

KWAME NKRUMAH UNIVERSITY OF SCIENCE AND TECHNOLOGY



Predicting student course aptitude using Machine Learning
models

By:

Philemon Johnson Mawunya

4707615

A final year project proposal submitted to the Department of
Computer Science, Kwame Nkrumah University of Science
and Technology, Kumasi, Ghana

May 2019

ABSTRACT

The Field of Machine Learning over the past ten years has seen a major increase in popularity in mainstream media. It goes without a fact that this rise in popularity is due in part to the rise of ‘Big Data’, which is just another term for large datasets, without which Machine Learning models cannot be trained. Machine Learning can be defined as the intersection of computer science and statistics that brings a data driven, probabilistic approach to Artificial Intelligence. It is for this reason Machine Learning has become almost synonymous with the term “Artificial Intelligence”.

This project mainly focuses on the way machine learning can be leveraged in the field of Education. A Machine Learning Model has been developed for this purpose, using classification analysis to predict the potential course aptitude for new applicants to the undergraduate programme at the Kwame Nkrumah University of Science and Technology’s Department of Computer Science. This project seeks to demonstrate the potential application of Machine Learning to the field of Education.

First a Machine Learning model is trained on a dataset obtained from the Knowledge Media Institute of The Open University UK. This model is based on existing Machine Learning models and Artificial Neural Networks.

For purposes of demonstration, a RESTFUL API and Prediction system were developed and tested in a simulated environment to show feasibility.

Keywords: Machine Learning, statistics, data science, education, aptitude, Neural Networks

ACKNOWLEDGEMENT

To The Almighty Lord, who has shown me Grace, given me His Strength and Guidance throughout my work

I am also honoured to work with my supervisor, the esteemed Dr. Najim Ussiph, whose constant drive for perfection has kept me on my feet in order to produce the best I possibly can.

To my family, friends and especially Kennie who has given me emotional, physical and spiritual support in the course of this work, I say God bless you.

DECLARATION

Without any reservations, I hereby declare that we personally undertook the project submitted under the supervision.

JOHNSON PHILEMON MAWUNYA

DATE

SUPERVISOR

I declare that I have personally supervised the students undertaking the submitted project and confirm that the students have my permission to present it for further assessment.

DR. NAJIM UISSPH

DATE

TABLE OF CONTENTS

ABSTRACT	i
ACKNOWLEDGEMENT	ii
LIST OF FIGURES	vi
LIST OF TABLES	viii
Introduction	1
1.1 Background to the Subject area	1
1.2 Overview of the Subject Area	2
1.3 Problem Definition	4
1.4 Motivation for the project.....	4
1.5 General Objectives	4
1.6 Specific Objectives	5
1.7 Project Scope	5
1.8 Project activity plan	5
1.9 Summary and presentation of Project Report.....	7
1.10 Conclusion	7
Literature Review.....	8
2.1 Introduction	8
2.2 Technologies.....	8
2.3 Highlights of similar implementations	10
2.4 Review of similar implementations	11
2.5 Benefits and challenges of implementations	13
2.6 Research Issues.....	14
2.7 Future trends in Machine Learning	14
2.8 Summary.....	14
Methodology	15
3.1 Introduction	15
3.2 Architectural Design.....	15
3.3 Requirements Specification.....	17
3.4 UML Diagrams.....	18
3.5 Advantages and Limitations	23
3.6 Development Tools.....	24
System Implementation	25

4.1 Introduction	25
4.2 Model Construction	25
4.3 System Construction.....	29
Findings and Conclusions	37
5.1 Introduction	37
5.2 Model Testing.....	37
5.3 System Testing	38
5.4 Findings	39
5.5 Future works	39
5.6 Conclusions	39

LIST OF FIGURES

Figure 1.1: An illustration of alpha-beta pruning	1
Figure 1.2: The line of best fit for a polynomial regression algorithm	3
Figure 1.3: Proposed Gantt Chart for the project	6
Figure 2.1: The performance for C4.5 and 1-NN algorithms respectively	12
Figure 2.2: ROC curve for the SHRINK algorithm	12
Figure 2.3: Performance of the model using the three performance measures	13
Figure 3.1: The proposed methodology for Deep Learning	15
Figure 3.2: Architectural design for the proposed system	16
Figure 3.3: Database schema for students	19
Figure 3.4: Database schema for Users	19
Figure 3.5: Database schema for dataset	20
Figure 3.6: Sequence diagram for prediction function	21
Figure 3.6: Use case diagram for prediction system	22
Figure 4.1: An Example of One Hot encoding	27

Figure 4.2: A snapshot of the first 10 entries of the studentInfo.csv dataset	27
Figure 4.3: A snapshot of the dataset after data pre-processing	28
Figure 4.4: An example of a Random Forest	29
Figure 4.5: The landing page	30
Figure 4.6: The sign-in page	31
Figure 4.7: The register page	31
Figure 4.8: The homepage	32
Figure 4.9: The predictor	32
Figure 4.10: The predictor with a prediction recommendation	33
Figure 4.11: The report page	33
Figure 4.12: A list of previous predictions	34
Figure 4.13: The help page	34
Figure 4.14: Snapshot of the Student Class	35
Figure 4.15: Snapshot of the User Class.	35
Figure 4.16: A snapshot of the Storage console	36

LIST OF TABLES

Table 1.1: Proposed Action Plan for the project

6

1.2 Overview of the Subject Area

The proposed Machine Learning model seeks to leverage the predictive capabilities of Machine Learning. In essence, Machine Learning is a means of detecting discrete patterns in data. Machine Learning achieves this by performing one or more tasks, these tasks are classified into several categories:

- Supervised Learning, where the computer is presented both input and output, the Machine Learning model then generalises a relationship between them.
- Unsupervised learning, where only inputs are given, the computer is left to find the structure of the data itself
- Semi-supervised learning, which will be used in this project, where the computer is first given a complete set of inputs and outputs to learn, it is then trained on an incomplete set of inputs only.
- Active learning, is a very experimental kind of Machine Learning in which the computer selects what portions of the given input data to learn. It then learns the data by querying an information source for their corresponding output.
- Reinforcement learning, which is used for Machine Learning Implementations like self-driving cars, it uses the concept of weighted scores as feedback to train itself.

Machine Learning models are used in a variety of applications; these can be generally classified under the following:

- Classification, where inputs are divided into two or more classes, the computer must learn to classify new inputs into the stated classes. Most of the time, classification problems are tackled with supervised learning.
- Regression, which will be used in this project, continuous outputs are derived from discrete or continuous inputs. This kind of problem is solved with supervised learning and can be used for predictions
- Clustering, is very similar to classification problems. However, clustering has no predefined classes and so the computer is left to determine the classes on its own.
- Density estimation, finds the distribution of inputs in a sample space
- Dimensionality reduction, which will also be used in this project, maps inputs to a lower dimensional plane to simplify them.

Regression is a method of modelling a target value based on independent predictors. Regression is used as a method used for forecasting. In Regression, a number of already established algorithms are used to fit data. These tried and tested algorithms include the popular linear regression, and polynomial regression which fits data points to a curved line of best fit (Fig. 2). Lesser known regression algorithms like stepwise regression, which specifically deals with multiple independent variables which will also be used in the model proposed for the project.

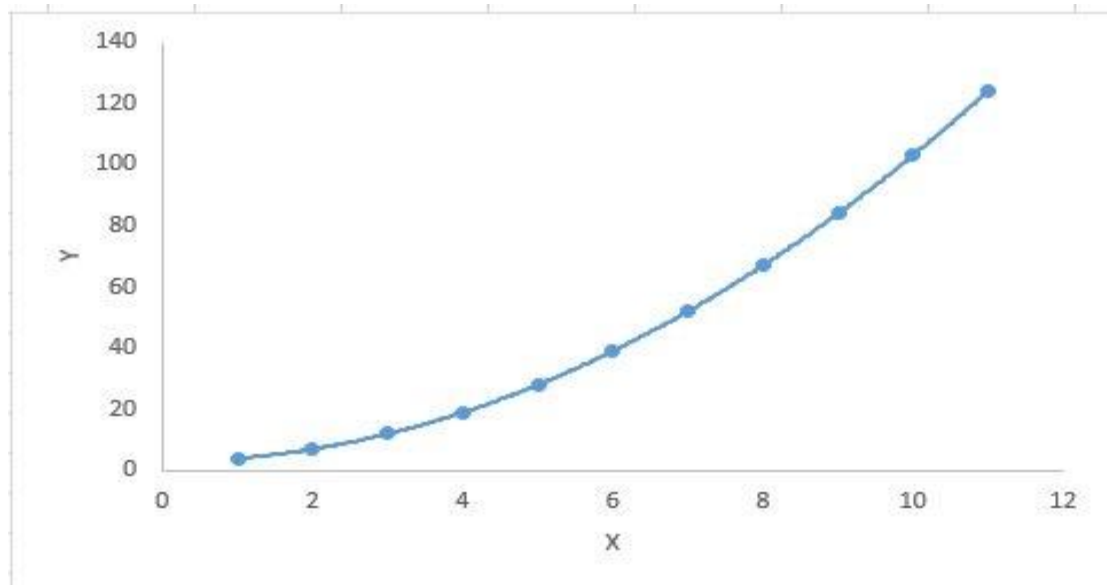


Figure 1.2: The line of best fit for a polynomial regression algorithm. Note the exponential slope of the line of best fit.[5]

Zhu said, “Semi-supervised learning uses both labelled and unlabelled data to perform an otherwise supervised learning or unsupervised learning task.” (Zhu, 2005)[3]. Semi supervised learning is relatively cheap compared to supervised learning and is more robust than unsupervised learning. The reason is that, in the real world, the data is mostly incomplete and so the best task will be to use supervised learning.

The Machine Learning model will be built with a functional and reflective programming language called Python, a Python framework for building Machine Learning models called Scikit-Learn will be used to build the model. Data visualization will be done through the Python Libraries Matplotlib and Scikit-Yellowbrick, a data visualization framework for Scikit-Learn. Data manipulation will be done using a Python mathematical library called Numpy.

serialization is the process of translating data structures or object state into a format that can be stored (for example, in a file or memory buffer) or transmitted (for example, across a network connection link) and reconstructed later (possibly in a different computer environment) [4]. Serialization in the Python Language is done through a third-party serialization library called Pickle.

A RESTFUL API is an Application Program Interface that uses HTTP request to transfer data through POST, GET, PUT and DELETE requests. A REST API is based off a programming architecture style called the REpresentational State Transfer (REST) Technology. The completed model has been serialised and an RESTFUL API was developed to interface with the model.

1.3 Problem Definition

The Department of Computer Science, College of Science, KNUST, over the years, has been using an archaic application process to select undergraduate students for their undergraduate programme, resulting in a varying degrees of success. Ideally, the department should rather have a steadily increasing pass rate year on year. In reality, this could be attributed to a number of factors including the archaic entry requirements that place emphasis on the sum of aggregates obtained in the West African Senior Secondary Certificate Examination (WASSCE). This leads to varying performance of the application process as the student course aptitude is not readily obtained from sum aggregates alone.

1.4 Motivation for the project

Having taken an interest in data science and Machine Learning, I started to look for ways in which Machine Learning can be leveraged to improve the education sector. One major problem in Machine Learning is the scarcity of readily available data in Ghana and that provides an extra challenge in this project. Machine Learning is the future of data science and analytics, and with this project, I hope to demonstrate a small portion of its immense capabilities if used well.

1.5 General Objectives

The proposed project aim is to improve the existing student application process in the Department of Computer Science by using a Machine Learning Model to predict final Cumulative Weighted Averages (CWAs) of applicants to the Department of Computer Science undergraduate program from applicant's West African Senior Secondary Certificate Examination (WASSCE) results thereby giving a measurement of student course aptitude for the programme.

1.6 Specific Objectives

The general objective will be achieved through the following objectives

- i. The creation of a base Machine Learning model for prediction
- ii. The creation of an Artificial Neural Network (ANN) for Evaluation
- iii. The training and testing of the base Machine Learning model and Artificial Neural Network
- iv. The Serialization of the Complete Machine Learning model
- v. The Development of A RESTFUL API to consume the serialized Model

1.7 Project Scope

This project will focus solely on the Department of Computer Science, specifically the application process of applicants to the Department of Computer Science undergraduate program.

1.8 Project activity plan

The project is estimated to take a total of 28 weeks to complete. A proposed Action Plan and Gantt Chart have been provided below.

Table 1.1: Proposed Action Plan for the project

Symbol	Task	Precedence	Estimated duration	Cost
A	Download, install and configure Machine Learning libraries	-	1 week	GHS 20
B	Research on optimal Machine Learning algorithms	-	2 weeks	NIL
C	Create And Configure base Machine Learning model	A,B	1 week	NIL
D	Generating the training data set	-	1 week	NIL
E	Train the model on the training data set	C	2 weeks	NIL
F	Refine and optimize the model	E	6 weeks	NIL
G	Generate the Test data set	-	1 week	NIL
H	Test the Refined Machine Learning model	F,G	2 weeks	NIL
I	Finalize and serialize Machine Learning model	H	2 weeks	NIL
J	Writing and reviewing final report	I	10 weeks	NIL
K	Create RESTFUL API to consume serialized model	I	2 weeks	NIL
L	Editing, Printing, binding and submission of final report	I,K	3 weeks	GHS 50

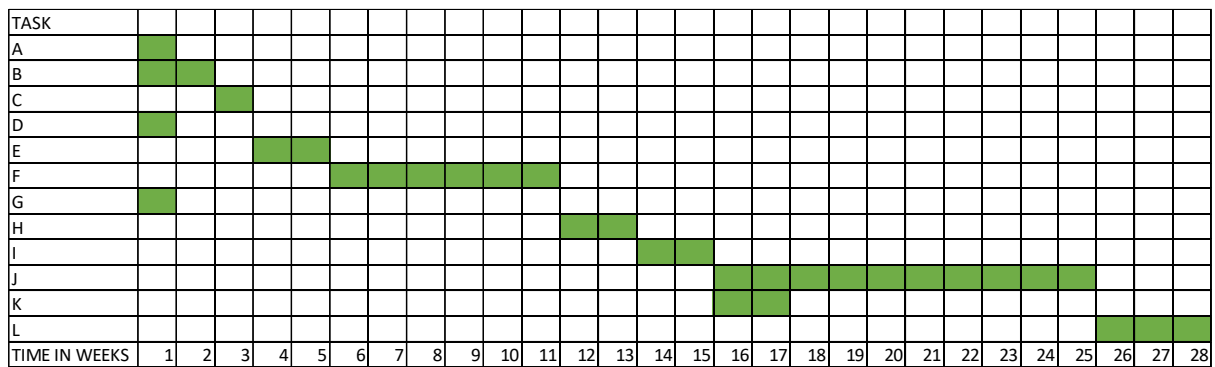


Figure 1.3: Proposed Gantt Chart for the project

1.9 Summary and presentation of Project Report

The following describes the structure of the report which will comprise of the contents of the various chapters in the system development.

Chapter one entails the overview of the proposed system, motivation for the proposed system and the objectives to be achieved at the end of the project. It also consists of the project scope as to who the proposed system applies to, activity plan and beneficiaries of this system.

Chapter two introduces review of similar systems where an overview of the subject area will be discussed. Past and present technologies that applies to the proposed system will also be discussed. This chapter will also include the review of existing implementations including highlights from similar vendors. The benefits and challenges of implementing the system will also be discussed.

Chapter three, the development methodologies and development tools that will be used in the design and implementation of the system will be looked at in detail. A brief description of the various methodologies including the advantages and limitations of alternate methodologies will also be considered.

Chapter five will introduce the implementation of the system. It will talk about the Machine Learning model and how it will be developed including the various methods of training the model. It also includes the RESTFUL API, its requirement specifications and how it will be developed and deployed.

Chapter six which is the final chapter will consist of findings and conclusion. This will include summary of various problems faced in project, achievements and challenges, recommendations, enhancements that can be made to the project in the future and a conclusion to the project.

1.10 Conclusion

In conclusion, This Machine Learning model is designed to be used by the Department of Computer Science to improve its application process. This will be used as a demonstration of the potential of Machine Learning in the Education sector.

CHAPTER 2

Literature Review

2.1 Introduction

Machine Learning has been used in a number of data heavy problems. In this chapter, a number of Machine Learning technologies used both in the past and present will be discussed. Implementations of various Machine Learning algorithms created from such technologies will also be discussed along with their merits and demerits. Current issues in research will be discussed as well as the benefits and challenges of the various implementations of Machine Learning algorithms.

2.2 Technologies

Machine Learning is classified generally into 5 broad categories; these are Supervised learning, Semi-supervised learning, Unsupervised learning, Active learning, and Reinforcement learning.

2.2.1 Supervised learning

Supervised learning is the machine learning task of learning a function that maps an input to an output based on example input-output pairs (Russel & Norvig, 2012) [1]. It uses training data which is made up of various unique examples to determine a function that can best fit that data and map new training or test data. In modern Machine Learning the training data can usually be divided into two parts namely; features and labels being inputs and outputs respectively. Machine Learning algorithms created within the scope of Supervised learning usually use inductive bias to generalise unseen test data to reach conclusions.

Supervised learning has a number of steps used to solve a given problem. First the type of training examples is determined. The kind and structure of data to be used is decided in this step. Next the training data is gathered from the appropriate sources. Dummy data can be generated in this step, however special care must be given in this case as wrongly generated training data may adversely affect the performance and accuracy of the Machine Learning model. Next the training data is divided into features and labels. A feature vector is produced in this step. The final step is the selection of the Machine Learning algorithm based on the structure of the training data, this allows for optimal performance of the Machine Learning model.

2.2.2 Semi-supervised learning

Semi-supervised learning is a variation of Supervised learning that makes use of unlabelled training data with a small amount of labelled data. Using unlabelled data along with a small amount of labelled data has brought about a record improvement in the learning accuracy of a Machine Learning model over other models like Unsupervised learning. Semi-supervised learning also reduces the cost of gathering labelled data as that requires a lot of time and skilled labour.

Semi-supervised learning may either be transductive or inductive. Transductive Supervised learning infers the correct label for given unlabelled data. Inductive Supervised learning infer the correct mapping from examples to labels.

In order to use unlabelled data, some assumptions must be made about the underlying structure of the data. Continuity of assumption states that “data points that are close together most likely share a label.”. The second assumption is Cluster assumption, which states “The data tend to form discrete clusters, and points in the same cluster are more likely to share a label”.

2.2.3 Active learning

Active learning uses learning algorithms that are able to interactively query a user to obtain desired outputs at new data points (Settles, 2010)[2]. Active learning is used when there is an abundance of unlabelled data but manually labelling that data is expensive, the algorithm actively queries the user for the labels iteratively.

In the case of Active learning, a number of query strategies are used for determining which data points should be labelled (Settles, 2010) [2]. Bouneffouf et al. (2014) proposed a sequential algorithm named Active Thompson Sampling which assigns a sampling distribution on the dataset, samples a point from that distribution and queries based on that sample [3]; this algorithm uses the balance exploration and exploitation query strategy that seeks to balance exploration of the data and exploitation of data. Another query strategy is Pool-Based Sampling, in this strategy, instances are taken form the entire dataset or pool and assigned a score which is a measure of how well the Machine Learning model understands the data. The Machine Learning model then selects the instance with the highest score and queries the user for labels [4]. An opposing query strategy will be Uncertainty sampling, which queries the user for the labels of data points of which the Machine Learning model is least certain of.

2.2.4 Unsupervised learning

Unsupervised learning learns from wholly unlabelled training data. Unsupervised learning identifies common features of the training data and reacts to the presence ir absence of such data in test data. Unsupervised learning is mainly used in the statistical field of density estimation [5]. Because of the unusual nature of Unsupervised Learning, algorithms that are

based of this technology try to mimic human logical patterns of searching for hidden patterns to analyse new data [6].

One approach to Unsupervised Learning is the method of moments. The method of moments relates unknown parameters in a Machine Learning model to the statistical moments of one or more random variables such that the unknown parameters can be estimated from the moments. The first and second moments are the arithmetic mean and the covariance of the random variable. In the case of higher order moments, mathematical constructs known as tensors are used to represent moments.

One of the most important examples of unsupervised Learning is the Neural Network study conducted by Canadian psychologist Donald Hebb (Hebb, 1949)[7]. The Self-Organising Map (SOM) and Adaptive Resonance Theory (ART) are used in unsupervised learning algorithms. The SOM organises inputs into a topographical map in which adjacent points represent inputs with similar properties. ART networks are used in a lot of pattern recognition tasks like seismic signal processing (Carpenter & Grossberg, 1988)[8].

2.2.5 Reinforcement learning

Reinforcement learning uses rewards and weights to guide a Machine Learning model's learning. Markov Decision Process (MDP) is used to represent basic Reinforcement Learning. A set of state agents, actions, a reward for a successful state change and rules that describe the model's observations are used. Rules are often random or Stochastic in nature. A reinforcement learning agent interacts with its environment in discrete time steps. An agent receives an observation which includes a reward. The goal of a Reinforcement learning model is to collect as much reward as possible by changing environment state using a set of predefined actions.

2.3 Highlights of similar implementations

Many implementations of Machine Learning Technologies exist, from engineering to healthcare, politics to social science, these Machine Learning models show off the power of Machine Learning in everyday life. However, there are currently no readily available examples of Machine Learning technologies being used in education. It is necessary however to demonstrate the benefits of Machine Learning, and for that matter, a number of successful implementations will be highlighted and reviewed in this chapter.

The first implementation is an oil spill detection model which uses satellite radar images for detection of oil spills in maritime borders (Kubat, Holte, & Matwin, 1998)[9]. This Machine Learning model was developed by Kubat et al. (1998) for Macdonald Dettwiler Associates and is implemented in the Canadian Environments Hazards Detection System (CEHDS) this model uses Supervised learning with a classification algorithm to identify potential oil spills and present them for man examination. This is a classic example of Machine Learning being harnessed to aid or improve an existing process as it does not seek to replace the existing manual oil detection system, but only to streamline it.

Menden et al. (2013) used Machine Learning to predict the sensitivity of cancer cells to drugs based on genomic and chemical properties [10]. Predictions were generated using Neural Network models for each defined drug to determine the IC_{50} profile across the panel of cell lines based on the genomic background of each cell, as characterised by a number of external factors. Necessary assumptions were made when training the Neural Networks and a number of performance measures were used including the Root Mean Square Error, which depicts the difference between predicted values and actual observed values. This kind of Machine Learning model utilises Unsupervised learning, which greatly benefits from the inference power of a Neural Network.

2.4 Review of similar implementations

Every Machine Learning project is a unique process that creates an equally unique model. All Machine Learning projects however, have common features or procedures that can be identified. In this chapter, two implementations were highlighted and will now have their data, methodologies and results will now be reviewed.

In the case of Kubat et al. (1998), one major problem that was determined was a scarcity of good data. This was due to the nature of the population of the problem set, where there was a large imbalance of negative samples (oil spill lookalikes) as compared to positive samples (actual oil spills). In their sample, the number of negative samples were as much as 96% of the total sample size of their data ($n=896$). Feature Engineering was utilized to prepare the data for the classification algorithm. Their data was organised into batches which further threatened the accuracy of their model, as there were great variations between batches of data and not enough variation of data within the batches. Performance measures of the Machine learning model was done using a Receiver Operating Characteristic (ROC) curve to find the accuracy of the algorithm. This performance measure however was used quite peculiarly as a tuning function for the specificity of the model due to the final product requiring the ability to arbitrarily set the accuracy of the model. The first algorithms used were the C4.5 and the 1-Nearest Neighbour (1-NN) algorithms.

Figure 2.1: The performance for the C4.5 and 1-NN algorithms respectively. The C4.5 algorithm clearly performs better than the 1-NN algorithm with increasing number of negative samples (Kubat et al., 1998)[9].

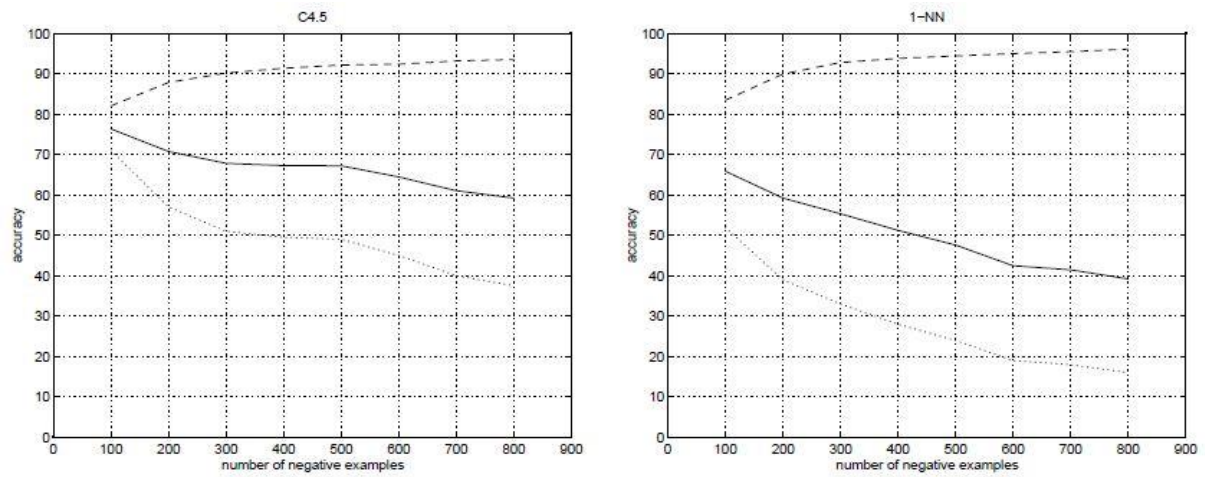


Figure 2.1: The performance for the C4.5 and 1-NN algorithms respectively. The C4.5 algorithm clearly performs better than the 1-NN algorithm with increasing number of negative samples [9].

A third algorithm was developed called SHRINK for the purpose of countering an imbalanced set of data. This algorithm had an accuracy of 70% but performance only declined marginally as more negative samples were added.

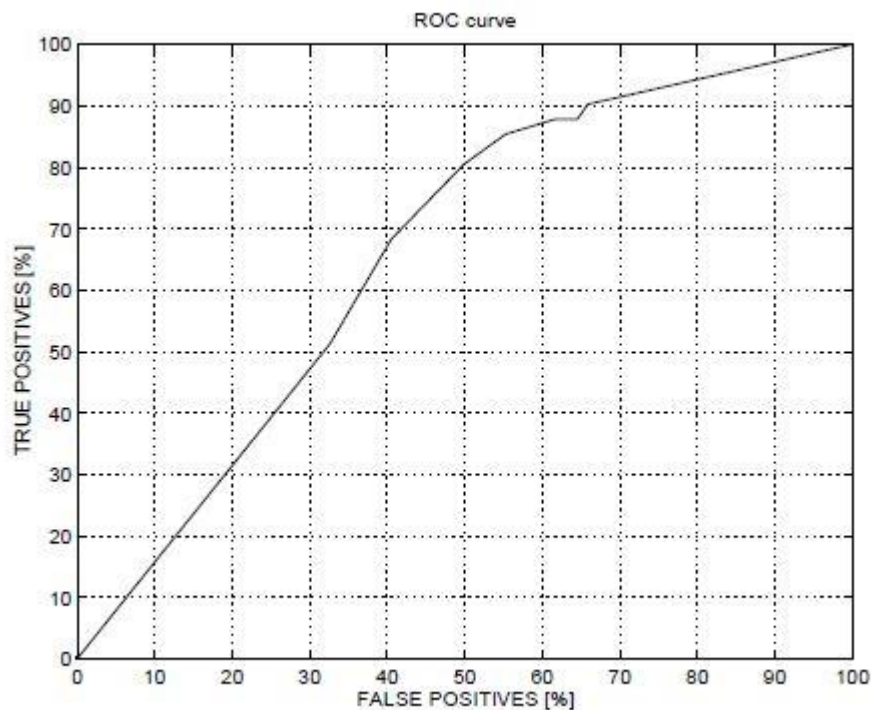


Figure 2.2: ROC curve for the SHRINK algorithm. Specificity changes along the gradient of the line [9].

Menden et al. (2013) were using Neural Networks to predict cancer cell sensitivity. Unlike Kubat et al., their data was very comprehensive as it was the drug screening dataset from the “Genomics of Drug Sensitivity in Cancer” (GDSC) project (Garnett et al., 2012)[11]. There was a total of 827 features comprising of 689 chemical descriptors of the given drugs, and 138 genome features that aid in developing the cell lines. This data set was a combination of the GDSC project data set and a data set that contained the chemical features of the drugs. This form of Feature Engineering was necessitated by the fact that the GDSC data set alone could not be used to predict cell sensitivity. Pearson correlation coefficient (R_p) which measures the linear correlation between two variables X and Y, coefficient of determination (R^2) which is the proportion of the variance in the dependent variable that is predictable from the independent variable and RMSE were the performance measures used in this project. The goal of the project was to create an accurate predictor of cell sensitivity to drugs; to achieve this, 8-fold cross-validation was performed; after which neural networks were used to compute and input missing IC_{50} values on the test set. The RMSE of the model had a staggering score 0.84 across all drugs. A score of 0.85 and 0.72 for R_p and R^2 were also recorded

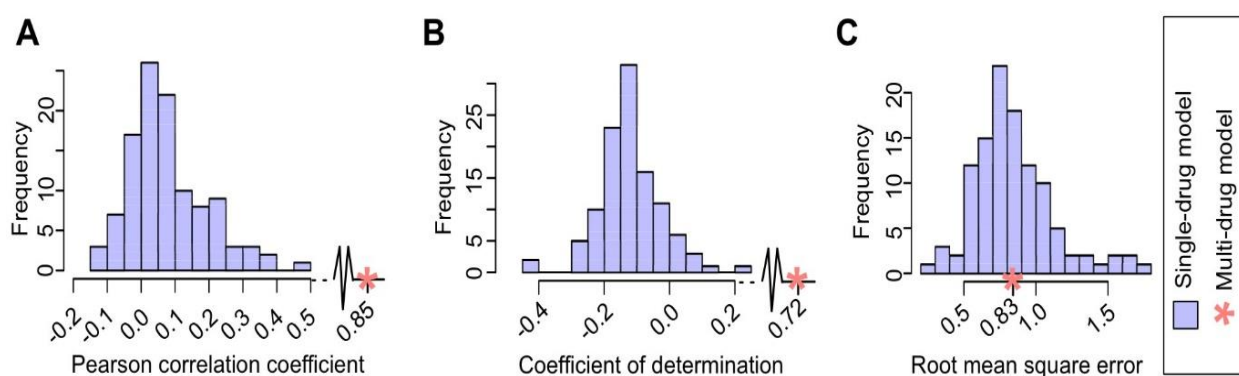


Figure 2.3: Performance of the model using the three performance measures [10].

2.5 Benefits and challenges of implementations

Kubat et al. described a number of challenges faced in the course of the project. The most critical challenge was the scarcity of data. An imbalanced training set could drastically reduce model accuracy without early detection and the data was unclean and could not be used by the classification algorithm, thus requiring extensive Feature Engineering. Another problem was that at the time of the project completion, there were no readily available algorithms that could cater for the imbalanced data. This was not so in the case of Menden et al. where readily available data sets were usable with a relatively small amount of cleaning and Feature Engineering done on them. However, assumptions were still made about the data during Feature Engineering and that could introduce human bias into the model. One of the main benefits of the implementation of Kubat et al. was the development of the SHRINK algorithm

which caters for imbalanced data sets. Menden et al. have now provided a key component of personalised cancer treatment which is expected to raise cancer remission rates and life expectancy of cancer patients.

2.6 Research Issues

The major hurdle in all machine learning research and projects is the scarcity of viable data. This is even more prevalent in education, healthcare, law enforcement and other fields which require a high level of privacy and confidentiality. Dietterich et al. (1997) suggests creating artificial data sets by modelling key characteristics of the required data as a workaround to this problem [12]. While there are a lot of questions about this workaround, it is quickly becoming a more popular alternative to real data with some data scientists supplementing inadequate data sets with artificial data. This could lead to a lot of bias and bad Machine Learning models and therefore a lot of care must be taken when creating artificial data sets.

Another issue is the need for assumptions during Feature Engineering. Without proper domain knowledge of the problem, feature engineering can produce very biased and unusable features that may hamper the performance of the Machine Learning model.

2.7 Future trends in Machine Learning

Machine Learning require a large amount of computing power to analyse large scale data. With the rise of cloud computing, Machine Learning facilities are now being modelled as cloud-based Software as a Service (SaaS) packages. This allows Machine Learning to be used more and more in everyday life.

The introduction of quantum computer also brings about another opportunity for Machine Learning. Quantum computing can greatly reduce the time taken to train Machine Learning models and with the rise in automated feature engineering, will greatly reduce the time taken to build Machine Learning models.

2.8 Summary

A number of Machine Learning implementations exist, each with varying levels of success. Two successful Machine Learning projects were Highlighted and Reviewed in detail. Machine Learning technologies were also discussed, as well as problems and challenges with the various implementations of Machine Learning technologies. Issues pertaining to research were discussed and Future trends in Machine learning were looked at.

CHAPTER 3

Methodology

3.1 Introduction

In this chapter, the project methodologies as well as the different Machine Learning technologies that will be used in this project will be stated and discussed. The various advantages and disadvantages of the Machine Learning model will be discussed as well as the various assumptions that will be made about the model. The various development tools that will be used for this project will also be discussed.

3.2 Architectural Design

Since the start of software engineering and the formulation of the Software Development Methodologies (SDM) framework in the 1960s, there have been a number of software development

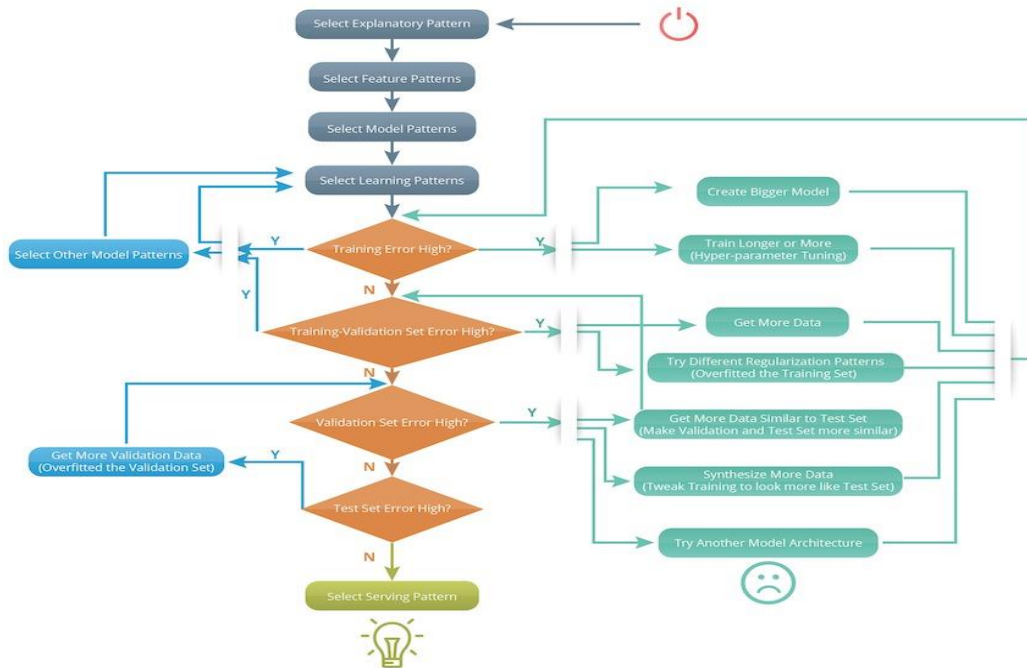


Figure 3.1: The proposed methodology for Deep Learning [1].

methodologies that have been proposed to streamline the software development life cycle. Of the various methodologies like waterfall, prototyping and incremental approaches to software development, projects that develop Machine Learning models generally use the agile approach to software development. A number of alternative development methodologies designed

specifically for Deep Learning projects have been proposed. One such methodology was proposed by Perez (2017) who insisted that Deep Learning was rich and complex enough to warrant its own development methodology [1].

In this project, the agile software development methodology will be used. In chapter 2, a number of Machine Learning methodologies were discussed, in this project, supervised learning will be used as well as the various algorithms that are used to train Machine Learning models.

The proposed system will be built using the 3-tier client-server architecture. A system that uses the client-server architecture is arranged as a set of servers and services, with clients using those services. The major components of this architecture include; a set of servers that offer services to other parts of the system, a set of clients that call on the services that are offered by the servers, and a network that allows clients and servers to communicate with each other. There may be multiple instances of clients and most client-server systems are implemented as distributed systems.

The proposed system is made up of a client-side application interface to the Machine Learning model, a server to handle data pre-processing and inference for report generation, and a Machine learning model with its own database to perform predictive analysis.

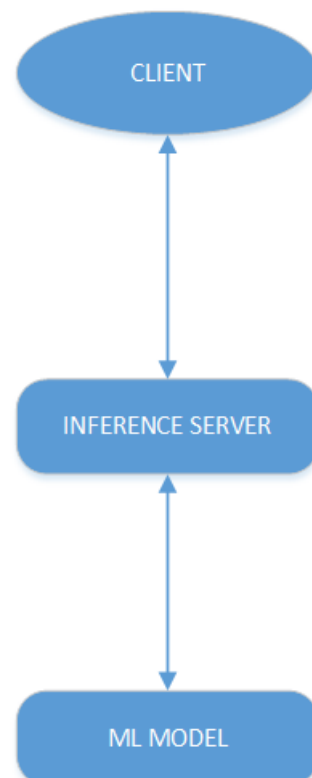


Figure 3.2: Architectural design for the proposed system. The client does not directly interface with the Machine Learning model.

3.3 Requirements Specification

Requirements specification is the process of stating the user requirements of a system in a clear, concise manner. The user requirements for a system should describe the functional and non-functional requirements so that they are understandable by system users who don't have detailed technical knowledge. Ideally, they should specify only the external behaviour of the system (Sommerville, 2011)[3].

3.3.1 Functional Requirements

The functional requirements of the system are as follows:

- The system shall authenticate the users of the system.
- A user may register to use the system.
- A user may make predictions on student course aptitude.
- A user may view previous predictions on student course aptitude.
- A user may delete previous predictions on student course aptitude.
- A user may view a report generated by the system based on the prediction made.

3.3.2 Non-functional Requirements

The non-functional requirements of the system are as follows:

- Security: A system is termed as secure if there are systems that may check authenticity of the user. Any information that is entered into the systems has to be done by an authenticated user. The system is also run on a local server, and accessed using a computer connected to the Department's Local Area Network
- Performance: The system is optimised for performance by caching predictions in a local database and storing generated reports in a cloud based storage facility for faster retrieval. Bulk processes are also shifted to an asynchronous task queue to prevent poor performance.
- Ease of use: The system provides an easy to use form for predictions, while handling all data processing in a server. The system also allows for bulk predictions via the uploading of a csv format file.
- Portability: the system uses the 3-tier client-server distributed architecture. This allows for portability, as many users can use the system anywhere, provided that they are connected to the department's Local Area Network.

3.4 UML Diagrams

Unified Modelling Language(UML), is the standard language for specifying, visualizing and documenting all the parts of a system. UML is a general purpose language that includes a graphical notation used to create an abstract model of a system, commonly referred to as a UML diagram or Model.

UML models emphasize the requirements of the system using objects, attributes, operations and relationships and the dynamic behaviour of the system by showing collaborations among objects and changes to the internal states.

Various UML diagrams have been created for this project. A few are shown below:

3.4.1 Database Schemas

A database schema is a tabular representation of a database in the system. It shows information like requirements and restrictions of the database. For this system, a total of three Database schemas were produced.


Student	
 PK	reference_id
gender	
disability	
highest_education	
imd_band	
region	
age_band	
previous_attempts	
mathematics	
science	
english	
physics	
emathematics	
prediction	

Figure 3.3: Database schema for students, this is the information stored by the system about each student after prediction has been made.


User	
 PK	id
username	
password_hash	

Figure 3.4: Database schema for Users. This schema consists of the username and passwords of the various users of the system.


Student Dataset	
 PK	id_student
	gender
	region
	highest_education
	imd_band
	age_band
	num_of_prev_attempts
	studied_credits
	disability
	Mathematics
	Science
	English
	Physics
	Elective Mathematics
	final_result

Figure 3.5: Database schema for dataset. This schema consists of what information was collected about each student in order to make the Machine Learning Model.

3.4.2 Sequence Diagrams

A sequence diagram shows the interactions between objects in a time sequence. They depict objects involved in a scenario and depicts the sequence of messages exchanged between the objects needed to carry out the scenario.

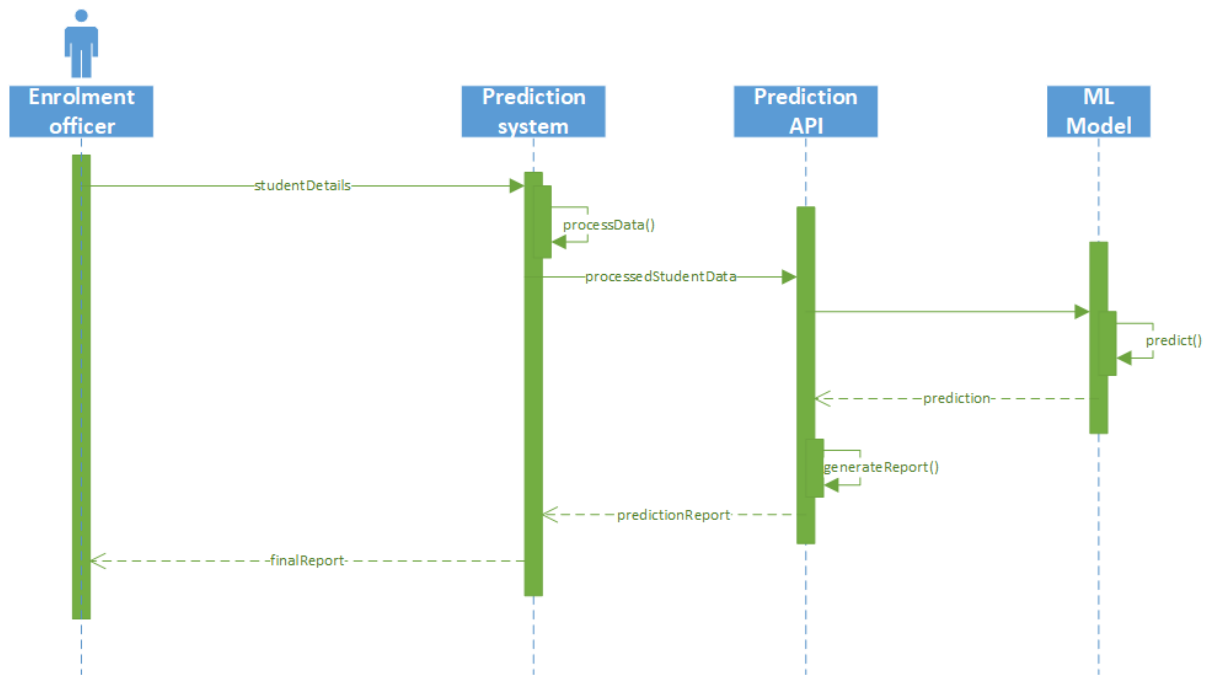


Figure 3.6: Sequence diagram for prediction function. This sequence diagram shows the communication path flow between the various objects in the system.

3.4.3 Use case Diagrams

A use case diagram captures the piece of functionality that a system provides. A use case diagram is the functional description of a system and its major processes. It provides a graphic description of who will use the system and what kind of interactions to expect within that system.

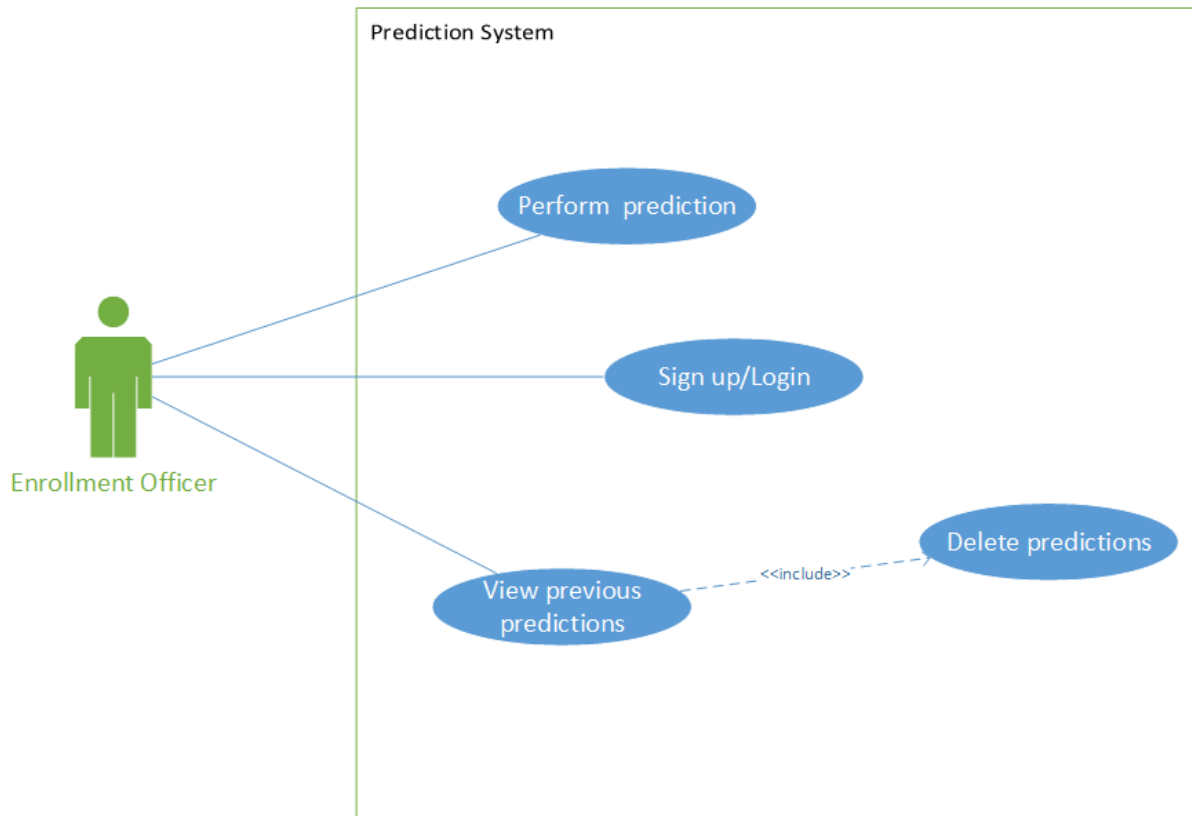


Figure 3.7: Use case diagram for prediction system. The use case shows the various services the enrolment officer can perform with the system.

3.5 Advantages and Limitations

There are many advantages of Machine Learning as well as demerits. Only advantages and disadvantages pertaining the machine Learning model created in this project will be discussed.

One advantage of the proposed system is the ability to loosely predict a student's final Cumulative Weighted Average(CWA), based on a number of factors like entry grades, schools attended, category of the Schools attended and other factors. This predictive power is essential to the determination of the student's course aptitude.

Another advantage of the project is the analysis of trends in the department's application process. A product of Exploratory Data Analysis, this trend analysis will use visualizations of data to detect and examine various hidden trends as well as suggestions that can be given on the data.

A major limitation of the Machine Learning model is the lack of an explanation facility. Unlike in Expert Systems, Machine Learning models do not use an inference engine for logical conclusions. It is due to this fact that an explanation facility cannot be built into a machine learning model as there is no logical inference to explain. Another major limitation is the use of assumptions in this project, assumptions play a big role in all Machine Learning projects [1]. This could lead to a potentially bad Machine Learning model that would not perform well in real world instances.

The principal advantage of the 3-tier client-server architecture is that it can be distributed across a network, allowing for multiple users of the system at once.

A major disadvantage of the 3-tier client-server architecture is that each service is a potential point of failure in the system. Also because of the propagation on a network, performance may be unpredictable.

3.6 Development Tools

The proposed Machine Learning model will be written in Python using an already established Machine Learning Framework Called Scikit-Learn.

All data will be handled by the Python mathematics libraries Numpy and Pandas. All data visualizations will be done through another library called Matplotlib and a Machine Learning data visualization framework called Scikit Yellowbrick, as well as a data analysis engine called Orange.

In computer science, in the context of data storage, serialization is the process of translating data structures or object state into a format that can be stored (for example, in a file or memory buffer) or transmitted (for example, across a network connection link) and reconstructed later (possibly in a different computer environment) [2]. Serialization in the proposed project will be done using a python serialization tool called Pickle.

The Restful API that will be developed to consume the Machine Learning model will be constructed using a python web framework called Flask.

CHAPTER 4

System Implementation

4.1 Introduction

System implementation is the process of developing a system based on the structure created during the system design phase of the software development life cycle. System implementation involves the coding of the individual functional units of a software. The system must be developed based on the functional user requirements stated in the system design phase

This chapter is about the implementation of the Machine Learning model.

4.2 Model Construction

The Machine Learning model was constructed using the python framework called Scikit-Learn. Scikit-Learn is a free software machine learning library that features various classification, regression and clustering algorithms like Support Vector Machines (SVM), random forests, gradient boosting and k-means. Performance of algorithms is optimised with Cython, which is a C based python wrapper.

4.2.1 The Dataset

The Dataset was acquired from the Knowledge Media Institute of The Open University, UK.

It is an anonymised Open University Learning Analytics Dataset (OULAD). This dataset consists of seven files stored in the CSV format, containing data about courses, students and their interaction with a virtual learning environment for seven selected courses.

For the purpose of this project, demographic information as well as student results was taken from the StudentInfo.csv file. The data set is made up of the following columns.

- Code_module: this is the identification code of the presentation module for which the student is registered.
- Code_presentation: the identification code of the specific presentation for which the student is registered
- Id_student: A unique identification number for each student.
- Gender: the sex of the student

- Region: the geographic region where a student taking the presentation module lived while taking the presentation module
- Highest_education: the highest educational level achieved by the student on entry to the presentation module
- IMD_band: this specifies the Index of Multiple Deprivation band of the place where the student lived during the presentation module
- Age_band: For anonymity, the ages of each student was binned into three ranges or bands.
- Num_of_previous_attempts: the number of times the student has attempted the presentation module.
- Studied_credits: the total number of credits for the modules the student is currently studying.
- Disability: this indicates whether the student is suffering from a disability
- Final_result: this is the students final result in the presentation module.

The model was trained on the following columns: gender, region, highest_education, IMD_band, age_band, num_of_previous_attempts and disability as features, and final_result as the Target Variable.

For the region, 14 specific regions in the UK were identified. These were the East Anglian Region, East Midlands Region, Ireland, London Region, North Region, North Western Region, Scotland, South East Region, South Region, South West Region, Wales, West Midlands Region and Yorkshire Region. Each student lived in one of the aforementioned regions. There were five major levels of education that students belonged to; these were: No Formal Qualifications, Lower than A Level, A level or Equivalent, Higher Education Qualifications, Post Graduate Qualification. The Index of Multiple Deprivation, is the official measure of the relative deprivation for small areas in England. There are 32,844 small areas in England, and the IMD ranks them from 1 (most deprived) to 32,844 (least deprived). The index, combine the information gathered from seven domains to produce the overall relative measure of deprivation; these are the Income deprivation, Employment deprivation, Education skills and training deprivation, Health Deprivation and Disability, Crime, Barriers to Housing and Services and Living Environment Deprivation (Deprivation & Area, 2016)[1]. The IMD rankings are however very long, and thus an aggregate was developed to simplify the IMB rankings, the 32,844 small areas were binned into 10 different bands or ‘deciles’, from the 10% most deprived areas, to the 10% least deprived areas [1]. The ages of the students were binned into three bins, namely 0-35 years of age, 35-55 years of age and 55 years or older. This was done for the purpose of anonymity. Individual disabilities as well as the degree of disability were not stated; only the presence of a disability was stated. The final_result for each student was also discretized into four bins; withdrawn, fail, pass and distinction.

4.2.2 Data pre-processing

A quick snapshot of the dataset reveals that most of the data is categorical in nature. The gender, region, highest_education, IMD_band, age_band and disability features, as well as the final_result target variable, were all string variables. Data pre-processing is necessary for the preparation of the model as Machine Learning models can only understand numerical data.

Categorical data represents types of data which may be divided into groups. Ordinal data is categorical data that can be ranked in order of Increasing value, nominal data is data that as no intrinsic rank to them. The highest_education, IMD_band, age_band as well as the final_result value are ordinal categorical variables, while the gender, region and disability variables were nominal.

Data pre-processing for categorical data involves transforming it into a numerical representation. Ordinal categorical data can be transformed using Ordinal encoding techniques, while Nominal categorical data can be transformed using techniques such as One Hot encoding. One Hot encoding encodes categorical integer features as a one-hot numeric array. The encoder takes m unique features from a list of n features and constructs an nxm array with binary

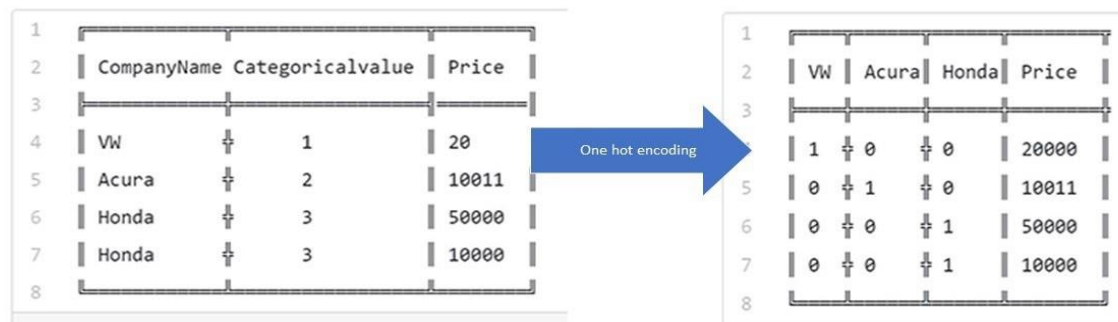


Figure 4.1: An Example of One Hot encoding. An nxm binary array is produced from the first list.

encoding.

Ordinal encoding takes m unique features from a list of n features and assigns an ordered number to each unique feature. This allows the Machine Learning model to understand the order between each feature.

Other pre-processing transformations include Standardization of data, that centres data by removing the mean value of each feature and then scaling it by dividing each feature by their standard deviation [2]. The aim of standardization is to improve the performance of models. Making all features equal so that no one feature has dominance in representation.

The gender, region, and disability features were One Hot Encoded using Scikit-Learn's inbuilt pre-processing OneHotEncoder, while ordinal encoding was performed on the imd_band, highest_education, age_band, and final_results features through pandas' map function.

code_moc	code_pres	id_student	gender	region	highest_education	imd_band	age_band	num_of_prev_att	studied_cr	disability	final_resul
AAA	2013J	11391	M	East Anglian Regi	HE Qualification	90-100%	55<=	0	240	N	Pass
AAA	2013J	28400	F	Scotland	HE Qualification	20-30%	35-55	0	60	N	Pass
AAA	2013J	30268	F	North Western Re	A Level or Equivalent	30-40%	35-55	0	60	Y	Withdrawn
AAA	2013J	31604	F	South East Region	A Level or Equivalent	50-60%	35-55	0	60	N	Pass
AAA	2013J	32885	F	West Midlands Re	Lower Than A Level	50-60%	0-35	0	60	N	Pass
AAA	2013J	38053	M	Wales	A Level or Equivalent	80-90%	35-55	0	60	N	Pass
AAA	2013J	45462	M	Scotland	HE Qualification	30-40%	0-35	0	60	N	Pass
AAA	2013J	45642	F	North Western Re	A Level or Equivalent	90-100%	0-35	0	120	N	Pass
AAA	2013J	52130	F	East Anglian Regi	A Level or Equivalent	70-80%	0-35	0	90	N	Pass

Figure 4.2: A snapshot of the first 10 entries of the studentInfo.csv dataset.

F	M	Disability/	Disability/	highest_ec	imd_band	East Angli	East Midla	Ireland	London Re	North Reg	North Wes	Scotland	South East	South Regi	South Wes	Wales	West Midl	Yorkshire	age_band	num_of_p	final_resul
0	1	1	0	3	10	1	0	0	0	0	0	0	0	0	0	0	0	0	2	0	2
1	0	1	0	3	3	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	2
1	0	0	1	2	4	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0
1	0	1	0	2	6	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	2
1	0	1	0	1	6	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	2
0	1	1	0	2	9	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	2
0	1	1	0	3	4	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	2
1	0	1	0	2	10	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	2
1	0	1	0	2	8	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2

Figure 4.3: A snapshot of the first 10 entries of the studentInfo.csv dataset after data pre-processing.

4.2.3 Machine Learning Model

The machine learning model was created using the Scikit-Learn python framework for Machine Learning. The model was trained on the Pre-processed Dataset CleanData2.csv. The performance metrics used to evaluate the Machine Learning model were the coefficient of determination (R^2 score) and the mean squared error.

The coefficient of determination is the proportion of the variance in the dependent variable that is predictable from the independent variable(s). R^2 is a statistical measure of how close the data are to the fitted regression line. This metric is specially for predictive statistical methods to test the outcomes of such models. A key limitation to R^2 is that the metric cannot determine whether the coefficient estimates and predictions are biased.

This model was trained using the Support Vector Classification (SVC) algorithm, which belongs to a class of algorithms called the Support Vector Machines (SVM) and A Random Forest Classifier. The SVM range of algorithms seek to draw a hyperplane, separating two or more groups of points. The SVC Algorithm does this by trying to maximize the distance between sample points and the hyperplane. A Random Forest Classifier is a meta estimator that take a number of decision tree based classifiers and trains them on unique subsets of the provided dataset, using averaging to improve the predictive accuracy of the model as well as to prevent overfitting.

The model's predictive functions were serialized for later use by the API using a python library called pickle.

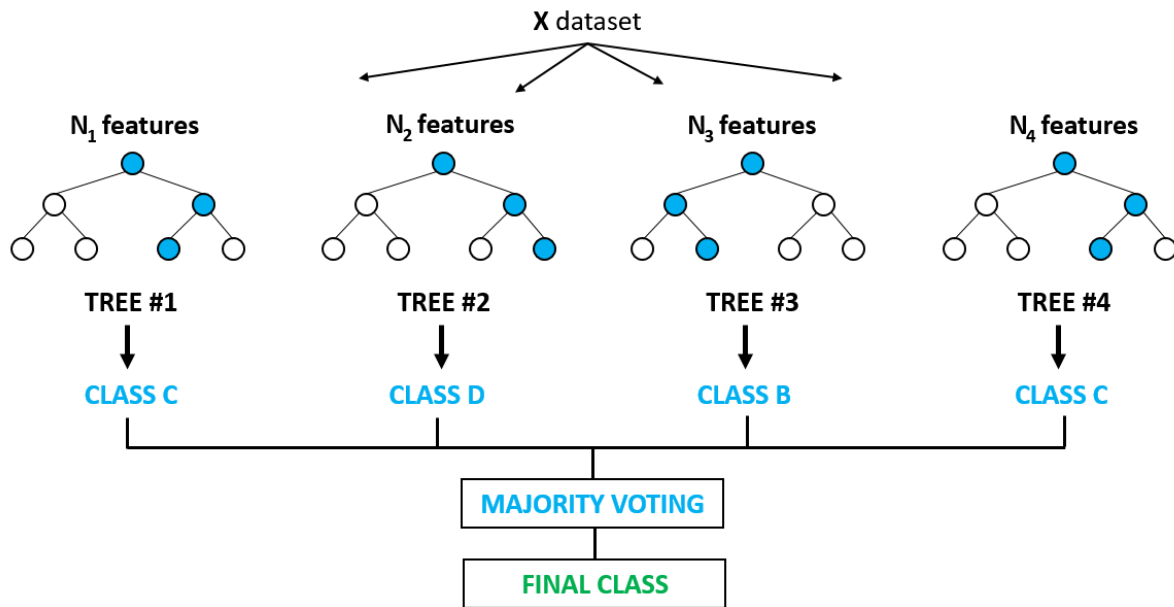


Figure 4.4: An example of a Random Forest. the individual Decision Trees in the "forest" are averaged to improve accuracy.

4.3 System Construction

The system is constructed using the 3-tier client-server architecture model. The system is divided into two separate servers, The Inference server and the Machine Learning model as well as the client-side interface.

4.3.1 Client Interface and Server

The client interface was built using a data driven frontend framework developed for python called Dash. Dash is a framework for building analytical web applications. A python web framework called Flask served as a wrapper for the dash frontend interface, the Flask wrapper enables user authentication, the viewing of previous predictions and viewing of reports.

The Flask wrapper contains a server for client interactions, the server also stores all user information in an SQLite Database, and performs all user authentication tasks as well as interfacing with the inference server.

The client interface server is broken into separate functional units called routes. Routes are endpoints in a web server which can be hit to perform a specific task. A total of six of such routes are used for the client interface server. The root route `'/'` renders the home page of the client interface, `'/login'` renders the login page of the client interface as well as handles all user authentication, validation and session handling. The `'/logout'` route handles the logging out of the user, the `'/register'` route handles user registration as well as server side validation. The `'/list'` route renders the view of all previous predictions made as well as handles search. The `'/help'` route renders the help page.

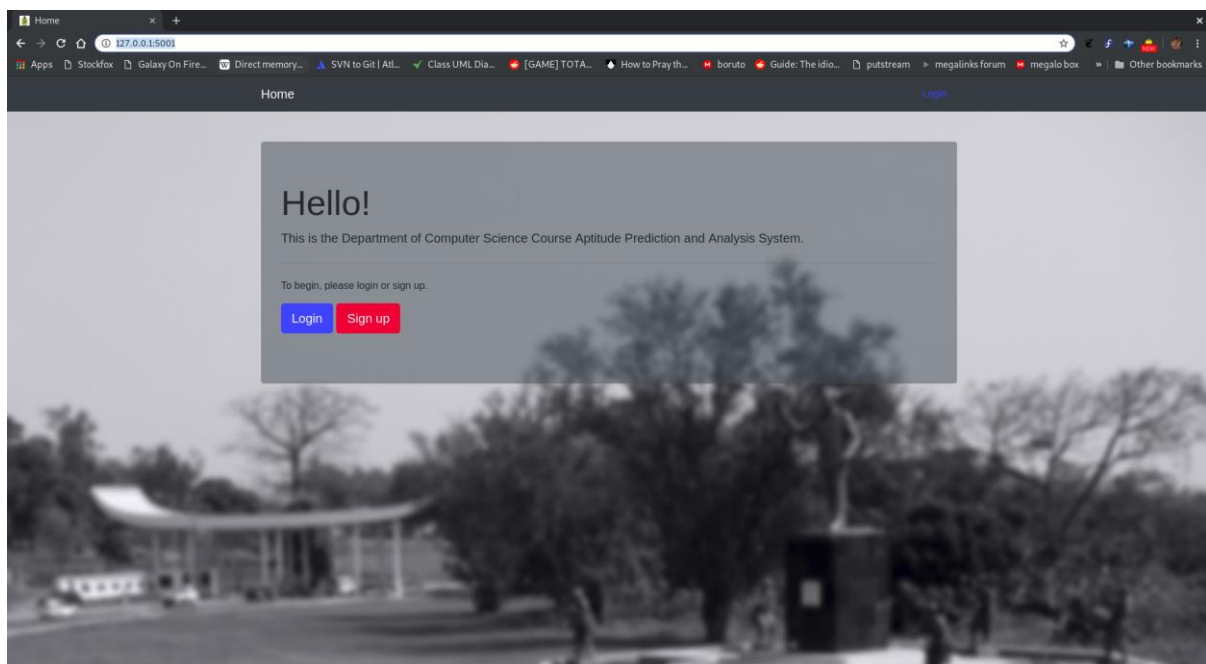


Figure 4.5: The landing page. this is the page that is used to login or sign up.

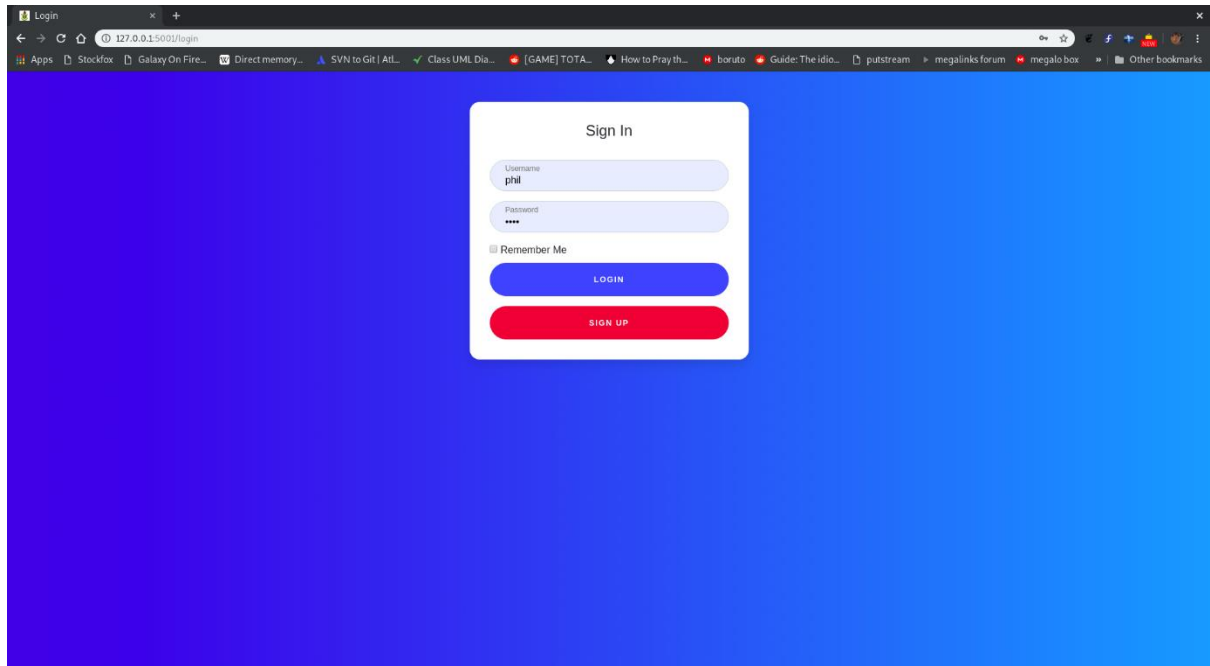


Figure 4.6: The sign-in page. This page takes user input for login.

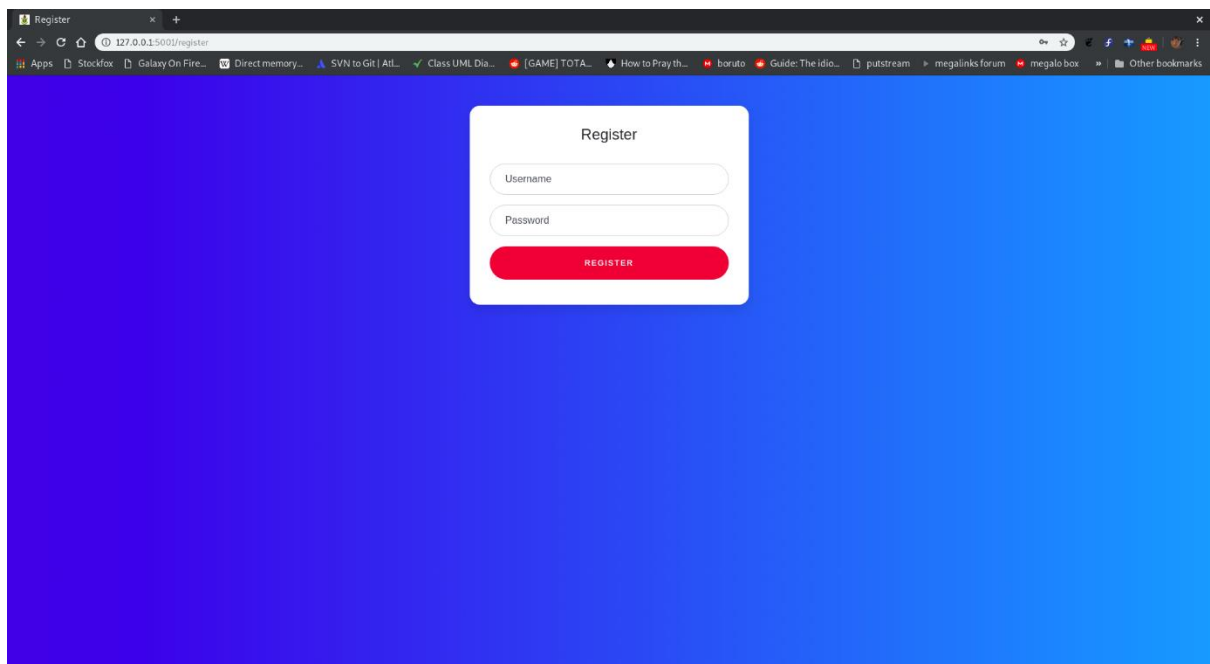


Figure 4.7: The register page. this is page allows users to register to use the system.

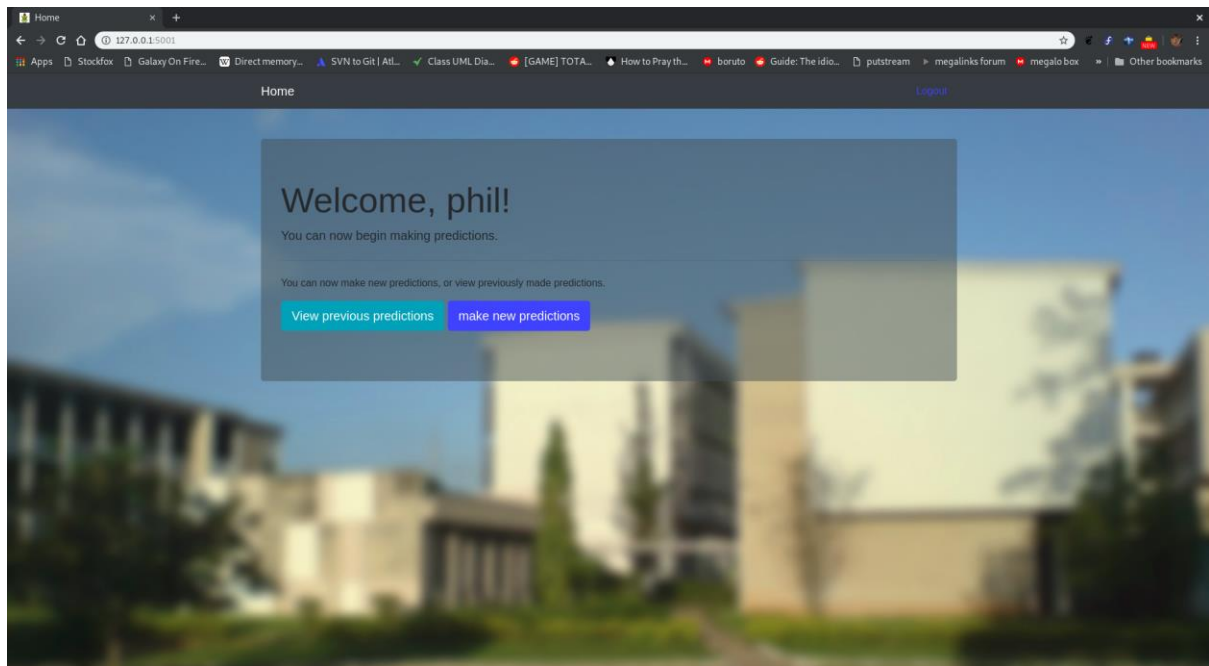


Figure 4.8: The homepage. From here, a user can make new predictions and view previous predictions.

Figure 4.9: The predictor. The predictor allows a user to make a prediction on course aptitude.

The screenshot shows the Predictor web application interface. On the left, a form contains the following information:

- Reference ID: 1245 (with a green checkmark)
- High School: 0WASS
- Gender: ☐ Female, ☒ Male
- Region: London (dropdown menu)
- Qualifications: No Formal Qualifications (dropdown menu)
- Disability: ☒ No, ☐ Yes
- IMD Band: A slider between 'Most Deprived' and 'Least Deprived'.
- Age Range: A slider between '0-35 years', '35-55 years', and '55 years or more'.
- GCE Grades:

Mathematics: 23 (green checkmark)	English: 43 (green checkmark)	Elective Mathematics: 56 (green checkmark)
Science: 46 (green checkmark)	Physics: 33 (green checkmark)	

On the right, a message states: "The Student with the ID 1245 is **NOT RECOMMENDED** for this course". Below this message is a blue button labeled "View Report".

Figure 4.10: The predictor with a prediction recommendation.

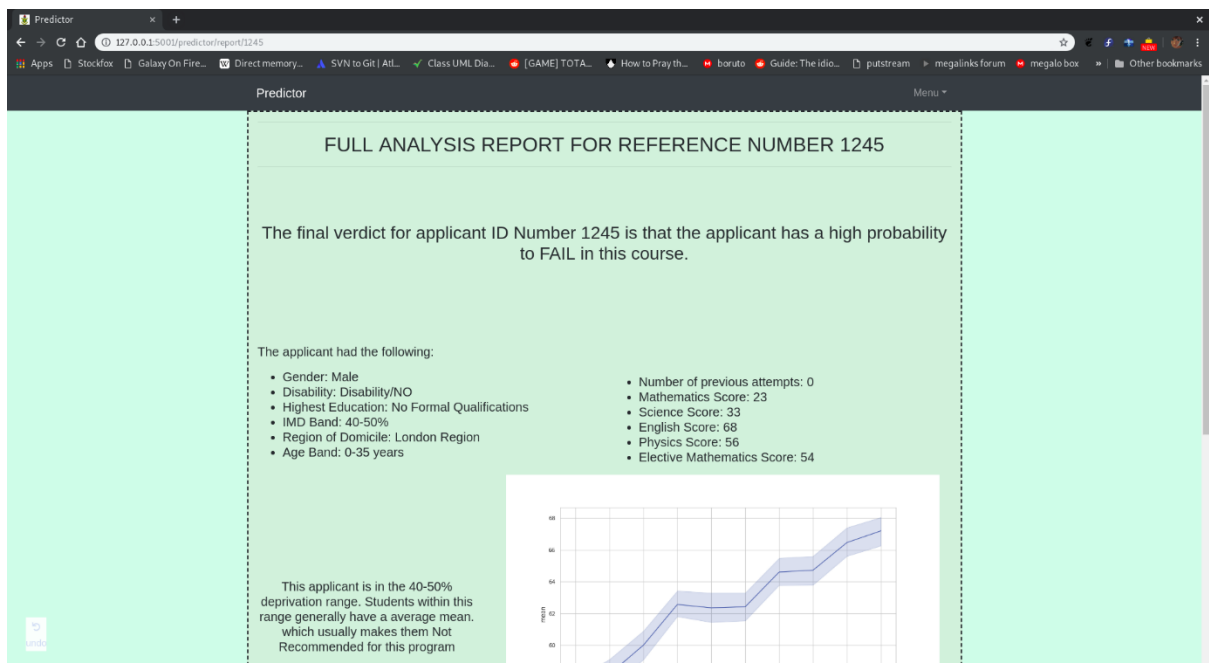


Figure 4.11: The report page. This page shows a report on the prediction made on a student.

Reference ID	High School	Gender	Disability	Highest Education	IMD Band	Region	Age Range	Previous Attempts	Mathematics	Science	English	Physics	Elective Mathematics	Prediction	Action
1245	M	Achmota	NO	No Formal Qualifications	5	London Region	0-35 years	0	23	33	68	56	54	Fail	View Report Delete
2039	M	Adisadel	YES	A level or Equivalent	1	Wales	0-35 years	0	98	90	79	80	98	Distinction	View Report Delete
2355	M	GHANASS	NO	No Formal Qualifications	5	London Region	0-35 years	0	67	75	89	96	67	Pass	View Report Delete
3452	M	GIS	NO	Higher Education Qualification	10	London Region	35-55 years	0	76	56	86	74	80	Pass	View Report Delete
5665	M	OWASS	NO	No Formal Qualifications	5	London Region	0-35 years	0	45	34	76	76	45	Fail	View Report Delete
65487	M	Achimota	NO	No Formal Qualifications	5	London Region	0-35 years	0	56	78	78	45	56	Pass	View Report Delete

Figure 4.12: A list of previous predictions. Search functionality has been implement for data retrieval.

Help

- » [What is this system?](#)
- » [Who can use this system?](#)
- » [How to use this system](#)

This system relies on data collected about the applicant from the application form. The following fields are required to be filled properly for the system to make predictions.

Reference ID: for identification of the student
 Gender of the student
 Disability: Whether the student is disabled or not
 Region of domicile: picked from a list 16 preselected regions
 Age band: Ages are banded into 3 tiers. Younger than 35 years, 35 to 55 years, and older than 55 years
 IMD band: the index of of multiple deprivation band rsng of the students domicile. See 'link ' for more
 The 5 raw scores for the following subjects: mathematics, science, english, physics, elective mathematics

Upon the collection of above data, the data is passed to the predictor.
 The predictor then makes one of the following recommendations: Not recommend, Recommended and Highly Rec

A report can then be viewed about the student in detail.

- » [Caveats](#)
- » [Disclaimer](#)

Figure 4.13: The help page. This page shows the user how to use the system.

4.3.2 Inference Server

The inference server handles all input passed to it from the Client interface and Server; this includes all inputs into the predictor made by a client, the server can then make predictions on the data provided by the client inputs using the Machine Learning model. The inference server is also capable of making inference based on the predictions made by the Machine Learning model, as well as storing the prediction data in a SQLite database handled by an Object Relational Mapper (ORM) called SQLAlchemy, which uses a class based approach to database management.

```
class Student(db.Model):
    reference_id = db.Column(db.Integer(), primary_key=True)
    school = db.Column(db.String(), nullable=True)
    gender = db.Column(db.String(), nullable=False)
    disability = db.Column(db.Integer(), nullable=False)
    highest_education = db.Column(db.String(), nullable=False)
    imd_band = db.Column(db.Integer(), nullable=False)
    region = db.Column(db.String(), nullable=False)
    age_band = db.Column(db.String(), nullable=False)
    previous_attempts = db.Column(db.Integer(), nullable=False)
    mathematics = db.Column(db.Integer(), nullable=False)
    science = db.Column(db.Integer(), nullable=False)
    english = db.Column(db.Integer(), nullable=False)
    physics = db.Column(db.Integer(), nullable=False)
    emathematics = db.Column(db.Integer(), nullable=False)
    prediction = db.Column(db.Integer(), nullable=False)
```

Figure 4.14: Snapshot of the Student Class. An ORM is used to handle database operations

```
class User(UserMixin, db.Model):
    id = db.Column(db.Integer, primary_key=True)
    username = db.Column(db.String(64), index=True, unique=True)
    password_hash = db.Column(db.String(128))

    def set_password(self, password):
        self.password_hash = generate_password_hash(password)

    def check_password(self, password):
        return check_password_hash(self.password_hash, password)

    def __repr__(self):
        return '<User {}>'.format(self.username)
```

Figure 4.15: Snapshot of the User Class.

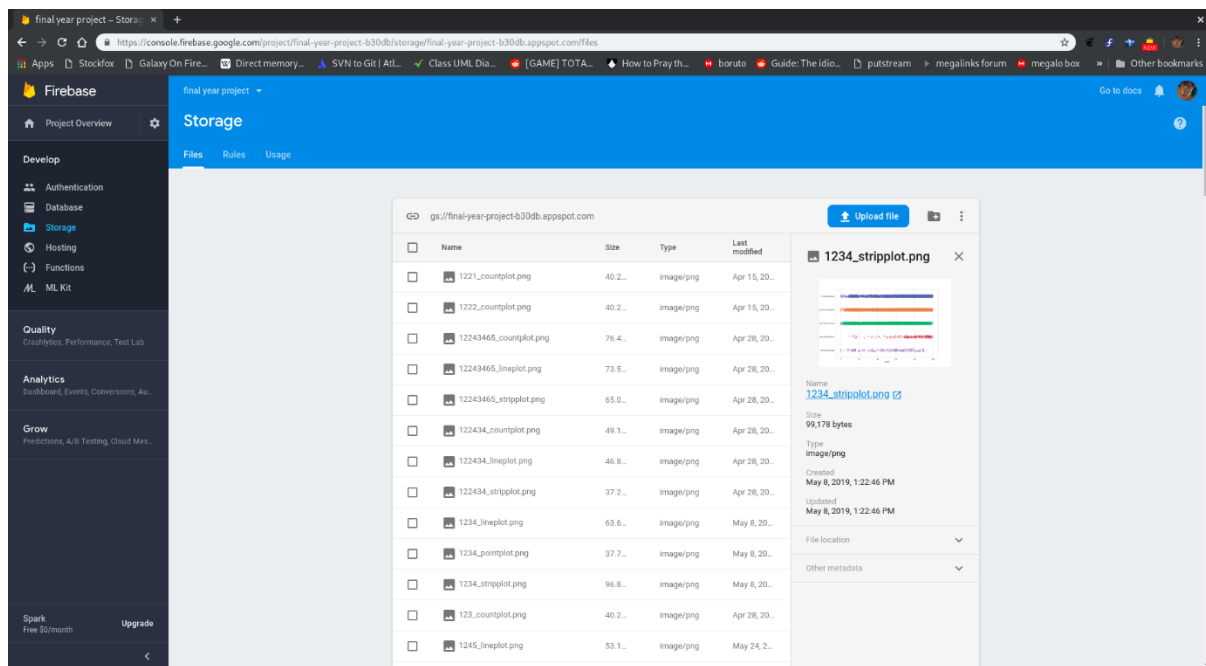


Figure 4.16: A snapshot of the Storage console. Graphs pertaining to each unique student is stored in Google Cloud storage

4.3.3 The Machine Learning model

The Machine Learning model after development must be deployed in the inference server. To that end, a serialization python library called pickle was used to deploy the model. The model was first serialized and the deployed. Upon every prediction, the model is deserialized and predictions are made.

CHAPTER 5

Findings and Conclusions

5.1 Introduction

Software testing is an investigation conducted to provide stakeholders with information about the quality of the software product or service under test (Jorgensen & Jorgensen, 2019)[1]. Software testing can also allow a business to quantify the risk involved in software implementation. Testing involves the process of executing a program with the intent of finding errors or software bugs in the system. Testing also involves verifying that the system is ready to use.

In software testing, the system is evaluated to check if it meets the Functional user requirements stated during the software design phase. Many levels of testing occurring during the testing phase; the first is unit testing which takes each functional unit of a system and tests it independent of all other units. Upon the successful completion of the unit tests, the system then undergoes integration or system testing; in this level of testing, the entire system is tested as a whole to check whether all functional units of the system are integrated correctly.

In this chapter, the Machine Learning Model, as well as the proposed system will be tested, findings, conclusions and future work will also be discussed in this chapter.

5.2 Model Testing

The Machine Learning model was constructed using the python framework called Scikit-Learn, and was trained using a Random Forest Classification algorithm. The model was trained on 70 percent of the obtained dataset and rest of the dataset was used for testing purposes. The Machine learning model had a prediction accuracy score of 64.7 (approximated to 65%) percent accuracy. This falls within the accepted accuracy range.

5.3 System Testing

The system is constructed using the 3-tier client-server architecture model. The system is divided into two separate servers, The Inference server and the Machine Learning model as well as the client-side interface. All interfaces are tested separately and then integrated testing is performed.

5.3.1 Unit Testing

In computer programming, unit testing is a software testing method by which individual units of source code, sets of one or more computer program modules together with associated control data, usage procedures, and operating procedures, are tested to determine whether they are fit for use (Huizinga & Kolawa, 2007)[2]. A unit is the smallest testable part of an application, in object-oriented programming, a unit is often an entire interface, such as a class, but could be an individual method (Xie, Taneja, Kale, & Marinov, 2007)[3].

A status code is a code issued by a server in response to a client's request made to the server. The Internet Engineering Task Force (IETF) created these status codes as means of standardizing error messages across the internet. Every status code is part of the HTTP/1.1 standard, and the status code registry is maintained by the Internet Assigned Numbers Authority (IANA).

A webserver sends a status code of 200 when the server completes a task correctly. This will be used as the evaluation metric.

5.3.2 System testing

System tests is testing that is conducted on an integrated system to evaluate the systems readiness and compliance with the functional requirements stated during the product design phase of the software development process.

All servers are integrated into one functional system and testing is performed. In this test case, a student, with the id 1245 has data run through the predictor; the expected recommendation should be 'NOT RECOMMENDED', a report on the recommendation will then be viewed.

5.4 Findings

In this project, the possibility of using Machine Learning models in the field of education has been proved to be successful. It is for that fact that the use of Machine Learning models as a form of decision support to help speed up enrolment processes in Universities has been the perfect proof of concept.

5.5 Future works

Plans for future work include the ability to suggest courses, based on comparison with Models generated for other Courses, Machine learning can also be used in the field of education to generate personal models to help predict missing scores. This potentially erases human bias in the event of filling out missing scores.

5.6 Conclusions

The successful implementation of the Machine Learning model has shown the potential use of Machine Learning in the Field of Education. However, limitations on the availability of data due to confidentiality concerns, the lack of viable, complete information, and the general unwillingness to devote time and monetary resources into Machine Learning can ultimately doom its application in the Education Sector.

References

- Carpenter, G. A., & Grossberg, S. (1988). The ART of Adaptive Pattern Recognition by a Self-Organizing Neural Network. *Computer*. <https://doi.org/10.1109/2.33>
- Deprivation, M., & Area, L. S. O. (2016). *The English Indices of Deprivation 2015 – Frequently Asked Questions (FAQs)*. (December), 1–21.
- Garnett, M. J., Edelman, E. J., Heidorn, S. J., Greenman, C. D., Dastur, A., Lau, K. W., ... Benes, C. H. (2012). Systematic identification of genomic markers of drug sensitivity in cancer cells. *Nature*. <https://doi.org/10.1038/nature11005>
- Hebb, D. O. (1949). The Organization of Behaviour. In *John Wiley and Sons*. <https://doi.org/citeulike-article-id:1282862>
- Huizinga, D., & Kolawa, A. (2007). Automated Defect Prevention: Best Practices in Software Management. In *Automated Defect Prevention: Best Practices in Software Management*. <https://doi.org/10.1002/9780470165171>
- Jorgensen, P. C., & Jorgensen, P. C. (2019). Exploratory Testing. In *Software Testing*. <https://doi.org/10.1201/b15980-20>
- Kohavi, R., & Provost, F. (1998). Glossary of terms: Editorial for the Special Issue on Applications of Machine Learning and the Knowledge Discovery Process. *Journal of Machine Learning*. [https://doi.org/10.1016/S0146-6453\(00\)80006-8](https://doi.org/10.1016/S0146-6453(00)80006-8)
- Kubat, M., Holte, R. C., & Matwin, S. (1998). Machine Learning for the Detection of Oil Spills in Satellite Radar Images. *Machine Learning*.
- Russel, S., & Norvig, P. (2012). Artificial intelligence—a modern approach 3rd Edition. In *The Knowledge Engineering Review*. <https://doi.org/10.1017/S0269888900007724>
- Samuel, A. L. (2000). Some studies in machine learning using the game of checkers. *IBM Journal of Research and Development*. <https://doi.org/10.1147/rd.441.0206>
- Settles, B. (2010). *Active Learning Literature Survey*.
- Sommerville. (2011). Software engineering. 9th ed. In *Monthly Notices of* <https://doi.org/10.1111/j.1365-2362.2005.01463.x>
- Xie, T., Taneja, K., Kale, S., & Marinov, D. (2007). Towards a framework for differential unit testing of object-oriented programs. *Proceedings - International Conference on Software Engineering*. <https://doi.org/10.1109/AST.2007.15>
- Zhu, X. (2005). [F] Semi-Supervised Learning with Graphs. *Machine Learning*. <https://doi.org/10.1023/A:1022653227824>