CCT College Dublin

Assessment Cover Page

Module Title:	Strategic Thinking
Assessment Title:	Strategic Thinking Continuous Assessment 2
Lecturer Name:	James Garza
Student Full Name:	Anna Georgieva; Nick McNamara; Octavio Rieu
Student Number:	sbs23039@student.cct.ie; sbs23022@student.cct.ie; , sbs23024@student.cct.ie
Assessment Due Date:	14/11/2023 @ 11.55pm
Date of Submission:	14/11/2023

Nick McNamara
Anna Georgieva
Octavio Rieu

Declaration

By submitting this assessment, I confirm that I have read the CCT policy on Academic Misconduct and understand the implications of submitting work that is not my own or does not appropriately reference material taken from a third party or other source. I declare it to be my own work and that all material from third parties has been appropriately referenced. I further confirm that this work has not previously been submitted for assessment by myself or someone else in CCT College Dublin or any other higher education institution.

Prediction for lending loans

Word count: 7,091

- Github repository: Link
- Presentation : Link

Table of contents

List of Figures

Introduction

Word count: 1,003

This semester, our team should conduct a comprehensive assessment of the capstone project from Semester One, refining its strengths and identifying areas for improvement. Equipped with the insights and expertise gained from successfully completion of Class Assignment One in the first semester, combined with the advanced knowledge we have gathered throughout the second semester, we are in a position to enhance and augment the project. Our approach will involve the deployment of at least three machine learning models for two datasets and optimise their performance through hyperparameter tuning. We will validate the outcomes to ensure the robustness of our results. We are committed to advancing the project. Our efforts from the last semester were foundational, setting the stage with a well-crafted hypothesis and a suite of strategic questions, leading us toward meaningful solutions. The current semester's objective is to solidify these initial findings, ensuring that our position is both well-supported and resilient. There is mutual agreement within our team to continue our focus on loan defaulting within the financial services sector, a subject that remains exceedingly relevant and compelling.

The recent surge in the Central Bank's interest rates has triggered a domino effect, prompting local Irish banks to implement corresponding increases in their rates. This economic instability and inflation make our project not only timely but critical, providing a rich context for our continued exploration and analysis. According to the article of (O'Halloran, 2023), increasing interest rates have unsettled prospective homeowners, with a majority now opting for fixed-rate mortgages. The research, released by Myhome.ie, a property portal owned by The Irish Times, indicates that surging borrowing costs along with worries about housing availability are putting pressure on the real estate market. The survey reveals that *66%* of aspiring homebuyers are concerned about how escalating interest rates might impact their home purchasing capabilities, a significant increase of *20%* from the previous survey in March. Myhome.ie reports that the climb in interest rates has alarmed mortgage seekers, noting that over half of them are in search of mortgage agreements extending beyond five years, and three-quarters are intent on obtaining a fixed-rate mortgage. In response to the change of ECB, (Healy, 2023) *Bank of Ireland*, for example, has increased its variable mortgage rates by *0.25%* from 27th of October this year, whilst Allied Irish Banks(AIB) increased its variable interest rates for mortgages by *0.55%* for Private Dwelling Homes(PDH) since 14th August, 2023 (AIB, 2023).

AIB PDH Variable Rates:

PDH Variable Rates	Current Rate			New Rates (Effective from 14 August 2023)		
	>80%	50% - 80%	<50%	>80%	50% - 80%	<50%
LTV Variable Rate	3.50%	3.30%	3.10%	4.15%	3.95%	3.75%
Standard Variable Rate	3.50%			4.15%		

Figure 1: AIB PDH Variable Rates

Regardless of the gloomy forecast for mortgage lending, it appears that first-time buyers purchasing power remains solid, no matter that the mortgage activity slowed down in Q2 (Murphy, 2023). The drawdown value is €284,397, which is at its peak since 2003, and

there are a total of 9,896 new approved mortgages. The number of mortgage loans dropped by 5.7% and by value with 3.6% in comparison to Q1, 2023.

We attempted to find datasets that represent a sample of loan borrowers for the Irish market, especially data, which is recent and actual with the current mortgage climate, however it has proven difficult as there are no public repositories available for use. If we did manage to find this kind of dataset, it would have been in juxtaposition with our hypothesis, because our findings supported the theory that loans are granted to borrowers, who have prior credit history and track record of a stable repayment capability. An important aspect to examine in our review of 'Dream Housing Finance Company' (Kaggle datasets, 2023) loan application process is whether their loan approval criteria align with the established standards for the Irish market.

In the previous semester, we conducted an examination of Lee and Lee research from 2018, which confirms the hypothesis that credit history plays an important role in the loan approval process for a variety of groups of borrowers, such as individuals, partnerships, corporations, clubs, societies, and trusts. Their findings emphasise the significance of credit history in financial decisions. According to Lee and Lee, a credit score, which reflects a person's previous and ongoing credit, can forecast their probability of repaying financial obligations. Lenders leverage these scores to assess the viability of a loan applicant, to decide on the interest rates, and to determine the extent of credit limits. The credit score is influenced by several factors: payment history on various accounts such as credit cards and mortgages accounts for *35%* of the score; legal and financial blemishes like lawsuits, bankruptcies, and court judgments make up *30%*; and the effect of opening multiple new credit accounts suggests a negative impact, as it might signal repayment challenges, contributing to *15%* of the score. Our capstone project hypothesis corresponded with the conclusions drawn by Lee and Lee (2018).

Nevertheless, considering the statistical overview for mortgage lending in Ireland, can the findings of Lee and Lee (2018) be considered universally applicable and serve as foundational principles across diverse banking systems? Does the discretion in approving loans depend on the economic climate of the country or is it tailored to the financial circumstances of individual applicants? We will evaluate all datasets features again and determine what factors weigh in these decisions. In addition to analysing Dream Housing Finance Company datasets, we will also explore the '*Loan Default Prediction*' dataset from Kaggle. This particular dataset allows us to confront a critical and industry-relevant machine learning challenge—the prediction of loan defaults. It encompasses a distinctive set of 255,347 records across 18 different attributes, providing a rich testing ground for enhancing our predictive modelling skills. By examining both datasets, we aim to further validate our hypothesis for a second time. The newly included datasets have features, which are lacking in the previous datasets and are quite interesting to analyse, such as '*CreditScore*', '*Months Employed'*, '*NumCreditLines'*(the number of credit lines the borrower has opened), '*InterestRate'*(a feature which is essential when evaluating the decision to lend a loan to an individual), *'DTIRatio'*(Debt-to-income ratio, which indicates the person's debt in comparison to their income), *HasMortgage*, *'HasCoSigner'*(whether the loan has co-sign) and lastly, but not the least by importance *'Default'*.

Business understanding

Word count: 712

For this project we utilised the CRISP DM methodology, an accepted standard, in data mining. Overall all this methodology will allow us, through our research, to minimise the lending risk associated with loan defaulters. Our strategy involved gaining an understanding of the challenges within the lending industry. Then we proceeded to gather and analyse data such, as credit history, marital status and financial indicators of borrowers.

The first dataset consists of borrowers whose loan status is approved or rejected and the second one consists of borrowers, who defaulted on their loan or not. We will evaluate all the variables in the datasets and determine which ones have the highest impact on the decision to approve or decline a loan. The data mining process will involve data preparation, such as cleaning the datasets from NaN or missing values, data normalisation, analyse the relationships between the variables, train and test the datasets with a couple of models such as Random Forest Classifier, Linear Regression, Artificial Neural Network, Support Vector Machine, and Support Vector Classifier and based on the results make a conclusion, which model is the best performing one.

Hypothesis

For this semester, our focus will be on analysing two distinct datasets. The first dataset contains loan applications with their respective approval statuses, which seemed to be influenced by the applicants' Credit History. The second dataset offers insight into which loan applications resulted in defaults. Our objective is to assess the potential risk faced by banks when approving loans to clients. Through our analysis, we aim to validate or refute the hypothesis that the Credit History attribute is a significant predictor in determining loan approval outcomes.

General goal

In our efforts to reduce biases and maintain objectivity when working with data we have implemented an approach that combines procedural safeguards. This includes using datasets to minimise the impact of information, applying cross validation techniques to ensure our models are applicable, in various scenarios and prioritising transparency in our algorithmic processes so that others can scrutinise our machine learning methods. Upholding considerations is of importance to us which involves addressing any imbalances in the data and striving for fairness. We have continuously evaluated our methods and findings to incorporate insights ensuring that our conclusions are based on unbiased analysis.

Success criteria/indicators

Success of this project would be determined by finding a correlation between the individual customer circumstances and a proclivity to default on a loan. To achieve the best predictions and results with higher accuracy, we will use machine learning models and

algorithms such as linear regression, random forest algorithms and others to predict the outcome. We would hope one of these algorithms would provide us with test results in excess of 98%.

Technologies

In addition to Python and Jupyter Notebook, our team used Google Colab, which provides a cloud-based Python programming environment with robust computational capabilities. For team communication and coordination, Google Meet has been our virtual meeting room. It has allowed for effective and efficient remote collaboration, enabling us to conduct regular check-ins, discuss project progress, and make collective decisions. The combination of Google Colab and Google Meet, along with GitHub Desktop for version control, has created a cohesive and dynamic development environment that is adaptive to our needs as a distributed team working on a data-intensive project.

Libraries

The libraries used for this project include Numpy, Pandas, Matplotlib.pyplot, Plotly express, Bokeh, Seaborn, Counter, ListedColormap, and mean_squared_error. For preprocessing the data, the following libraries are employed: LabelEncoder, StandardScaler, SimpleImputer, KNNImputer, SMOTE and LinearSegmentedColormap. The train_test_split function is used for splitting the data into training and testing sets. To assess model accuracy, the following metrics and functions are used: cross_val_score, accuracy_score, confusion_matrix, and recall_score, F1-score and make_classification. Machine learning model libraries are: Support Vector Machine, Logistic Regression, Random Forest Classifier, KNeighborsRegressor, Support Vector Machine, Artificial Neural Networks.

Datasets and source

The first dataset, which we use for this project is called '*Home Loan Approval*' and the source is from Kaggle (Konapure, 2023). This dataset is owned by a finance company, which lends loans to people who want to buy properties in rural, semi-urban and urban areas. They need to automate the approval process by segmenting the customers' eligibility. The second dataset called '*Loan Default Prediction Dataset*' is also from Kaggle, however it contains data of individuals who defaulted on their loans (NIKHIL, 2023).

Data understanding

Word count: 1,841

Loan Approval Cleaning

Pandas package for Python was used to analyse and handle the datasets. This library is suitable and efficient in handling data (Müller and Guido, 2017). NumPy is the numerical package for Python, and is also used. We select the seaborn library for data visualisation, which generates fascinating and practical statistical visualisations like heatmaps, bar charts, pie charts, scatter plots, and others. As an alternate library for producing high-quality graphs and charts, Matplotlib, Bokeh and Plotly express were also imported.

We have two types of datasets belonging to 'Home Loan Approval'. One CSV is used to train data and one CSV is used to test data, so we can evaluate the success of the trained data. We use loan = pd.read_csv('loan_train.csv) code and load the datasets into pandas dataframe and we name it *'loan'*. The CSV files are in the same GitHub Desktop directory as the Jupyter notebook, which contains this code.

We run *loan.head*() to display the first 5 rows and all columns, so we can quickly check what kind of data we work with. The attributes in this dataset are categorical *Loan_ID, Gender, Married, Dependents, Education, Self_Employed, Credit_History, Property_Area, Loan_Status* and numerical attributes - *ApplicantIncome, CoapplicantIncome, LoanAmount* and *Loan_Amount_Term*. It is a common issue to handle missing values in a dataset and we will use the **isnull** method in order to identify them (Harrison, 2019). This dataset is not a high-dimensional one as we have only 11 attributes to analyse and 614 observations. We should exclude Loan status, because it is a dependent variable in this data mining process and Loan ID, because it doesn't have impact on the analysis and it is an identifier of the loan application. The data type of '*ApplicantIncome*' is int64 and it is converted to float64.

We checked the count of missing values.

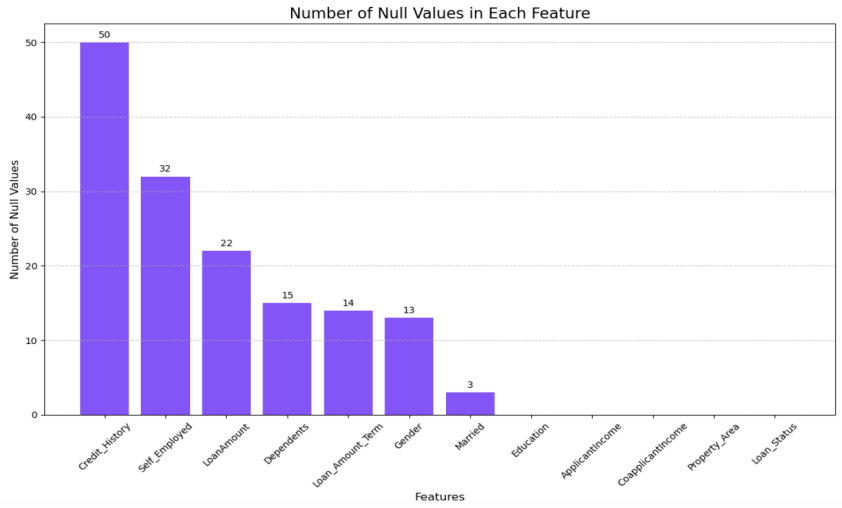

Figure 2: Totals of null values

In the previous semester we dropped any row which contained a missing value. For this capstone project, it has been decided that instead of dropping all unknown values, it will be a

better approach to impute all of them, because this could impact the overall results with such a small dimensional dataset. It is essential to understand the importance of these features and the approach is taken accordingly. For example, at the beginning of this report, we mentioned that *'Credit History'* is the most important influencer on credit loan approvals. There are 84 rows with missing values in the specified columns: *'Gender'*, *'Married*', '*Dependents'*, '*Self_Employed*', *'LoanAmount', 'Loan_Amount_Term'*, but with '*Credit_History'* data present. If the unknown values are dropped from all these features, we can impact significantly on the machine learning models outcomes. This is the reason why, by employing SimpleImputer, the most frequent values were used for the categorical features, such as *'Gender', 'Married'*, *'Dependents'* and *'Self_Employed'* and the mean or median for numerical data - *'LoanAmount'* and *'Loan_Amount_Term'*. The sklearn.impute.SimpleImputer in scikit-learn provides a convenient way to impute missing values in a dataset. By default, for numerical data, it can replace missing values using the mean or median of each column, while for categorical data, it can use the most frequent value (mode). As of sklearn's documentation in 2023, these strategies are widely applied due to their simplicity and effectiveness in many scenarios.

```
There are 84 rows with missing values in the specified columns but with Credit_History data present.
```

Figure 3: Checking missing values in key columns and cross-reference the count in Credit History

However, for a critical feature like *'Credit History'* that has 50 missing entries, a more sophisticated approach may be necessary. Given the feature's significance, a multivariate imputation method could be more appropriate. This would involve using the entire dataset and employing all available features to estimate the missing *'Credit History'* values, thus preserving the underlying data structure and relationships (scikit-learn, 2022). In the section for 6.4.4. Nearest neighbours imputation article, it states that the KNNImputer class utilises the k-Nearest Neighbors technique to input missing values in a dataset. It employs the 'euclidean_distances' metric by default to estimate the closest neighbours, even when there are missing values. For imputing a specific feature, the algorithm considers values from the nearest 'neighbours' that have non-missing values for that feature. These neighbour values are then averaged, either equally or weighted by their distance, to fill in the missing entry. If multiple features are missing from a sample, the set of neighbours used for imputation may vary for each feature. In situations where the available neighbours are fewer than 'n_neighbors,' the overall mean of the feature across the training set is used. However, if there is at least one neighbour within a certain distance, the average of these neighbours are either weighted or not and then is used for imputation. Features that are consistently missing across the training data are excluded in the transformation process.

In our prior project, we employed dimensionality reduction techniques on a dataset and removed 134 observations. This method, however, may not represent the most accurate approach to managing the missing values present within the dataset. Our analysis reveals that out of these observations, 84 rows are with missing values across various features, with the exception of 'Credit History,' where complete data exists for the corresponding rows. This suggests the need for a more careful strategy in data handling to ensure the integrity and utilisation of the dataset.

Cleaning Loan Default

We created a new Jupiter notebook and imported Loan_default.csv. There are 17 features in the dataset.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 255347 entries, 0 to 255346
Data columns (total 18 columns):
 #   Column           Non-Null Count   Dtype
---  ------           --------------   -----
 0   LoanID           255347 non-null  object
 1   Age              255347 non-null  int64
 2   Income           255347 non-null  int64
 3   LoanAmount       255347 non-null  int64
 4   CreditScore      255347 non-null  int64
 5   MonthsEmployed   255347 non-null  int64
 6   NumCreditLines   255347 non-null  int64
 7   InterestRate     255347 non-null  float64
 8   LoanTerm         255347 non-null  int64
 9   DTIRatio         255347 non-null  float64
 10  Education        255347 non-null  object
 11  EmploymentType   255347 non-null  object
 12  MaritalStatus    255347 non-null  object
 13  HasMortgage      255347 non-null  object
 14  HasDependents    255347 non-null  object
 15  LoanPurpose      255347 non-null  object
 16  HasCoSigner      255347 non-null  object
 17  Default          255347 non-null  int64
dtypes: float64(2), int64(8), object(8)
memory usage: 35.1+ MB
```

Figure 4: Descriptive table of datasets

This datasets doesn't contain any missing values as stated above and the data types are, object, int64 and float64. This is a high-dimensional dataset consisting of 255,347 observations. Our analytical focus is on the subset of data related to individuals who have defaulted and not defaulted on mortgage loans, distinctively excluding any data related to personal, educational, automobile loans or other types of loans.

```
Counts of unique values in the 'LoanPurpose' column:
Business     51298
Home         51286
Education    51005
Other        50914
Auto         50844
Name: LoanPurpose, dtype: int64
```

Figure 5: Checking the count of Mortgage Loans

There are **51,286** mortgage loans observations in this dataset and a new subset of a data frame is created '***defaultloan***'.

Data Visualisations *'Home loan approvals'*

We need to check approved loans vs rejected loans. Last semester we identified 69.2% of the loan applications were approved, and this semester we can see 68.7% approved applications, whereas 30.8%(last semester) of the applications were rejected and this semester 31.3%.

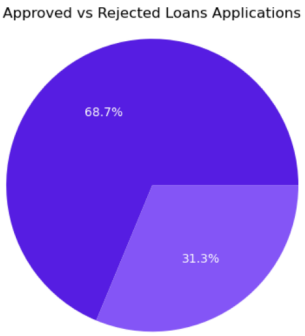

Figure 6: Percentage of approved and denied loan

The current treatment of missing values decreased by 0.5 points for *'Approved'* category and increased with the same percentage for '*Rejected'*. Bokeh and Plotly express are imported in Jupyter notebook for further analysis.

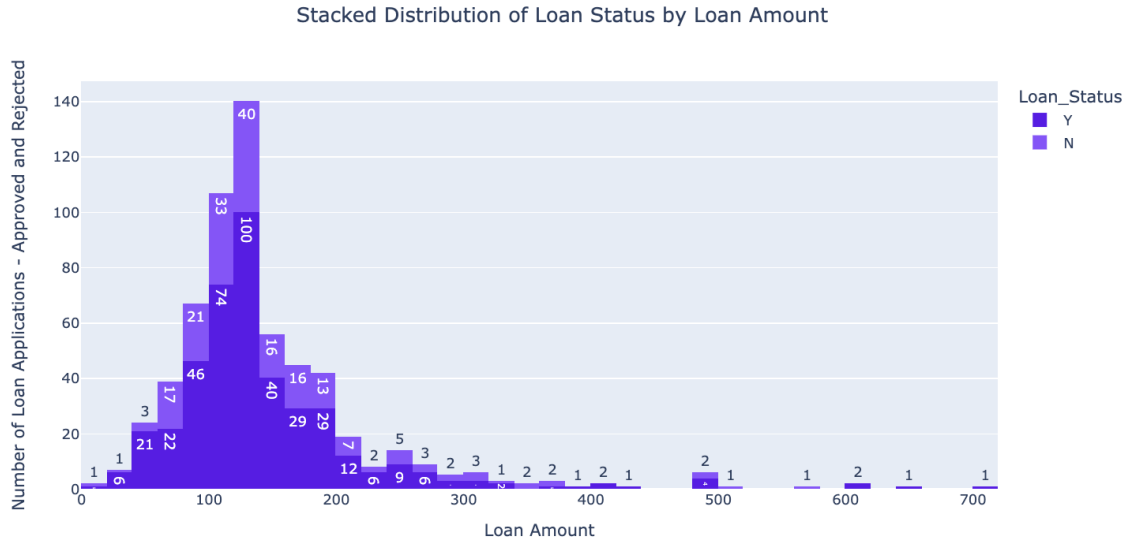

Figure 7: Stacked Distribution of Loan Status by Loan Amount

We created a stacked interactive chart, which displays clearly the distribution of *'Loan Amount'* among all of the applicants. Another stacked interactive chart was created for '*Applicant Income*' and the distribution shows that most loan applications are in the lower band of income.

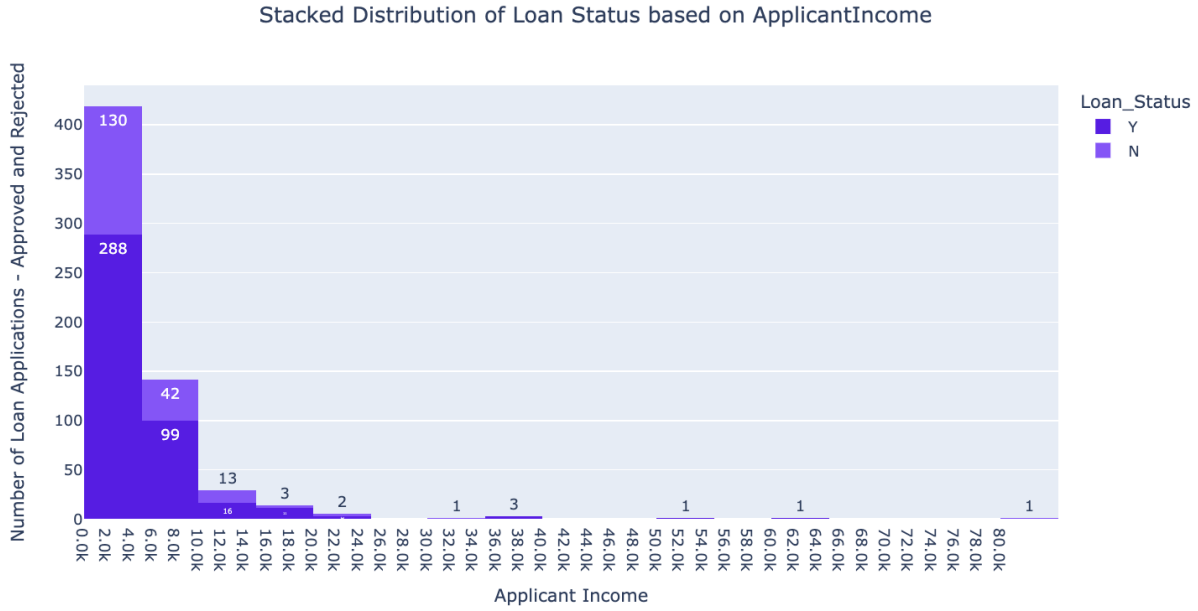

Figure 8: Stacked Distribution of Loan Status by Applicant Income

We analysed the distribution of CoapplicantIncome and it shows that most of the applicants do not have co-applications.

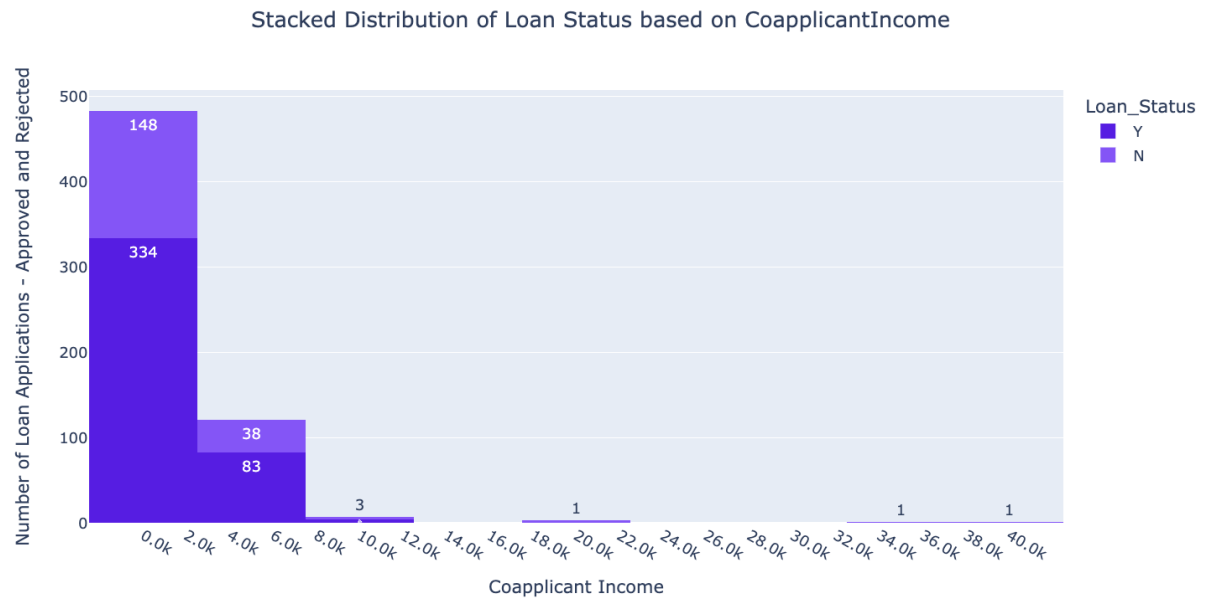

Figure 9: Stacked Distribution of Loan Status by Coapplicant Income

Most of the applicants have Credit history in place when they submitted their mortgage loan applications.

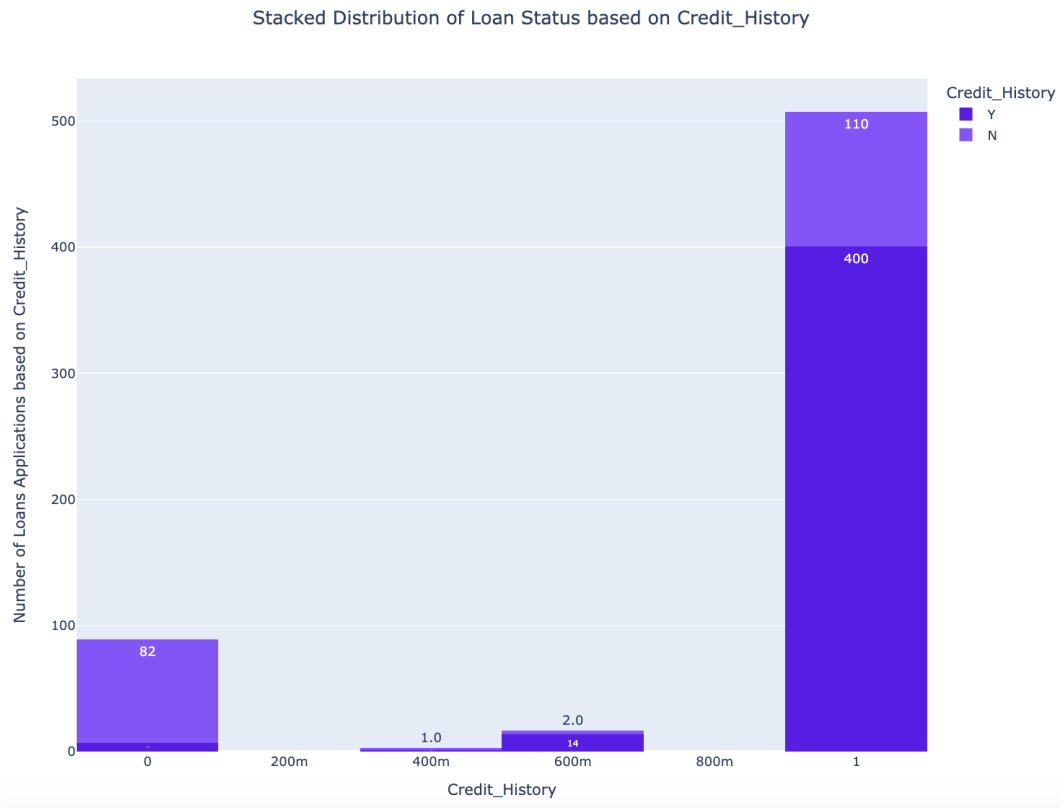

Figure 10: Stacked Distribution of Loan Status by Credit History

When we implemented K-Nearest Neighbors imputation on 50 values in Credit history , 18 of these entries were estimated and filled with values that represent the median of their closest neighbours in the dataset. This method leverages the similarity between entries,

ensuring that the imputed values are consistent with the underlying data distribution. The rest of the Credit History values were imputed with *'0'* or *'1'*.

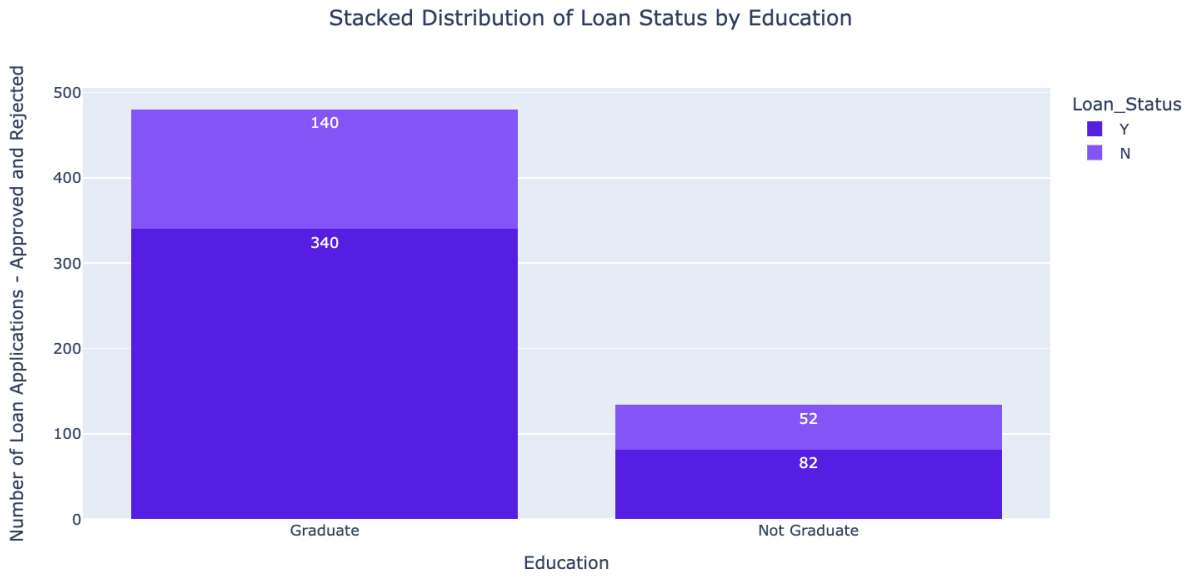

Figure 11: Stacked Distribution of Loan Status by Education

The above data is with categorical variables *'Education'* and *'Loan Status'*. We compare the two variables, so we can see that there is an apparent relationship between the education vs approval of the loan application. The same observations are with married couples, where we can see that Married couples are more likely to receive loan approval.

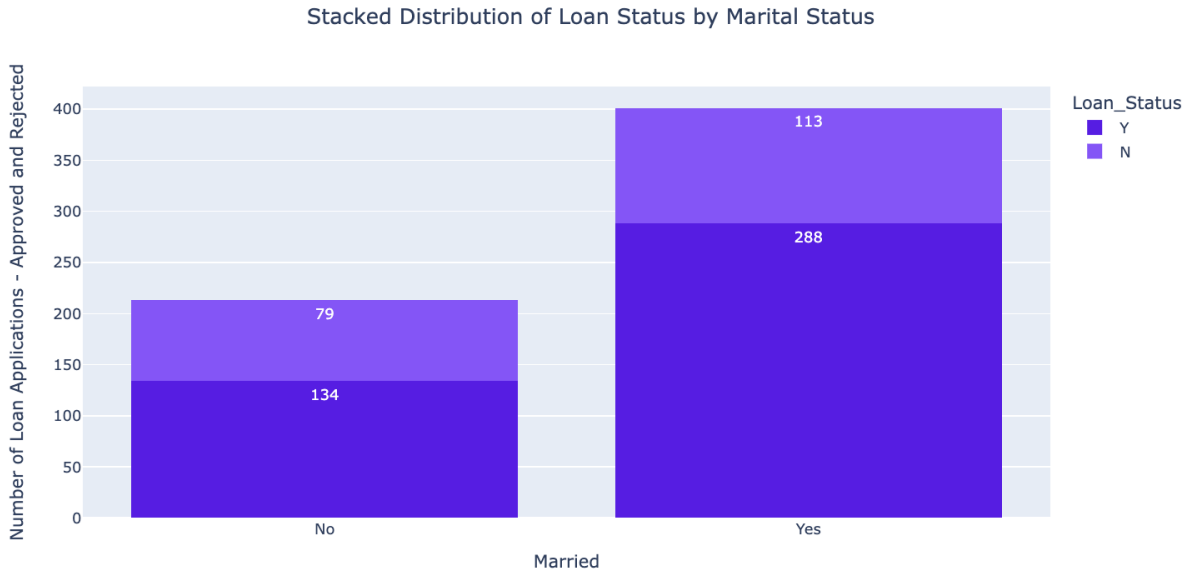

Figure 12: Stacked Distribution of Loan Status by Marital status

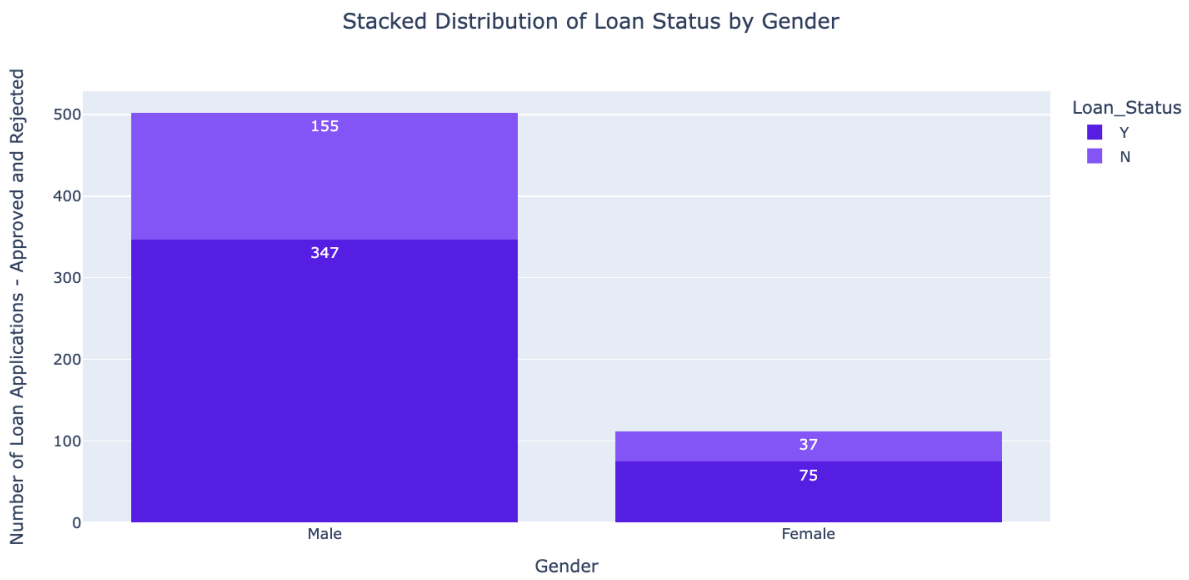

Figure 13: Stacked Distribution of Loan Status by Gender

We used *plotly express* in order to visualise the above chart where we have done comparison of Males' vs Females' loan application approval status and we can see that males' approved applications prevail. We compared the variable 'Dependents' vs 'Loan status'. 247(which is 40% of the total observations) approved loan applications belonging to individuals with no children.

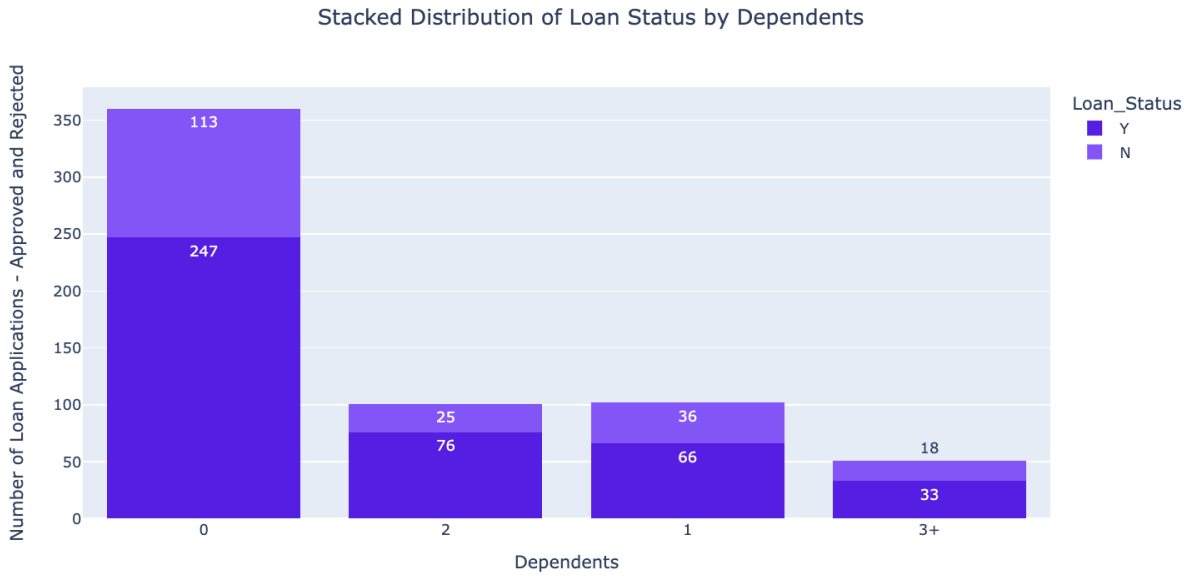

Figure 14: Stacked Distribution of Loan Status by Dependents

Data Visualisations 'Loan Default'

In the bar chart generated from the *'defaultloan'* dataframe, it is evident that within the subset of loans filtered by the presence of a mortgage, approximately 10% have been marked as defaulted. The analysis indicates a progressive rise in the rate of defaults correlating with

larger loan sizes, implying that higher borrowing amounts are associated with an increased risk of default. The likelihood of default increases with the borrower's youthfulness, low income, high interest rates and lesser time being employed.

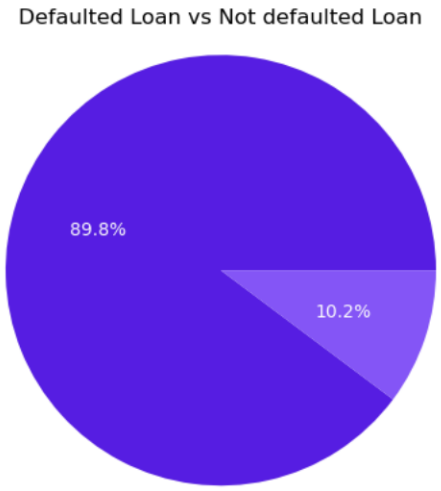

Figure 15: Defaulted and not defaulted loans

Data preparation for 'Loan Approvals'

A crucial step in pre-processing data for a machine learning model is feature scaling. Since the machine learning model sees only numbers, if the spectrum of numbers is quite broad from tens to thousands, the assumption will be that the higher ranging numbers have a superiority (Roy, 2020).

For pre-processing the data, StandardScaler was used. By using this method, all the numerical values are transformed into values between 0 and 1. It is important to reduce the dimensionality of the data, because by compressing it we get a better representation of each feature (Müller and Guido,2017).

StandardScaler *is applied on:*

'ApplicantIncome', 'CoapplicantIncome', 'LoanAmount','Loan_Amount_Term', 'Credit_History'.

Label-encoder is used to encode all the categorical variables to numerical values, so the dataset is balanced and ready to be used for the machine learning models (Scikitlearn, 2019). *'Loan_Status'* column is the dependent variable encoded, too. The rest of the features are used to recognize the factors impacting banks' decisions to approve or reject customers' loan applications.

Label-encoder *applied on:*

'Gender', 'Married', 'Dependents', 'Education', 'Self_Employed','Property_Area', 'Loan_Status'

Following the data preprocessing stage, we created a Spearman correlation heatmap to identify the relationships between different variables within our dataset. This visual analysis revealed a strong correlation of 49% between *'Loan_Status'* and *'Credit_History'*. It is worth noting that compared to the previous semester's analysis, which showed a slightly stronger correlation of 53%, the current correlation has shown a reduction by 4 percentage points. This suggests that the modifications applied to the data processing methods in the current process may have influenced the relationship between these variables. Additionally, the

heatmap provided insights into the relationship between *'Applicant Income'* and *'Loan_Amount'*, where a significant correlation of 50% is observed. This substantial correlation coefficient indicates a moderate-to-strong linear relationship, suggesting that as the income of the applicants increases, there is a tendency for the loan amount to increase correspondingly.

By analysing the interactive plots and map, we can identify that there is a strong positive correlation between the approval of the loan amount and the credit history. The interactive plots showcase that males receive more loan approvals than females. Married couples get more approved applications than single individuals. If there is a mortgage application, the property area matters, as there are more approvals for semi-urban areas. Applicants who don't have children or are not self-employed are more likely to receive a loan approval.

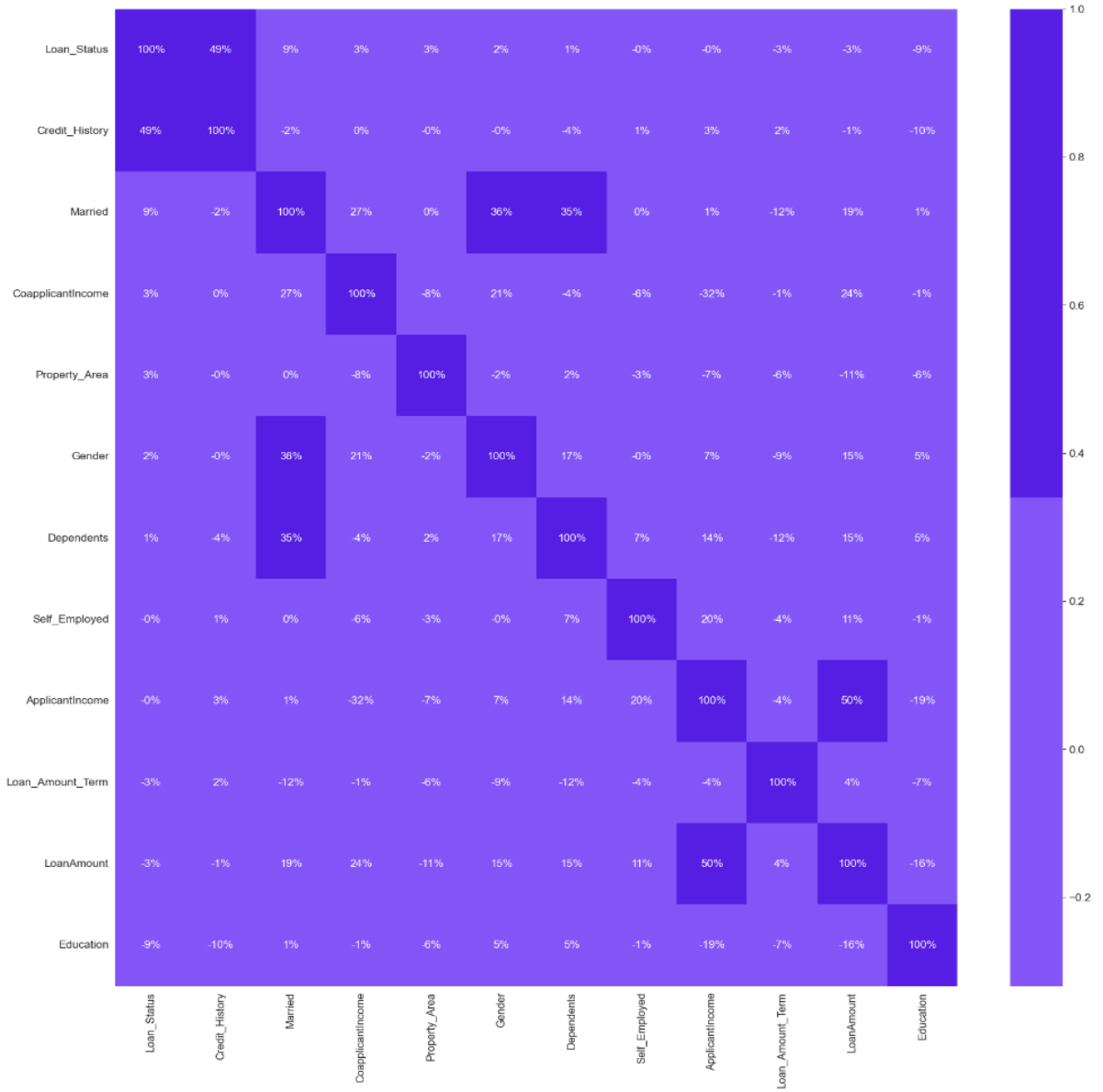

Figure 16: Spearman correlation

Data preparation for 'Loan Default'

For the numerical variables, the pipeline first imputes missing values with the column's mean. Following this, it standardised these values to have a mean of zero and a unit variance. This standardisation is essential for many machine learning models as it normalises the scale of the data. On the other hand, for the categorical variables, the pipeline initially imputes missing values with a placeholder 'N/A'. It then applies one-hot encoding, which transforms these categorical values into a numerical format by creating binary columns for each category.

Model implementation

Word count: 1,954

Loan Approval

After conducting this thorough data visualisation analysis, the next step in the process is to drop the dependent variable *'y' (Loan_Status)* from the independent variables *'X'(the rest of the features in the dataframe)*. ***Random forest classification*** is initiated and after fitting, the model can list the features by importance. The top 3 features are selected. After applying two different methods - Spearman correlation and Feature Importance, we are confident enough to proceed with the implementation of a machine learning model, by selecting the most strongly associated independent variables to the target variable, *'Credit History', 'Applicant Income'* and *'Loan Amount'*. These numerical variables are appropriate for prediction of the dependent variable 'y' = 'Loan_Status'.

The Random Forest algorithm is suitable at handling both categorical and continuous variables, performing well on both classification and regression tasks. This machine learning model operates by employing diverse random subsets of observations and features to construct multiple decision trees. Each tree independently classifies the data, contributing to the overall decision-making process. Upon finalising predictions, the algorithm aggregates the outcomes of these individual trees, typically through majority voting for classification or averaging for regression, to derive a more accurate and reliable final prediction. This approach takes into account the strengths of multiple trees, reducing the likelihood of overfitting and enhancing the model's generalisability to new data. (Shafi, 2023)

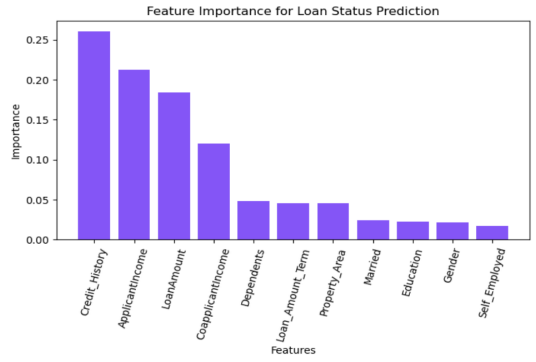

Figure 17: Feature importance

At the beginning of our project, it became evident that the dataset is highly- imbalanced where there are 68.7% approved loan applications and 31.3% rejected loan applications. If we

do not apply SMOTE, the model predictions and classification could lead to bias results and not satisfactory outcomes (Brownlee, 2020). SMOTE uses k-nearest neighbours, draws a line between them, and chooses a point along that line (Harrison, 2019). SMOTE is ideal for generalising on imbalanced data where the minority of the data is oversampled, meaning that new cases of this sample are generated, but it doesn't increase the number of the majority sample (Microsoft, 2021).

Random Forest Classification (SMOTE) has been used and the results are as follows:

- Accuracy: 0.66
- Precision: 0.78
- Recall: 0.70
- Cross-validations: 0.79

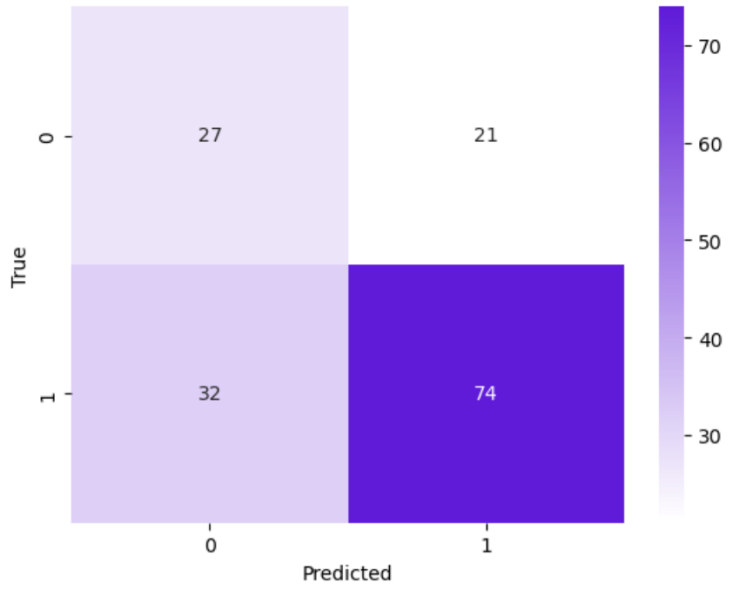

Figure 18: Random Forest Classification confusion matrix (SMOTE)

The confusion matrix shows that true positives(TP) are 74 cases, where the model correctly predicted that an applicant's loan would be approved, and this was indeed the case. True negatives(TN) is 27, where it was predicted that an applicant's loan would not be approved, which aligned with the actual outcomes. The confusion matrix further reveals that there were 32 false negative cases, indicating instances where the model incorrectly predicted that applicants' loan applications would be rejected when, in fact, they were approved. Additionally, there were 21 false positive cases, where the model wrongly predicted loan approval for applicants whose applications were actually rejected. Cross-validation accuracy is 0.79. It can be concluded that, given the overall outcomes of the confusion matrix and the cross-validation accuracy, refinement of the machine learning model is necessary. This includes addressing the false positives and false negatives to enhance the model's precision and reliability.

The next machine learning model in our project is **Logistic Regression** with SMOTE. By utilising SMOTE, we improve the model one step further and we ensure that there is a balanced set of class representatives. Logistic regression examines the relationship between the existing variables and the dependent variable. The results can lead to straightforward conclusions between the two options. (Lawton, 2022) Logistic regression(SMOTE) estimates probabilities and utilizing class weights proportional to class distribution. Class weights essentially determine the degree of penalty the algorithm incurs for erroneous predictions associated with a particular class (Yadav, 2020).

Logistic Regression (SMOTE) has been used and the results are as follows:

- Accuracy: 0.78
- Precision: 0.79
- Recall: 0.93
- Cross-validated Accuracy: 0.80

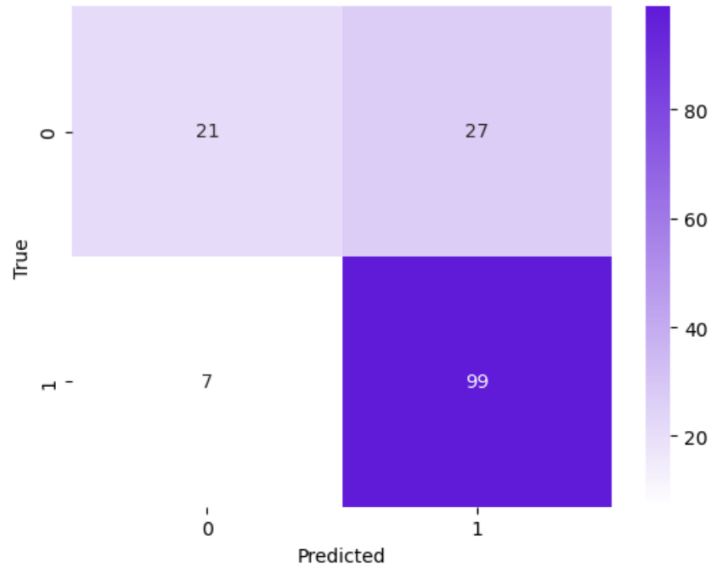

Figure 19: Logistic regression confusion matrix (SMOTE)

The confusion matrix effectively showcases the performance of our predictive model. It reveals 99 true positive cases, where the model correctly predicted loan approvals, and these predictions were accurate. Similarly, there were 21 true negative cases, where loan rejections predicted by the model were indeed observed. However, the matrix also indicates areas for improvement: there were 7 false negative cases, where loan rejections were predicted but the applicants actually had their loans approved, and 27 false positive cases, where the model incorrectly predicted approvals for loans that were actually rejected.

The upcoming phase we applied **Support Vector Classification**, augmented with SMOTE. We firstly tested the model with SMOTE only and in order to improve its performance we used hyperparameter tuning with GridSearchCV. Support Vector Machine is observed to perform well, just like Random Forest, in regression and classification tasks (scikit learn -

SupportVectorMachines, 2018). SVC tries to fit a line between different classes and maximise the distance from the line to the points of the classes. A robust separation between classes is achieved in this way (Harrisson, 2019). Testing and training sets are splitted 50/50.

Support Vector Classification (SMOTE) results are as follows:

- Accuracy: 0.79
- Precision: 0.78
- Recall: 0.95
- Cross-validated accuracy: 0.80

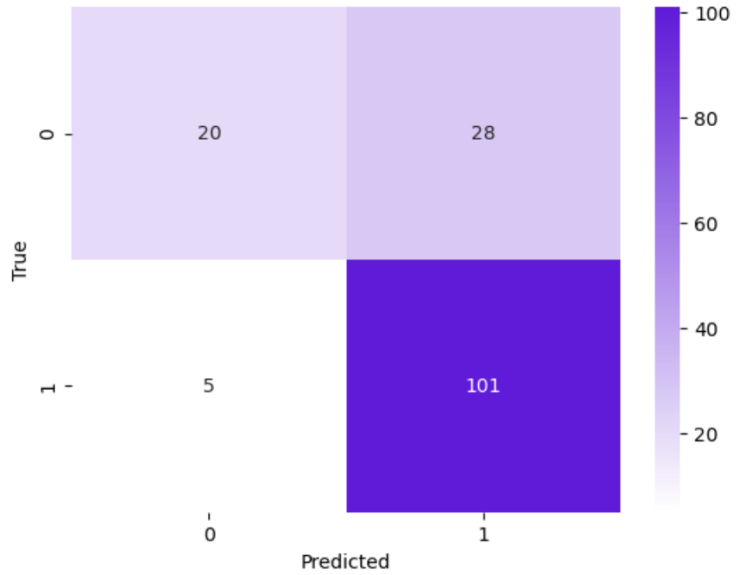

Figure 20: Support Vector Classification (SMOTE)

The confusion matrix clearly demonstrates the outcome of our predictive model. It identifies 101 instances as true positives, accurately forecasting loan approvals. Likewise, 20 cases were correctly classified as true negatives, with the model's predictions of loan rejections aligning with actual outcomes. Yet, the matrix also highlights opportunities for improvement: there were 5 instances of false negative, where the model erroneously forecasted loan rejections but the loans were approved, and 28 instances of false positives, where it inaccurately predicted loan approvals that were in fact rejected. These are results based on default SCV parameters. However, in order to refine its performance, we used **GridSearchCV**. GridSearchCV is a cross-validation technique, which is trying to find the optimal parameter values from a given set of parameters in a grid. It is a hyperparameter tuning process to determine the optimal values in a specific model. It uses different combinations of parameters and ensures that the model is performing at its peak efficiency. The right balance of parameters can significantly impact its performance positively (Great Learning, 2020)

The Support Vector Machine, enhanced with GridSearchCV, stands out with its impeccable accuracy, precision, and recall scores, positioning it as the top-performing model across these three crucial metrics among the evaluated models. However, achieving perfect

scores in every metric raises concerns about overfitting, particularly if these results pertain solely to the training set.

Support Vector Classification (SMOTE and GridSearchCV) results are as follows:

- Accuracy: 1
- Precision: 1
- Recall: 1
- Test accuracy: 1
- Cross-validated accuracy: 1

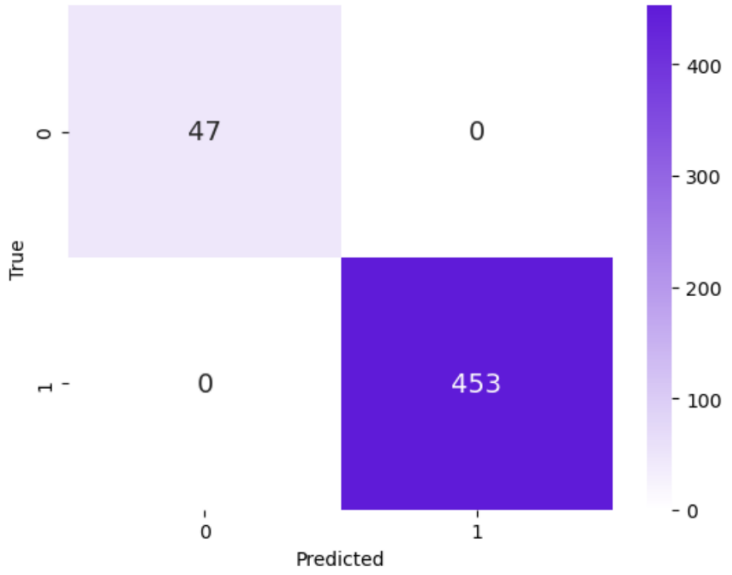

Figure 21: Support Vector Classification (SMOTE and GridSearchCV)

Support Vector Classification with SMOTE and GridSearchCV performs the best. It identifies 453 instances as true positives, accurately forecasting loan approvals. Likewise, 47 cases were correctly classified as true negatives, with the model's predictions of loan rejections aligning with actual outcomes and no false positives or false negatives at all.

Hyperparameter tuning is also applied with Random Forest Classification.

Random Forest Classification (SMOTE and GridSearchCV) results are as follows:

- accuracy: 0.91
- precision: 0.91
- recall: 1.00
- Accuracy on test predictions: 0.90
- Cross-validated accuracy: 1

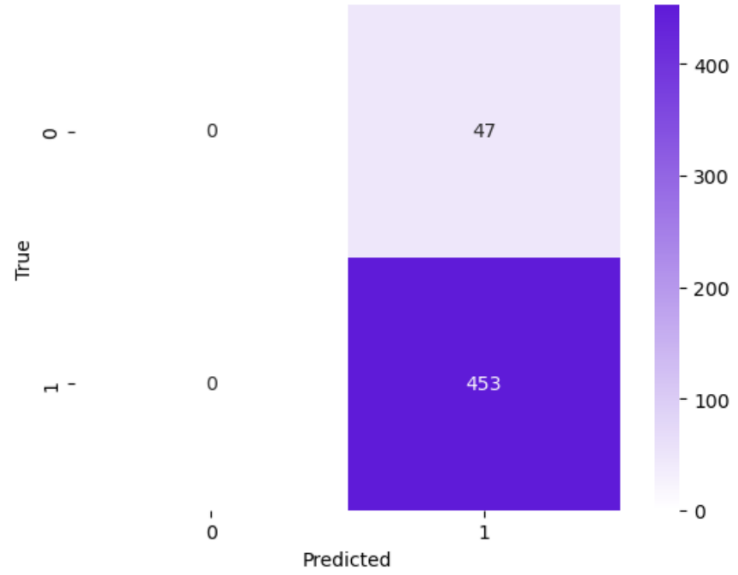

Figure 22: Random Forest Classification (SMOTE and GridSearchCV)

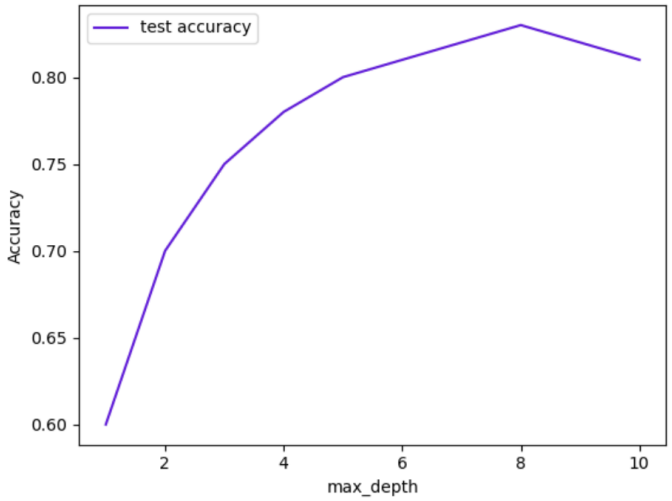

Figure 23: Random Forest Classification - Impact of Max Depth on Model Accuracy

Random Forest Classification with SMOTE and GridSearchCV displays the outcomes that there are 453 instances as true positives, accurately forecasting loan approvals. There are no true negatives and false negatives, however there are 47 false positive instances, where the model inaccurately predicted loan approvals when it should not have. Cross-validated accuracy is 0.99. Table 21 showcase the impact of max depth on model accuracy. After the model was hypertuned, there is a high performance observed across different max_depth values, which suggest that the model captures well the underlying patterns in the data effectively, however since all the values seems with perfect score, it can be inferred that the model is overfitting, especially when we have such as a small datasets as Loan approval one.

Artificial Neural Networks Machine Learning Model has also been used for this project. Artificial neural networks process information through multiple layers of mathematical

computations. These networks typically consist of numerous artificial neurons, known as units, which can range from tens to millions in number. These units are organised in several layers. The initial layer, known as the input layer, gathers diverse types of data from external sources – this is the information that the network is designed to analyse or learn from. Following the input layer, the data is passed through one or more hidden layers. The primary function of these hidden layers is to modify the input data into a format that is suitable for the output layer to utilise (Marr, 2018). When running this model, the dataset is splitted into 20% for testing purposes and 80% for training purposes.

Artificial Neural Networks results are as follows:

Training accuracy: 100% with loss 1.2141e-05

Such a small loss could be a sign for overfitting and that the model learned on training data very well. The accuracy score indicates that the model has perfectly classified all training data.

Testing accuracy: 99.50% with loss 0.0319

The loss here is higher than the one of the training set and this is usual because the model is performing well on training data. Since the loss is relatively small, this indicates that the performance is good. High accuracy suggests that the model generalises very well on unseen data. Even though there's a slight chance of overfitting due to perfect training accuracy, the high test accuracy tends to mitigate this concern. Overall, the model performed excellent.

Artificial Neural Network results are as follows:

- Training Accuracy: 100.00% - 0s 1ms/step - loss: 1.2141e-05 - accuracy: 1.0000
- Testing Accuracy: 99.50% - 0s 2ms/step - loss: 0.0319 - accuracy: 0.9950
- Accuracy: 0.99
- Precision: 1.00
- Recall: 0.99

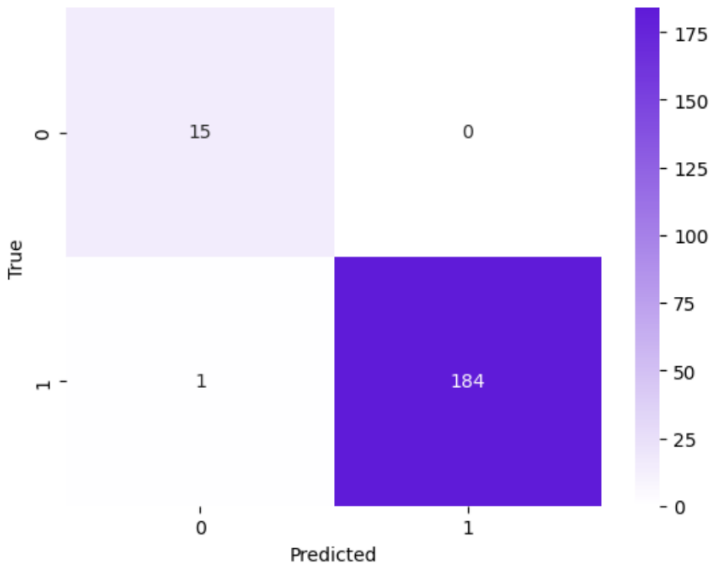

Figure 24: Artificial Neural Networks - Confusion Matrix

The Artificial Neural Networks Confusion matrix that true positives(TP) are 184 cases, where the model correctly predicted that an applicant's loan would be approved, and this was indeed the case. True negatives(TN) is 15, where it was predicted that an applicant's loan would not be approved, which aligned with the actual outcomes. The confusion matrix further reveals that there is one false negative case, indicating an instance where the model incorrectly predicted that applicant' loan application would be rejected when, in fact, they were approved. There are no false positives whatsoever.

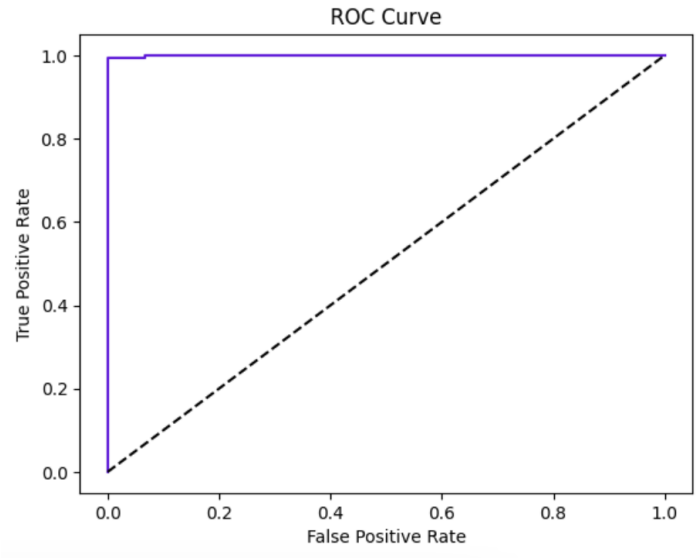

Figure 25: Artificial Neural Networks - ROC curve

The ROC curve above is the same graphical representation of the confusion matrix for Artificial Neural Network model classified the dependent variable 'Loan_Status'. (Bhandari, 2020)

Loan Default

We applied a Random Forest Classifier to the 'Loan Default' dataset, utilising RandomizedSearchCV to hyper-tune the model. RandomizedSearchCV performs a randomised search across a defined grid of hyperparameters, effectively balancing thoroughness and computational efficiency by limiting the number of iterations and cross-validation folds (sklearn, 2019). The best cross-validation score, which was 89.78%, indicates the robust performance of our model and the effectiveness of the selected hyperparameters. Further analysis was conducted using SHAP (SHapley Additive exPlanations), a tool for interpreting machine learning models (SHAP 2023). SHAP values clarify the contribution of each feature to the model's prediction. We ran a code, which was provided during the Strategic thinking lectures. SHAP identified 'Age', 'Interest Rate', and 'Income' as key features influencing the likelihood of loan default. These features were visualised using Matplotlib and SHAP in descending order of importance.

Contrary to our initial hypothesis, 'Loan Amount' and 'Credit History' ranked differently than expected in the feature importance plot, indicating that 'Credit Score' is not as significant predictor as presumed for loan approvals or defaults.

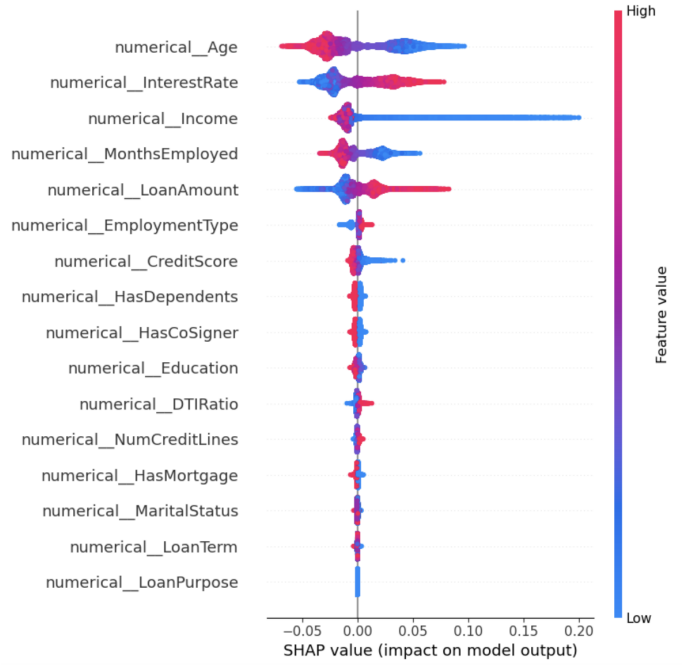

Figure 26: Feature Impact on Loan Default Prediction: A SHAP Value Analysis

To validate the hyper-tuned model's accuracy, we tested a specific instance (Instance 2) from the dataset. The model predicted a 12% probability of loan default for this borrower. This instance's analysis showed that 'Interest Rate' and 'Loan Amount' are prominent predictors of default, with red values in the SHAP visualisation indicating features that reduce the predicted probability of default.

Model deployment

Word count: 541

During the semester in Class Assignment 1, we embarked on a journey of exploration to find the model for our dataset. Our main focus was on classification techniques. We experimented with well-known algorithms; Decision Tree Classification, Random Forest Regressor, Random Forest Classification and K Nearest Neighbors. Each of these models has its strengths and is suitable for different types of data and problems. The Decision Tree Classification is simple and interpretable but may overfit complex datasets. Random Forest Regressor, typically for regression, can perform classification tasks by aggregating multiple decision trees to improve prediction stability. The Random Forest Classification variant is better for categorising, as it uses a decision tree ensemble for higher accuracy and less overfitting. The K Nearest Neighbors (KNN) algorithm classifies based on case similarity, handling intricate boundaries effectively. Consequently, we opted for the Random Forest Classification model for deploying on our test datasets due to its robustness in handling features and its effectiveness in producing predictions.

In this semester, we introduced different types of models from the previous semester such as Support Vector Machine and Artificial Neural Network and used a confusion matrix to

validate the outcome of each model. Last semester we didn't apply Synthetic Minority Over-sampling Technique, which was a tremendous shortcoming for treating datasets of this kind where a high imbalance is observed between the two classes. Another technique, which was not used previously, but it was used for this semester class assignment is hypermeter tuning with GridSearchCV. GridSearch improved the results for Support Vector Machine and Random Forest Classification. Considering that the datasets were heavily imbalanced we conducted several machine learning model comparisons and we visualised the outcomes.

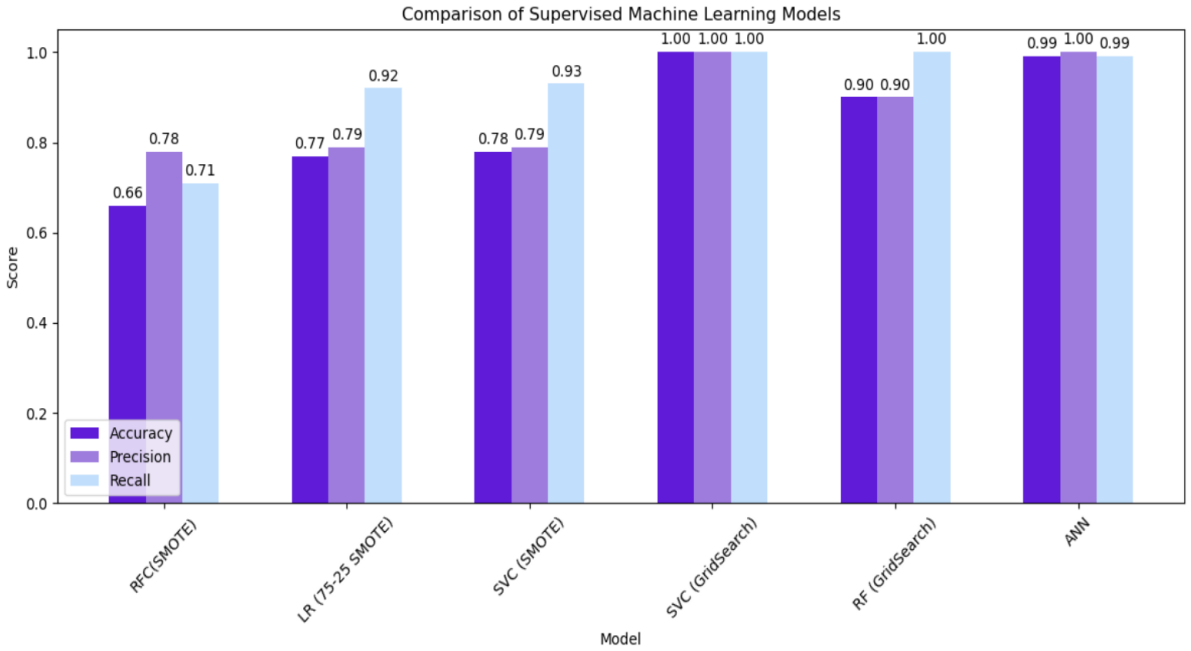

Figure 27: Model comparison

In summary, comparing all the employed methods and learning models, the Support Vector Machine with SMOTE and GridSearchCV effectively segregates and nests the values and it appears to be the best classifier for applicant's who had their application approved or rejected. It is also important to highlight that since the datasets are of a small scale, there could be overfitting. We saved the Support Vector Machine model in a pickle format, in order to deploy it to the test datasets.

We used the same Jupiter notebook in order to proceed with the deployment. We imported *'loan_test.csv'* dataset and named the dataframe *'loantest'*. This dataset is smaller than the training dataset with 367 observations and 12 features. In our dataset, we found missing values: 11 in *'Gender'*, 10 in *'Dependents'*, 23 in '*Self-Employed*', 5 in *'LoanAmount'*, 6 each in *'Loan_Amount_Term'* and *'Credit_History'*. To fill these, we used SimpleImputer for most fields and KNNimputer for *'ApplicantIncome'*, *'LoanAmount'*, *'Loan_Amount_Term'*, and *'Credit_History'*. We applied label encoder on all categorical values and StandardScaler on all numerical columns. We also dropped the *'Loan_ID'* column from the dataset. We added a column for *'Loan_Status'*. We dropped the dependent variable from the dataset *'Loan_Status'* - *'y'* and for independent variables we used only three columns *'ApplicantIncome'*, *'LoanAmount'*, *'Credit_History'*. Afterwards we loaded the model *'finalized_svm_model.pkl'*.

When we initiated the individual instance predictions, it became evident that the classification was highly accurate. Specifically, 21 instances were categorised as 'Approved' for loan applications, while the remaining instances were classified as 'Rejected'. This indicates that 6% of the total instances fall into the 'Approved' loan category.

Results, Discussions and Challenges

Word count: 422

Last semester, a Linear Regression classifier was the top-performing model, evaluated using mean square error with a score of 0.14. However, this semester, after applying the Synthetic Minority Oversampling Technique (SMOTE), the performance of the Linear Regression model dropped, as evidenced by a comparative visualisation of all models. It now ranks fourth based on accuracy, precision, and recall metrics. The hypertuned Support Vector Classifier (SVC) emerged as the best model, closely followed by the Artificial Neural Network (ANN). The Random Forest Classifier with GridSearchCV came in third. Prior to machine learning model deployment, we carefully preprocessed the test dataset to handle missing values using SimpleImputer and KNNimputer and scaled the data with StandardScaler. Additionally, we encoded categorical variables using LabelEncoder. It was necessary to modify the dataset by adding a 'Loan_Status' column and removing the *'Loan_ID'* column to ensure compatibility with our model. Deploying the SVC, we found that it predicted *'Loan_Status'* for each instance, identifying 21 instances (or 6% of the dataset) as approved loan applications and the rest as rejected. It's critical to note that the training dataset was relatively small, with only 614 observations, which may limit the reliability of our model's performance and generalizability of the conclusions.

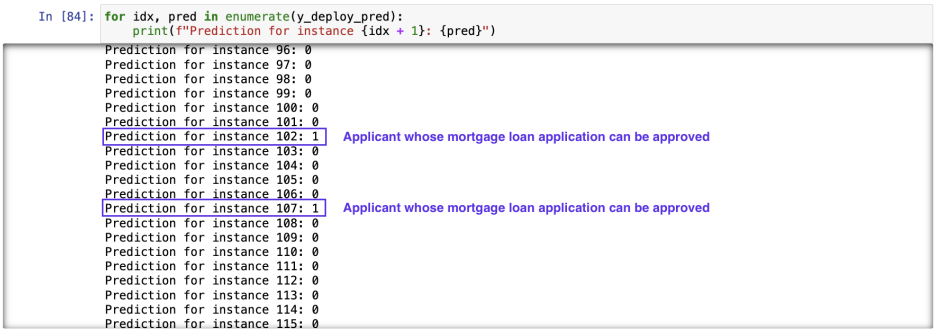

Figure 28: Support vector machine learning model deployment

To test our hypothesis regarding the characteristics of defaulted loans—be they personal, car, or home loans—we introduced a new dataset specifically comprising instances of mortgage default. We thought that analysing this dataset would either corroborate or challenge our initial assumptions. The dataset contains variables like *'Interest Rate', 'Credit Score'*, and *'Debt-to-Income Ratio'*, among additional factors, which facilitated this analysis. The goal was to identify a specific group of borrowers who had failed to repay their mortgage loans. A pie chart was created to illustrate the percentage of defaults, showing that they made up 10% of the loans in our database.

Numerical and categorical values were imputed with StandardScaler and OneHotEncoder. We implemented a hypertuned Random Forest Classification model and the cross-validation is 89.78%. We utilised the SHAP library to visually represent feature importance, revealing that variables like *'Age'*, *'Interest Rate'*, and *'Income'* are crucial in assessing the likelihood of a borrower defaulting on a loan. We also tested the Random forest prediction capability on instance '**two'** of our datasets. For this particular borrower, the model forecasts a default risk of 12%. The analysis highlights *'Interest Rate'* and *'Loan Amount'* as key indicators influencing the likelihood of default. In the SHAP visualisation, features marked with red suggest a mitigating effect on the default prediction, lowering the probability of default.

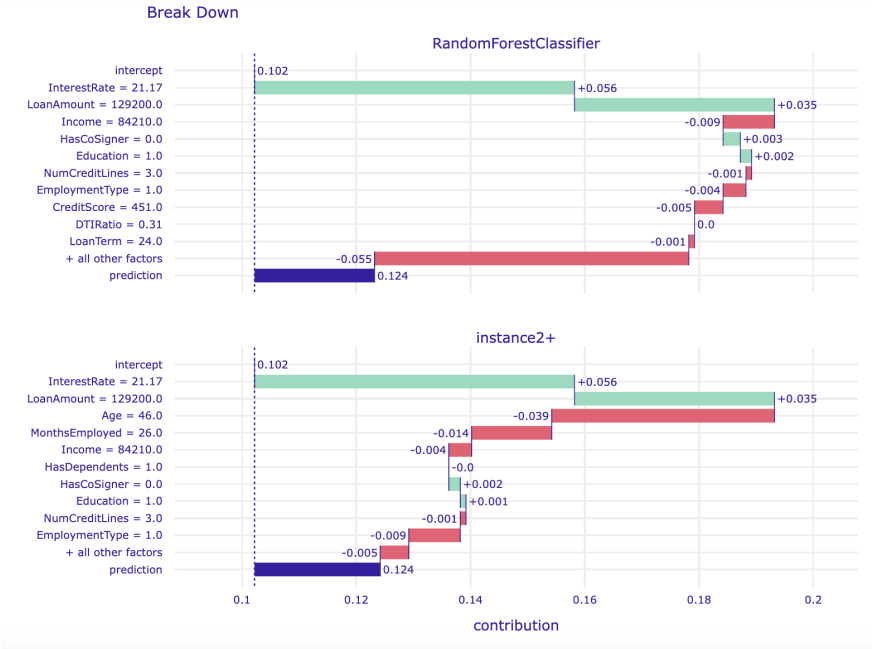

Figure 29: Shap Values and predictions for instance 'two'(*Loan Default Dataset*)

Conclusions

Word count: 618

The article by (O'Regan, 2023) examines the repercussions for Irish mortgage payers due to the rise in interest rates by the European Central Bank (ECB). It points out that those with variable-rate mortgages are likely to experience a swift surge in their monthly payments, potentially challenging their financial capacity to meet these payments. Those with fixed-rate mortgages won't be affected immediately, but may encounter increased rates when it's time to renew. The ECB's decision to elevate interest rates is a strategy to fight inflation, yet it also means increased payment obligations for mortgage holders. Consequently, this could impose economic burdens on families with variable-rate mortgages or those transitioning from fixed-rate terms.

Specifically, our dataset underscores the heightened risk of default among borrowers facing higher interest rates, particularly those with lower income or younger age. This ties into the broader discussion of our analysis of the second dataset, where we could conclude that ***'Interest rate'*** is second in feature importance ranking, signalling potential borrowers whose

higher interest rates could lead to incapability of mortgage repayments. Overall, the signals for default are the lesser income and younger a borrower is, the more likely it is for them to default on their loan. However, *'Credit Score'* appears in the 7th position in feature importance ranking and results differ from the research we have done in the first dataset *'Loan approval'*. In their 2019 article, Kagan explains that Credit Scores span from 300 to 850, with higher scores increasing the likelihood of loan approval and potentially resulting in lower interest rates. In the first dataset *'Loan Approval',* we do have information if an applicant has 'Credit history', but we do not have their score, which is not sufficient to make a thorough assessment of a borrower's profile. We believe that the first dataset should contain information such as leading features from the second dataset and also it will be crucial to know if borrowers rent or live in a second property.

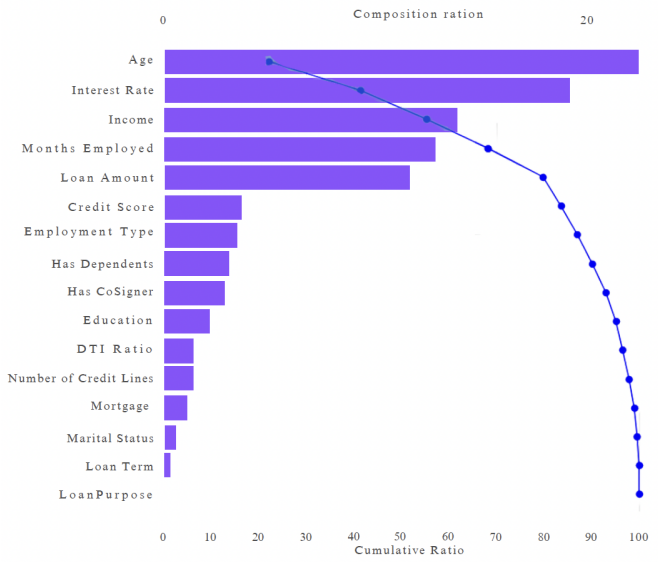

Figure 30: SHAP Waterfall Plot, visualising the most important features in ***Loan default*** dataset

In conclusion, the journey we embarked on for our capstone project was not just challenging, but a meaningful learning experience. It pushed us to explore the boundaries of our creativity and adaptability. As a small team working remotely, we had to find ways to collaborate despite the barriers. We relied on meetings using Google Meet, which proved to be an invaluable tool for maintaining clear and consistent communication among team members. To enhance our efforts further we utilised Google Docs for real time sharing and editing of documents as well as GitHub for version control of our project files. This setup allowed us to seamlessly work across the various team members schedules.

Following the CRISP methodology throughout the project provided us with an approach although we did encounter challenges along the way. One of the hurdles was acquiring up to date datasets that accurately reflected the current economic conditions, particularly in the Irish mortgage market. The scarcity of relevant data highlighted a gap in timely financial information available for academic research, which presented a substantial obstacle in our analysis. Moreover the unpredictable nature of market trends added a layer of complexity to our project. We had to develop models that could effectively adapt to market volatility and incorporate it into their calculations.

Throughout the process we were acutely aware of the risks of overfitting which is a common challenge in machine learning projects. Overfitting occurs when models perform well on training data but struggles to generalise to new, unseen data. Maintaining a balance between data analysis and the reliability of our models required planning and execution. We had to make decisions regarding how we explored the data while also ensuring that our models were robust enough to handle real world scenarios.

Reference list

Works Cited

AIB. "Variable Rate Mortgage Change | AIB Mortgages." *AIB*, 2023, aib.ie/our-products/mortgages/variable-rate-mortgage-change. Accessed 6 Nov. 2023.

Brownlee, Jason. "SMOTE for Imbalanced Classification with Python." *Machine Learning Mastery*, 16 Jan. 2020, machinelearningmastery.com/smote-oversampling-for-imbalanced-classification/.

Coursera. "Data Science Coding Challenge: Loan Default Prediction." *Coursera*, 2023, www.coursera.org/projects/data-science-coding-challenge-loan-default-prediction.

Dabbura, I. (2019). *Predicting Loan Repayment*. [online] Medium. Available at: https://towardsdatascience.com/predicting-loan-repayment-5df4e0023e92 [Accessed 6 April, 2023]

Grace-Martin, Karen. "Seven Ways to Make up Data: Common Methods to Imputing Missing Data." *The Analysis Factor*, 4 Feb. 2009, www.theanalysisfactor.com/seven-ways-to-make-up-data-common-methods-to-imputing-missing-data/#:~:text=Imputation%20simply%20means%20replacing%20the.

Harrison, M. (2019). *Machine learning pocket reference : working with structured data in Python*. North Sebastopol, Ca: O'reilly Media, Inc.

Hastie, T. (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction (2nd ed.). Springer*.

Healy, Alan. “Bank of Ireland Raises Variable Mortgage Interest Rates by 0.25%.” *Irish Examiner*, 27 Oct. 2023, www.irishexaminer.com/business/economy/arid-41256781.html#:~:text=For%20existing%20mortgage%20customers%20already. Accessed 6 Nov. 2023.

Konapure, R. (2023). *Home Loan Approval*. [online] www.kaggle.com. Available at: https://www.kaggle.com/datasets/rishikeshkonapure/home-loan-approval.[Accessed 29 March, 2023]

Lawton, George. “What Is Logistic Regression? - Definition from SearchBusinessAnalytics.” *SearchBusinessAnalytics*, Jan. 2022, www.techtarget.com/searchbusinessanalytics/definition/logistic-regression.

Lee, N. and Lee, V. (2018). *Bank Lending : Principles and Practice*. [online] Ebscohost.com. Available at: https://eds.p.ebscohost.com/eds/ebookviewer/ebook/ZTAyMG13d19fMTg3MDI2MV9fQU41?sid=471e9fd1-48c4-422b-b036-f17a767f9dfd%40redis&vid=1&format=EB&rid=1 [Accessed 5 Apr. 2023].

Microsoft. “SMOTE - Azure Machine Learning.” *Learn.microsoft.com*, 4 Nov. 2021, learn.microsoft.com/en-us/azure/machine-learning/component-reference/smote?view=. Accessed 7 Nov. 2023.

Murphy, Fiona . “BPFI Mortgage Approvals - June 2023.” *Banking & Payments Federation Ireland*, 2023, bpfi.ie/publications/bpfi-mortgage-approvals-june-2023/. Accessed 6 Nov. 2023.

Müller, A.C. and Guido, S. (2017). *Introduction to machine learning with Python : a guide for data scientists*. Beijing: O’reilly.

NIKHIL. “Loan Default Prediction Dataset.” *Www.kaggle.com*, 2023, www.kaggle.com/datasets/nikhil1e9/loan-default.

Roy, B. (2020). *All about Feature Scaling*. [online] Medium. Available at: https://towardsdatascience.com/all-about-feature-scaling-bcc0ad75cb35 [Accessed 9 Apr. 2023]

O’Halloran, Barry. “Interest Rate Hikes Spooking Homebuyers, Report Claims.” *The Irish Times*, 31 Oct. 2023,

www.irishtimes.com/business/economy/2023/10/31/interest-rate-hikes-spooking-home-buyers-report-claims/. Accessed 6 Nov. 2023.

O'Regan, Ellen. "What Does ECB Rates Holding High Mean for Irish Mortgage Holders?" *The Irish Times*, 11 Nov. 2023, www.irishtimes.com/business/2023/11/10/what-does-ecb-rates-holding-high-mean-for-irish-mortgage-holders/? Accessed 11 Nov. 2023.

scikit. "6.4. Imputation of Missing Values — Scikit-Learn 0.22.2 Documentation." *Scikit-Learn.org*, 2022, scikit-learn.org/stable/modules/impute.html.

---. "Sklearn.impute.SimpleImputer — Scikit-Learn 0.24.1 Documentation." *Scikit-Learn.org*, 2023, scikit-learn.org/stable/modules/generated/sklearn.impute.SimpleImputer.html.

scikit learn - SupportVectorMachines. "1.4. Support Vector Machines — Scikit-Learn 0.20.3 Documentation." *Scikit-Learn.org*, 2018, scikit-learn.org/stable/modules/svm.html . Accessed 9 Nov. 2023.

Shafi, Adam. "Sklearn Random Forest Classifiers in Python Tutorial." *Www.datacamp.com*, Feb. 2023, www.datacamp.com/tutorial/random-forests-classifier-python. Accessed 9 Nov. 2023.

Shap. "Welcome to the SHAP Documentation — SHAP Latest Documentation." *Shap.readthedocs.io*, shap.readthedocs.io/en/latest/. Accessed 11 Nov. 2023.

sklearn. "6.4. Imputation of Missing Values." *Scikit-Learn*, 2022, scikit-learn.org/stable/modules/impute.html#nearest-neighbors-imputation. Accessed 6 Nov. 2023.

---. "Sklearn.model_selection.RandomizedSearchCV — Scikit-Learn 0.21.3 Documentation." *Scikit-Learn.org*, 2019, scikit-learn.org/stable/modules/generated/sklearn.model_selection.RandomizedSearchCV.html. Accessed 11 Nov. 2023.

Team, Great Learning. “Hyperparameter Tuning with GridSearchCV.” *Great Learning Blog: Free Resources What Matters to Shape Your Career!*, 29 Sept. 2020, www.mygreatlearning.com/blog/gridsearchcv/#:~:text=GridSearchCV%20is%20a%20technique%20for. Accessed 9 Nov. 2023.

Yiu, Tony. “Understanding Random Forest.” *Medium*, Towards Data Science, 12 June 2019, towardsdatascience.com/understanding-random-forest-58381e0602d2. Accessed 9 Nov. 2023.