**Overview**

For this project we wanted to build a machine learning model to predict loan approval. This machine learning model considers 12 independent variables to determine the target variable which is the loan status. The table below is the data dictionary for the dataset we used.

## **Personnel**

1. Alexandra L. Dakhniuk
2. Nishant Patel
3. Siddhant Pillai
4. Vinu Ratnayake

**Why Python?**

There are several reasons why we used python for this project. Python is simple and consistent. Python is very intuitive and its simplicity makes it easy for us to build a machine learning model. This allowed us to focus solely on the machine learning problem instead of the technical nuances of the language.

Python also offers an extensive collection of libraries and frameworks. To reduce development time, programmers turn to a number of Python frameworks and libraries. Libraries are pre-written code that developers use to execute common programming tasks. Python, with its rich technology stack, has an extensive set of libraries for machine learning. Another reason we used Python is because of its platform independence. Python code is supported by several operating systems including windows, mac os and linux. Furthermore Python has a great community and popularity. Therefore if we ran into any problems during development we can look for solutions in websites like stack overflow.

**Data Dictionary**

| **Variable** | **Description** |
| --- | --- |
| Loan\_ID | Unique Loan ID for each loan application |
| Gender | Male/ Female |
| Married | Whether the Applicant is married or not (Y/N) |
| Dependents | Number of dependents in applicant family |
| Education | Level of Applicant Education (Graduate/ Undergraduate) |
| Self Employed | Is the applicant self-employed (Y/N) |
| ApplicantIncome | Applicant income |
| CoapplicantIncome | Co-applicant income |
| LoanAmount | Loan amount in thousands of Dollars |
| Loan Amount Term | Term of loan in months |
| Credit\_History | Credit history meets guidelines |
| Property\_Area | Urban/ Semi Urban/ Rural |
| Loan Status | (Target) Loan approved (Y/N) |

**Libraries and Packages**

For our project we used libraries and packages such as Pandas, Numpy, Seaborn and Scikit Learn.

**Pandas**

Pandas is a Python library used for working with data sets. It has functions for data cleansing, data fill, data normalization, merges and joins, data visualization, statistical analysis, data inspection and much more. Pandas is built on top of another python library called Numpy.

**Numpy**

Numpy is another python package that provides high-performance multidimensional array objects, and tools for working with these arrays.

**Seaborn**

Seaborn is a visualization library for plotting statistical graphics in Python. It provides various styles and color palettes to make statistical plots more attractive. It is built on the top of matplotlib library and also closely integrated to the data structures from pandas

**Matplotlib**

Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python. Matplotlib makes easy things easy and hard things possible. Create publication quality plots. Make interactive figures that can zoom, pan, update. Customize visual style and layout.

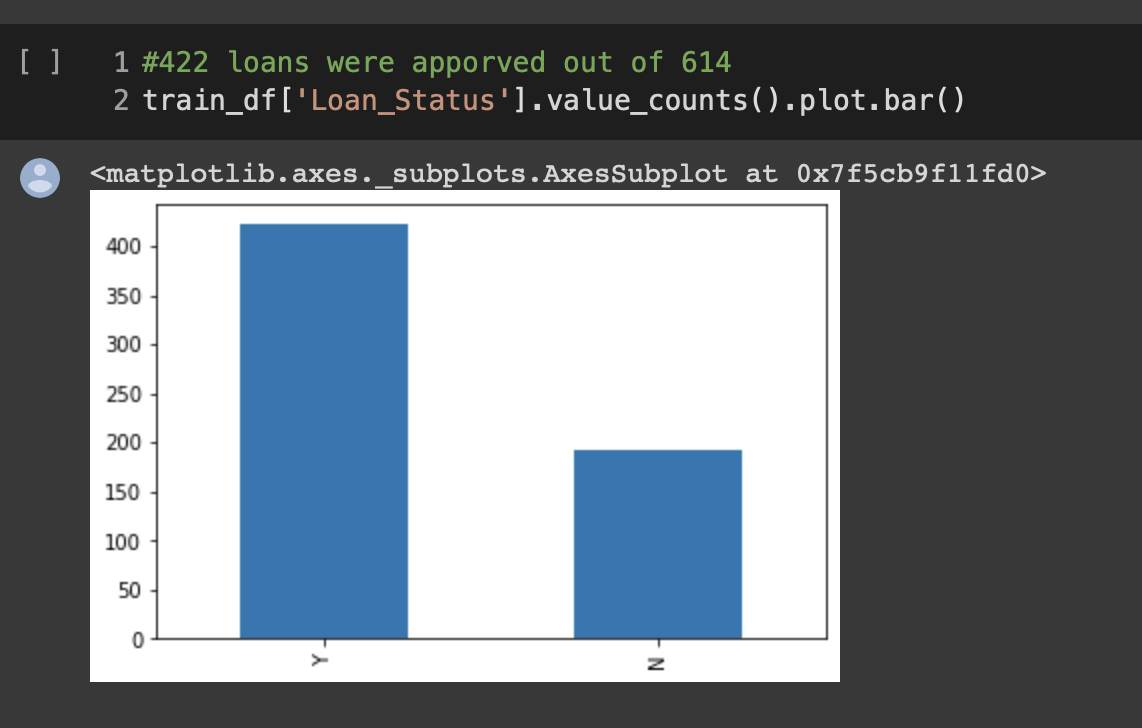
**Scikit Learn**

Scikit Learn is a library in Python that provides many unsupervised and supervised learning algorithms. Scikit provides functionalities like regression, classification, clustering, model selection and preprocessing.

**Dataset & Developing a High Performance Model**

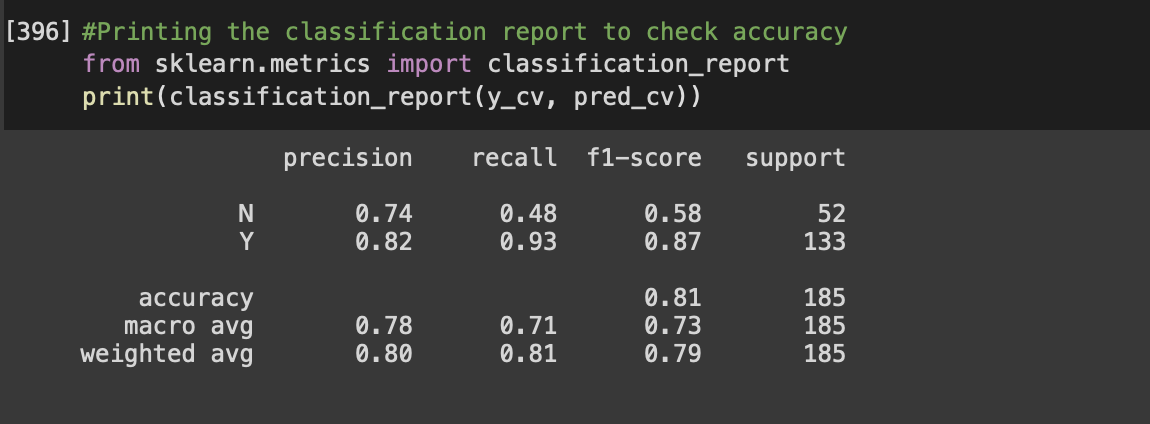
We used the dataset from Kaggle.com that included two subsets: testing and training. We mainly focused on the testing subset to create a model that would make the loan prediction efficient and accurate.

The ratio of approved vs. denied loans is 2:1. Therefore, the majority of loan applications were approved (almost 69%). The other types of visualizations are included in the report.



**LOGISTIC REGRESSION:**

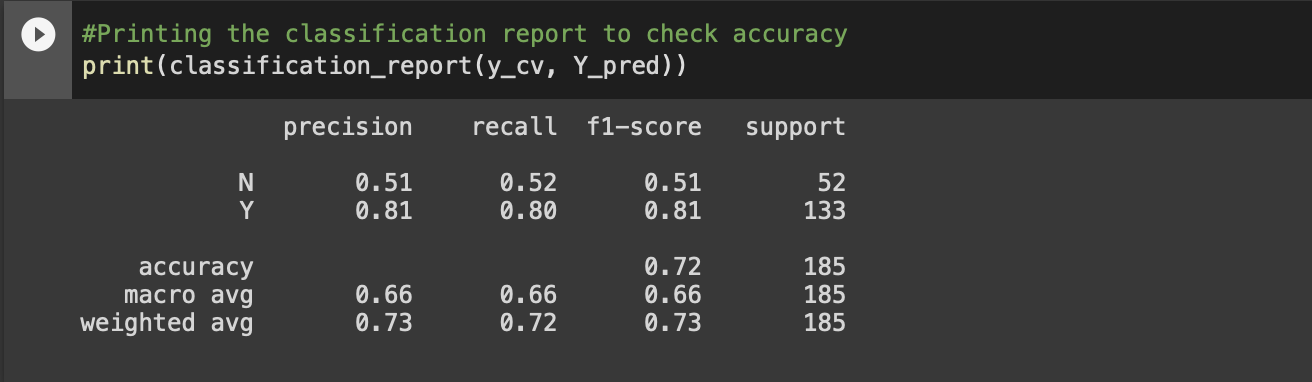
We have used Logistic Regression function (sigmoid function): 𝑓(𝐱): 𝑝(𝐱) = 1 / (1 + exp(−𝑓(𝐱)). As such, it's often close to either 0 or 1. The function 𝑝(𝐱) is interpreted as the predicted probability that the result for a given 𝐱 is equal to 1.



**DECISION TREE:**

Decision Tree is a versatile algorithm used to solve classification and regression problems. They consist of several branches, leaf nodes, and root nodes. The algorithm generates a structure like a tree by classifying the instances and utilizing a Recursive Partitioning Algorithm. A class label is represented by a leaf node and the branches represent test results. These tests are represented by internal nodes for an attribute.

The classification report of



REFERENCES

[1] <https://matplotlib.org/>

[2]<https://www.geeksforgeeks.org/introduction-to-seaborn-python/?ref=gcse>

[3]<https://www.geeksforgeeks.org/introduction-to-pandas-in-python/?ref=gcse>

[4]<https://www.geeksforgeeks.org/python-numpy/?ref=gcse>

[5]<https://www.geeksforgeeks.org/learning-model-building-scikit-learn-python-machine-learning-library/?ref=gcse>