, Sotiris. Patras : Emerging Artificial Intelligence Applications in Computer Engineering, 2007.

33. *An overview of classification algorithms for imbalanced dataset.* Ganganwar, Vaishali. 4, Pune : International Journal of Emerging Technology and Advanced Engineering, 2012, Vol. 2.

34. Monard, Maria C, Prati, Ronaldo C and Batista, Gustavo E.A.P.A. *Class imbalances versus class overlapping: an analysis of a learning system behavior.* Sao Paulo : University of S˜ao Paulo - Campus of S˜ao Carlos.

35. Scikit learn documentation 1.9.Naive Bayes. *Scikit learn.* [Online] [Cited: 10 july 2018.] http://scikit-learn.org/stable/modules/naive_bayes.html.

36. Rish, Irina. *An Empirical Study of the Naïve Bayes Classifier.* Hawthorne, NY : IBM, 2001.

37. *Comparative Study of K-NN, Naive Bayes and Decision Tree Classification Techniques.* Jadhav, Sayali D. and Channe, H P. 1, Pune : International Journal of Science and Research (IJSR) , 2016, Vol. 5. 2319-7064.

38. WÄRNLING, OSCAR and BISSMARK, JOHAN. *The Sparse Data Problem Within Classification Algorithms: The Effect of Sparse Data on the Naïve Bayes Algorithm.* Stockholm : KTH, 2017.

39. *Estimating continuous distributions in Bayesian classifiers.* John, John H. and Langley, Pat. San Mateo : Morgan Kaufmann Publishers Inc., 1995.

40. *Tackling the Poor Assumptions of Naive Bayes Text Classifiers.* Rennie, Jason D.M., et al. Cambridge, MA : ICML, 2003, Vol. 3.

41. *Random Decision Forests.* Ho, Tin Kam. Murray Hill, NJ : AT&T Bell Labroratories.

42. *The Random Subspace Methods for Constructing Decision Forests.* Murray Hill, NJ : Bell Labroatories, Lucent Technologies.

43. *Random Forests.* Breiman, Leo. 1, Berkeley, CA : Kluwer Academic Publishers, 2001, Vol. 45. 0885-6125.

44. Random Forest Simpliefied. *Medium.* [Online] Medium. [Cited: 15 July 2018.] https://medium.com/@williamkoehrsen/random-forest-simple-explanation-377895a60d2d.

45. Harrell, Frank E. *Binary Logistic Regression. In: Regression Modeling Strategies. Springer Series in Statistics.* s.l. : Springer, Cham, 2015. 978-3-319-19424-0.

46. Scikit learn. Scikit learn. *http://scikit-learn.org.* [Online] [Cited: 24 July 2018.] http://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html.

47. American Statistical Association. Past Winners. *Statistical Computing Statistical Graphics.* [Online] [Cited: 14 July 2018.] http://stat-computing.org/awards/jmc/winners.html.

48. Data Flair. XGBoost in Machine Learning – Features & Importance. *Data Flair.* [Online] [Cited: 29 July 2018.] https://data-flair.training/blogs/what-is-xgboost/.

49. Brownlee, Jason. A Gentle Introduction to XGBoost for Applied Machine Learning. *Machine Learning Mastery.* [Online] [Cited: 30 July 2018.] https://machinelearningmastery.com/gentle-introduction-xgboost-applied-machine-learning/.

50. —. Metrics To Evaluate Machine Learning Algorithms in Python. *Machine Learning Mastery.* [Online] [Cited: 31 July 2018.] https://machinelearningmastery.com/metrics-evaluate-machine-learning-algorithms-python/.

51. *Pattern Reognition: The use of the area under the ROC curve in the evaluation of machine learning algorithms.* Bradley, Andrew P. 7, s.l. : Elsevier, 1997, Vol. 30.

52. Sasaki, Yutaka. *The truth of the F-measure.* Manchester : University of Manchester, 2007.

53. Scikit learn. 1.1.11. Logistic regression. *Scikit Learn.* [Online] Scikit Learn. [Cited: 25 July 2018.] http://scikit-learn.org/stable/modules/linear_model.html#logistic-regression.

54. *Convergence Failures in Logistic Regression* . Allison, Paul D. 360, Philadelphia, PA : SAS, 2000.

55. Sci-kit Learn. 1.11.2. Forests of randomized trees¶. *Scikit learn.* [Online] [Cited: 24 July 2018.] http://scikit-learn.org/stable/modules/ensemble.html#forest.

56. XGBoost Developers. XGBoost Parameters. *XGBoost.* [Online] XGBoost. [Cited: 23 July 2018.] https://xgboost.readthedocs.io/en/latest/parameter.html#.

57. Lee, Muyueh. *Data Visualization.* s.l. : Muyueh Limited Company, 2014.

58. *An Introduction to Variable and Feature Selection.* Guyon, Isabelle and Elisseeff, Andre. 2003.

59. *The Feature Selection Problem: Traditional Methods and a New Algorithm.* Kira, Kenji and Rendell, Larry A. Urbana, Illionois : AAAI, 1992, Vols. AAAI-92 Proceedings.

60. *European Journal of Operational Research: A discrete particle swarm optimization method for feature selection in binary classification problems.* Unler, Alper. 3, 2010, Vol. 206.

61. Yang, Yiming and Pedersen, Jan O. *A Comparatice Study on Feature Selection in Text Categorization.* 1997.

62. *Proceedings of the 2012 SIAM International Conference on Data Mining: SOR: Scalable Orthogonal Regression for Non-Redundant Feature Selection and its Healthcare Applications.* Dijun Luo, Fei Wang, Jimeng Sun, Marianthi Markatou, Jianying Hu, Shahram Ebadollahi. s.l. : Society for Industrial and Applied Mathematics, 2012.

63. ***Bioinformatics: A review of feature selection techniques in bioinformatics.*** **Yvan Saeys, Iñaki Inza, Pedro Larrañaga. 19, s.l. : Oxford Academic, 2007, Vol. 23.**

64. ***IEEE Computer Society Conference: Feature selection for evaluating fluorescence microscopy images in genome-wide cell screens.*** **Kovalev, et al. 2006, Vol. 1.**

65. ***Airborne Lidar Feature Selection for Urban Classification Using Random Forests.*** **Chehata, Nesrine, Guo, Li and Mallet, Clement. 3, Paris : University of Bordeaux, GHYMAC Lab, 2009, Vol. 38.**

66. ***Remote Sensing of Environment: Object-based land cover mapping and comprehensive feature calculation for an automated derivation of urban structure types at block level.*** **Jena, Germany : Elsevier, Vol. 154.**

67. Learn, Scikit. Univariate Feature Selection. ***Scikit learn.*** **[Online] [Cited: 25 july 2018.] http://scikit-learn.org/stable/auto_examples/feature_selection/plot_feature_selection.html#sphx-glr-auto-examples-feature-selection-plot-feature-selection-py.**

68. Vlad, Cristi. CristiVlad25/ml-sklearn. ***Github.*** **[Online] [Cited: 10 July 2018.] https://github.com/CristiVlad25/ml-sklearn/blob/master/Machine%20Learning%20with%20Scikit-Learn%20-%2042%20-%20Automatic%20Feature%20Selection%20-%201.ipynb.**

69. —. ml-sklearn/Machine Learning with Scikit-Learn - 44 - Automatic Feature Selection - 3.ipynb. ***Github.*** **[Online] [Cited: 26 July 2018.] https://github.com/CristiVlad25/ml-sklearn/blob/master/Machine%20Learning%20with%20Scikit-Learn%20-%2044%20-%20Automatic%20Feature%20Selection%20-%203.ipynb.**

70. ***Robust Feature Selection Using Ensemble Feature Selection Techniques.*** **Saeys, Yvan, Abeel, Thomas and Van de Peer, Yves. Berlin : Springer, 2008. 978-3-540-87480-5.**

71. Scikit Learn. 1.13.3. Recursive feature elimination. ***Scikit Learn Documentation.*** **[Online] [Cited: 27 July 2018.] http://scikit-learn.org/stable/modules/feature_selection.html#rfe.**

72. ***Chemometrics and Intelligent Laboratory Systems: Recursive feature elimination with random forest for PTR-MS analysis of agroindustrial products.*** **Granitto, Pablo M., et al. 2, s.l. : Science Direct, Vol. 83.**

73. Abe, Shigeo. ***Feature Selection and Extraction.*** **London : Springer, 2010. 978-1-84996-097-7.**

74. ***An Introduction to Feature Extraction.*** **Guyon, Isabelle and Elisseeff, Andre. Berlin : Springer, 2006, Vol. 207.**

75. Liu, Huan and Motoda, Hiroshi. ***Feature Extraction, Construction and Selection: A Data Mining Perspective.*** **New York : Springer US, 1998. 0893-3405.**

76. Li, Haifeng, Tao, Jaing and Keshu, Zhang. ***Efficient and robust feature extraction by maximum margin criterion.*** **s.l. : NIPS, 2004.**

77. ***IEEE Transactions on pattern analysis and machine intelligence: KPCA plus LDA: a complete kernel Fisher discriminant framework for feature extraction and recognition.*** **Yang, Jian, Frangi, A.F. and Yang, Jing-Yu. 2, Nanjing : s.n., 2005, Vol. 27.**

78. Turk, Matthew A. and Pentland, Alex P. ***Face Recognition Using Eigenfaces.*** **Cambridge, MA : Vision and Modeling Group, The Media Laboratory: MIT, 1991.**

79. ***IEEE signal processing letters 9.2: Face recognition using kernel principal component analysis.*** **Joon, Kim Hang, Kim, Kwang In and Keechul, Jung. 2002.**

80. Nixon, Mark S. and Alberto, Aguado S. ***Feature extraction & image processing for computer vision.*** **s.l. : Academic Press, 2012.**

81. ***Proceedings of the workshop on Speech and Natural Language. Association for Computational Linguistics: Feature selection and feature extraction for text categorization.*** **Lewis, David D. 1992.**

82. ***Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing: Fog computing in healthcare internet of things: A case study on ecg feature extraction.*** **s.l. : IEEE International Conference, 2015.**

83. ***"Feature extraction and classification of blood cells for an automated differential blood count system." Neural Networks, 2001. Proceedings. IJCNN'01. International Joint Conference on. Vol. 4.*** **Ongun, Guclu, et al. s.l. : IEEE, 2001.**

84. Kopra, Toni, Mikko , Mäkipää and Mauri, Väänänen. ***"Mobile station and interface adapted for feature extraction from an input media sample.*** **2007. U.S. Patent 7,221,902,.**

85. Benz, Ursula C., Peter Hofmann, Gregor Willhauck, Iris Lingenfelder, and Markus Heynen. ***"Multi-resolution, object-oriented fuzzy analysis of remote sensing data for GIS-ready information.*** **s.l. : ISPRS Journal of photogrammetry and remote sensing 58, no. 3-4, 2004.**

86. Gong, Peng, Danielle J. Marceau, and Philip J. Howarth. ***"A comparison of spatial feature extraction algorithms for land-use classification with SPOT HRV data.".*** **s.l. : Remote sensing of environment 40:137-151 , 1992.**

87. Chapter 425. ***NCSS.*** **[Online] [Cited: 1 August 2018.] https://ncss-wpengine.netdna-ssl.com/wp-content/themes/ncss/pdf/Procedures/NCSS/Principal_Components_Analysis.pdf.**

88. Kröse, Ben, Ben Krose, Patrick van der Smagt, and Patrick Smagt. ***An introduction to neural networks 8th Edition.*** **Amsterdam : s.n., 1993.**

89. Stanford. CS231n: Convolutional Neural Networks for Visual Recognition. . ***Github.*** **[Online] [Cited: 5 August 2018.] http://cs231n.github.io/neural-networks-1/.**

90. Hagan, Martin T. , Demuth, Howard B. and Beale, Mark Hudson. ***Neural Network Design.*** **1996.**

91. Nielsen, Mikal. ***High-Dimensional Data Visualization.*** **Trondheim : Norwegian University of Science and Technology, 2017.**

92. *Processes of Creating Infographics for Data Visualization.* Mateusz, Szołtysik. s.l. : ISD, 2016.

93. Georges Grinstein, Marjan Trutschl, Urska, Cvek. *High-Dimensional Visualizations.* Amherst, MA : s.n., 2001.

94. Ultsch, Alfred and Morchen, Fabian. *ESOM-Maps: tools for clustering, visualization, and classification with Emergent SOM.* Marburg, Germany : University of Marburg, 2005.

95. *Intelligent Data Analysis: SOM-based data visualization methods.* Vesanto, Juha. 2, Helsinki : Elsevier, 1999, Vol. 3.

96. *viSNE enables visualization of high dimensional single-cell data and reveals phenotypic heterogeneity of leukemia. Nature biotechnology.* Amir EA, Davis KL, Tadmor MD, Simonds EF, Levine JH, Bendall SC, Shenfeld DK, Krishnaswamy S, Nolan GP, Pe'er D. s.l. : Nature Biotechnology, 2013, Vol. 6.

97. *Applications of Machine Learning and High-Dimensional Visualization in Cancer Detection, Diagnosis, and Management.* Mccarthy, John F., Kenneth A. Marx, Patrick E. Hoffman, Alexander G. Gee, Philip O'neil, M. L. Ujwal, and John Hotchkiss. 1020, New York, NY : Annals of the New York Academy of Sciences , 2004, Vol. 1.

98. *Visualisation and dimension reduction of high-dimensional data for damage detection.* Worden, K. and Manson, G. Kissimmee, FL : In 17th International Modal Analysis Conference, 1999.

99. *Trends in Ecoogy & Evoloution: Information visualisation for science and policy: engaging users and avoiding bias.* McInerny, G.J., Chen, M., Freeman, R., Gavaghan, D., Meyer, M., Rowland, F., Spiegelhalter, D.J., Stefaner, M., Tessarolo, G. and Hortal, J. 3, s.l. : Cell Press, 2014, Vol. 29.

100. *IEEE TRANSACTIONS ON VISUALIZATION AND COMPUTER GRAPHICS: Visual Exploration of Big Spatio-Temporal Urban Data: A Study of New York City Taxi Trips.* Ferreira, Nivan, et al. 12, New York, NY : s.n., 2013, Vol. 19.

101. *15th International Conference on Information Visualisation: Users' Responses to 2D and 3D Visualization Techniques.* Koramaz, Turgay Kerem and Gülersoy, Nuran Zeren. Istanbul : IEEE, 2011.

102. Principal Component Analysis: Explained Visually. *Setosa.* [Online] 12 August 2018. http://setosa.io/ev/principal-component-analysis/.

103. Articles - Principal Component Methods in R: Practical Guide. *Statistical tools for high-throughput data analysis.* [Online] [Cited: 7 August 2018.] http://www.sthda.com/english/articles/31-principal-component-methods-in-r-practical-guide/112-pca-principal-component-analysis-essentials/.

104. Van der Maaten, Laurens and Hinton, Geoffrey. *Visualizing Data using t-SNE.* Maastricht, Netherlands : Journal of Machine Learning Research 1, 2008.

105. Laurens Van der Maaten. t-SNE. *github.* [Online] [Cited: 12 August 2018.] https://lvdmaaten.github.io/tsne/.

106. Learn, Scikit. sklearn.manifold.TSNE. *Scikit Learn.* [Online] [Cited: 15 August 2018.] http://scikit-learn.org/stable/modules/generated/sklearn.manifold.TSNE.html.

107. *Toward Integrating Feature Selection Algorithms for Classification and Clustering.* Liu, Huan and Yu, Lei. Tempe, AZ : ASU, 2005.

108. *Statistics & Probability Letters: A note on margin-based loss functions in classification.* Lin, Yi. 1, s.l. : Elsevier, 2004, Vol. 68.

110. De Martino, Michael, Causa, Federico and Serpico, Sebastino B. *Classification of Optical High Resolution Images in Urban Environment Using Spectral and Textural Information.* Genova, Italy : Dept. of Biophysical and Electronic Engineering (DIBE), University of Genoa.