STUDENT ACADEMIC SKILL AND CAREER PREDICTION USING MACHINE LEARNING

THESIS REPORT

Submitted in partial fulfillment of the requirements for the award of the Degree of Master of Technology in Electronics and Communication Engineering with specialization in Communication Systems by the A P J Abdul Kalam Technological University

by

ANUROOPA SANTHOSH

Reg.No TKM20ECCS05

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

TKM COLLEGE OF ENGINEERING

KOLLAM 691 005

JULY 2022

STUDENT ACADEMIC SKILL AND CAREER PREDICTION USING MACHINE LEARNING

THESIS REPORT

Submitted in partial fulfillment of the requirements for the award of the Degree of Master of Technology in Electronics and Communication Engineering with specialization in Communication Systems by the A P J Abdul Kalam Technological University

by

ANUROOPA SANTHOSH

Reg.No TKM20ECCS05

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

TKM COLLEGE OF ENGINEERING

KOLLAM 691 005

JULY 2022

DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING

TKM COLLEGE OF ENGINEERING

KOLLAM 691 005

CERTIFICATE

Certified that this Project report titled **"STUDENT ACADEMIC SKILL AND CAREER PREDICTION USING MACHINE LEARNING"** is a bonafide record of the work done by **ANUROOPA SANTHOSH** (Reg.No.TKM20ECCS05) under my supervision, in partial fulfillment of the requirements for the award of the Degree of Master of Technology in Electronics and Communication Engineering with specialization in Communication Systems by the A P J Abdul Kalam Technological University.

Guide
Dr. ANZAR S M
Assistant Professor
Dept. of ECE, TKMCE

Coordinator
Dr. NISHANTH N
Associate Professor
Dept. of ECE, TKMCE

HoD
Prof. ABID HUSSAIN M
Head, Dept. of ECE
TKMCE

Acknowledgements

At the outset, I consider it my duty to thank Almighty God for giving me the necessary wisdom to successfully complete this project presentation.

I thank **Prof. ABID HUSSAIN MUHAMMED**, HOD, Department of Electronics and Communication, for his encouragement and support.

I express my sincere thanks to our PG coordinator, **Dr. NISHANTH N**, Associate Professor, Department of Electronics and Communication Engineering, for the support and encouragement during the course of this presentation.

I take this opportunity to express my sincere gratitude and profound thanks to my guide, **Dr. ANZAR S M**, Assistant Professor, Department of Electronics and Communication, for his advice, supervision and patience during the course of project preparation and presentation and for providing me guidance and critical inputs in the preparation and presentation of my project.

I would also like to express my sincere gratitude to all my teachers, friends and my parents for their much needed support during the preparation and presentation of the project.

ANUROOPA SANTHOSH
TKM20ECCS05

ABSTRACT

An important aspect of any educational institution is the academic performance of the students. Various factors can affect the academic performance of students. Understanding the performance of a student is beneficial to both the student and the educational institution. Most educators know that grades are an important performance indicator when it comes to monitoring the academic performance of students. Using artificial intelligence and machine learning, this work aims to predict students' careers based on their academic data. In particular, it helps calculate students' grades for their different skills by analysing patterns in the academic data of technical courses.

In this work, the academic grades and career of different students/subjects are collected and using regression models, these grades are assigned to six skill sets such as analytical skills, design skills, memory skills, numerical skills, presentation skills and programming skills. Based on these skills, different classifiers is implemented to predict the students' career. Linear Regression Regressor for skill prediction and Random Forest classifier for career prediction provide the best prediction performance, with accuracy scores of 0.999 and 0.962, respectively. So this work helps to calculate students' career for their different skills by analysing trends in large academic data of technical courses using artificial intelligence and machine learning.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

It is very important for students to assess their abilities and identify their interests during their studies so that they can find out what career field their interests and abilities will lead them to. In this way, they can improve their performance and motivate their interests, which will steer them towards their desired career. Early prediction of student performance can help higher educational institutions take immediate action, such as organising the right training, to increase student success rates. In addition, identifying at-risk students and taking preventative actions can significantly increase student success. The use of machine learning techniques for predictive purposes is widespread. Moreover, with the help of these career recommendation systems, recruiters can decide where to hire based on the applicant's performance and other assessments after carefully considering all aspects of the applicant.

Our ultimate goal is to predict the career or most suitable work area for a student among four major options such as Software field, Hardware/design field, Lecturers/Professors/Assistant Professors and Public servant/Upsc aspirant/Others by first matching the core courses in electronics and communication engineering with the six skills such as analytical skills, design skills, memory skills, numerical skills, presentation skills and programming skills that are related to the concepts of machine learning, a subset of artificial intelligence, such as Linear Regression, Decision Tree Regressor, Random Forest Regressor, MultiOutput Regressor and K-Neighbors Regressor.The academic data was taken from the institute's own academic database consisting of 250 students from the Eelectronics and Communication Engineering department of our Institution. In addition, available historical data was collected from

alumni of the institution, including information about their current field of work, and pre-processed to remove invalid data such as null values and duplicate values.The results were analysed for accuracy and the algorithm was optimised based on these results. This work also aims to handle imbalanced data using the Synthetic Minority Oversampling Technique(SMOTE) and its influence on the accuracy of each model is also analysed. The processed data was used to train six machine learning classifiers, such as Support Vector Machine, Gaussian Naive Bayes, Decision Tree Classifier, Random Forest Classifier and K-Nearest Neighbors Classifier.

Based on the application of SMOTE, the proposed model shows a significant impact on improving the performance of student career prediction models. LR Regressor for skill prediction and RF classifier for career prediction provide the best predictive performance with accuracy of 0.999 and 0.962 respectively.The machine learning classifiers were implemented using Google Colab in Python 3.6.9 programming language.

Chapter 2

Literature Review

This section, presents an overview of some works related to the proposed approach.

T. Alam *et.al.*,and [1] proposed an effective ensemble method to solve the problem of imbalance between several classes. An effective recursive technique for imbalanced data regression and multiclass classification is proposed. The problem of data imbalance is a critical problem in dealing with academic data, and this ensemble approach uses a recursive strategy to overcome it. Results show that the strategy works and improves performance. Several balanced results are created from the imbalanced data. This process is repeated until equilibrium is reached. A regression analysis is then performed to obtain the predicted value.

Y. Nieto *et.al.*, [2] they proposed a model to predict the number of students who will or will not graduate. Three supervised classification methods are used to predict the graduation rate of engineering students in South America using real data. The accuracy of the decision tree, logistic regression and random forest was also compared. The model is trained on 70 per cent of the 55200 records, which is a large dataset, and the rest is tested on the remaining records. They arrived at the results because RF shows the best result. The area under the curve (AUC) is used as an evaluation criterion in this paper. The AUC helps to reduce the ROC (Receiver Operating Characteristic Curve) to a single value that represents the expected performance of the classifier. The ROC curves for the three algorithms are compared and the prediction accuracy is evaluated. The false positive rate is shown on the x-axis, while the true positive rate is shown on the y-axis. It is clear that the logistic AUC is slightly higher than that of Random Forest.

In analysing data from computer science students, Abana *et.al.*, [3] used classification, a data mining technique, to identify students in need of academic counselling in the course. For this study, 133 individuals were selected. A ML collection of algorithms from open source software under the GNU general public licence WEKA (Waikato Environment for Knowledge Analysis) in Java is used for the classification model. The predictive performance of the classification models is then evaluated with a 10-fold cross-validation. From the results, RT (Random tree) has an accuracy of 75.2 per cent. A software tool was developed to identify students in need of academic counselling. They concluded that the accuracy of the classification model after cross-validation suggests that even more samples and attributes are needed to make a very accurate prediction.

Brijmohan Lal Sahu *et.al.*, [4] proposed model for predicting the possibility of students placement.For the dataset, data was collected from 100 students through a survey using a Google form, and information was also collected on students from the Department of Computer Science and Engineering and the Department of Information Technology from various colleges and universities in Chhattisgarh. The different types of Naive Bayes algorithm are the most efficient and simple algorithms for supervised labelled data sets. Gaussian Naive Bayes, Multinomial Naive Bayes and Bernoulli Naive Bayes are used for student placement prediction. Bernoulli NB has the highest accuracy rate of 0.75, which means that it makes a correct prediction of 75%. The authors conclude that it could become even more accurate with the availability of additional data.

Hussain S *et.al.*, [5]proposed model, which examines the quality of education based on the grades and prediction of students with their historical education data, is a well-known and valuable application in the field of education data mining. In particular, the concepts of preprocessing methods used to process data deal with instances with missing values and pull-out attributes that remove students' personal information, data removal, etc. Regression models and classification algorithms are also used to provide a framework for predicting scores. The results of the proposed models show that the decision tree model based on genetic algorithm has high accuracy.

The data prediction model proposed by Patel *et.al.*, [6] takes into account students' academic grades and the weighted skills they acquire and enabled the creation

of a decision-making tool to improve students' resource utilisation by predicting their preferred areas of work based on their academic performance. They also created a graphical user interface (GUI) that provided both input and output fields for the website. Based on our skills as input, it assigns us the optimal work category.

Sripath Roy *et.al.*, [7] aims in developing a model that allows prediction for the career field of computer science candidates.The data was trained and tested using three algorithms and the results show that SVM has the highest accuracy (90.3%) and XG Boost has the lowest accuracy (88.33%). Since SVM provided the highest accuracy, it was decided that all subsequent data predictions should be done using SVM. A web application was also created to provide the students' input parameters and the final result. The results are presented in a graph and it is obvious that the SVM algorithm hits the predictions better.

Laxmi Shanker Maurya *et.al.*, [8] proposed a mechanism for prioritising academic performance parameters relevant to a student's placement to predict a student's placement in IT based on his academic performance in grades ten to twelve, his graduation and his backlog to graduation. Again, the results of many proposed classifiers were compared. It was found that the percentage in grade ten was most important, followed by the percentage in grade twelve and the backlog in B Tech. This was confirmed by applying the classifiers to new data and performing a decision tree analysis. Gradient Descent has the highest precision of 0.9117, while Support Vector Machine and Logistic Regression have the best value for the area under the curve of 0.86.

Khadilkar *et.al.*, [9] aims to predict whether a person is employable or needs further training, both employability and job hopping are predicted. For predicting employability, 5000 examples of CVs were collected. For job hopping prediction, which predicts whether a worker will quit their job or not, 100 reviews of workers about their companies were collected and a satisfaction score was derived from the workers' reviews using text mining and a csv file was created. The accuracy, misclassification rate, confusion matrix, F1 score and precision of each model were then calculated. The data is then analysed by graphing both the prediction of employability and the prediction of whether or not an employee will quit. With an accuracy of 89 per cent, the Naïve Bayes strategy appears to be the most accurate for predicting employability. The decision tree, with an accuracy of 85%, seems to be a good tool for predicting

Paper	Contribution	Algorithms used	Results	Limitation
T. Alam et.al. [1]	Ensemble method for solving multi-class imbalanced problem in regression analysis, new recursive approach, experimental analyses.	SMOTE Bagging and Split Bal. Linear Regression, REPTree, Locally Weighted Regression, Piecewise Regression	Effective for multi-class imbalance classification and data imbalance regression problem	Proposed method is external.
Y. Nieto et.al. [2]	Predict the number of students going to graduate or not	DT, Logistic Regression, RF	RF had the highest overall accuracy	Analysis of other effectiveness metrics such as F-Measure or Specificity not considered, comparison with 3 algorithms is only used.
Abana et.al [3]	Identify students who need academic counselling in the subject	Decision tree algorithms, Random Tree, REPTree and J48	Random Tree decision tree shows highest accuracy	Limited to the use of only three decision algorithms, more samples and more attributes is still needed to arrive with a highly accurate prediction.
Brijmohan Lal Sahu et.al [4]	Predicting the possibility of placement of a student	SVM, feature selection algorithms, Neural Network, Fuzzy System and Deep Learning.	Bernoulli Naïve Bayes model is best over Gaussian and Multinomial	Dataset is limited to 100 students.
Hussain S et.al [5]	Predicting students' performance, Regression will forecast marks, while grade will be forecasted by classification system.	Regression model and DT-classifier, Attributes Optimization Using Genetic Algorithm	GA based decision tree classifier and regression achieved remarkable results.	Big-Data issue, algorithms of ML doesn't play adequately.
Patel et.al [6]	Predict preferred work domain	Keras machine learning module	Web application which caters to the academic need of the student and the faculty by predicting suitable job domains and academic grade.	Accuracy comes to around 70 % where the probability depends on the keyword entered by the user
Sripath Roy et.al [7]	Career area prediction of computer science domain candidates.	SVM, XG Boost	SVM having the highest accuracy (90.3\%) and XG Boost having the lowest accuracy (88.33\%)	Less reliability of system due to input directly given.
Laxmi Shanker Maurya et.al [8]	Prioritize the academic performance parameters relevant for student placement	Gradient Descent, SVM, Logistic Regression	Gradient Descent has the highest accuracy score	Lack of attributes in dataset.
Khadilkar et.al [9]	Person is employable or needs more training, both employability and job-hopping prediction is done	KNN, RF, NB for employability prediction and DT for job hopping.	Naïve based strategy and DT appears to be the most accurate	Used the resumes in the same format or template, Need for more proper data.

Figure 2.1: Taxonomy of related studies

whether a person will quit their employment or not.

Giri *et.al.*, [10] proposed a Placement Prediction System that predicts the probability of a student getting placed. The input attributes are the student's academic history, skills such as programming skills, communication skills, analytical skills and teamwork. The data used is from the placement statistics of PES Institute of Technology's Bangalore South Campus for the last two academic years. The basic model in this case is the K-nearest neighbour classification. Here, for each new sample, the votes of nearest neighbours from other classes are searched and counted. The accuracy in this case was 78.57 per cent.

A taxonomy of related studies is shown in 2.1.

Chapter 3

Proposed Model

Traditionally, questionnaires and other skills analysis tests are used to predict a student's skill and preferred career or field of work. However, this process is time-consuming. Nowadays, computer technologies are indispensable in many fields. Machine learning is used in today's digital world in many different sectors and industries, e.g. clinical analysis, image processing, classification and regression. Put simply, machine learning is the study of how to learn from experience and act like human. There are three types of machine learning algorithms: supervised machine learning, unsupervised machine learning and reinforcement learning. It is crucial to assess students' abilities so that they can be steered in the right direction [11].

Figure 3.1 shows a simple representation of the model, illustrating the initial mapping of courses to skills derived from the model. For each skill we take input from the appropriate courses, e.g. for analytical skills we take courses such as Solid State Devices,Electromagnetic Theory, Microwaves and Radar,Nanoelectronics; for design skills we take Engineering Graphics, Design Engineering, Logic Circuit Design and Electronic Circuits. For memorisation skill, we take grades from courses like Business Studies, Principles of Management, Disaster Management and Life Skills. For presentation skills: Seminar, Project, Design Project and Comprehensive Viva. For programming skills, Object-oriented Programming, Microcontroller lab and Matlab are chosen and finally for numerical skills, courses like Calculus, Network theory, Signals and Systems and Control Systems.

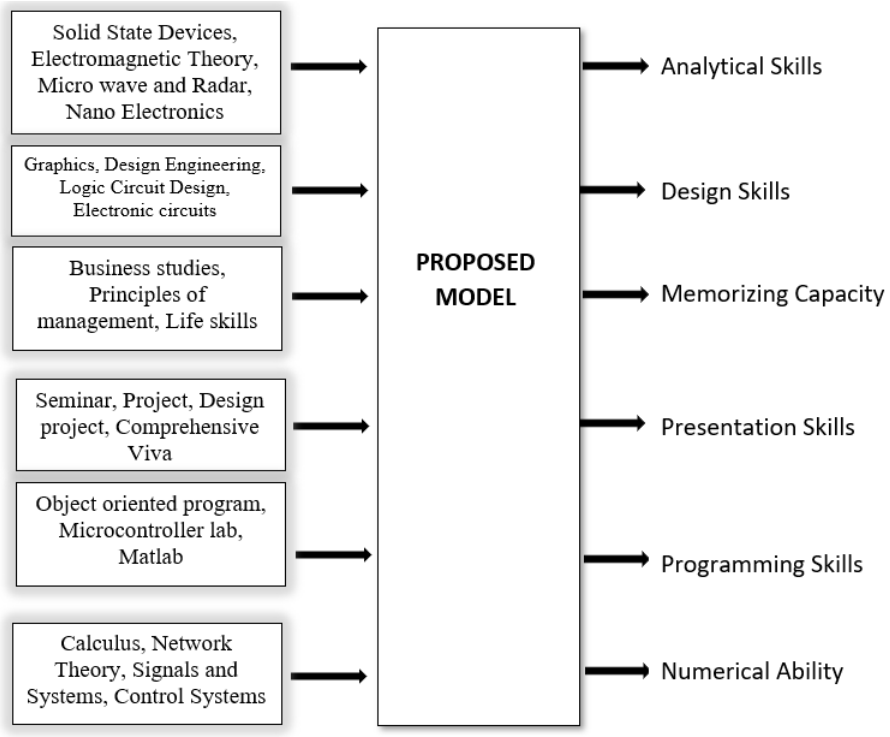


Figure 3.1: Proposed Model

3.1 Dataset

Data collection is the methodical process of gathering and analysing information from a variety of sources to obtain a comprehensive and accurate picture of a particular course of events. By collecting data, an individual or organisation can analyse outcomes, make predictions about future probabilities and respond to relevant queries [12]. The dataset, consisting of 250 students from our institution's ECE department, as well as available historical data from graduates containing information about their current field of work, was also collected and compiled into an Excel file that can be read with Python code.

3.2 Software Tool Used

We chose Google Colab, a free, fully cloud-hosted Jupyter notebook environment in which to write and run Python programmes. The screen shot 3.2 shows the Google Colab platform and part of the programme showing the call to the dataset.

Python is an interpreted high-level general-purpose programming language. Its design philosophy emphasises code readability and uses distinct indentation. The

language constructs and object-oriented approach are designed to help programmers write clear, logical code for small and large projects. Python uses garbage collection and has dynamic typing. It supports a variety of programming paradigms, including procedural, object-oriented and structured programming (in particular). The scientific discipline of machine learning enables computers to learn without explicit programming. They are typically used to solve numerous problems in everyday life. In the past, machine learning tasks were performed manually by coding each algorithm and mathematical and statistical formula, which was tedious, time-consuming and ineffective. But thanks to various Python libraries, frameworks and modules, this is now much easier and more effective than before. Today, Python is one of the most widely used programming languages for this task and has displaced several others in the industry, partly due to its enormous library collection.
Python libraries needed for use in this Machine Learning project are:

- **NumPy**: is a very popular Python library for large multidimensional arrays and processing matrices using a large collection of high-level mathematical functions. It is very useful for basic scientific computations in machine learning. Its capabilities in linear algebra, Fourier transform and random numbers are very useful. High-end libraries like TensorFlow manipulate tensors internally with NumPy.

- **TensorFlow**: is a very popular open-source library for high-performance numerical computation developed by the Google Brain team in Google. It is a framework for specifying and performing Tensor-based computations. It can be used to train and run deep neural networks that can be used to create a variety of AI applications. In the field of deep learning research and application, TensorFlow is widely used.

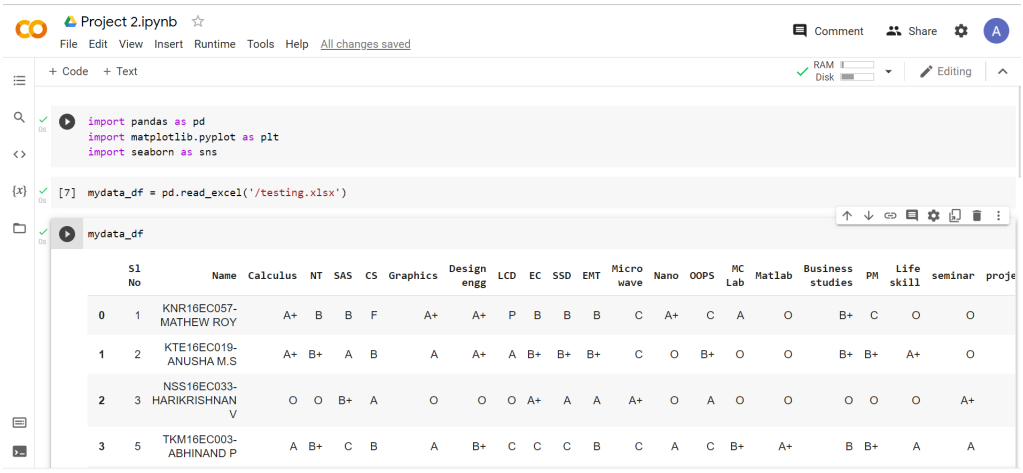
- **Pandas**: is a popular Python library for data analysis. It is not directly related to machine learning. The dataset needs to be prepared before training. Pandas is helpful in this case because it is primarily designed for preprocessing and extracting data. It offers a wide range of tools for data analysis as well as high-level data structures. It offers a variety of built-in techniques for searching, combining and filtering data.

- **Matplotlib**: is a very popular Python library for data visualisation. Like Pandas, it is not directly related to machine learning and is particularly useful when a programmer needs to recognise data patterns. It is a 2D graph library that allows us to create 2D graphs and plots. A module called pyplot provides tools for editing line styles, font attributes, axis formatting, etc. and makes it easy for programmers to plot data. For visualisation techniques, it offers a variety of charts and plots, including histograms, error charts and bar charts.

- **Seaborn**: is a high-level data visualization library, meaning it does many things for us automatically and adds a lot of aesthetic elements to our plots. Seaborn can be modified by using Matplotlib.

Google Colab: Google is quite aggressive in AI research. Over many years, Google has developed an AI framework called TensorFlow and a development tool called Colaboratory, which Google has made freely available to the public. Colaboratory is now known as Google Colab or simply Colab. The use of GPU(Graphics Processing Unit) is another attractive feature that Google offers to developers. Colab is completely free and supports GPUs [13]. The fact that Colab is being made available to the public for free could be aimed at establishing the software as a benchmark for university courses in data science and machine learning, or in the long run to create a user base for the paid Google Cloud APIs. Whatever the reason, Colab's debut has made it easier to learn about machine learning and build applications.
As a programmer, we can perform the following using Google Colab.

- Write and execute code in Python.
- Document your code that supports mathematical equations.
- Create/Upload/Share notebooks.
- Import/Save notebooks from/to Google Drive.
- Publish notebooks from GitHub.
- Import external datasets e.g. from Kaggle
- Integrate PyTorch, TensorFlow, Keras, OpenCV



	Sl No	Name	Calculus	NT	SAS	CS	Graphics	Design engg	LCD	EC	SSD	EMT	Micro wave	Nano	OOPS	MC Lab	Matlab	Business studies	PM	Life skill	seminar	proje
0	1	KNR16EC057-MATHEW ROY	A+	B	B	F	A+	A+	P	B	B	B	C	A+	C	A	O	B+	C	O	O	
1	2	KTE16EC019-ANUSHA M.S	A+	B+	A	B	A	A+	A	B+	B+	B+	C	O	B+	O	O	B+	B+	A+	O	
2	3	NSS16EC033-HARIKRISHNAN V	O	O	B+	A	O	O	O	A+	A	A	A+	O	A	O	O	O	O	O	A+	
3	5	TKM16EC003-ABHINAND P	A	B+	C	B	A	B+	C	C	C	B	C	A	C	B+	A+	B	B+	A	A	

Figure 3.2: Data into colab

- Free Cloud service with free Graphics processing unit.

3.3 Framework of Proposed Model

The first phase of work discusses the workflow of a machine learning project as a whole. This includes all the steps required to build a proper machine learning project from scratch, as well as data pre-processing, data cleaning, feature exploration and feature engineering. With this in mind, an initial workflow diagram for our proposed work was developed and shown in Figure 3.4.

We can define the machine learning workflow in stages:

1. **Gathering data**

   The method of data collection depends on the type of project we want to create. For example, if we want to create a real-time ML project like to design an IoT system that collects data from various sensors. The data set can come from a variety of sources, such as files, databases, sensors and many others. However, it cannot be used directly for the analysis process because it may contain large amounts of missing data, extremely high values, unordered text data or noisy data.To solve this problem, the data is prepared.

2. **Data pre-processing**

   One of the most important phases in machine learning is the preprocessing of data. It is the most important phase in the process of correctly building

machine learning models. In machine learning, the 80/20 rule applies. Every data scientist should spend 80% of the time on data preprocessing and 20% on the actual analysis. In our project, the data is processed by loading the dataset into Colab. During preprocessing, the data is cleaned by removing the null values and unwanted rows and columns.

The cleaning of raw data, or the transformation of data collected in the real world into a clean data set, is called data preprocessing. In other words: When data is collected from numerous sources, it is done in a raw format that prevents data analysis. Data pre-processing is the term for the stage of the process where various steps are taken to transform the data into a manageable, clean data set. In a classification task, we do not always end up with a situation where the distribution of the two classes is equal. So most of the time we get unbalanced data. The challenge in working with unbalanced data is that most ML techniques ignore the minority class and therefore perform poorly.

There are two techniques to overcome this and they are under sampling and over sampling,Undersampling is a technique to balance imbalanced data sets by keeping all the data in the minority class and reducing the size of the majority class. This reduces the number of samples in the majority class and important data is lost, while oversampling draws new samples from the minority class that could reach the length of the majority class samples,where comes SMOTE analysis which is been used in this work as the sampling method.

In the case of SMOTE, we create artificial samples of our minority class. For example, if we have two features X1 and X2 and the number of red samples is higher than the number of black samples, we will not decrease the number of red samples but increase the number of black samples representing the minority class by SMOTE. To do this, we take the four black samples and perform SMOTE analysis as shown in 3.3, to find the nearest neighbours of each point, i.e. for P1 we have P2, P3 and P4 as neighbours. Similarly, for P2, points P1, P2 and P3 are neighbours and so on for P3 and P4. Given the number of samples to be created, SMOTE first finds the lines connecting our minority class samples and we draw the same (artificial) instances somewhere on these lines.

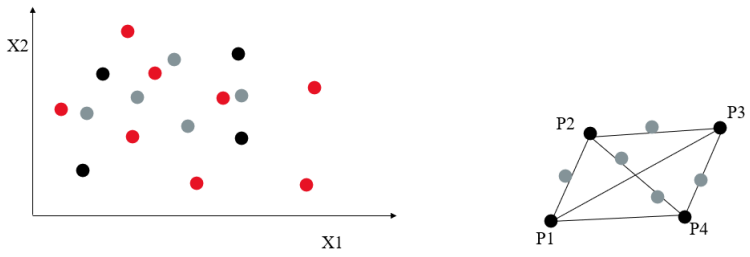


Figure 3.3: SMOTE Analysis

The grey points in the figure are also samples belonging to the black coloured samples. Depending on how many data points we want to have, we can have multiple samples on a single connection. So we can specify artificial samples that we want to increase to SMOTE to match our majority class.

To simplify the classification process, we have grouped related job titles into one class during data processing. Therefore, there are only the four main job options - Software field (Full stack Developer, Testing engineer) as Class A, Hardware/design field (UI Developer, Graphic design, PCB designer, Embedded hardware engineer, Hardware Engineer Vehicle Dynamics) as Class B, Lecturers/Professors/Assistant Professors as Class C and finally Public servant / Upsc aspirant/ Others (Central Government Jobs, Assistant Executive Engineers, Assistant Engineers, Electrical Sectional Engineer, Central Engineering Service, Telecommunication Engineer, Income Tax, Inspector in Central Excise, ISRO, BARC, Banking, Defence etc.) as Class D

Most of the real-world data is messy, some of these types of data are:

- Missing data: Missing data can be found if it is not continually created or if the application is experiencing technical difficulties (IOT system).
- Noisy data: Outliners are another name for this kind of data, which might happen as a result of human error (data collected manually) or a technical issue with the device at the time of data collection.
- Inconsistent data: Human error (mistakes with the name or value) or data duplication may have resulted in the collection of this type of information.

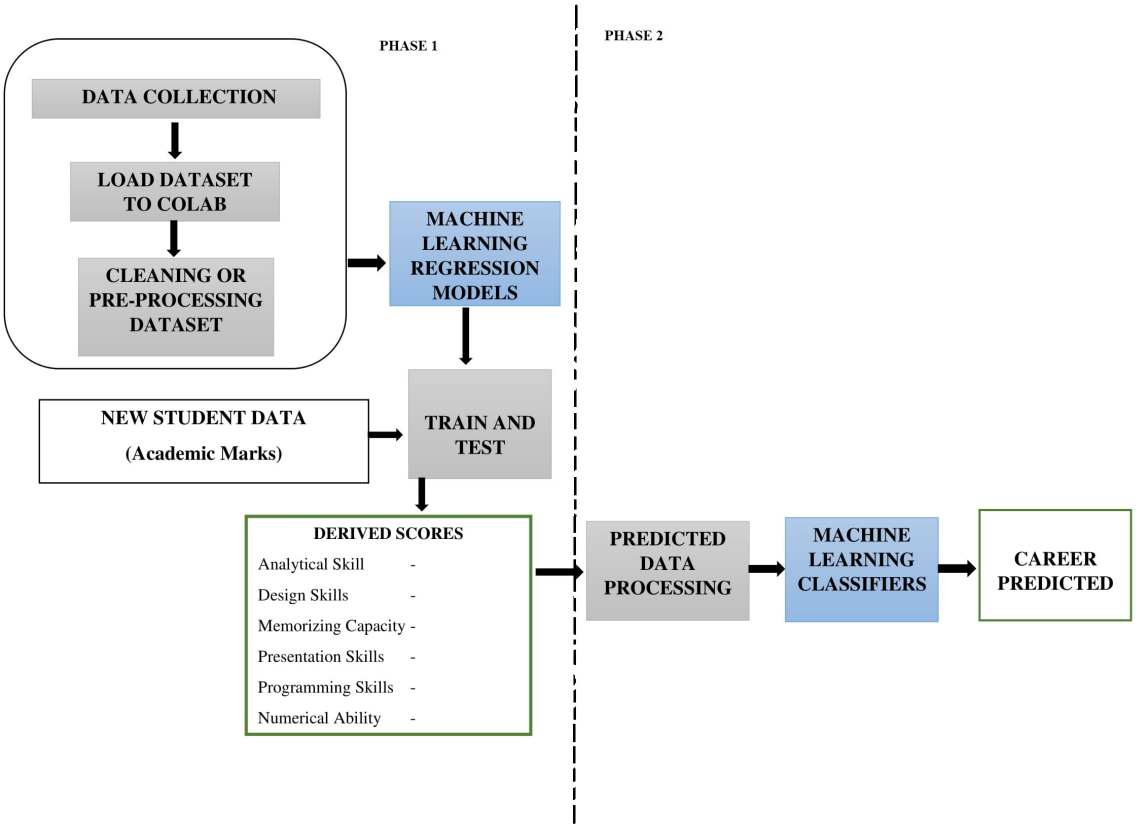


Figure 3.4: Framework of Proposed Model

3. **Researching the model that will be best for the type of data**

   Our prime aim is to use the pre-processed data to train the model that performs the best. Right here in this step we choose the appropriate ML Model. In figure 3.4 which shows the workflow of our proposed work, phase 1 is the flow of skill prediction and phase 2 is career prediction from phase 1. Each phase has to go through the above process, in our project we used 5 regression models like Linear Regression, Decision Tree Regressor, Random Forest Regressor, MultiOutput Regressor and K Nearest Neighbors Regressor for phase 1. For phase 2, we used 6 classifiers such as SVM, GaussianNB, DecisionTreeClassifier, Random Forest Classifier and K-Nearest Neighbors Classifier.

   In particular, the following are the theoretical model used as basis to construct the proposed prediction model:

   - **Regression Models**

     (a) **Linear regression** is a machine learning algorithm based on supervised learning. It performs a regression operation. Regression uses

independent variables to model a target predicted value. It is most often used to determine how variables and predictions relate to each other. Regression models differ depending on the number of independent variables used and the type of relationship they consider between the dependent and independent variables.

(b) **Decision Tree Regressor** creates tree-like models for classification or regression. It incrementally develops an associated decision tree as it divides a data set into smaller and smaller sections [14]. The result is a tree with leaf nodes and decision nodes. The ID3 algorithm can be used to construct a decision tree for regression by replacing information gain with standard deviation reduction.

(c) **Random Forest Regressor** is a meta estimator that fits a number of classifying decision trees on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting.

(d) **Multi-Output Regressor** strategy consists of fitting one regressor per target. This is a simple strategy for extending regressors that do not natively support multi-target regression.

(e) **K Nearest Neighbors Regressor** is a non-parametric technique that provides an intuitive approximation of the relationship between independent variables and the continuous outcome by averaging the data in the same neighbourhood. The analyst must determine the size of the neighbour, or cross-validation can be used to determine the size that minimises the mean squared error.

- **Classifiers**

(a) **SVM** The SVM represents each data set as a separate point in n-dimensional space that can be used to categorise the data. The Support Vector Machine (SVM) is based on the idea of decision planes that define decision boundaries and effectively handle classification problems. The SVM is a non-probabilistic binary classifier because it predicts from a sorted data set which of two possible classes contains the information.

(b) **The GaussianNB** method is based on Bayes' theorem and is popular because it is simple and has a fast prediction time. With its mixture of complexity and an adaptive probabilistic model, it is suitable for small data sets. There is not just one technique for training such classifiers, but rather a family of algorithms based on the premise that the value of a feature given a class variable is independent of the value of any other feature. For example, if a fruit is red and round and has a diameter of about 10 cm, it can be considered an apple. Despite possible correlations between the variables colour, roundness and diameter, a naive Bayes classifier assumes that each of these features contributes independently to the probability that this fruit is an apple.

(c) **Decision Tree Classifier** uses a branching strategy to show the possible outcomes of each decision. This type of supervised learning technique is most commonly used in classification problems. It can be used for both categorical and continuous dependent variables. With this approach, we divide the population into two or more homogeneous groups. To create as many unique groups as possible, this is done on the basis of the most important features/independent variables. [15].

(d) **Random Forest Classifier** RF is a group of decision trees. The Random Forest (RF) classifier uses a variety of decision trees on different subsets to discover the optimal features for high accuracy and avoid the problem of overfitting [16]. It is based on ensemble learning. The RF performs well in classification and is relatively insensitive to outliers and noise.

(e) **K Nearest Neighbors Classifier** KNN is one of the simplest Machine Learning algorithms based on Supervised Learning technique.The K-NN algorithm assumes that the new case and the existing cases are comparable and places the new case in the category that is most similar to the existing categories.The K-NN algorithm can be used for both regression and classification, but is mostly used for classification problems.

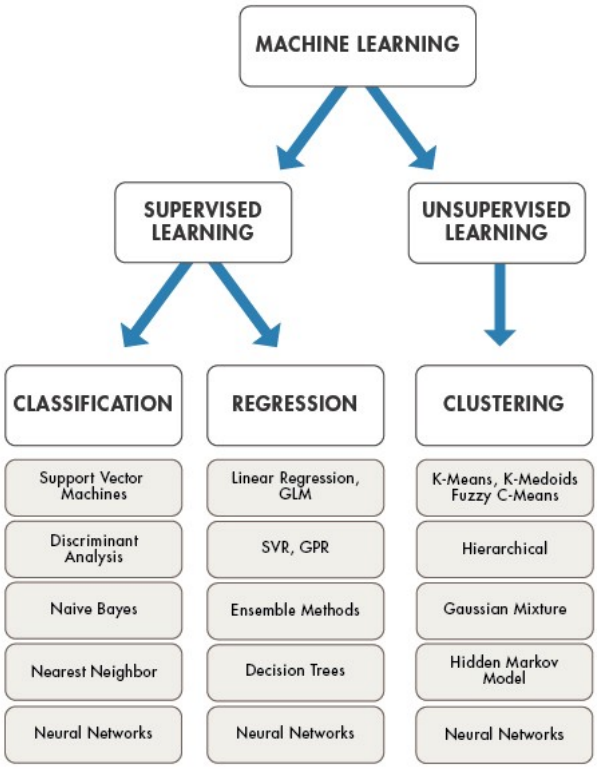


Figure 3.5: Overview of models

4. **Training and testing the model** To train a model, we first divide the data set into three sections: 'training data', 'validation data' and 'test data'. We use a "training dataset" to train the classifier, a "validation dataset" to fine-tune the parameters, and a "test dataset" to evaluate the performance of our classifier. Only the training and/or validation dataset is available to the classifier during training. The test data set is not be used for training the classifier. Only when the classifier is tested, the test dataset is made available. For our work, the entire dataset is divided into 80% for training and the rest for testing.

5. **Evaluation** Model evaluation is an essential part of the model development process. We can adjust the hyperparameters of the model to increase its accuracy, while at the same time examining the confusion matrix to see if we can increase the proportion of true positives and true negatives. In this project, we use k-fold cross-validation to evaluate the accuracy of each model.

    - **Cross-validation** is a statistical method for estimating the capabilities of machine learning models. It is simple to understand, easy to implement, and produces skill estimates that often have lower bias than other methods.

	Actually Positive (1)	Actually Negative (0)
Predicted Positive (1)	True Positives (TPs)	False Positives (FPs)
Predicted Negative (0)	False Negatives (FNs)	True Negatives (TNs)

Figure 3.6: Confusion Matrix

It is often used in applied machine learning to compare and select a model for a particular prediction problem.

- **Confusion Matrix** helps us visualise the classification performance of each predictive model. The instances in each actual class are shown in the rows of the matrix, while the instances in each predicted class are shown in the columns, or vice versa. The table shows the number of false positives, false negatives and true negatives. This allows for more detailed analysis and accuracy. If we compare the true classification set with the predicted classification set, there are 4 different outcomes that can occur in a given column, as shown in 3.6. A true positive result is one where both the actual and predicted classification are positive (1,1), which means that the classifier has correctly identified the positive sample. The positive sample is incorrectly classified as negative by the classifier when the actual classification is positive and the predicted classification is negative (1.0), which is called a false negative result. The four types of outcomes (true positive, false negative, false positive and true negative) and the positive and negative classifications serve as a template for each binary confusion matrix. The four outcomes can be formulated in a 2×2 confusion matrix.
  - **Accuracy**: A measure of how closely a value matches the real number. The proportion of all things that are correctly classified is this number. It is the proportion of the sum of the true positives and the true negatives.
  - **Precision**: Percentage of the total number of positives that have been accurately classified as such. It measures the proportion of true posi-

tives to the total number of true and false positives. The ratio of true positives to the sum of true and false positives is called precision.

- **Recall**: It is the number of items correctly identified as positive in relation to the total number of true positives. Since the penalties for precision and recall are opposite, the equations themselves are also opposite.
- **F-Measure**: The F-measure is the weighted harmonic mean of precision and recall The F-score or F-measure is a measure of the accuracy of a test. It is calculated from the precision and recall of the test.

Chapter 4

Result and Discussions

The results of the study are divided into subsections in which the performance of each model is assessed in detail. The results of all the Machine Learning models used were examined and compared. The impact of using oversampling SMOTE to solve the problem of unbalanced multiple classification with the same data set is then analysed and evaluated. A statistical investigation of the effectiveness of the model in determining the relevance of features is also conducted.

1. COMPARISON OF THE PREDICTIVE MODELS USING MACHINE LEARNING ALGORITHMS

   Here, the student dataset was trained with five selected algorithms in regression and classification for skill and career prediction and their prediction accuracy was evaluated. Performance accuracy is compared using ten-fold cross-validation to identify the best predictive model for the best results. To ensure that the predictive model is appropriate to produce accurate results, performance is assessed using several metrics, including classification accuracy, precision, recall (sensitivity) and f-measure. The predictive performance metrics for each classifier on the student dataset are summarised in 4.1 , 4.2 and 4.3.

   Also a correlation matrix is a table that shows the correlation coefficients for different variables. In the matrix, the correlation between all possible pairs of values is shown in a table. The correlation matrix is an effective tool for finding and displaying trends in the data provided and for summarising a large data set. For our career prediction dataset, we plotted the correlation matrix and

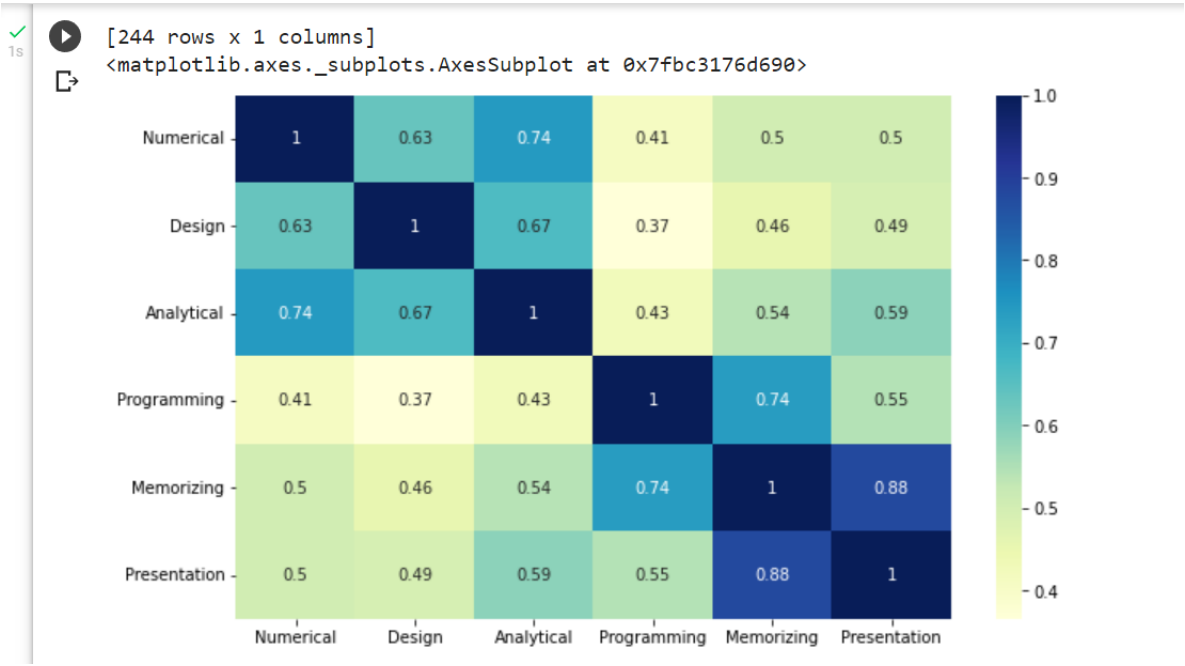


Figure 4.1: Correlation matrix of features

presented it in the figure 4.1. The correlation coefficient is included in each cell of a table. A completely negative linear correlation between two variables is represented by the value -1. No linear correlation between two variables is represented by zero . A perfect positive linear correlation between two variables is represented by the value 1.Since each skill has a positive correlation with the others, there are no features in our situation that can be considered excluded from our design. Each feature is equally important, so we cannot eliminate any of the skill features from the data set.

Table 4.1: Comparison of predictive model for Skill prediction

Metric	**LR**	**DT**	**RF**	**MultiOutputReg**	**KNN**
Accuracy	**0.999**	0.709	0.978	0.984	0.685

Table 4.2: Analysis based on Confusion matrix for Career Prediction

Predicted Class	**SVM**	**GaussianNB**	**DT**	**RF**	**KNN**	**SMOTE**
Accuracy	92.6	63.5	89.4	**96.2**	92.2	**After**
Accuracy	78.7	68.8	71.1	75.3	74.3	**Before**

The analysis shows that the Linear Regression regressor for predicting ability and the Random Forest classifier for predicting career provide the best predictive performance with an accuracy of 0.999 and 0.962, respectively. Gaussian

Table 4.3: Classification report Analysis based on Confusion matrix for Career Prediction after SMOTE

Predicted Class	**SVM**	**GaussianNB**	**DT**	**RF**	**KNN**
Accuracy	0.92	0.63	0.89	**0.96**	0.92
Precision	0.77	0.63	0.61	**0.85**	0.64
Recall	0.72	0.54	0.54	**0.84**	0.61
f1-score	0.73	0.57	0.54	**0.83**	0.62

```
Accuracy of  RandomForestClassifier : 0.967 (0.029)
                                precision    recall  f1-score   support

          Hardware/Design Engineer       1.00      0.57      0.73         7
              Lecturers/Professors       0.50      0.50      0.50         6
Public Servant/Upsc aspirant/Others      0.50      1.00      0.67         1
                 Software Engineer       0.89      0.94      0.92        35

                          accuracy                           0.84        49
                         macro avg       0.72      0.75      0.70        49
                      weighted avg       0.85      0.84      0.83        49
```

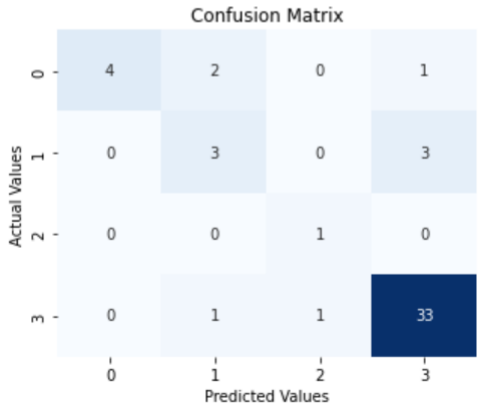


Figure 4.2: Confusion matrix of RF Classifier

Naive Bayes scored the worst model with a score of 0.589. However, the highly imbalanced classes in the dataset often resulted in incorrect decisions for the minority class developed during training of the dataset. Further experiments addressing the problems were conducted for the purpose of generalisability to lower the ratio of each class.Also, the confusion matrix of Random Forest Classifier with classification report is shown in Figure 4.2. The confusion matrix provides a comparison between actual and predicted values. This results in a 4×4 confusion matrix, which calculates an accuracy of 0.962.

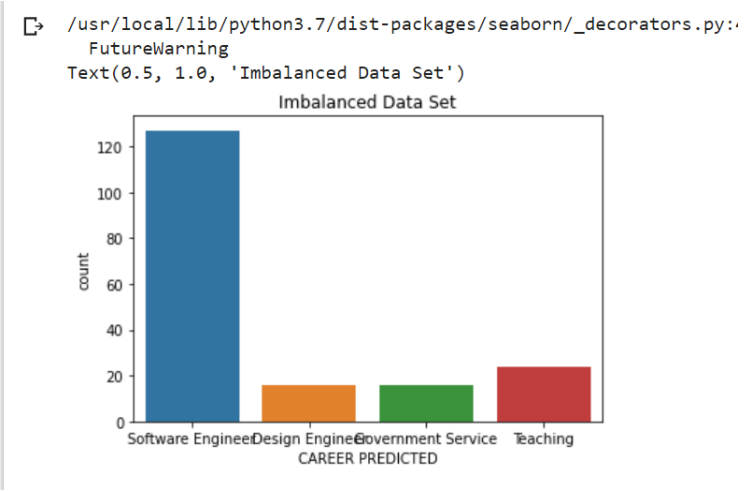


Figure 4.3: Imbalanced data

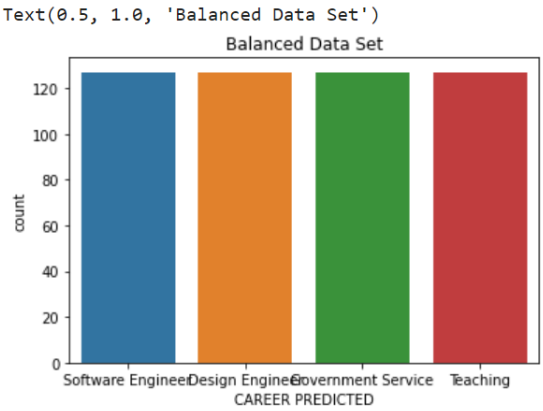


Figure 4.4: Balanced data

2. IMPACT OF OVERSAMPLING AND COMPARISON WITH SMOTE APPLIED

The most appropriate approach to solving the problem of imbalanced data sets based on a random sampling algorithm is the so-called SMOTE or Synthetic Minority Oversampling Technique. By applying a synthetic sampling technique, new instances of minority classes can be constructed from an existing unbalanced dataset and the distribution can be made more balanced. The confusion matrices of all predictive models have been shown to improve the classification accuracy for minority classes, as shown in the graph 4.6 and the table 4.2. The data balancing after SMOTE is shown in figure 4.5. Also the results in change of dataset is plotted in graph using matplotlib python in figure 4.3 and 4.4.

Before sampling, the data set is unbalanced because the software engineer is the

```
x=df[["Numerical","Design","Analytical","Programming","Memorizing","Presentation"]]
y=df["CAREER PREDICTED"]

from sklearn.model_selection import train_test_split
xtrain,xtest,ytrain,ytest=train_test_split(x,y,train_size=0.75)

from collections import Counter
print("Before oversampling: ", Counter(ytrain))
from imblearn.over_sampling import SMOTE
SMOTE = SMOTE(k_neighbors=1)
xtrain_SMOTE,ytrain_SMOTE = SMOTE.fit_resample(xtrain, ytrain)
print("After oversampling: ",Counter(ytrain_SMOTE))
```

```
Before oversampling:  Counter({'Software Engineer': 126, 'Teaching': 23, 'Government Service': 18, 'Design Engineer': 16})
After oversampling:  Counter({'Government Service': 126, 'Software Engineer': 126, 'Teaching': 126, 'Design Engineer': 126})
```

Figure 4.5: Data Balancing

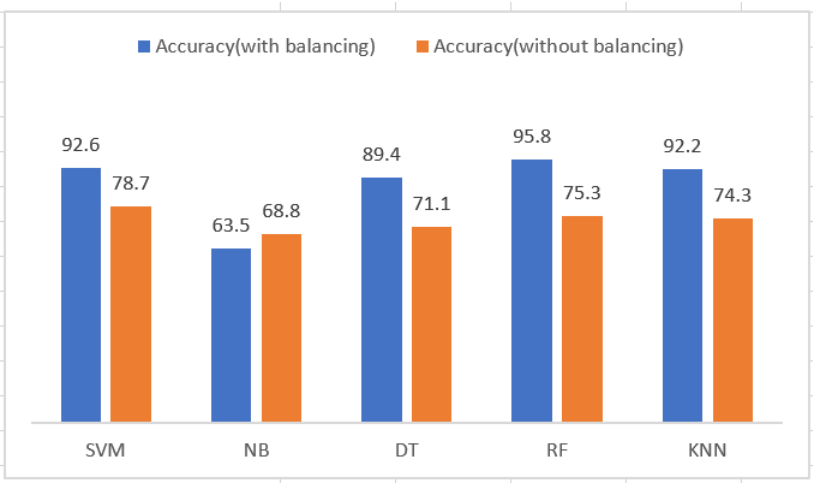


Figure 4.6: Sampling effect on Accuracy

majority class and all three other classes are in the minority. Applying models at this stage results in lower performance than after applying SMOTE oversampling. After applying SMOTE the data set is balanced and all four classes are present in equal numbers. Applying ML here leads to better performance, which is shown in the graph 4.6.

3. STATISTICAL EXAMINATION AND FEATURE RELEVANCE

   The term "feature importance" describes methods that assign a score to each input feature for a given model; the scores indicate only the "importance" of each feature. A higher score means that the feature in question has a greater influence on the model used to predict a particular variable.

   Understanding the data that goes into a model is different from creating a model. Just as a correlation matrix allows us to understand the link between the features and the target variable, feature mining does the same. In model

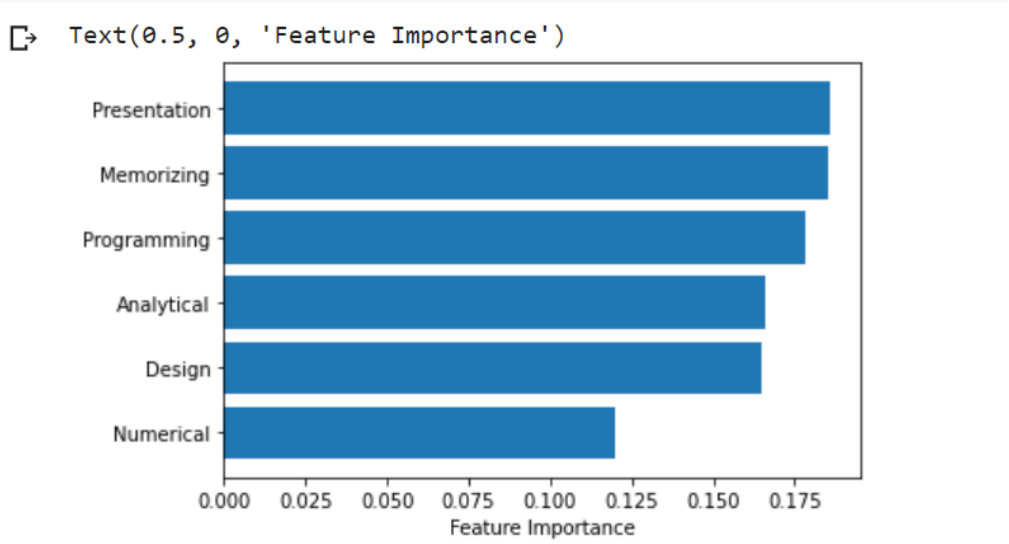


Figure 4.7: Feature Importance in Career Prediction

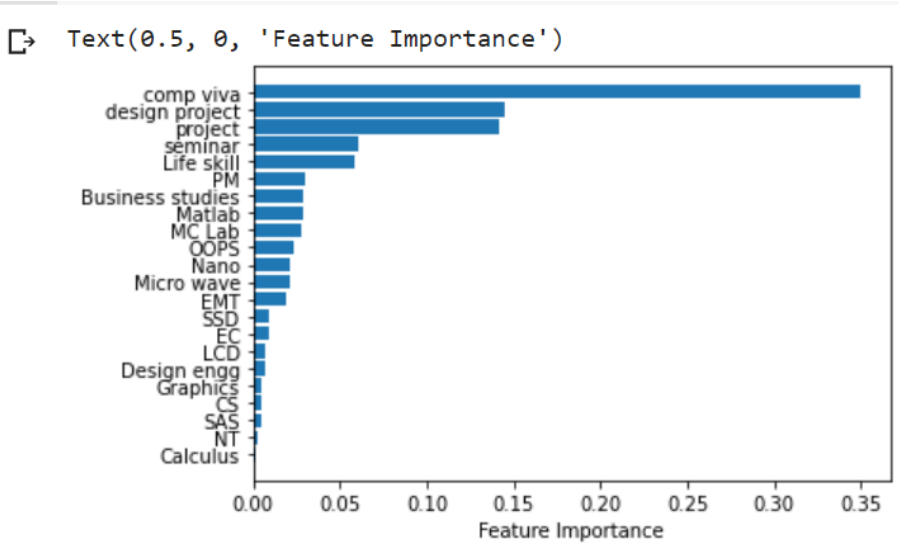


Figure 4.8: Feature Importance in Skill Prediction

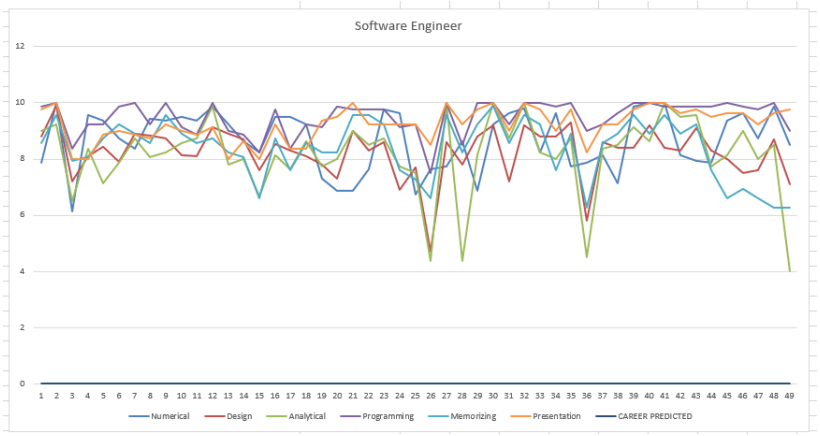


Figure 4.9: Feature Importance of Software engineer

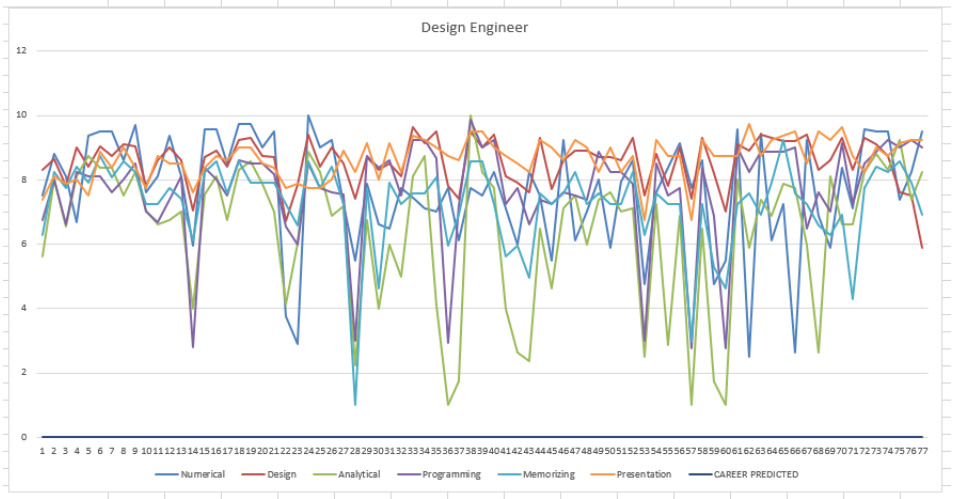


Figure 4.10: Feature Importance of Hardware/design field

improvement, we can use the scores calculated from feature importance to reduce the dimensionality of the model. Feature importance is also useful for interpreting and communicating our model to other stakeholders.

We tried the permutation of feature importance for this project to find the most relevant feature, and the result shows up as 4.7 where presentation skill and memorizing skill have the most relevance. And for skill prediction figure 4.8 shows that Comprehensive viva results gives out the most contributing feature followed by projects and seminar scores and courses like Calculus, NT, signals etc gives out the less importance.

Permutation Feature Importance There is a simple logic underlying the concept of the meaning of permutation features. When we permute the values of a feature, we can determine the relevance of the feature by observing the

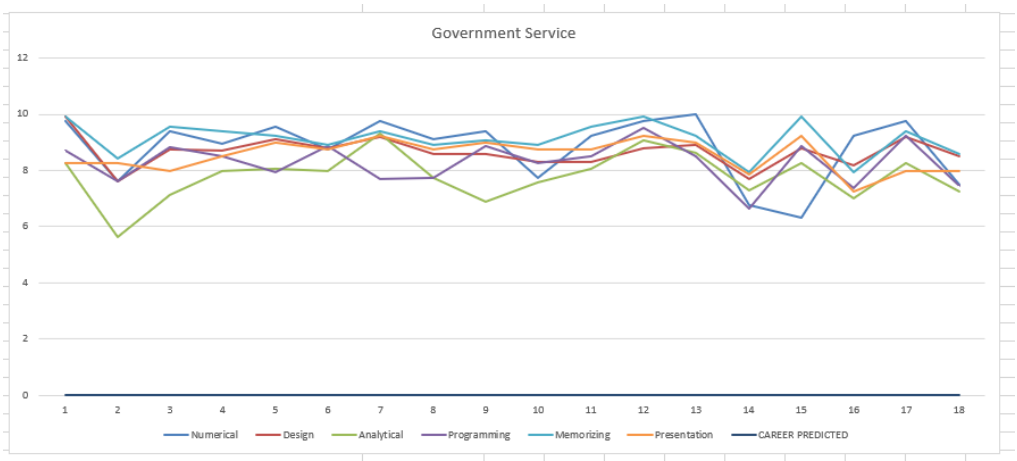


Figure 4.11: Feature Importance for Public servant/Upsc aspirant/Others

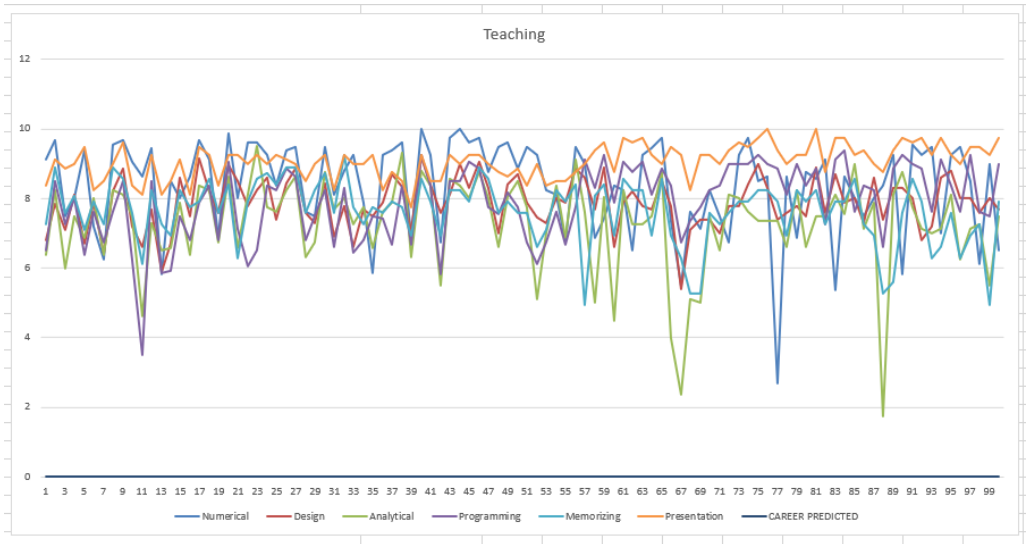


Figure 4.12: Feature Importance for Teaching

increase or decrease in the error. The feature is critical to our model if permuting the values significantly changes the imprecision. The best thing about this approach is that it works with all machine learning models. The method is model-independent, which gives you a lot of freedom. It is not based on any complicated mathematical formulae.

Moreover, the relevance of each feature with respect to each class is shown separately in the diagrams 4.10, 4.9, 4.12 and 4.11. The results show that programming and presentation skills are more prominent in the software industry. In the design industry, presentation and design skills are emphasised. For public sector jobs, all skills are equally represented, although memorisation is more prominent, and in teaching, presentation skills are more prominent.

Chapter 5

Future Scope

The abundance of data in educational databases makes predicting students' academic ability difficult. In higher education, student performance is critical. Our main goal was to assess student performance and predict their level of achievement and preferred field of work. Predicting students' performance and abilities from their academic data is very useful to help teachers and learners to improve their learning and teaching process, and it is also very useful for students to choose a better career in the future by identifying their outstanding abilities with our proposed model.

We concluded that by assigning different academic skills to technical subjects, the achieved skill scores can be predicted by using different ML techniques for the model, and a career field can also be predicted from the ML model. Pre-processing of the data is an important and challenging procedure.

A user-friendly interface that can be used by students or academic institutions to check their skills and find a preferred career option can be created as future work. In addition, the accuracy of the models can be further increased by advanced data balancing techniques and feature selection methods. Also, deep learning methods can be used instead of ML to increase the accuracy of performance.

Chapter 6

Conclusion

Predicting students' skills is one of the most important areas of research in today's digital world. To predict a student's career, many survey methods have been used in the past. However, it takes a long time to predict the outcome using these methods. In the modern digital age, a variety of computer techniques are used to estimate outcomes across a range of domains and it is important to have a predictive model that can reduce the uncertainty of the outcome for an imbalanced data set. Techniques such as SMOTE in conjunction with machine learning models have been used to predict student career outcomes.

In this work, we conducted a comparative analysis of the different skill prediction regression models and career prediction classifiers by combining oversampling SMOTE to evaluate the accuracy of career prediction for students. The field of work or recommended career can be predicted based on the skill scores obtained. Pre-processing the data was a vital and lengthy process.

We concluded that the Linear Regreession regressor for skill prediction and the Random Forest classifier for career prediction gave the best predictive performance with an accuracy of 0.999 and 0.962 respectively. For future work, a user-friendly GUI can be created for predicting new data. Also, many feature selection methods can be used in the future to improve the accuracy.

References

[1] T. Alam, C. F. Ahmed, S. A. Zahin, M. A. H. Khan, and M. T. Islam, "An effective recursive technique for multi-class classification and regression for imbalanced data," *IEEE Access*, vol. 7, pp. 127 615–127 630, 2019.

[2] Y. Nieto, V. Gacía-Díaz, C. Montenegro, C. C. González, and R. González Crespo, "Usage of machine learning for strategic decision making at higher educational institutions," *IEEE Access*, vol. 7, pp. 75 007–75 017, 2019.

[3] Abana and Ertie, "A decision tree approach for predicting student grades in research project using weka," *International Journal of Advanced Computer Science and Applications*, vol. 10, 07 2019.

[4] B. L. Sahu and D. A. Tiwari, "Student placement possibility prediction using naive bayes algorithm," *International Journal of Advance Research, Ideas and Innovations in Technology.*

[5] Hussain, S. Khan, and M.Q, "Student-performulator: Predicting students' academic performance at secondary and intermediate level using machine learning," *Ann. Data. Sci.*, 2021.

[6] Patel, Ankit, Mascarenhas, Savio, Thomas, Akhil, Varghese, and Ditty, "Student performance analysis and prediction of employable domains using machine learning," in *Proceedings of the International Conference on Recent Advances in Computational Techniques (IC-RACT)*, 2020.

[7] K. S. Roy, K. Roopkanth, V. Teja, V.Bhavana, and J. Priyanka, "Student career prediction using advanced machine learning techniques," *International Journal of Engineering and Technology*, April 2018.

[8] L. S. Maurya, M. S. Hussain, and S. Singh, "Developing classifiers through machine learning algorithms for student placement prediction based on academic performance," *Applied Artificial Intelligence*, vol. 35, no. 6, pp. 403–420, 2021.

[9] N. Khadilkar and D. Joshi, "Predictive model on employability of applicants and job hopping using machine learning," *International Journal of Computer Applications*, vol. 171, no. 1, pp. 37–41, Aug 2017.

[10] A. Giri, M. V. V. Bhagavath, B. Pruthvi, and N. Dubey, "A placement prediction system using k-nearest neighbors classifier," in *2016 Second International Conference on Cognitive Computing and Information Processing (CCIP)*, 2016, pp. 1–4.

[11] N. VidyaShreeram and D. A. Muthukumaravel, "Student career prediction using machine learning approaches," *I3CAC*, june 2021.

[12] D. Chaudhary, H. Prajapati, R. Rathod, P. Patel, and R. Gurjwar, "Student future prediction using machine learning," *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*, pp. 1104–1108, 04 2019.

[13] A. K. Akhilesh Kumar, "Human sentiment analysis on social media through naïve bayes classifier," *Journal of Scientific Research of The Banaras Hindu University*, 2022.

[14] B. Antony, "Prediction of the production of crops with respect to rainfall," *Environmental Research*, vol. 202, p. 111624, 2021.

[15] E. Fantin Irudaya Raj, *Implementation of Machine Learning Techniques in Unmanned Aerial Vehicle Control and Its Various Applications*. Cham: Springer International Publishing, 2022, pp. 17–33.

[16] Q. A. X. Victor Chang, Keerthi Kandadai and S. Guan, "Development of a diabetes diagnosis system using machine learning algorithms," *IJDST*, vol. 13, pp. 1–22, 2022.