SOLENT UNIVERSITY

FACULTY OF BUSINESS,

LAW AND

DIGITAL TECHNOLOGIES

MSc Applied AI and Data Science
Academic Year 2021-2022
A. Yusuf

An intelligence loan approval system for

financial institutions

SOLENT UNIVERSITY

FACULTY OF BUSINESS LAW AND DIGITAL TECHNOLOGIES

MSc Applied AI and Data Science

Academic Year 2021-2022

A. Yusuf

An intelligence loan approval system for financial institutions

Supervisor: Prof Shakeel Ahmad

September 2022

This report is submitted in partial fulfilment of the requirements of Solent University for the degree of MSc Artificial Intelligence and Data Science

ACKNOWLEDGEMENT

Thanks to Allah Almighty who enable me to research in this project.

I revere the patronage and moral support extended with love by my family and friends whose support and passionate encouragement made it possible for me to complete this project.

I humble dedicate this to my Late Father and all concern person who understand and cooperate with me in this regard

Yusuf Amusa

List of Contents

CHAPTER 1

CHAPTER 2

CHAPTER 3

CHAPTER 1

INTRODUCTION AND BACKGROUND

The primary loan providers are often commercial and non-commercial banks. Lending Club (LC), a peer-to-peer online lending marketplace, stands in contrast to others. The largest marketplace in the world for connecting borrowers and investors, it offers consumers and small business owners reduced credit costs and a better lending experience than traditional banks while also providing investors with attractive risk-adjusted returns.

American peer-to-peer lender LendingClub has its main office in San Francisco, California. [3] It was the first peer-to-peer lender to provide loan trading on a secondary market and to register its offerings with the Securities and Exchange Commission (SEC) as securities. The biggest platform for peer-to-peer lending in the world is LendingClub. According to the business, as of December 31, 2015, loans totaling $15.98 billion had originated using its platform.

Investors can now easily access this alternative investment asset class by making loans to individual borrowers through websites like LendingClub, Prosper Marketplace, and Upstart, or too small businesses through Funding Circle, thanks to the rise in popularity of peer-to-peer lending platforms in recent years.

Borrowers submit loan applications to the platform, which conducts credit assessments and accepts or rejects each application based on the results. To calculate the interest rate for accepted loans based on the creditworthiness of borrowers, the platform also uses a proprietary methodology.

The portal then lists approved loans for funding from investors. By merely investing a little sum, such as $25, in each loan, investors typically seek to diversify their portfolio.

RESEARCH QUESTIONS

The questions for research that are meant to be the bone of contention for this project are:

1. What are the relationships between the features and details of the borrower?
2. What are the relationships between the details of the borrower concerning if he will repay completely or not?
3. What is the prediction? Would he pay in full or not?
4. What are the factors contributing to the prediction

AIM AND OBJECTIVE

To invest in loans with lower perceived risks, investors should be able to swiftly and independently assess the credit risk of a large number of listed loans.

This encourages the building of machine-learned classification models that can predict credit risk with a historical loan dataset from LendingClub.

CHAPTER 2

LITERATURE REVIEW

Numerous studies have been done on classification models that forecast LendingClub loan default. The geometric mean of true positive and true negative rates, or G-mean score, was developed by Chang et al. using Logistic Regression, Naive Bayes, and SVM classifiers.

We challenge the treatment of loans with a Current status and Fully Paid debts as instances of good practice. This technique unavoidably misrepresents some real downsides as positives because existing debts could go into default in the future. We choose to limit our dataset to only finished loans in light of this.

Along with Random Forest, Tsai et al. experimented with the three models mentioned above, but with a focus on precision at the expense of recall and negative predictive value (i.e. precision for the negative class). They discover that Logistic Regression outperforms the other models in terms of precision. They also categorize the metrics according to the loan grades (A-G) and subgrades (such as A-1) that LendingClub has allocated to each loan. As a naive model that always predicts positively already obtains a good precision because the vast majority of cases are positive, we think that accuracy for both classes and their recalls are equally relevant metrics to optimize for.

Since the majority of examples in both classes are positive, a naive model that always predicts positively already achieves a good precision, but its negative predictive value would be 0, we feel that precision for both classes and their recalls are equally relevant metrics to improve for.

Gutierrez and Mathieson developed regression models that forecast the annualized return of a specific loan in addition to categorization models that forecast loan default. The loan selection strategy created by combining these models was successful in outperforming the baseline in terms of investment performance as indicated by the Sharpe ratio. This motivates us to develop regression models and assess an investment approach that chooses loans with adequate annualized return forecasts.

Classification and regression models were created by Pujun et al. to forecast LendingClub loan approval and the associated interest rates. To identify hidden tendencies in LendingClub authorized loans, they used k-means clustering and PCA algorithms. Their most intriguing discovery is that over time, loan approval standards have been gradually loosened. This confirms the necessity and benefit of creating an impartial and reliable methodology for assessing credit risks.

CHAPTER 3

METHODOLOGY

This initiative uses a quantitative experimental research methodology. This research method was chosen because algorithms are being developed to predict a classification dataset based on research. Metrics from the output data are used to measure how well various algorithms are performing. It is impossible to look at a problem and find an ideal solution because all dimensions are interrelated. Therefore, testing many algorithms and models, experimenting with different parameters and finding the optimal solution is the best way to solve these problems.

We also chose an experimental research method because our research involves machine learning. Many machine learning algorithms have hyperparameters for which ideal values are often not calculated for real data. So, your only option is to try different values for the hyperparameter and see which one works best. The project uses a deductive research methodology. This strategy was chosen because we could get historical data set of more than 9,000 customers applying for loan. By running algorithms that use historical data and extract specific values from the results, you can test your system and predict whether your system will continue to perform well in the future. This initiative uses an experimental research methodology.

This research approach was chosen because all variables and factors are controlled during algorithm testing. The output of this algorithm depends on parameters and historical data. In these cases, you can explore the relationship between variables and outcomes and make behavioural inferences.

DATASET:

Read in from CSV format into the notebook using Python's Pandas library

Here are what the columns of the data represent:

- credit. policy: 1 if the customer meets the credit underwriting criteria of LendingClub.com, and 0 otherwise.
- purpose: The purpose of the loan (takes values "credit_card", "debt_consolidation", "educational", "major_purchase", "small_business", and "all_other").
- int_rate: The interest rate of the loan, as a proportion (a rate of 11% would be stored as 0.11). Borrowers judged by LendingClub.com to be riskier are assigned higher interest rates.
- instalmentnt: The monthly instalments owed by the borrower if the loan is funded.
- log.annual.inc: The natural log of the self-reported annual income of the borrower.
- diti: The debt-to-income ratio of the borrower (amount of debt divided by annual income).
- fico: The FICO credit score of the borrower.
- days.with.cr.line: The number of days the borrower has had a credit line.
- revol.by: The borrower's revolving balance (amount unpaid at the end of the credit card billing cycle).
- revol.util: The borrower's revolving line utilization rate (the amount of the credit line used relative to total credit available).
- innq. last.6 months: The borrower's number of inquiries by creditors in the last 6 months.
- delinq.2yrs: The number of times the borrower had been 30+ days past due on a payment in the past 2 years.
- pub.rec: The borrower's number of derogatory public records (bankruptcy filings, tax liens, or judgments).

FRAMEWORKS, LIBRARIES AND TECHNOLOGIES USED

Languages Used

1. Data Wrangling: Python
2. Data exploration and visualization: Python
3. Machine Learning: Python
4. Front end : HTML, CSS, JavaScript

Frameworks and Libraries used

5. Numpy
6. Pandas
7. Matplotlib
8. Seaborn
9. Scikit Learn
10. Flask
11. Treeinterpreter

PYTHON IN DATA AND MACHINE LEARNING

Python is a high-level, open-source, interpreted language that offers a fantastic approach to object-oriented programming. Data scientists utilize it as one of the best languages for a variety of projects and applications. Python has excellent capabilities for working with mathematical, statistical, and scientific functions. It offers excellent libraries for dealing with applications of data science.

Because of its simplicity and ease of use, Python is one of the most popular programming languages in the scientific and research sectors. People without engineering backgrounds may easily learn how to use it because of this. Additionally, it is better suited for rapid prototyping.

Engineers from academia and industry claim that deep learning frameworks made available with Python APIs, along with the scientific packages, have greatly increased Python's productivity and versatility. Python deep learning frameworks have seen significant change, and they are rapidly evolving.

ML scientists favour Python as well in terms of application domains. Developers tended to lean toward Java when it came to areas like creating fraud detection algorithms and network security, but they chose Python for applications like sentiment analysis and natural language processing (NLP) because it has a large library of tools that make it easier to solve complex business problems and create robust systems.

BRIEFLY ABOUT THE USED LIBRARIES

1. NUMPY:

The Python package NumPy is used to manipulate arrays.

Additionally, it has matrices, Fourier transform, and functions for working in the area of linear algebra.

The equivalent of arrays in Python lists, although they take a long time to execute. The goal of NumPy is to offer array objects that are up to 50 times faster than conventional Python lists. The NumPy array object is referred to as ndarray, and it has several supporting methods that make using ndarray relatively simple.

In data research, arrays are often employed when speed and resources are crucial.

Because NumPy arrays are continuously maintained in memory rather than lists, processes can access and work with them quite effectively.

Computer science refers to this behaviour as the locality of reference.

NumPy outperforms lists in terms of speed primarily due to this. It is also designed to operate with the most recent CPU architectures.

Although part of the NumPy library is written in Python and requires quick processing, the majority of it is written in C or C++.

2. PANDAS:

Python's Pandas package is used to manipulate data sets. It offers tools for data exploration, cleaning, analysis, and manipulation.

With the aid of Pandas, we can examine large data sets and draw conclusions based on statistical principles. Pandas can organize disorganized data sets, making them readable and useful. In data science, relevant data is crucial.

Pandas provide you with information on the data. Like:

a. Does a relationship exist between two or more columns?

b. What is the median value?

c. The maximum?

d. Minimum value

Rows that are irrelevant or contain incorrect data, such as empty or NULL values, can also be deleted by Pandas. This process is known as data cleaning.

3. MATPLOTLIB:

A tool for visualizing data, Matplotlib is a low-level graph charting framework written in Python.

Since Matplotlib is open source, we are allowed to utilize it.

For platform compatibility, Matplotlib is primarily written in Python, with a small amount of code written in C, Objective-C, and Javascript.

Python's Matplotlib is a fantastic visualizing package that is simple to use. It is constructed using NumPy arrays, intended to operate with the larger SciPy stack, and includes several graphs, including lines, bars, scatters, histograms, and others.

4. SEABORN:

An open-source Python library based on matplotlib is called Seaborn. It is utilized for data exploration and data visualization. With data frames and the Pandas library, Seaborn functions with ease. The generated graphs are also easily customizable. Several advantages of data visualization are listed below.

In any machine learning or forecasting project, graphs can assist us in identifying helpful data trends.

It is simpler to convey your data to non-technical folks when you use graphs.

Readers will find presentations and reports much more enticing when the graphs are visually pleasing.

5. FLASK:

Python is used to create the Flask web application framework. It has a variety of modules that make it simpler for a web developer to construct applications without needing to focus on the specifics, such as protocol management, thread management, etc.

Flask offers us a range of options for creating web apps and provides us with the necessary tools and libraries to do so.

6. TREEINTERPRETER:

Treeinterpreter is the software for interpreting scikit-decision learns tree and random forest predictions.

You can use a Python module to determine the effect of each feature you choose for the Random forest method.

DATA CLEANING

The practice of correcting or deleting inaccurate, damaged, improperly formatted, duplicate, or incomplete data from a dataset is known as data cleaning. There are numerous ways for data to be duplicated or incorrectly categorized when merging multiple data sources. Even if results and algorithms appear to be correct, they are unreliable if the data is inaccurate. Because the procedures will differ from dataset to dataset, there is no one definitive way to specify the precise phases in the data cleaning process.

The process involved in cleaning the dataset includes:

- Renaming the columns of the dataset: The full stop'.' sign was replaced with underscore '_'. For example, ['credit.policy', 'purpose', 'int.rate', 'installment', 'log.annual.inc', 'dti', 'fico', 'days.with.cr.line', 'revol.bal', 'revol.util', 'inq.last.6mths', 'delinq.2yrs', 'pub.rec', 'not.fully.paid'] became:

['credit_policy', 'purpose', 'int_rate', 'installment', 'log_annual_inc', 'dti', 'fico', 'days_with_cr_line', 'revol_bal', 'revol_util', 'inq_last_6mths', 'delinq_2yrs', 'pub_rec', 'not_fully_paid']

This was done so as not to confuse the model and the python syntax.

EXPLORATORY DATA ANALYSIS

Data analysis utilizing visual methods is called exploratory data analysis (EDA). The use of statistical summaries and graphical representations, it is used to identify trends, and patterns, or to verify assumptions.

Exploratory data analysis is the crucial process of doing preliminary analyses on data to find patterns, identify anomalies, test hypotheses, and double-check assumptions with the aid of summary statistics and graphical representations.

Understanding the data first and attempting to glean as many insights from it as possible is a smart strategy.

THE DATA:

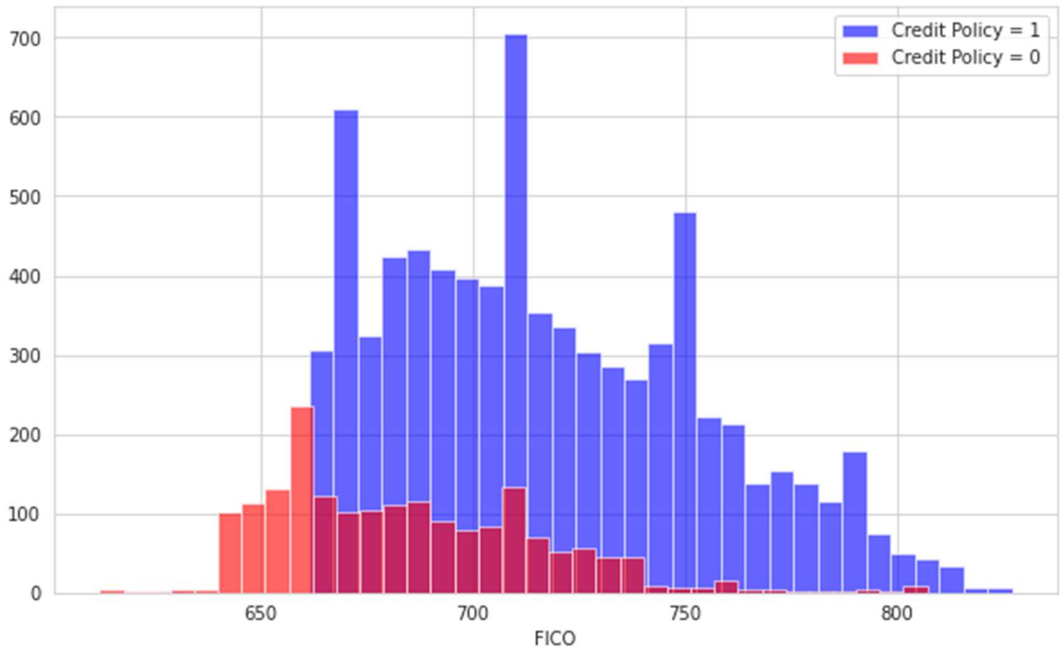
The dataset comprises 14 columns and 9578 rows.

One of the columns (‘purpose’) is categorical and the rest are numerical.

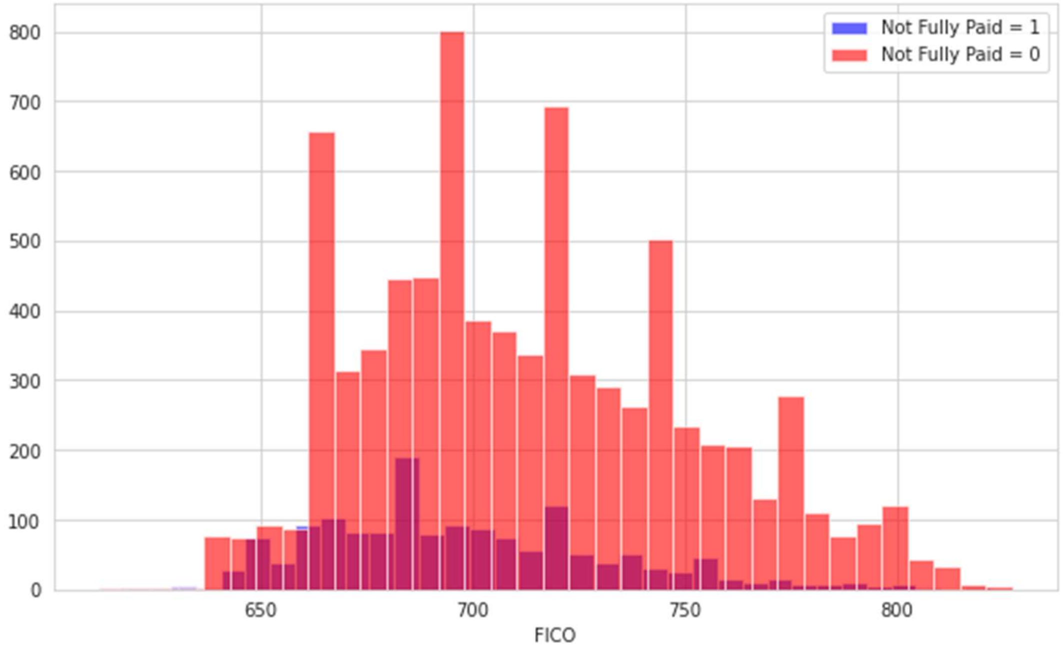
UNIVARIATE ANALYSIS

This is the simplest form of data analysis where the data being analyzed contains only one variable

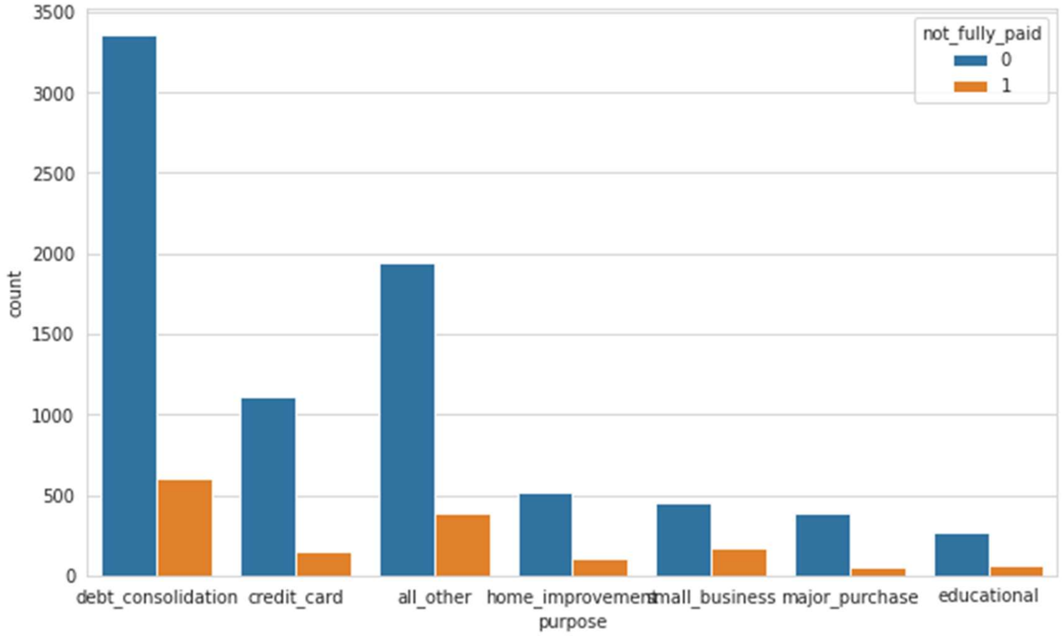
1. A histogram of two FICO distributions on top of each other, one for each credit.policy outcome



2. A similar figure, except this time selected by the not_fully_paid column



3. A countplot was created using seaborn showing the counts of df by purpose, with the colour hue defined by not_fully_paid



4. Each value count was displayed in a data frame and the distributions were noted.

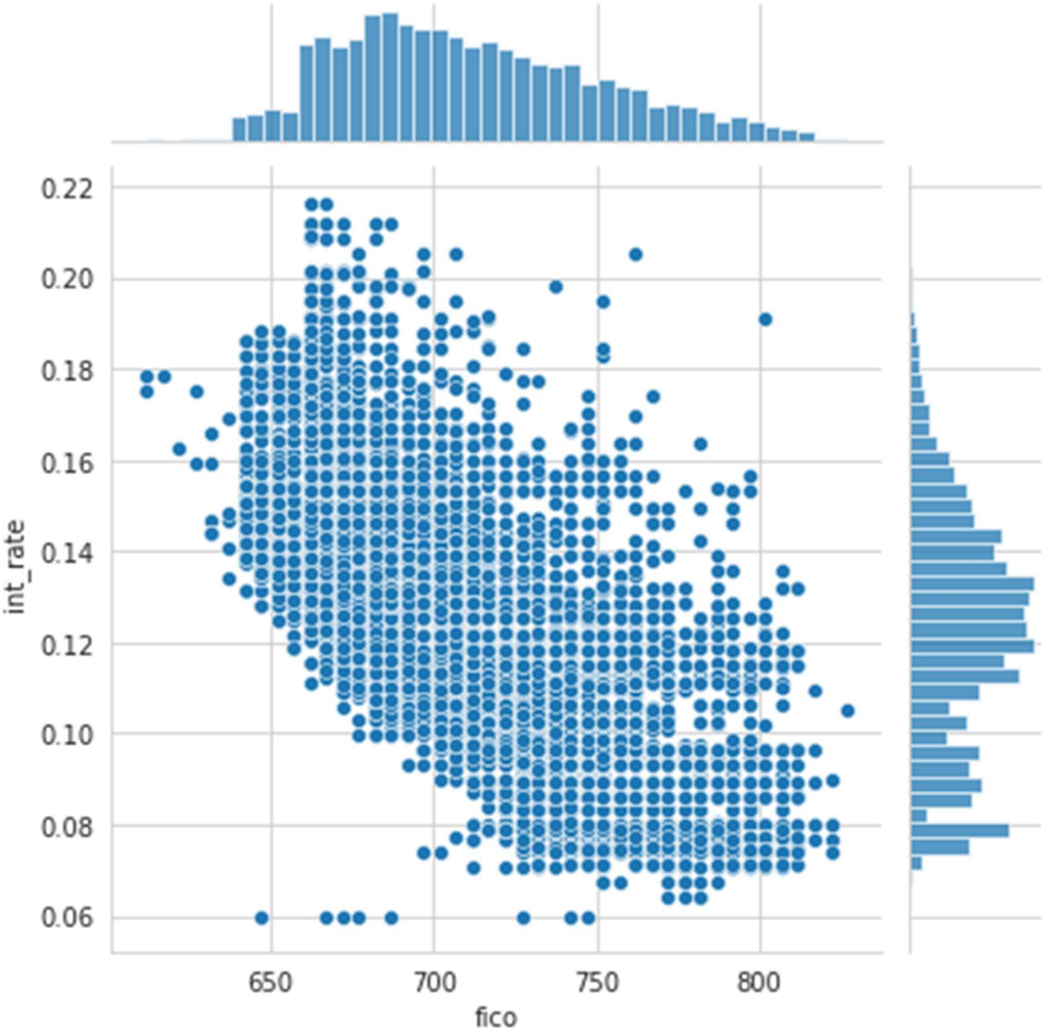
Findings from the analysis:

- People who met the underwriting criteria of LendingClub.com have higher FICO credit scores.
- People who met the underwriting criteria of LendingClub.com are significantly more than those who did not.
- People who did not pay fully are more than those who paid fully.
- The major purpose for borrowing money was due to Debt consolidation and the majority of them did not pay back fully.
- Majority of those that borrowed money because of Credit card issues did not pay it back fully.

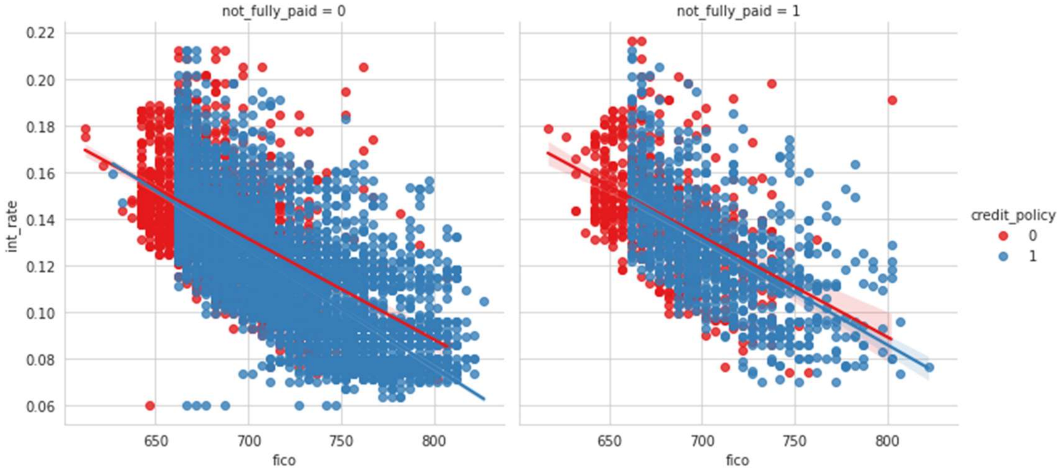
- Majority of those that borrowed money because of Home improvement purposes did not pay it back fully.
- Majority of those that borrowed money because of Major purchases did not pay it back fully.
- Majority of those that borrowed money because of Educational purpose did not pay back fully.
- Et cetera

BIVARIATE and MULTIVARIATE ANALYSIS

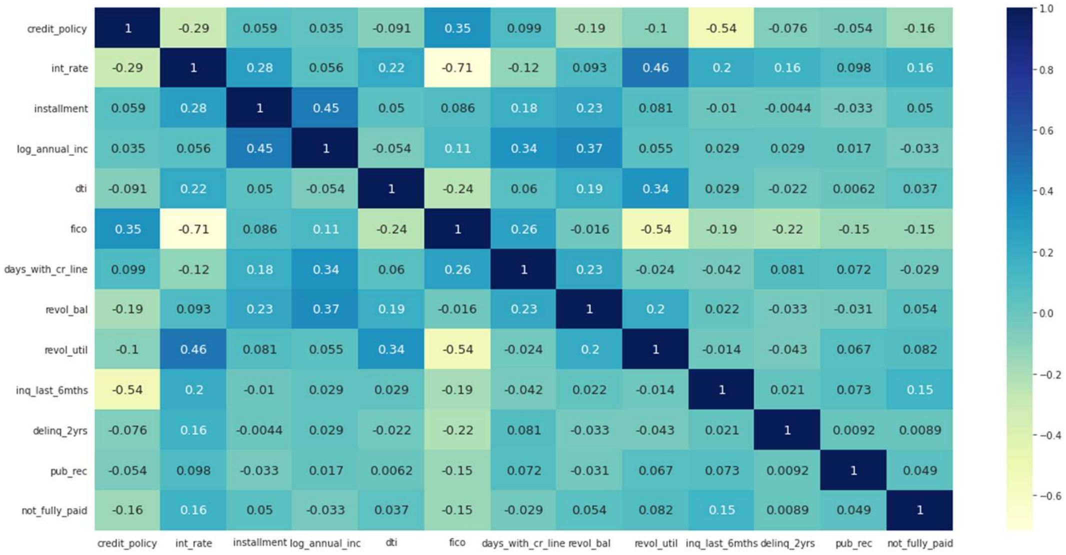
1. The trend between the FICO score and the interest rate was plotted in a joint plot.



2. Created lmplots to see if the trend differed between not_fully_paid and credit. policy.



3. Heatmap was plotted to visualize the correlation



	credit_policy	int_rate	installment	log_annual_inc	dti	fico	days_with_cr_line	revol_bal	revol_util	inq_last_6mths	delinq_2yrs	pub_rec	not_fully_paid
credit_policy	1	-0.29	0.059	0.035	-0.091	0.35	0.099	-0.19	-0.1	-0.54	-0.076	-0.054	-0.16
int_rate	-0.29	1	0.28	0.056	0.22	-0.71	-0.12	0.093	0.46	0.2	0.16	0.098	0.16
installment	0.059	0.28	1	0.45	0.05	0.086	0.18	0.23	0.081	-0.01	-0.0044	-0.033	0.05
log_annual_inc	0.035	0.056	0.45	1	-0.054	0.11	0.34	0.37	0.055	0.029	0.029	0.017	-0.033
dti	-0.091	0.22	0.05	-0.054	1	-0.24	0.06	0.19	0.34	0.029	-0.022	0.0062	0.037
fico	0.35	-0.71	0.086	0.11	-0.24	1	0.26	-0.016	-0.54	-0.19	-0.22	-0.15	-0.15
days_with_cr_line	0.099	-0.12	0.18	0.34	0.06	0.26	1	0.23	-0.024	-0.042	0.081	0.072	-0.029
revol_bal	-0.19	0.093	0.23	0.37	0.19	-0.016	0.23	1	0.2	0.022	-0.033	-0.031	0.054
revol_util	-0.1	0.46	0.081	0.055	0.34	-0.54	-0.024	0.2	1	-0.014	-0.043	0.067	0.082
inq_last_6mths	-0.54	0.2	-0.01	0.029	0.029	-0.19	-0.042	0.022	-0.014	1	0.021	0.073	0.15
delinq_2yrs	-0.076	0.16	-0.0044	0.029	-0.022	-0.22	0.081	-0.033	-0.043	0.021	1	0.0092	0.0089
pub_rec	-0.054	0.098	-0.033	0.017	0.0062	-0.15	0.072	-0.031	0.067	0.073	0.0092	1	0.049
not_fully_paid	-0.16	0.16	0.05	-0.033	0.037	-0.15	-0.029	0.054	0.082	0.15	0.0089	0.049	1

Findings:

- There is an inverse variation relationship between the FICO credit score and the interest rate.
- Credit policy and Interest rate correlated the most to the values of the target.

MODELLING

The problem is a binary classification problem, requiring us to predict whether or not the borrower returned the borrowed money in full.

A quite number of classification algorithms were used which include: 'Logistic Regression', 'Decision Tree Classifier', 'Support Vector Machine', 'Random Forest Classifier', 'AdaBoost Classifier', 'Gradient Boosting Classifier', 'K Neighbors Classifier', 'Gaussian Naive Bayes'.

The data were preprocessed before being fitted into the model.

The preprocessing method includes:

1. Scaling of the numerical columns using standard scaler from the scikit learn preprocessing library.
2. OneHot encoder was used to encode the categorical data.

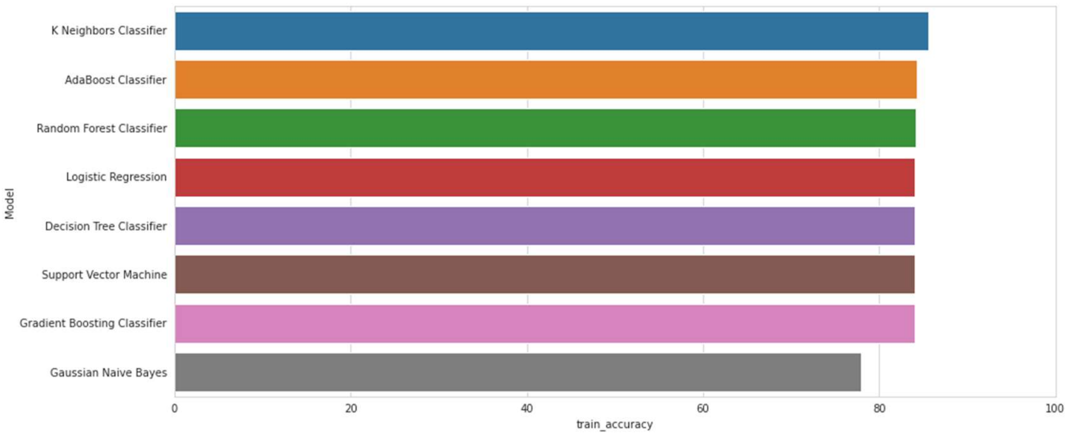
The data was then split into test and train sets in the ratio of 20 percent to 80 per cent.

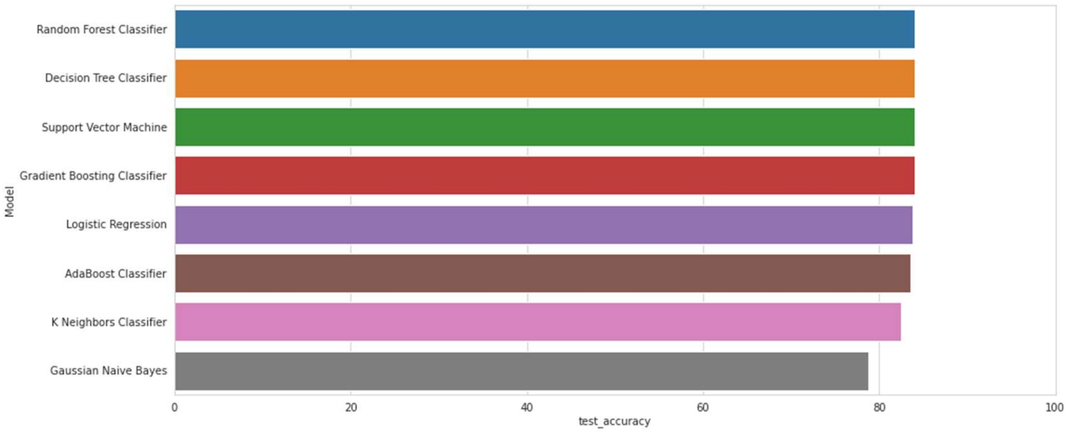
Cross-validation was implemented with Kfold values with a split number of 5.

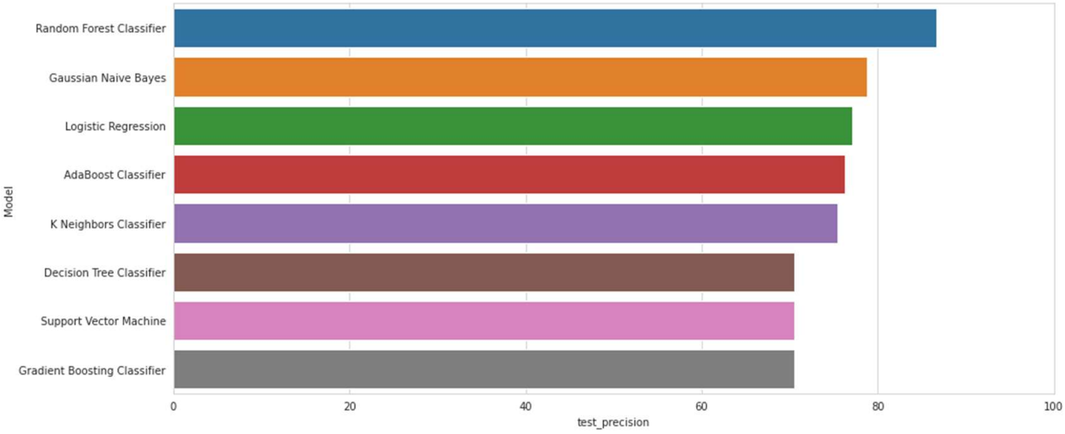
The preprocessing methods and the compiled models were then fitted in a pipeline which was then used to fit then train the model.

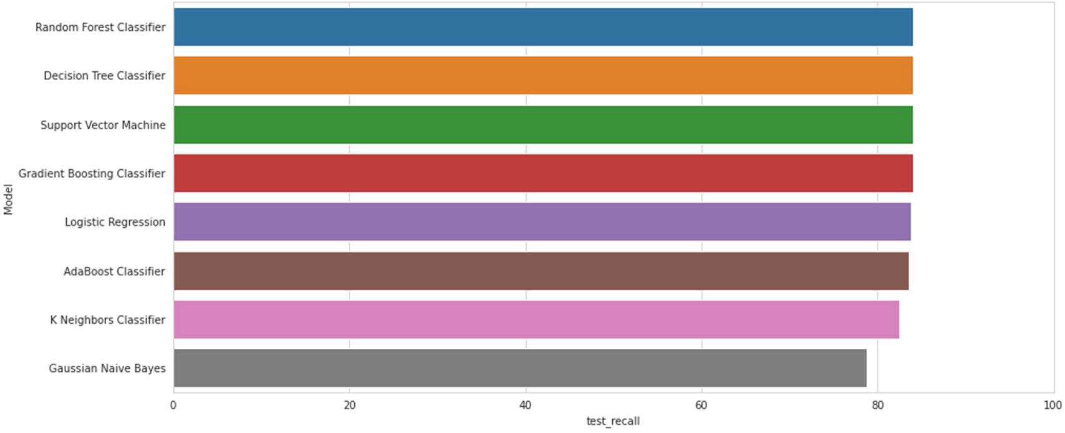
The metrics were then gathered and compared across the classification algorithms used.

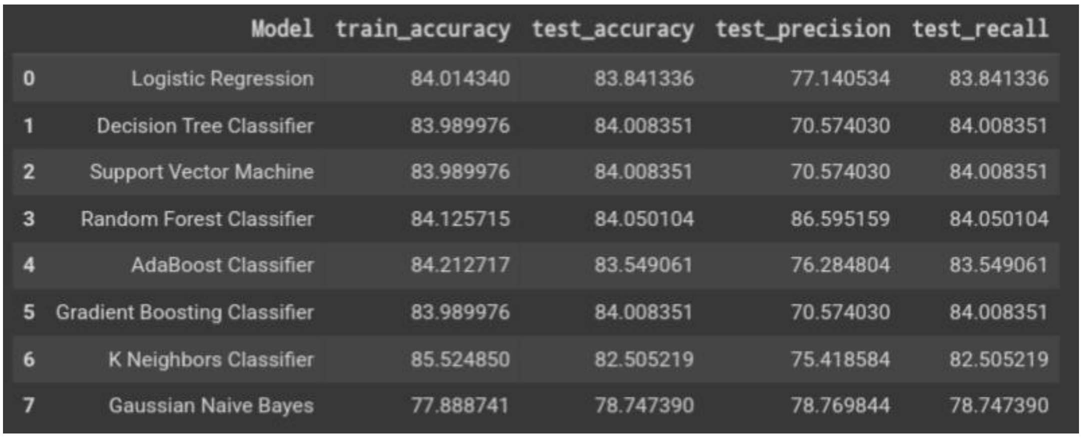
Graphical representations were done to appreciate the metrics comparisons.


K Neighbors Classifier
AdaBoost Classifier
Random Forest Classifier
Logistic Regression
Decision Tree Classifier
Support Vector Machine
Gradient Boosting Classifier
Gaussian Naive Bayes
Model
0
20
40
60
80
100
train_accuracy


Random Forest Classifier
Decision Tree Classifier
Support Vector Machine
Gradient Boosting Classifier
Logistic Regression
AdaBoost Classifier
K Neighbors Classifier
Gaussian Naive Bayes
Model
0
20
40
60
80
100
test_accuracy







	Model	train_accuracy	test_accuracy	test_precision	test_recall
0	Logistic Regression	84.014340	83.841336	77.140534	83.841336
1	Decision Tree Classifier	83.989976	84.008351	70.574030	84.008351
2	Support Vector Machine	83.989976	84.008351	70.574030	84.008351
3	Random Forest Classifier	84.125715	84.050104	86.595159	84.050104
4	AdaBoost Classifier	84.212717	83.549061	76.284804	83.549061
5	Gradient Boosting Classifier	83.989976	84.008351	70.574030	84.008351
6	K Neighbors Classifier	85.524850	82.505219	75.418584	82.505219
7	Gaussian Naive Bayes	77.888741	78.747390	78.769844	78.747390

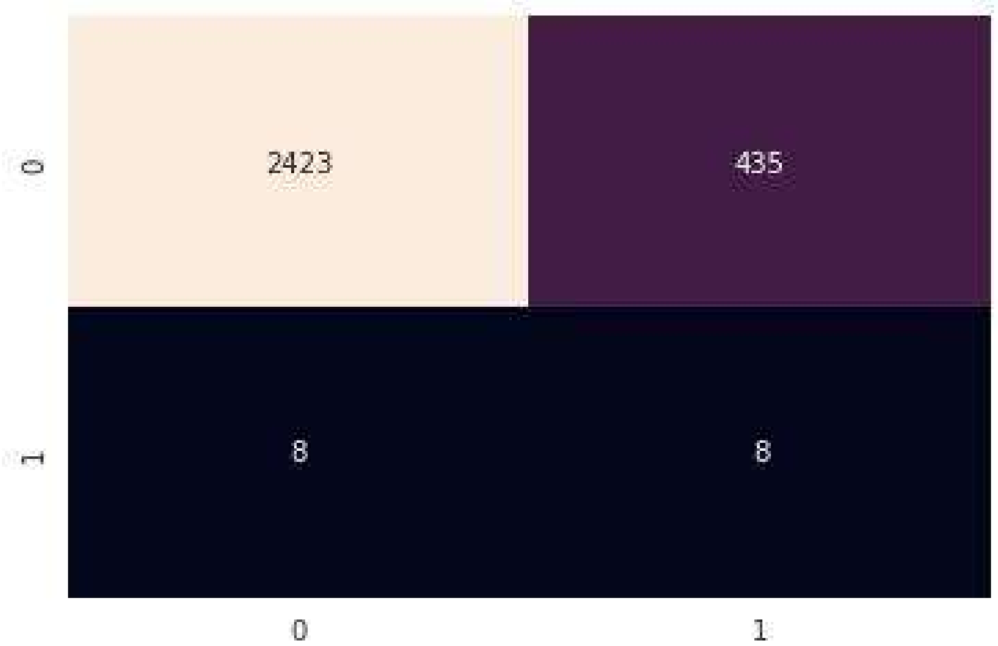
Findings

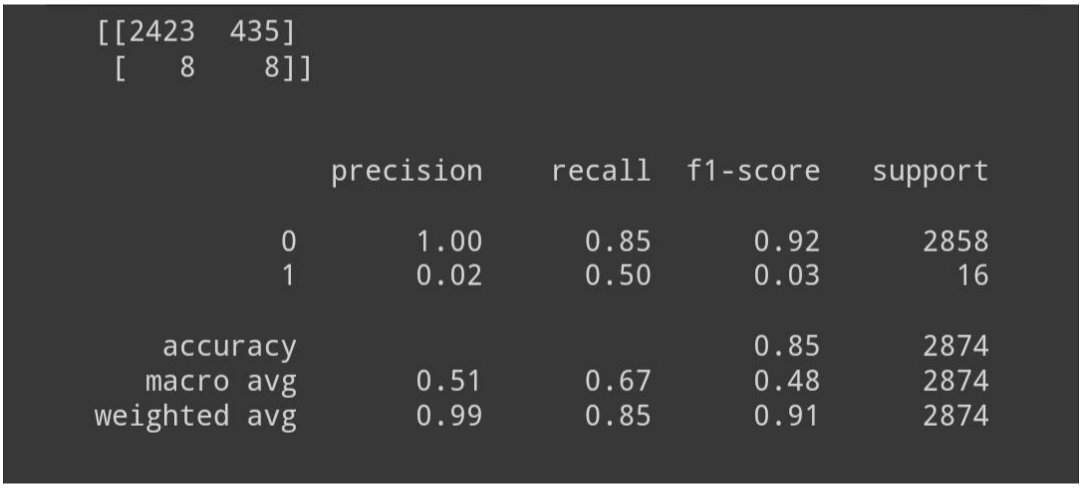
- Random Forest Classifier was noticed to have outperformed the other algorithms.
- K Nearest Classifier was noticed to have overfitted.

Way forward

Seeing that the random forest classifier performed better than the others, we then trained it and tweak hyperparameters to increase the accuracy and other metrics' performances.

A heatmap was plotted for the confusion matrix and the accuracy score, precision score, and F1 score were gotten from the classification report.





```
[[2423  435]
 [   8    8]]

              precision    recall  f1-score   support

           0       1.00      0.85      0.92      2858
           1       0.02      0.50      0.03        16

    accuracy                           0.85      2874
   macro avg       0.51      0.67      0.48      2874
weighted avg       0.99      0.85      0.91      2874
```

GETTING THE MOST IMPORTANT CONTRIBUTIONS

To improve the user performance and experience, there is a need to address the most important contributor to the prediction as a form of telling the user to pay more attention to those factors.

In this case, this tells the companies the reason why the borrower would most likely pay the money back in full or not.

The important library of python used for this is the TREEINTERPRETER LIBRARY.

Messages were rewritten to be sent to the user instead of just returning the suggested

to column(s) to the user.

The re-written messages go thus:

msg = (

    'Whether the borrower meets the credit underwriting criteria of LendingC lub.com',

    'The purpose of the loan',

'The interest rate of the loan',

'The monthly instalments ($) owed by the borrower if the loan is funded',

'The self-reported annual income of the borrower.',

'The debt-to-income ratio of the borrower',

'The FICO credit score of the borrower',

'The number of days the borrower has had a credit line.',

'The borrower’s revolving balance (amount unpaid at the end of the credit card billing cycle).',

'The borrower’s revolving line utilization rate (the amount of the credit line used relative to total credit available).',

'The borrower’s number of inquiries by creditors in the last 6 months.',

'The number of times the borrower had been 30+ days past due on a payment in the past 2 years.',

'The borrower’s number of derogatory public records (bankruptcy filings, tax liens, or judgments).')

These messages were then mapped with the respective columns using python’s zipped function.

However, it is important to note that the treeinterpreter library is not pre-installed in most IDEs, hence it should be installed.

BUILDING THE APP

1. Frontend: HTML, CSS, Javascript

HTML

The preferred markup language for documents intended to be viewed in a web browser is HTML or HyperText Markup Language. Technologies like Cascading Style Sheets (CSS) and scripting languages like JavaScript can help.

HTML documents are downloaded from a web server or local storage by web browsers, who then turn them into multimedia web pages. HTML originally featured cues for the document's design and semantically explains the structure of a web page.

The foundation of HTML pages are HTML components. Images and other objects, like interactive forms, may be embedded within the produced page using HTML techniques.

By indicating structural semantics for text elements like headings, paragraphs, lists, links, quotations, and other objects, HTML offers a way to generate structured texts. Tags, which are written in angle brackets, are used to distinguish HTML elements. Input and image tags, for example, add content directly to the page. Other tags, like p>, enclose and describe the text of the document. They may also contain other tags as sub-elements. Browsers employ HTML tags to decipher the page's content rather than displaying them.

CSS

A style sheet language called Cascading Style Sheets (CSS) is used to describe how a document produced in a markup language like HTML or XML is presented (including XML dialects such as SVG, MathML or XHTML). [1] The World Wide Web's foundational technologies, along with HTML and JavaScript, include CSS. [2]

Layout, colour, and font can all be separated from content and presentation using CSS.

By declaring the pertinent CSS in a separate.css file, which eliminates complexity and redundancy in the structured content, numerous web pages can share formatting, improving accessibility and giving more freedom and control in the specification of presentation features.

and make the.css file cacheable to speed up the page loads for pages that share the file's formatting.

The ability to offer the same HTML page in several styles for various rendering techniques, including on-screen, in print, by voice (using a speech-based browser or screen reader), and on Braille-based tactile devices, is also made possible by the separation of formatting and content. If a user accesses the material on a mobile device, CSS additionally provides rules for different formatting. [4]

When many style rules match an element, the priority system is used to determine which rule should be applied, hence the name cascading. This priority hierarchy is predictable.

JAVASCRIPT

Along with HTML and CSS, the computer language known as JavaScript, or JS, is one of the foundational elements of the World Wide Web. 98% of websites will utilize JavaScript on the client side by the year 2022 to control webpage functionality, frequently integrating third-party libraries. [13] A dedicated JavaScript engine is available in every major web browser and is used to run the code on users' devices.

JavaScript is an ECMAScript-compliant high-level, frequently just-in-time compiled language. It features first-class functions, prototype-based object orientation, and dynamic typing. It supports event-driven, functional, and imperative programming paradigms and is a multi-paradigm. It offers application programming interfaces (APIs) for using the Document Object Model, regular expressions, dates, and standard data structures (DOM).

2. Backend: Flask was used for the backend

- Functions were written to train the model on a random forest classifier.

- Functions were written to find the best contributors to the predictions to be implemented in the app API as to what to do to have better results and performance.

3. Files and folders required for the app to be deployed to the cloud: This includes:

- The static folder containing the images, CSS files and javascript files.
- The template folder containing the HTML files.
- The requirements.txt file containing all required dependencies required for the app to survive in the cloud.
- The runtime.txt file containing the specified version of python used for building the app and supported by the browser.
- The Procfile.

PROCFILE

A Procfile file lists the commands that a Heroku app will run when it first launches.

The instructions that are run by Heroku apps when they launch are listed in a Procfile.

Although a Procfile is not required for Heroku deployment, it does provide more initial configuration options and enables the development of several processes that run distinct dynos.

The root of an application's file system contains a file called the Procfile, which is used to specify a variety of processes.

The app was launched by a Heroku dyno that belongs to one of the specified process types.

REQUIREMENTS.TXT FILE

A requirement.txt file in Python is a specific type of file that often contains data about all the libraries, modules, and packages that are utilized while creating a specific project. Additionally, it keeps all of the files and packages needed for the project to function or on which it depends. The "requirement.txt" file is often located in the root directory of your projects.

As it resolves nearly all compatibility difficulties, it benefits us in several ways, even when we return our project in the future. If you've ever developed a project in Python or worked on one, you know that we typically need a large number of packages.

However, we often used a specific version of packages when working on a project. Later, the maintainer or package manager might make certain adjustments, and those changes could easily damage your entire application. Therefore, it would be very time-consuming to keep track of each package alteration. To avoid unpleasant surprises, it's critical to keep track of every package we use when the project is excessively large.

Making use of a virtual environment is one of the common solutions for these kinds of problems. We typically do not require all of these packages while working on a certain project because there are two primary sorts of packages and places where the Python libraries are typically housed;

4. Deployment: Apply was deployed on Heroku (Salesforce cloud).

A cloud platform as a service (PaaS) that supports many programming languages is called Heroku. Heroku, one of the first cloud computing platforms has been under development since June 2007, when it could only handle Ruby. Today, it can run Java, Node.js, Scala, Clojure, Python, PHP, and Go. Heroku is referred to as a polyglot platform as a result of its features that enable developers to create, deploy, and scale applications in a consistent way across the majority of languages.

Typically, applications that operate on Heroku have their own domain that is used to direct HTTP requests to the appropriate application container or dyno.

A "dyno grid" made up of many servers serves as the distribution point for each of the dynos. Pushes from authorized users' application repositories are handled by Heroku's Git server.

CONCLUSION

This strategy has been used by numerous company to win the battle, The Random forest performed generally well and better compared to the other classification algorithms.

With little tweaking of the hyperparameters, it gave a very good metric scores.

REFERENCES

1. Jean-Francois Darre, November 22, 2015, Analysis of Lending Club's data, https://www.datasciencecentral.com/profiles/blogs/analysis-of-lending-club-s-data
2. JFdarre, September 30, 2015, Project 1: Lending Club’s data, http://rpubs.com/jfdarre/119147
3. Credit Revolving Balance: https://www.creditcards.com/credit-card-news/glossary/term-revolving-balance.php
4. Lending Club: https://en.wikipedia.org/wiki/Lending_Club
5. F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, “Scikit-learn: Machine learning in Python,” Journal of Machine Learning Research, vol. 12, pp. 2825–2830, 2011.
6. S. Chang, S. D.-o. Kim, and G. Kondo, “Predicting default risk of lending club loans,” 2015.
7. K. Tsai, S. Ramiah, and S. Singh, “Peer lending risk predictor,” CS229 Autumn, 2014.
8. A. Gutierrez and D. Mathieson, “Optimizing investment strategy in peer to peer lending,” 2017.
9. B. Pujun, C. Nick, and L. Max, “Demystifying the workings of the lending club,”
10. “How we measure net annualized return — lending club.”