CCT College Dublin

Assessment Cover Page

| Module Title: | Strategic Thinking |
| --- | --- |
| Assessment Title: | Strategic Thinking Continuous Assessment 1 |
| Lecturer Name: | James Garza |
| Student Full Name: | Anna Georgieva; Nick McNamara; Octavio Rieu |
| Student Number: | sbs23039@student.cct.ie; [sbs23022@student.cct.ie](mailto: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:

## Table of contents

[Table of contents 3](#_w2b5qc91shel)

[List of Figures 4](#_qu0qt0jsoamm)

[Introduction 5](#_37041gfy3b00)

[Business understanding 7](#_f27wby1pvcsx)

[Data preparation for Loan Approvals 15](#_qekc2b3jdv3b)

[Model implementation 18](#_2nupq4rviyyh)

[Deployment 23](#_13tpygmpw3nl)

[Challenges 24](#_y7n4pc94gqui)

[Results and analysis 24](#_mab4aegwkzyj)

[Conclusion 26](#_nsqnf26gn17b)

[Reference list 27](#_3z44o5wi77sr)

## List of Figures

[Figure 1: AIB PDH Variable Rates 5](#_2n5gmn25efm2)

[Figure 2: Totals of null values 9](#_vf0urxfdo4rn)

[Figure 3: Checking missing values in key columns and cross-reference the count in Credit History 10](#_nonuu8byqmhn)

[Figure 4: Descriptive table of datasets 11](#_cgh66giefozy)

[Figure 5: Checking the count of Mortgage Loans 11](#_8gi46j62n9th)

[Figure 6: Percentage of approved and denied loan 12](#_miu8fg1m54fl)

[Figure 7: Stacked Distribution of Loan Status by Loan Amount 12](#_khvemvc7uvug)

[Figure 8: Stacked Distribution of Loan Status by Applicant Income 13](#_abi17q91t14x)

[Figure 9: Stacked Distribution of Loan Status by Coapplicant Income 13](#_d13t8j5zb2rr)

[Figure 10: Stacked Distribution of Loan Status by Credit History 14](#_g676jk5fcmai)

[Figure 11: Stacked Distribution of Loan Status by Education 14](#_5zoq87i3xmrn)

[Figure 12: Stacked Distribution of Loan Status by Marital status 15](#_dn66q5ybotv6)

[Figure 13: Stacked Distribution of Loan Status by Gender 15](#_hsoett17vf41)

[Figure 14: Stacked Distribution of Loan Status by Dependents 16](#_ud3ydxy3dwbf)

[Figure 15: Spearman correlation 17](#_c1vk3ixesyll)

[Figure 16: Feature importance. 18](#_hyirpukxfnit)

[Figure 24: Concad method 20](#_oao0287p03e0)

[Figure 25: Split dependent variable and independent variables 20](#_t5xcf3u9amhy)

[Figure 26: Decision tree classification 21](#_dyuwjhj64mfy)

[Figure 27: Linear regression, mean squared error 21](#_dpzwmdm5ejb1)

[Figure 28: Random forest regressor 21](#_4csz0jcmd5kf)

[Figure 29: Saving Linear regression model in .pkl format 22](#_281wdnpttb15)

[Figure 30: Linear regression deployed model 22](#_aq5w8durvhgb)

[Figure 31: Coefficient magnitudes 23](#_q2fjlispjprj)

[Figure 32: Additional features 23](#_50hyjtme80ik)

[Figure 33: Ascending order of coefficients magnitudes 24](#_x7il7d13iq83)

## Introduction

Word count: 998

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 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).



### 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. A relevant question to consider is whether the evaluation process for loan applications of ‘Dream Housing Finance Company’ (Kaggle datasets, 2023) is consistent with the standards applied to citizens within Ireland.

In the first semester, we examined the research conducted by Lee and Lee (2018), detailed in their book which asserts that credit history is the primary consideration in lending decisions for a diverse array of borrowers, including individuals, partnerships, corporations, clubs, societies, and trusts. Lee and Lee describe that a credit score is derived from a person's past and current credit and this can predict their likelihood of honouring debt commitments. 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: 2,270

We have used Cross-Industry Standard Process for Data Mining (CRISP) methodology for this project and our main objective is to reduce the risk of lending to borrowers who are likely to default on their loan.

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. In order to produce this project plan, we have used a Ghantt chart with a timeline on when each task should be completed by.

#### 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 the hypothesis that the Credit History attribute is the most significant predictor in determining loan approval outcomes.

#### General goal

Our primary path is to minimise biases and maintain objectivity in our handling and analysis of data throughout the duration of this project.

#### 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

The programming language of choice is Python. We use Jupyter Notebook in order to work on the datasets. For the project, version control of CA2 is required, hence we installed Github Desktop.

#### 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).

**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.

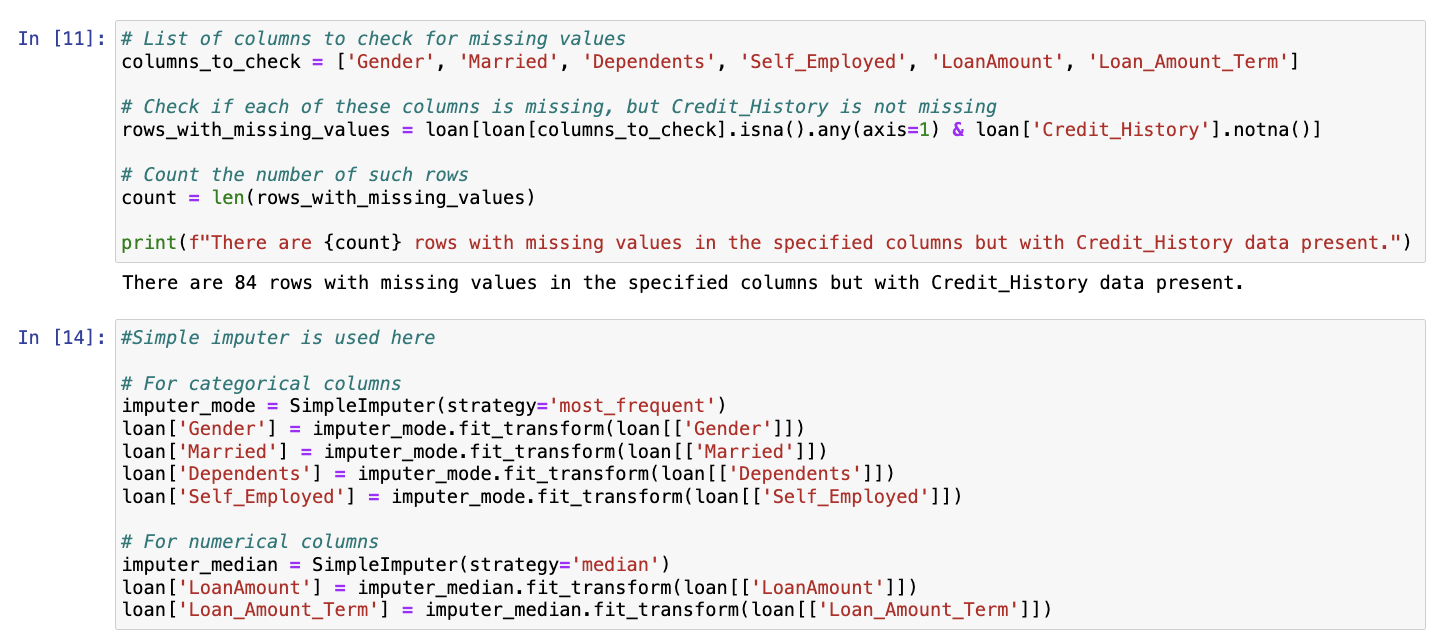
We run *loan.head*() to display the first 5 rows and all columns, so we can quickly check what kind of data we work with. The attributes in this dataset are categorical Loan\_ID, Gender, Married, Dependents, Education, Self\_Employed, Credit\_History, Property\_Area, Loan\_Status and numerical attributes - 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.



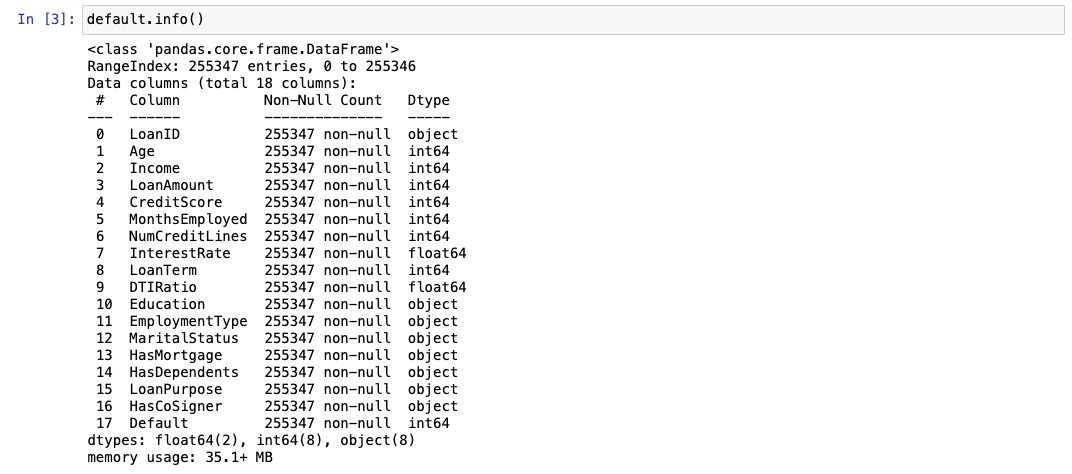
### Figure 3: Checking missing values in key columns and cross-reference the count in Credit History

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.

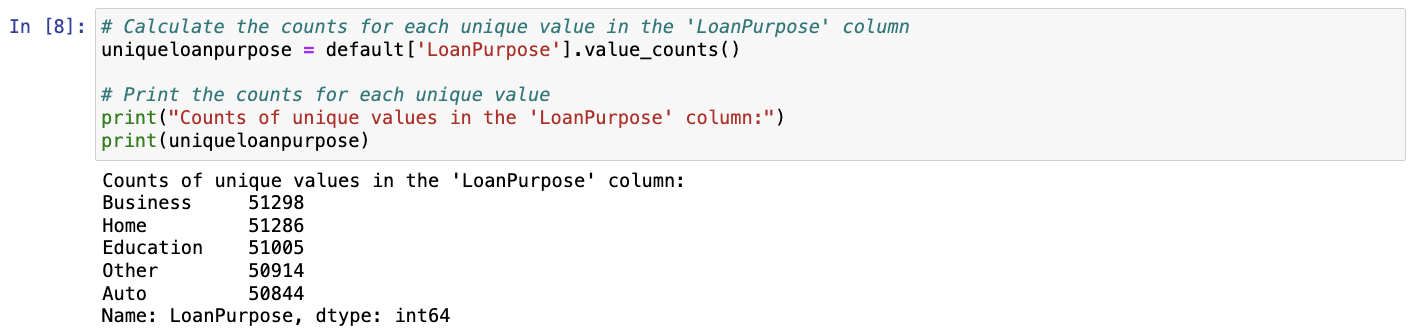
**Cleaning Loan Default**

We created a new Jupiter notebook and imported Loan\_default.csv. There are 17 features such as: *‘LoanID’, ‘Age’, ‘Income’, ‘LoanAmount’, ‘CreditScore’, ‘MonthsEmployed’, ‘NumCreditLines’, ‘InterestRate’, LoanTerm’, ‘DTIRatio’, ‘Education’, ‘EmploymentType’, ‘MaritalStatus’, ‘HasMortgage’, ‘HasDependents’, ‘LoanPurpose’, ‘HasCoSigner’, ‘Default’.*



### 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 type of loans.

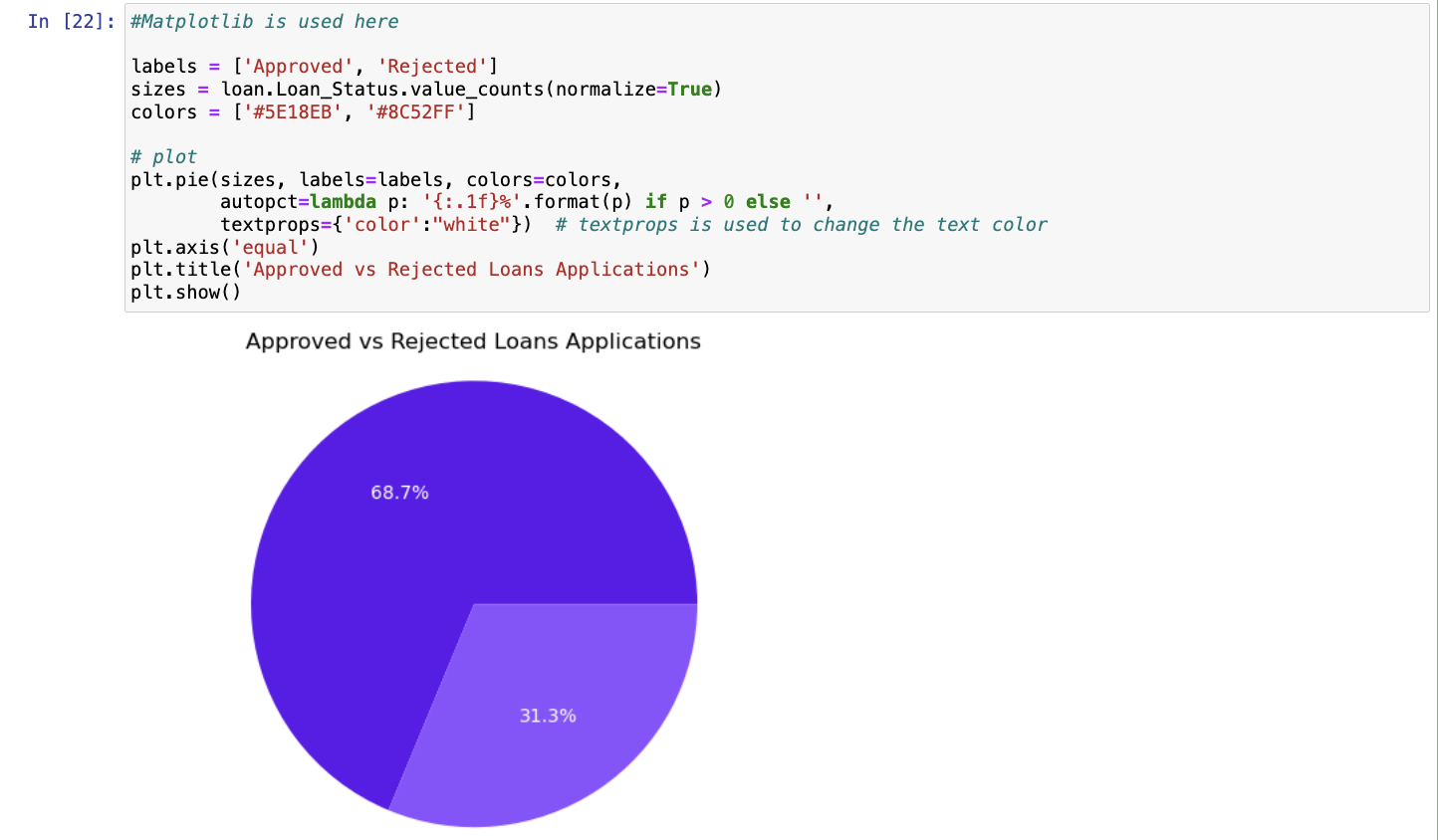


### Figure 5: Checking the count of Mortgage Loans

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

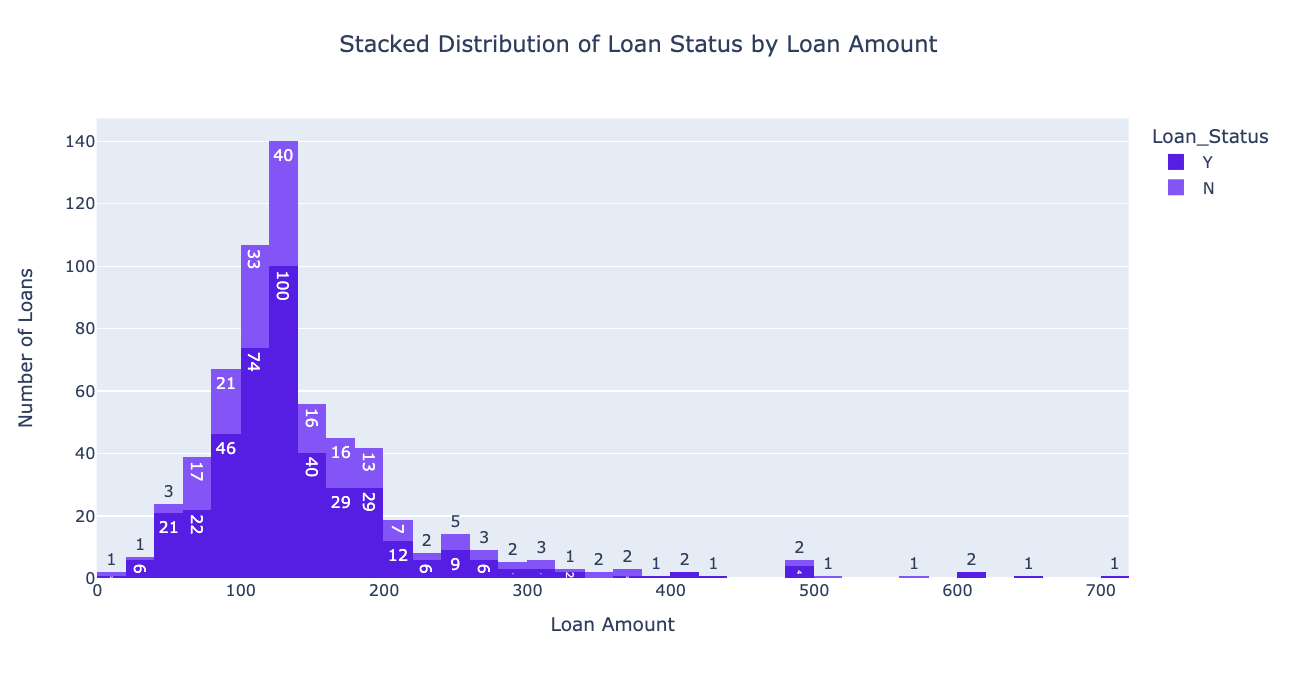
Data Visualisations *‘Home loan approvals’*

​​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%.



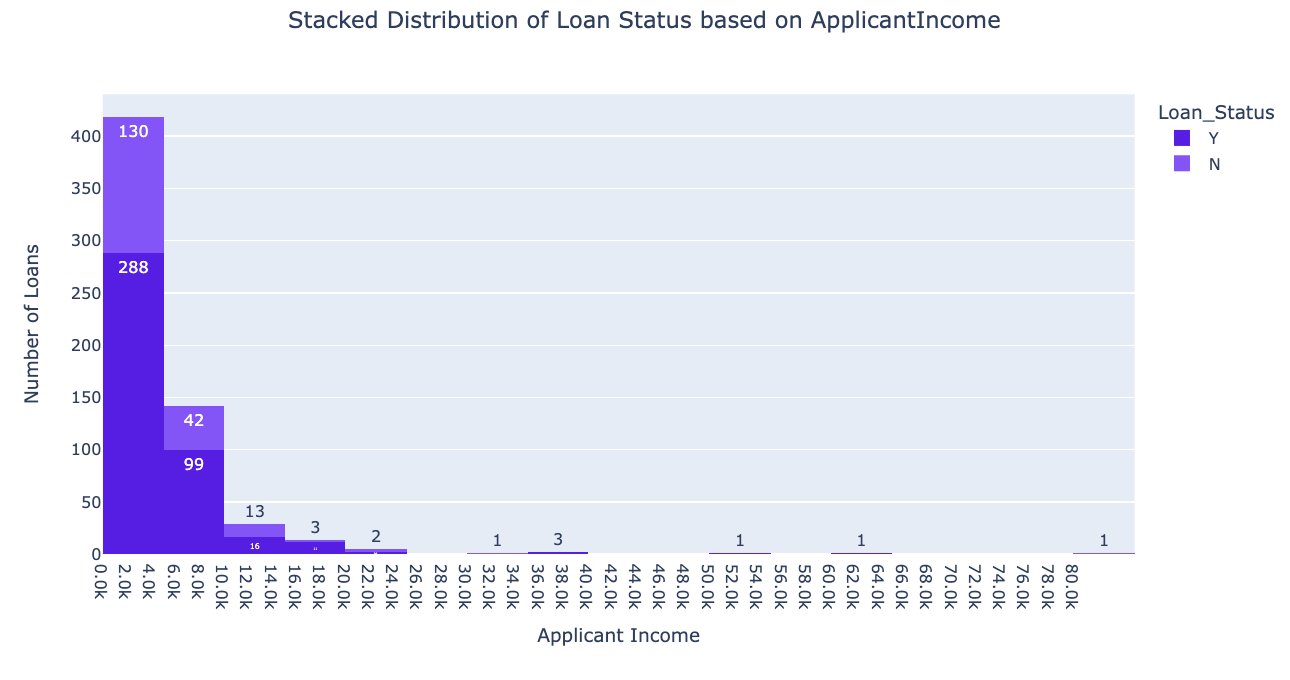
### Figure 6: Percentage of approved and denied loan

​​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 Jupiter notebook for further analysis.



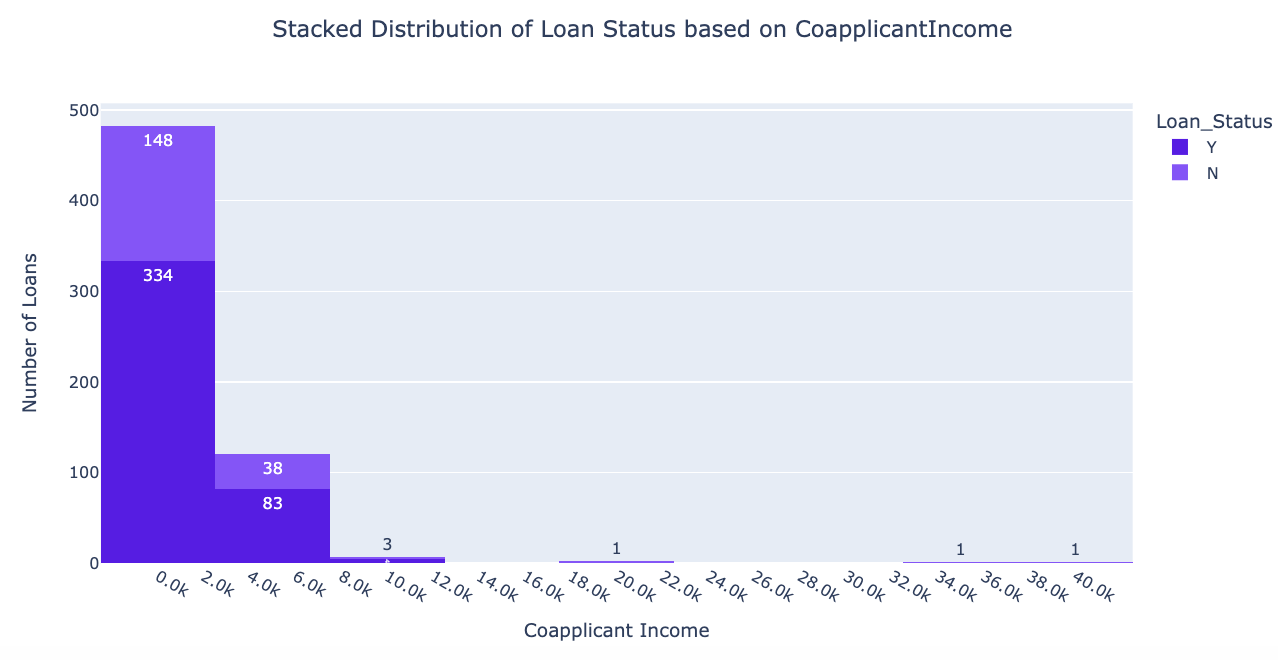
### Figure 7: Stacked Distribution of Loan Status by Loan Amount

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.*

**

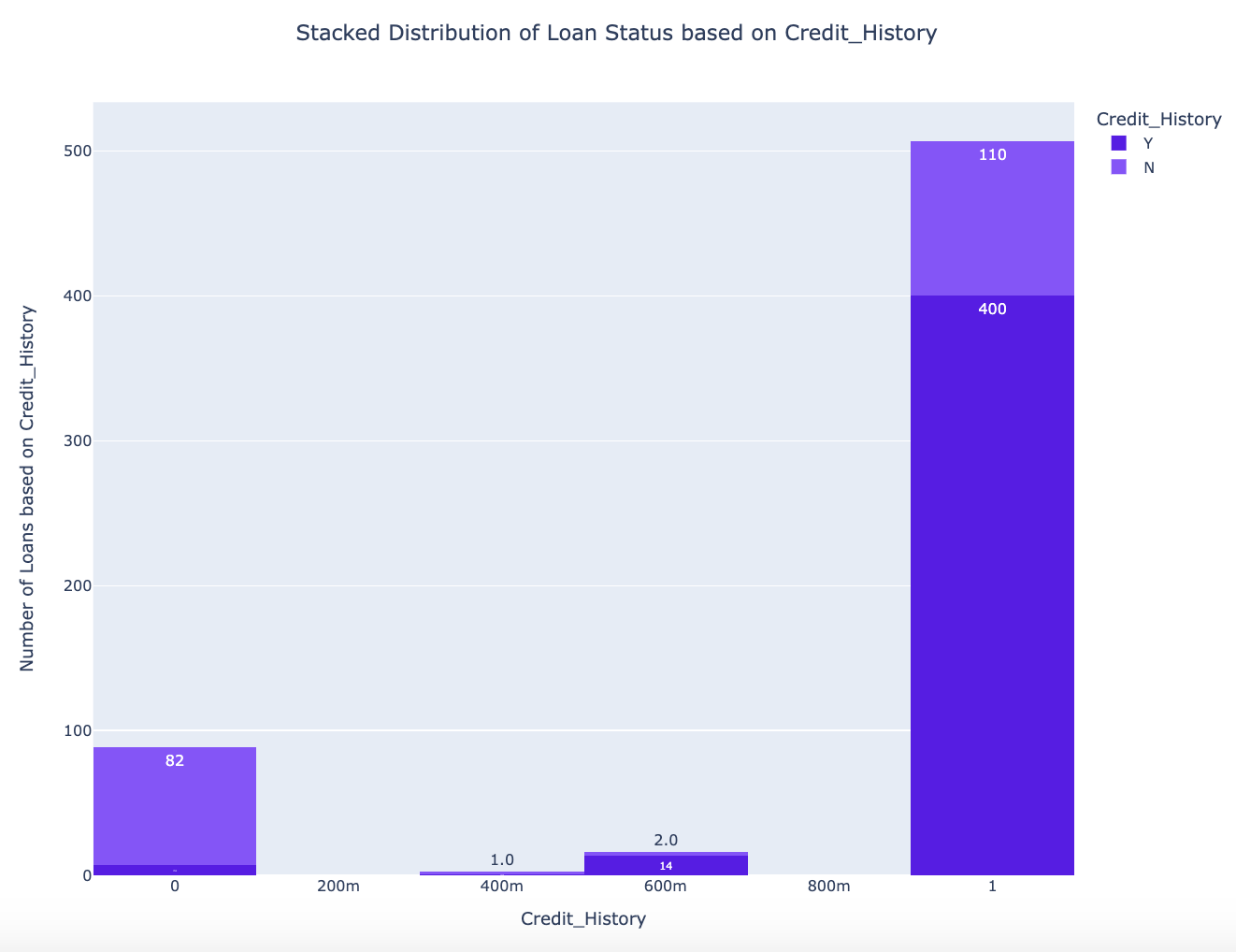
### Figure 8: Stacked Distribution of Loan Status by Applicant Income

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



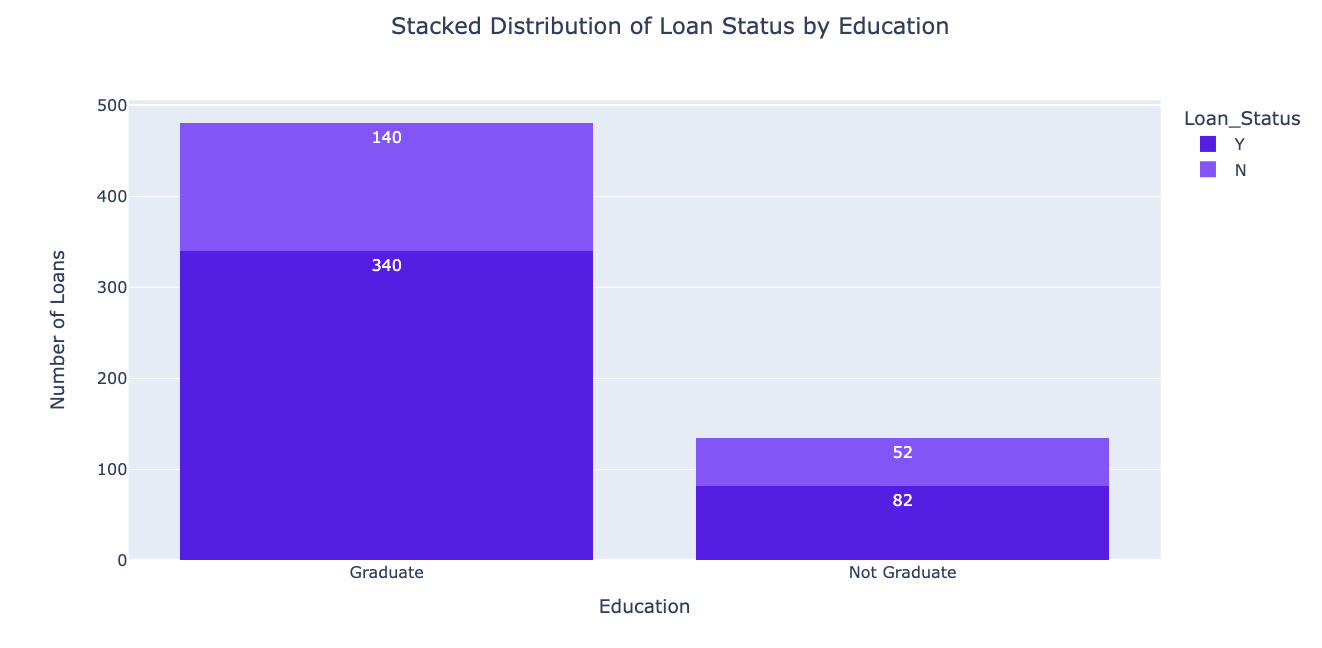
### Figure 9: Stacked Distribution of Loan Status by Coapplicant Income

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



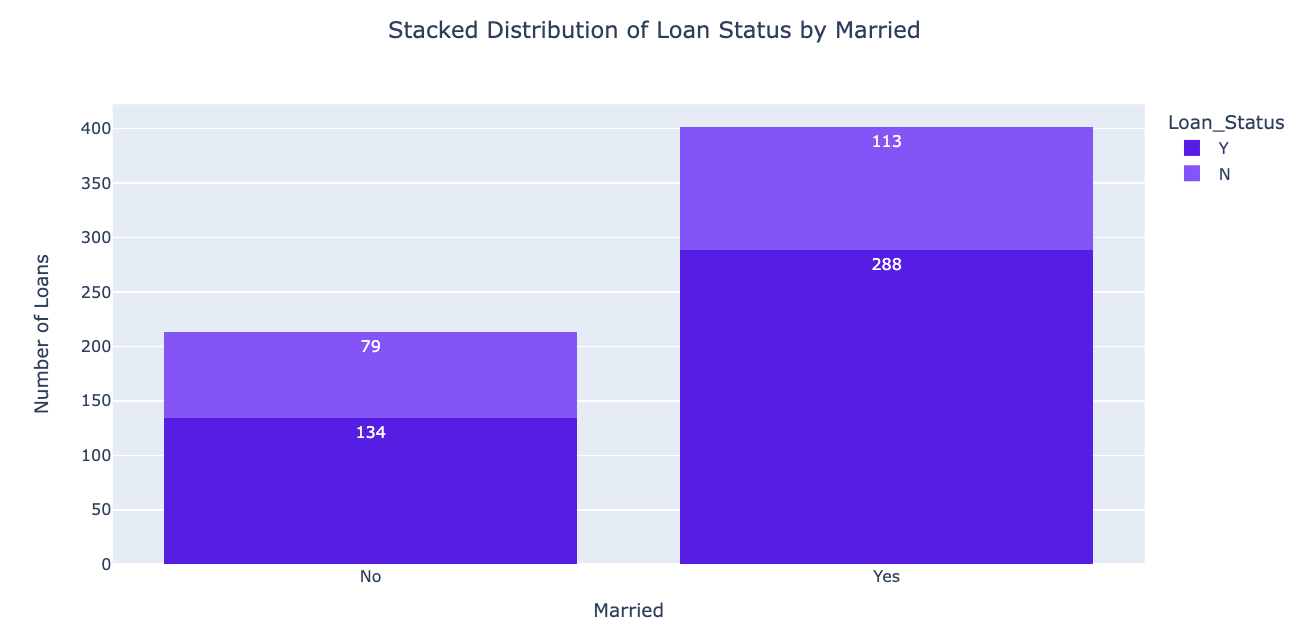
### Figure 10: Stacked Distribution of Loan Status by Credit History

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’.*

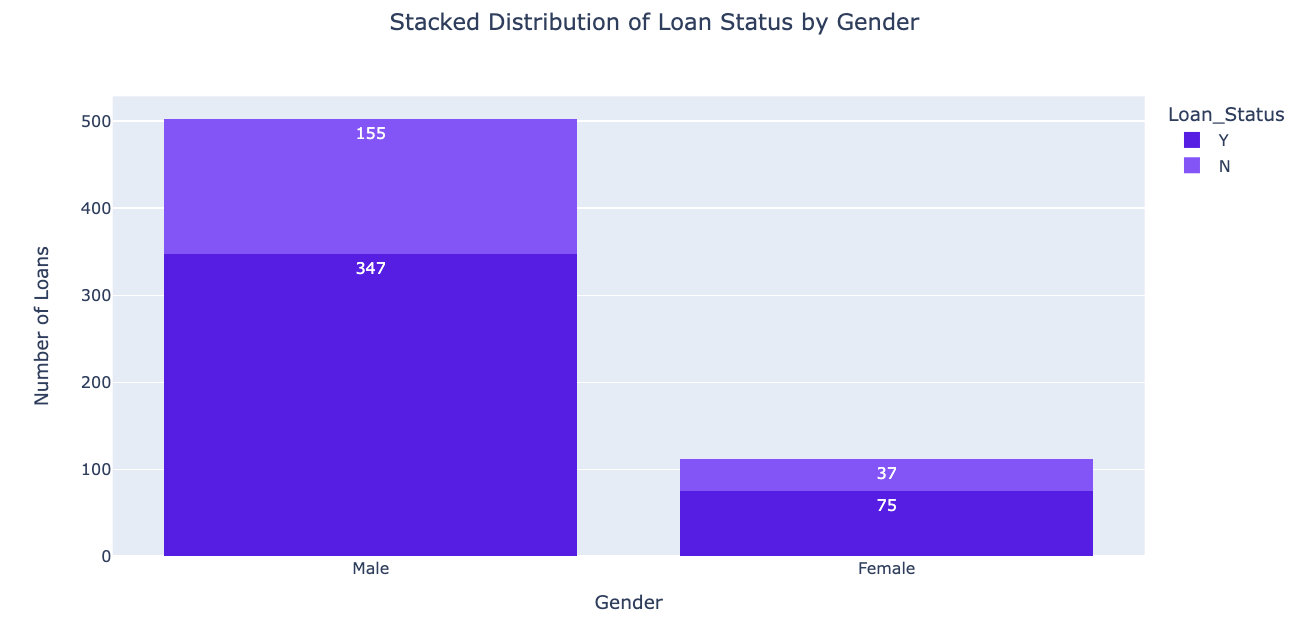


### Figure 11: Stacked Distribution of Loan Status by Education

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.

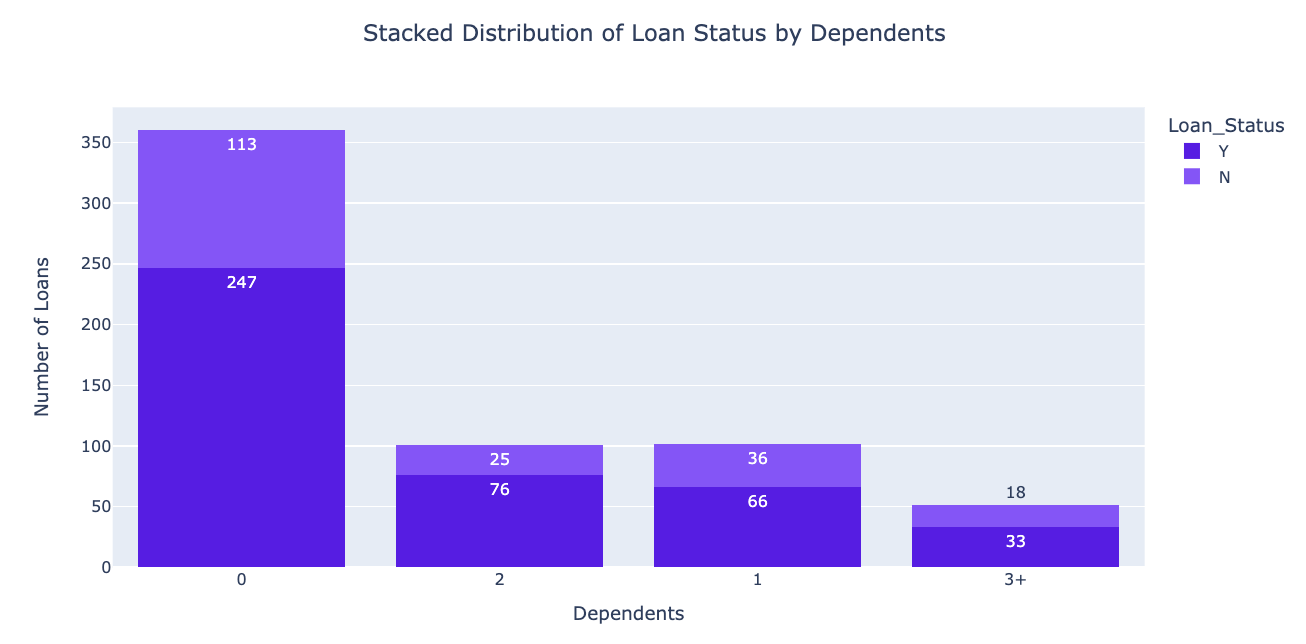


### Figure 12: Stacked Distribution of Loan Status by Marital status



### 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 belong to individuals with no children.



### Figure 14: Stacked Distribution of Loan Status by Dependents

## 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:

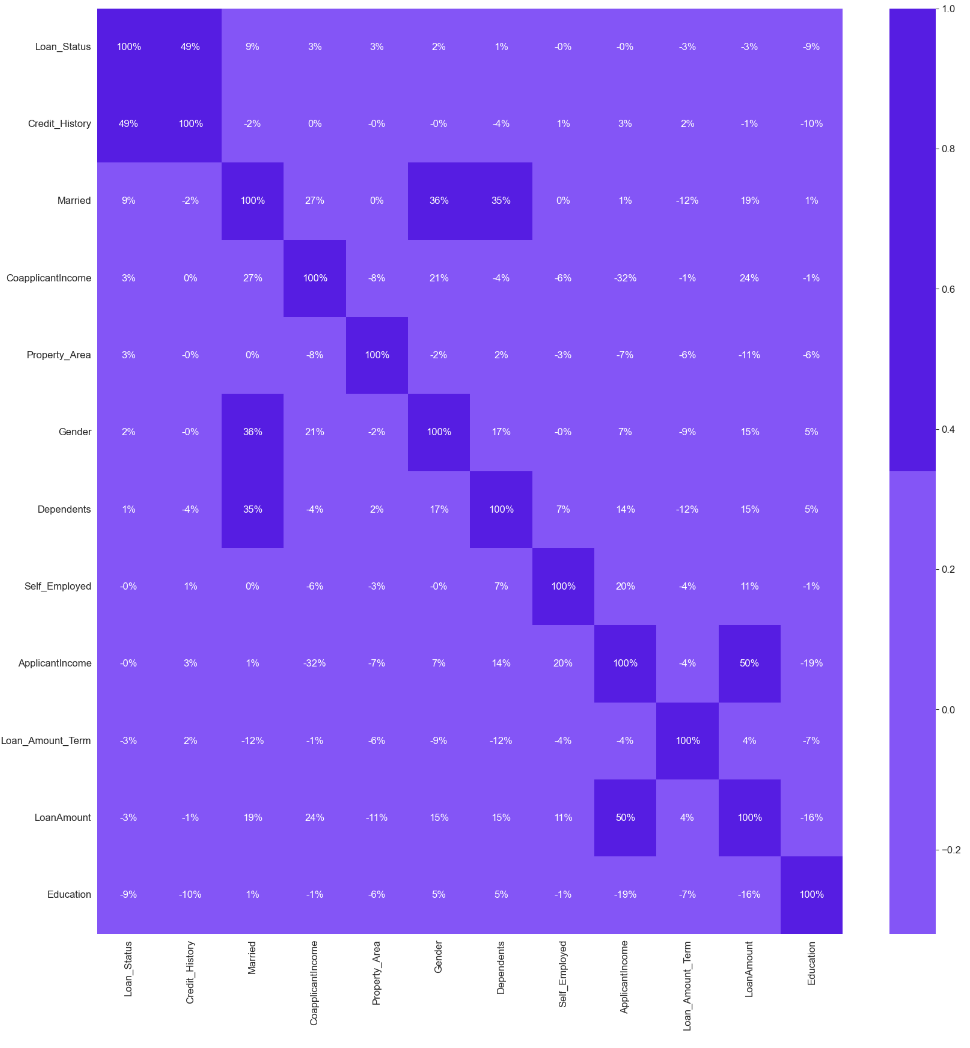
*'ApplicantIncome', 'CoapplicantIncome', 'LoanAmount','Loan\_Amount\_Term', 'Credit\_History'*.

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 customers' decisions to have their loan approved or rejected.

**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.



### Figure 15: Spearman correlation

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 scatter 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.

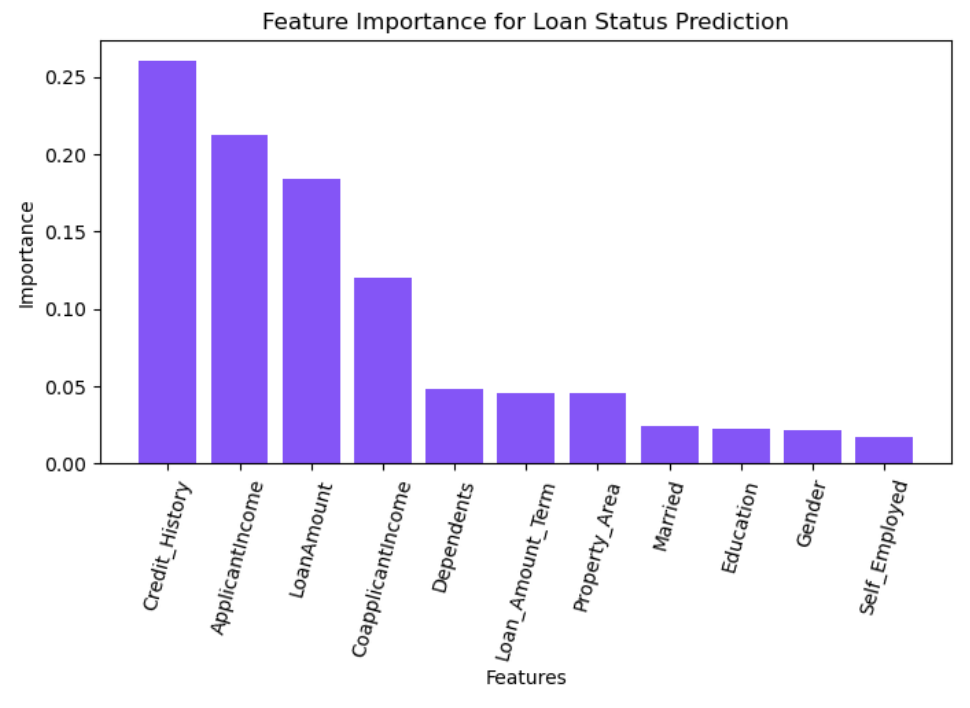
## Model implementation

Word count: 1,721

**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 generalizability to new data. (Shafi, 2023)

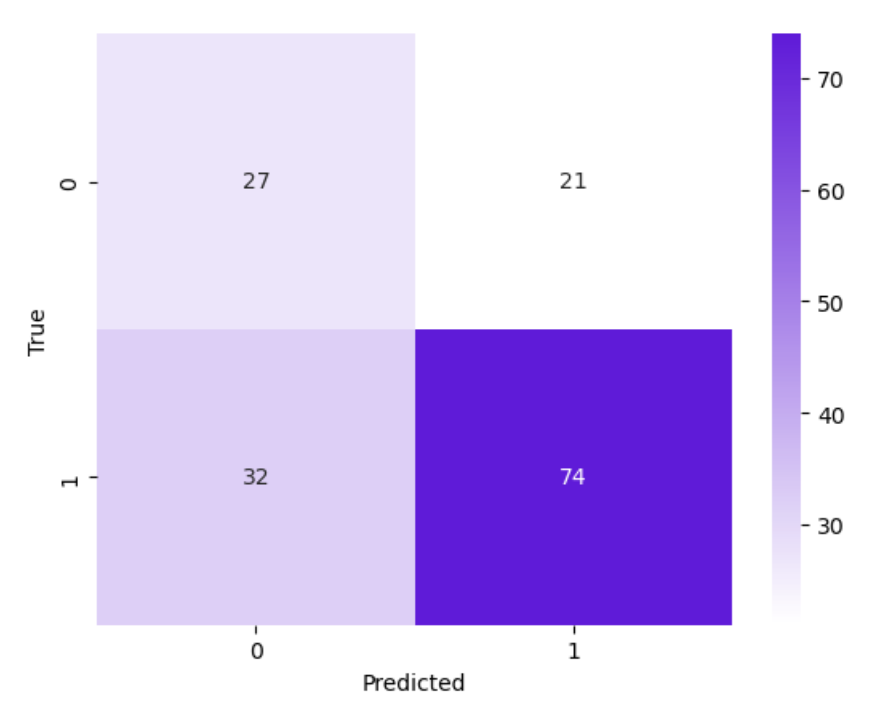


### Figure 16: Feature importance

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



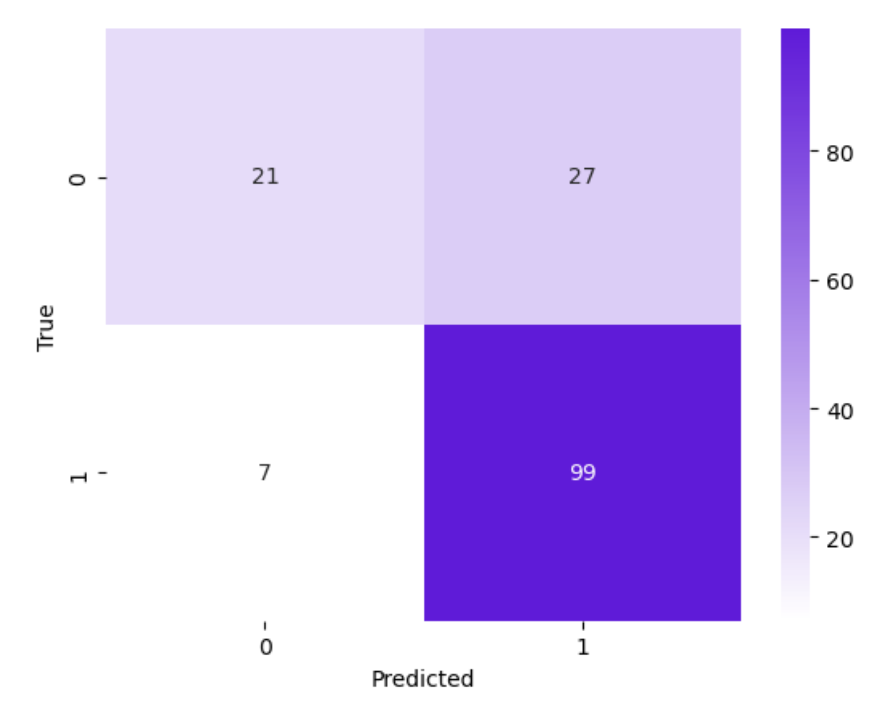
### Figure 17: Random Forest Classification confusion matrix (SMOTE)

The confusion matrix showcase 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 positive 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 negative 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).

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

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



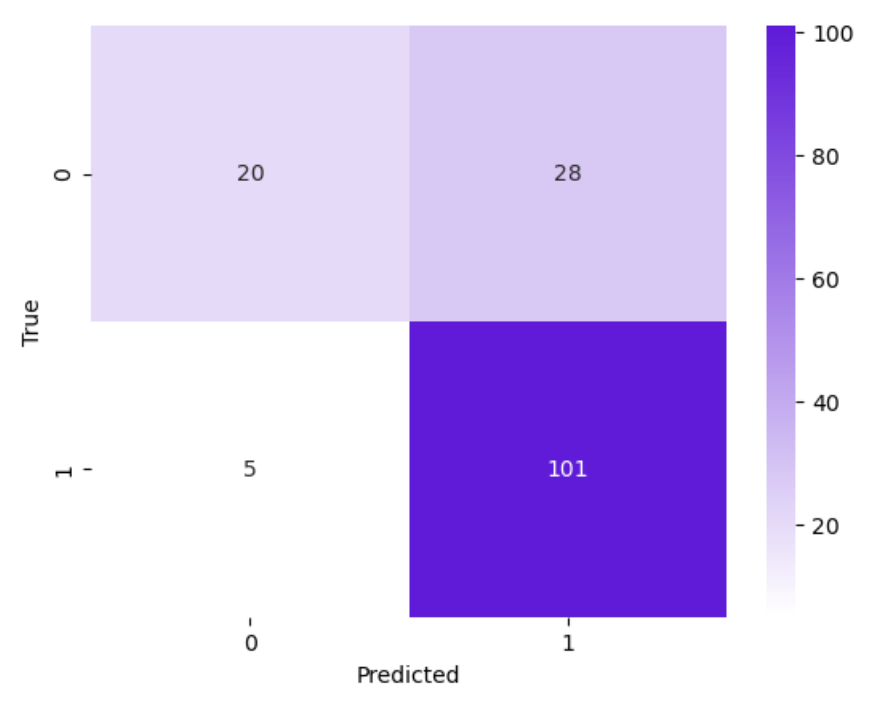
### Figure 18: Logistic regression confusion matrix (SMOTE)

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 positive cases, where loan rejections were predicted but the applicants actually had their loans approved, and 27 false negative 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



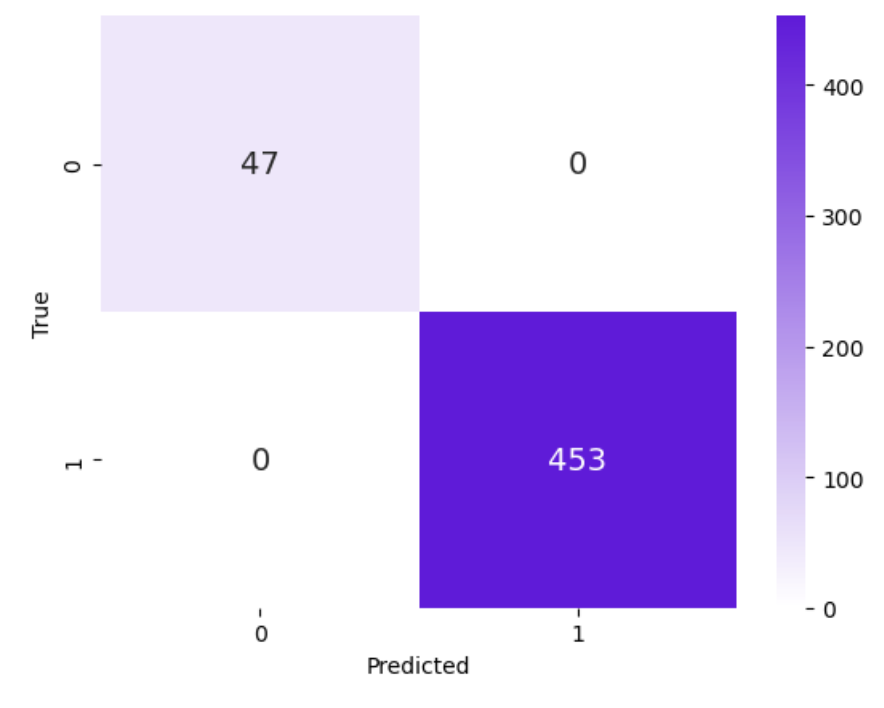
### Figure 18: Support Vector Classification (SMOTE)

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 positives, where the model erroneously forecasted loan rejections but the loans were approved, and 28 instances of false negatives, 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

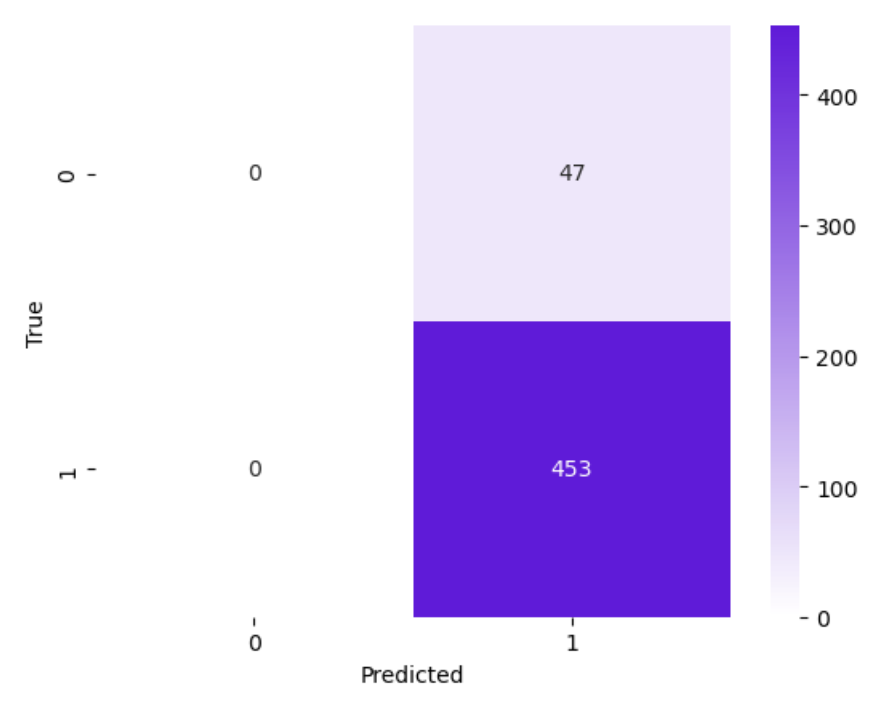


### Figure 19: Support Vector Classification (SMOTE and GridSearchCV)

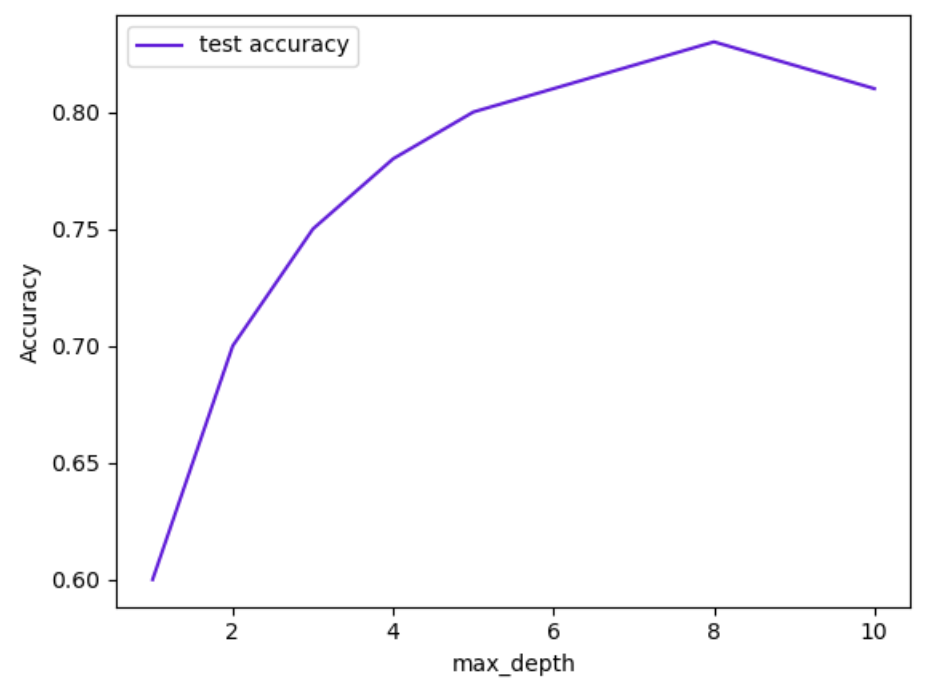
Support Vector Classification with SMOTE and GridSearchCV performs the best. It identifies 453 instances as true positives, accurately forecasting loan approvals. Likewise, 47 cases were correctly classified as true negatives, with the model's predictions of loan rejections aligning with actual outcomes 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 20: Random Forest Classification (SMOTE and GridSearchCV)



### Figure 21: 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 negative instances, where the model inaccurately predicted loan approvals that were in fact rejected. Cross-validated accuracy is 0.99. Table 21 showcase the impact of max depth on model accuracy. After the model was hypertunes, there is a high performance observed across different max\_depth values which suggest that the model captures well the underlying patters 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

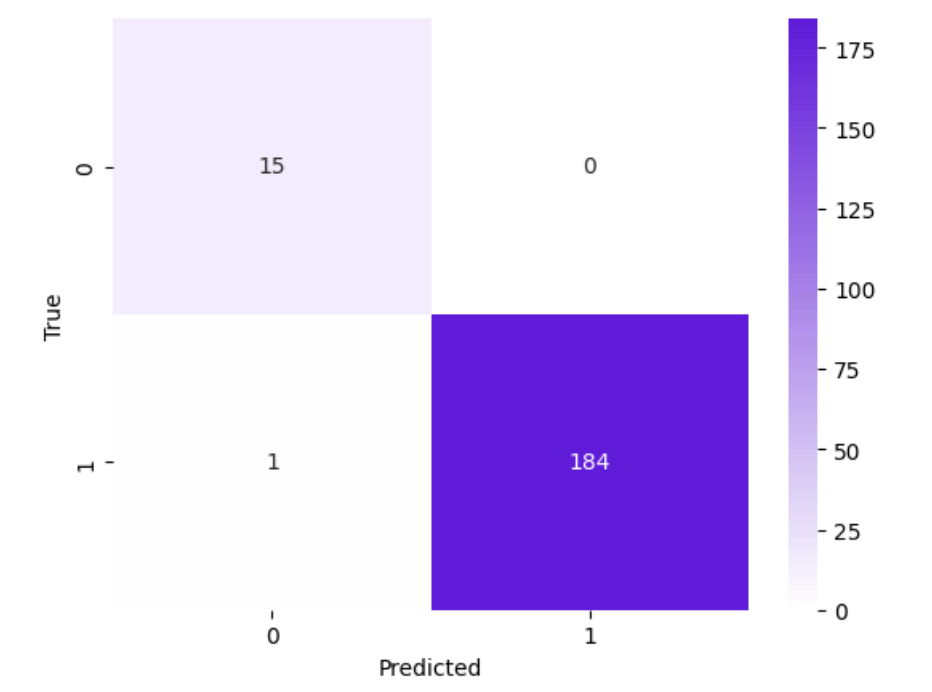
Such a small loss could be a sign for overfitting and that the model learned on training data very well. The accuracy score indicates that the model has perfectly classified all training data.

*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.

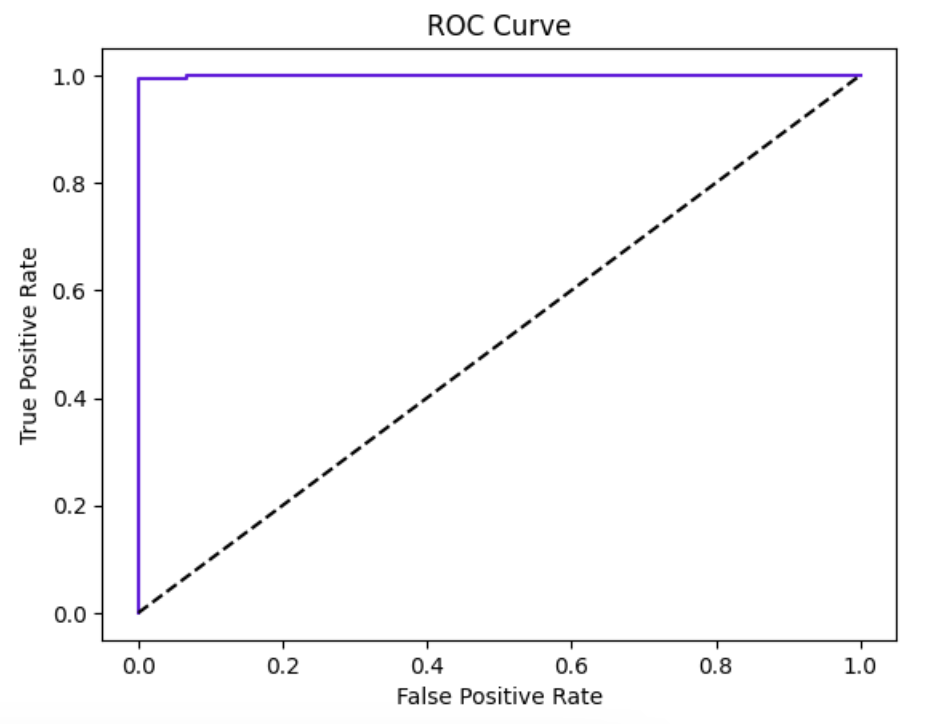
**Artificial Neural Network** results are as follows:

* 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 22: 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 positive 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 22: Artificial Neural Networks - ROC curve

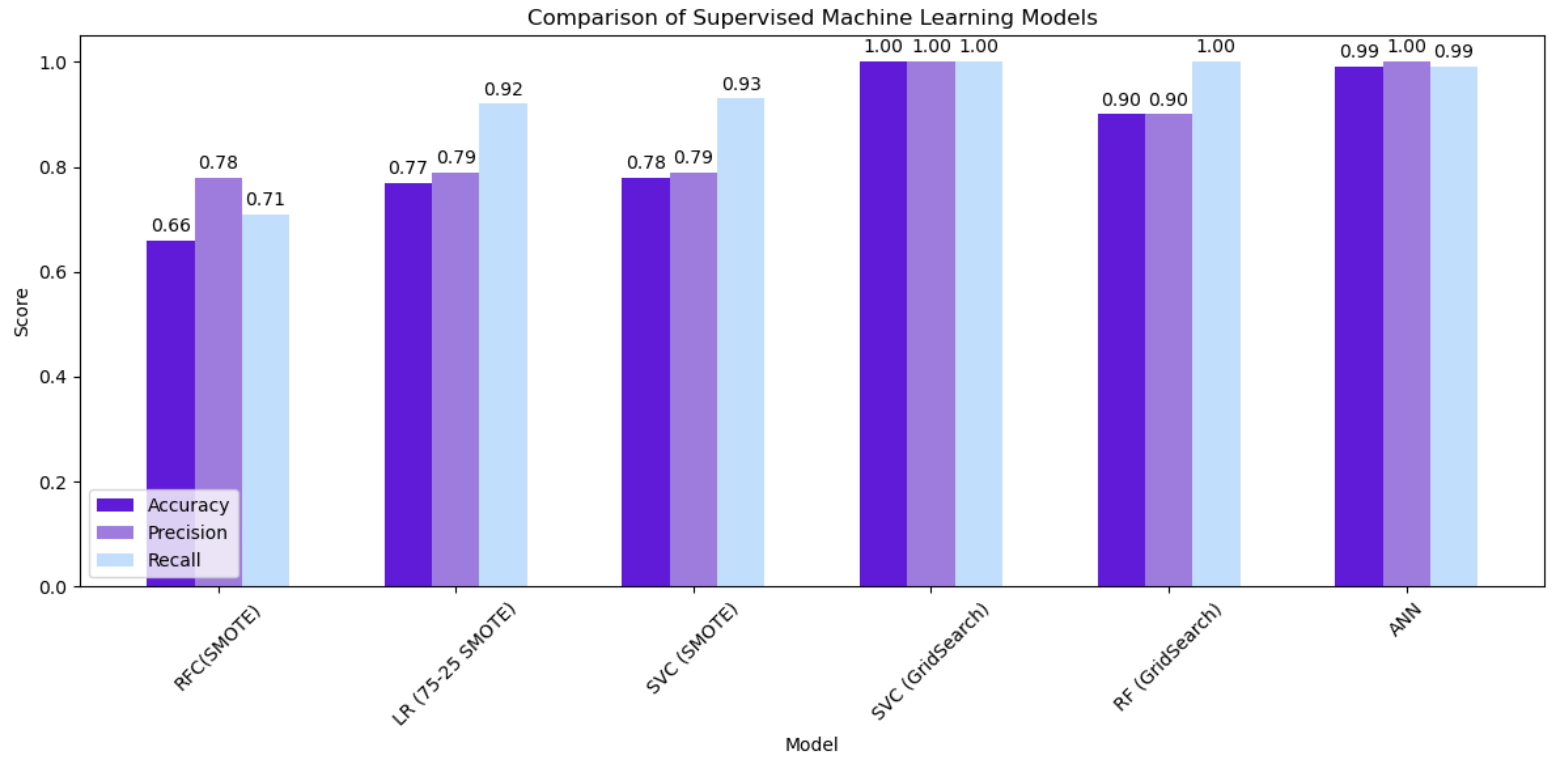
The ROC curve above is the same graphical representation of the confusion matrix for Artificial Neural Network model classified the dependent variable ‘Loan\_Status’. (Bhandari, 2020)

## Model deployment

Word count: 449

In the previous Semester in Class Assignment 1, we tested the classification with Decision tree classification, Random Forest Regressor, Random Forest Classification and KNeighbours. When comparing the results we concluded that Random Forest Classification would be the model to use for deploying on test datasets.

In this semester, we used a confusion matrix to validate the outcome of each model. We also introduced different types of models from the previous semester such as Support Vector Machine and Artificial Neural Network. 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 23: Model comparison

In summary, comparing all the employed methods and learning models, the SVM(SMOTE and GridSearch) model 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 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 datasets is smaller than the training dataset with 367 observations and 12 features. We checked if there are any missing values and we found 11 for *‘Gender’*, 10 for *‘Dependents’,* 23 for *‘Self-employed’,* 5 for *‘LoanAmount’*, 6 for *‘Loan\_Amount\_Term’ and ‘Credit\_History’* - 29. We used the same approach for treating null values and we applied SimpleImputer on *‘Gender’, ‘Dependents’, ‘Self\_Employed’, ‘LoanAmount’, ‘Loan\_Amount\_Term’* and KNNimputer for *‘ApplicantIncome’, ‘LoanAmount’, ‘Loan\_Amount\_Term’ and ‘Credit\_History’.* We applied label encoder on all categorical values: *'Gender', 'Married', 'Dependents', 'Education', 'Self\_Employed', 'Property\_Area'* and StandardScaler on all numerical columns: *'ApplicantIncome', 'CoapplicantIncome', 'LoanAmount', 'Loan\_Amount\_Term', 'Credit\_History'*. 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.

## —-------5438—-------

### 

## Results and Discussions

## Conclusions

## Challenges

We followed best practices for remote collaboration, using Google Meet to facilitate meetings once weekly. We also utilised Google Docs to share documents, allowing us all to collaborate on the document. We followed CRISP methodology for working on the project, which was easy to follow.

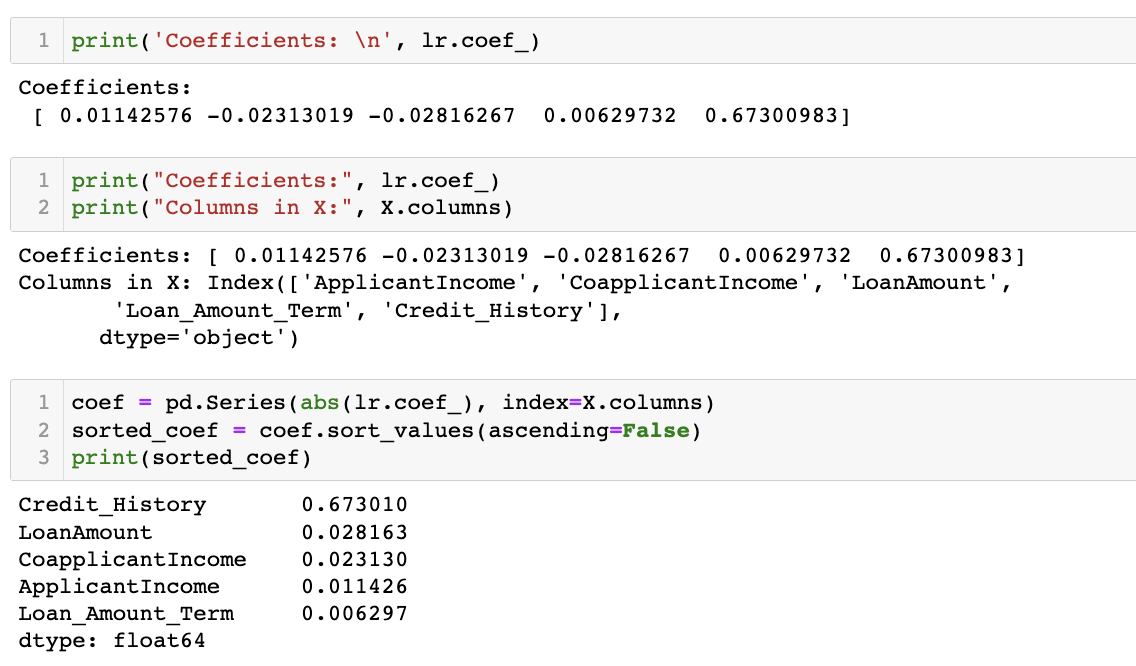
When we implemented the machine learning model, we received errors that there are still null values present. We repeated the process of cleaning the null values. The same applies for string data in column ‘Dependents’, where we treated the data with values 3+ and we needed to convert to integer. We had to repeat the steps.

It was challenging to implement the **deployment** of the machine learning model. We treated the new test data in a slightly different manner, where we imputed the missing values with ‘mean’ and also added a column ‘Loan\_Status’, as it was missing in the file. Even though we had all the libraries in place, we received an error that .pkl file is missing. After a couple of attempts, we managed to successfully deploy the Linear regression model and we received results that some of the instances are predicted with 98% accuracy.

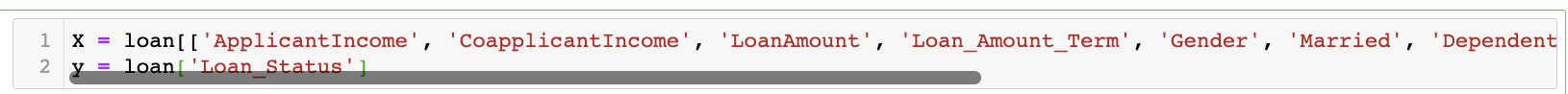
We outlined the feature importance for the best performing model and it has proven to be correct based on the research done by others in this area. We can not be fully confident that our data has been pre-processed with suitable methods or metrics, thus leading the outcome with biased results, however we used recommended approaches and accomplished satisfactory results.

## Results and analysis

Since the best performing score is for Linear regression model, we implemented coefficient magnitudes sorting values, based on the features we initially selected and we received the score as follows: Credit history 0.673, Loan amount 0.028, CoApplicant income 0.023, Applicant income 0.011 and Loan amount Term 0.006. We wanted to avoid biases and we added all of the attributes as features in this analysis and the outcome was slightly different.

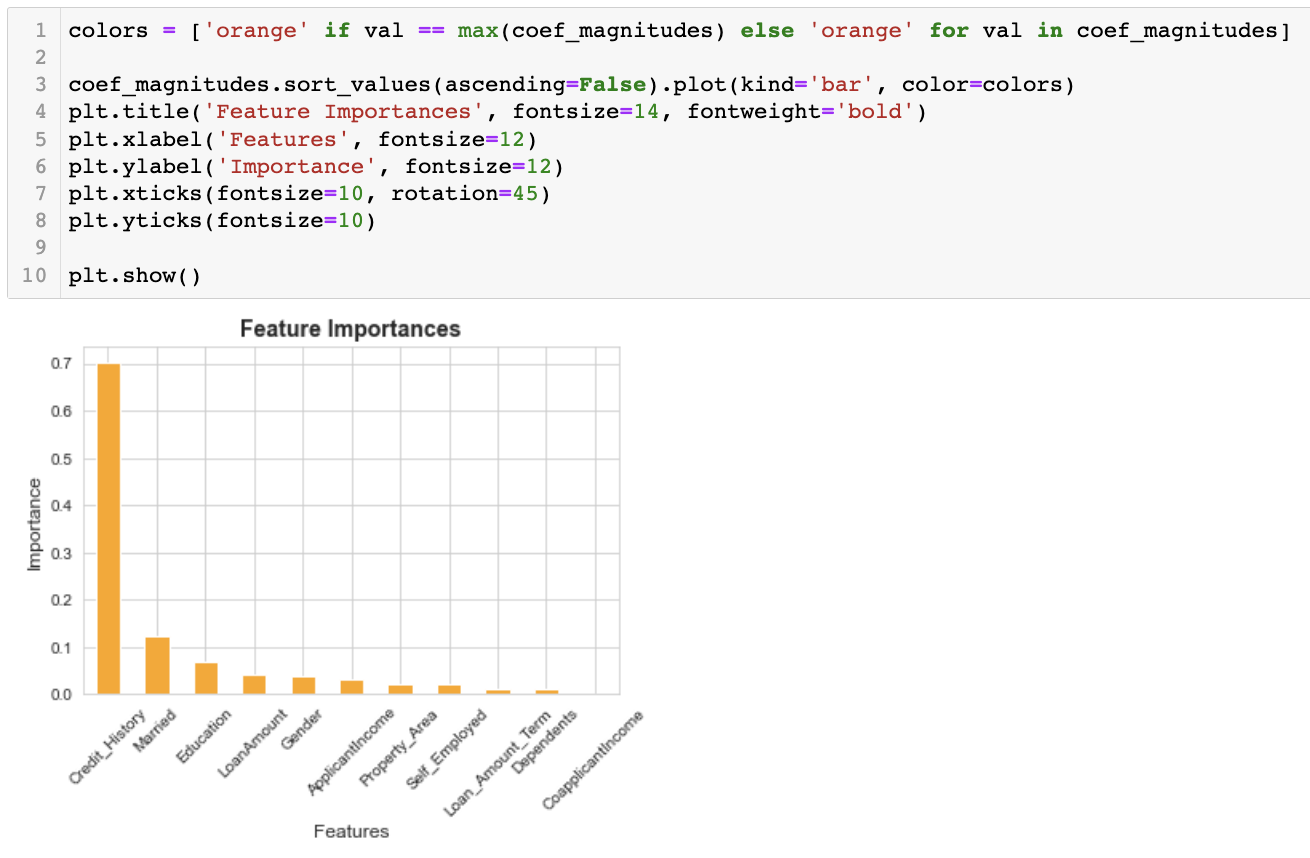


### Figure 31: Coefficient magnitudes



### Figure 32: Additional features

The model was tested with all of the features present in the datasets Loan and ‘Credit history’ stands out with its feature importance.



### Figure 33: Ascending order of coefficients magnitudes

The highest ranking is Credit history, followed by the marriage status of the applicants and the request loan amount. The results from this dataset analysis confirm the findings of (Lee and Lee, 2018) that Credit history impacts the decision to lend loans.

It is important to highlight that Heatmap displayed the same results, where we noticed high correlation between Loan status and Credit history. This feature also influence on the requested loan amount, where in the event that the applicant doesn’t have credit history, there is a higher likelihood of rejection of the loan application and if the applicant has credit history, there is a higher likelihood of approval of the loan application.

We believe that these datasets are missing key applicant’s information. For instance, if the applicant is owning already a home or renting as the asset owning can affect the individual’s financial capacity.

Overall, we are content with the results and the outcome is sufficient and proves the hypothesis.

## 

Using a large dataset, this project has demonstrated how machine learning techniques can be applied effectively to predict loan defaults. After carefully inspecting and preparing the data, which included handling missing values, encoding categorical variables, and normalizing numerical features, we began our analysis.

We tested the predictive power of several machine learning models, including Linear Regression, Decision Trees, and Random Forests. We evaluated and compared these models using various metrics in a methodical manner. We were able to improve the performance of our linear regression classifier by using feature selection approaches, and it proved that credit history is the most significant predictor for approving loans to applicants. Other factors that influence the decision to lend loans include marital status. Individuals who are married are more likely to get their applications approved because, in case one of the individuals defaults on the loan, the partner can support and continue paying the due credit. Education also has an effect on the decision, and it appears to be the third most important feature.

## 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](http://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](http://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](http://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](http://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.

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.

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