Give me Some Credit

Building a Classification Model to Detect the Likelihood of Borrower Default

Author: Dr. Jody-Ann S. Jones

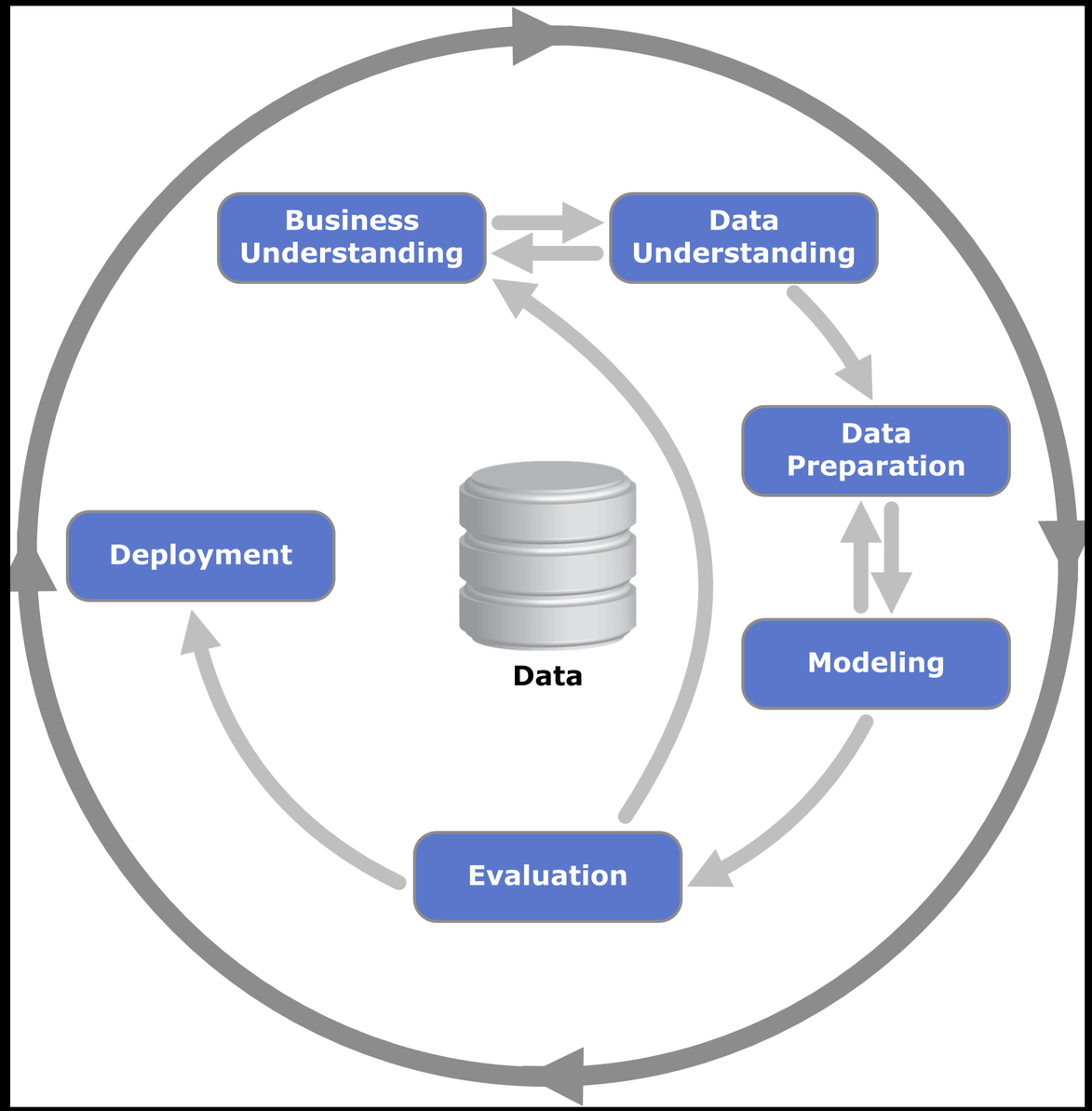
About this Project

- In this project, I will walk you through an exploratory data analysis of a dataset comprising of 11 features and 150,000 records.
- The dataset is taken from Kaggle and was originally created for the intended purpose of building credit scoring models. It consists of features that can be used to probabilistically determine the likelihood of a borrower defaulting on a loan or not.
- Accurately classifying a borrower is important to financial institutions as it mitigates the risk of credit default. Furthermore, the integrated adoption of such a model can help decision makers to recommend appropriate financial instruments to consumers.

Structure of the Project

This project utilises the CRISP-DM framework

- Check Data Integrity
- Exploratory Data Analysis
- Feature Engineering
- Model Training Preparation
- Model Training
- Model Evaluation



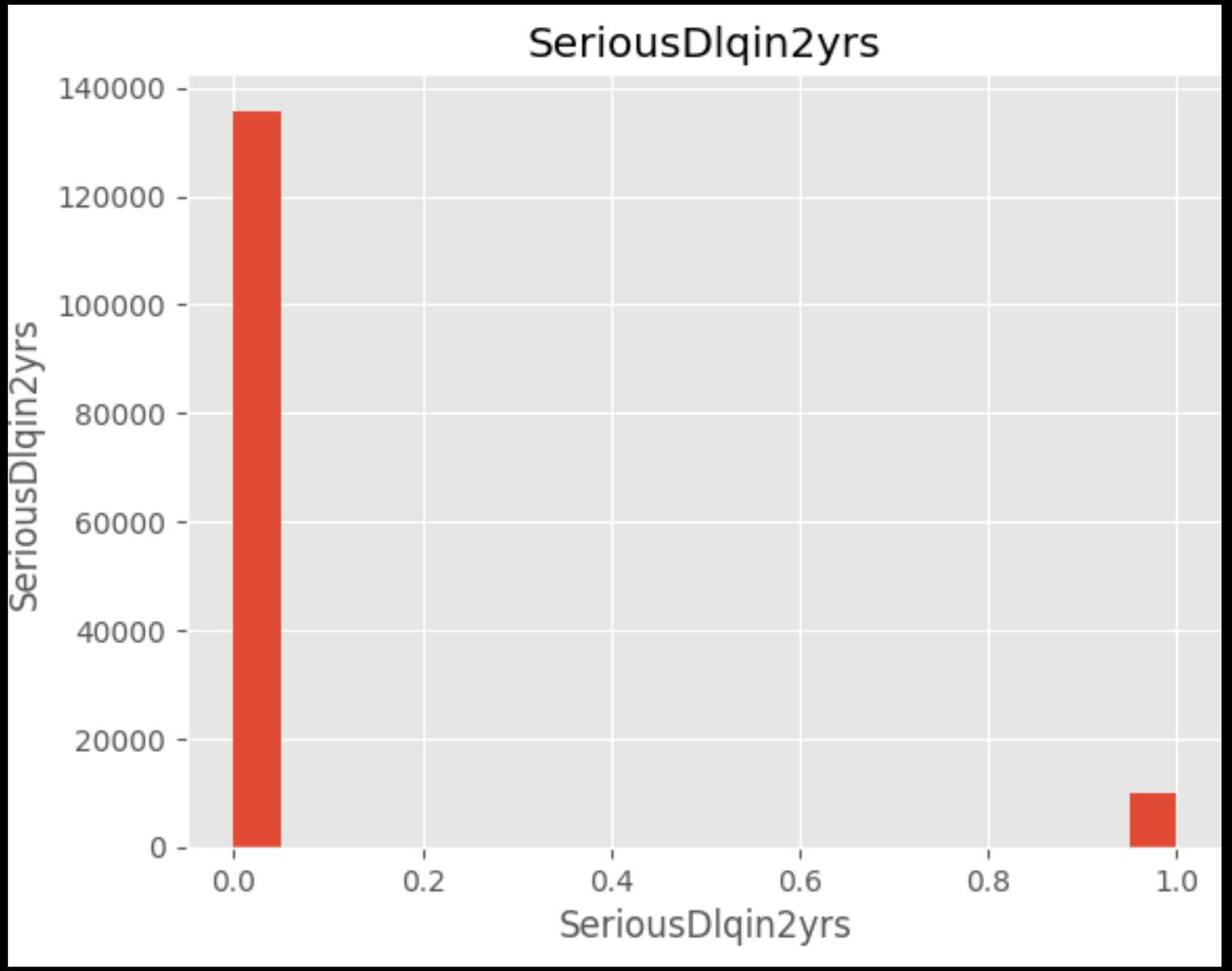
Exploratory Data Analysis

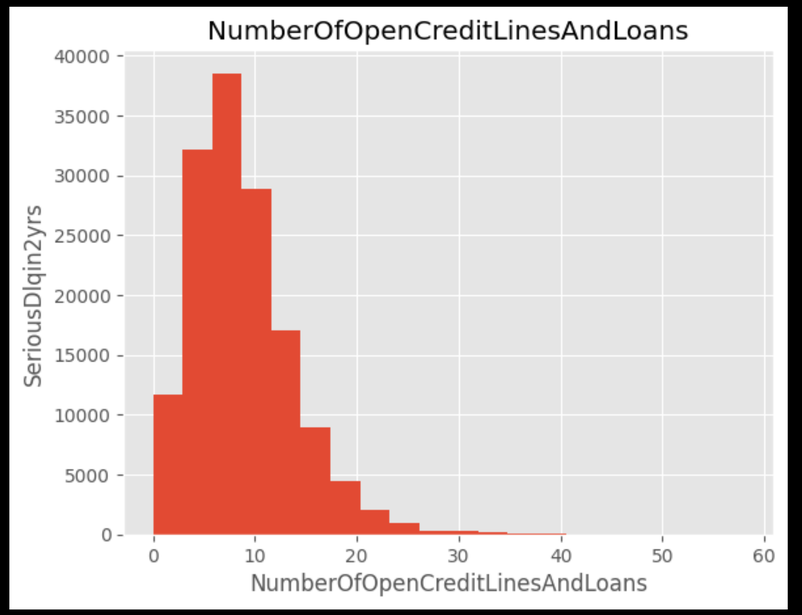
Descriptive Statistics

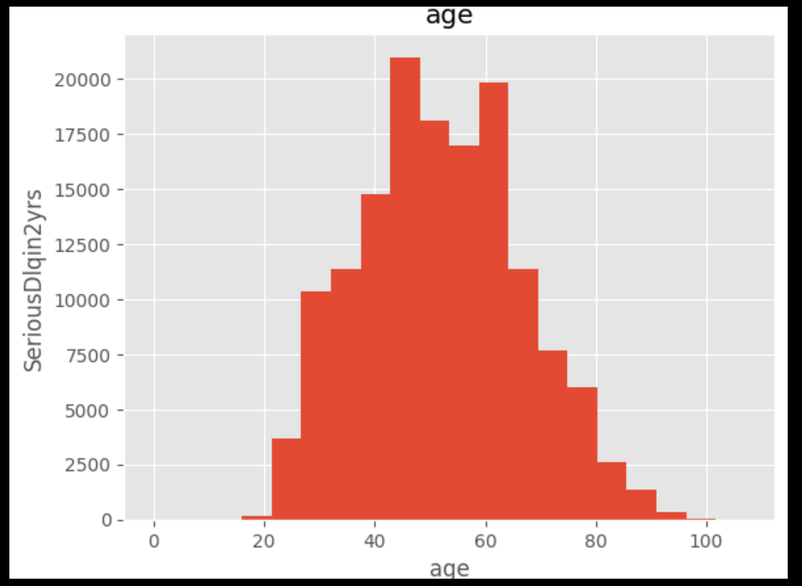
	count	mean	std	min	25%	50%	75%	max
SeriousDlqin2yrs	145,563.0	0.1	0.2	0.0	0.0	0.0	0.0	1.0
RevolvingUtilizationOfUnsecuredLines	145,563.0	5.9	250.5	0.0	0.0	0.2	0.6	50,708.0
age	145,563.0	52.1	14.6	0.0	41.0	52.0	62.0	107.0
NumberOfTime30-59DaysPastDueNotWorse	145,563.0	0.4	3.8	0.0	0.0	0.0	0.0	98.0
DebtRatio	145,563.0	334.6	1,947.2	0.0	0.2	0.4	0.8	329,664.0
MonthlyIncome	145,563.0	6,452.7	13,083.3	0.0	3,816.0	5,400.0	7,500.0	3,008,750.0
NumberOfOpenCreditLinesAndLoans	145,563.0	8.6	5.1	0.0	5.0	8.0	11.0	58.0
NumberOfTimes90DaysLate	145,563.0	0.2	3.7	0.0	0.0	0.0	0.0	98.0
NumberRealEstateLoansOrLines	145,563.0	1.0	1.1	0.0	0.0	1.0	2.0	54.0
NumberOfTime60-89DaysPastDueNotWorse	145,563.0	0.2	3.7	0.0	0.0	0.0	0.0	98.0
NumberOfDependents	145,563.0	0.8	1.1	0.0	0.0	0.0	1.0	20.0

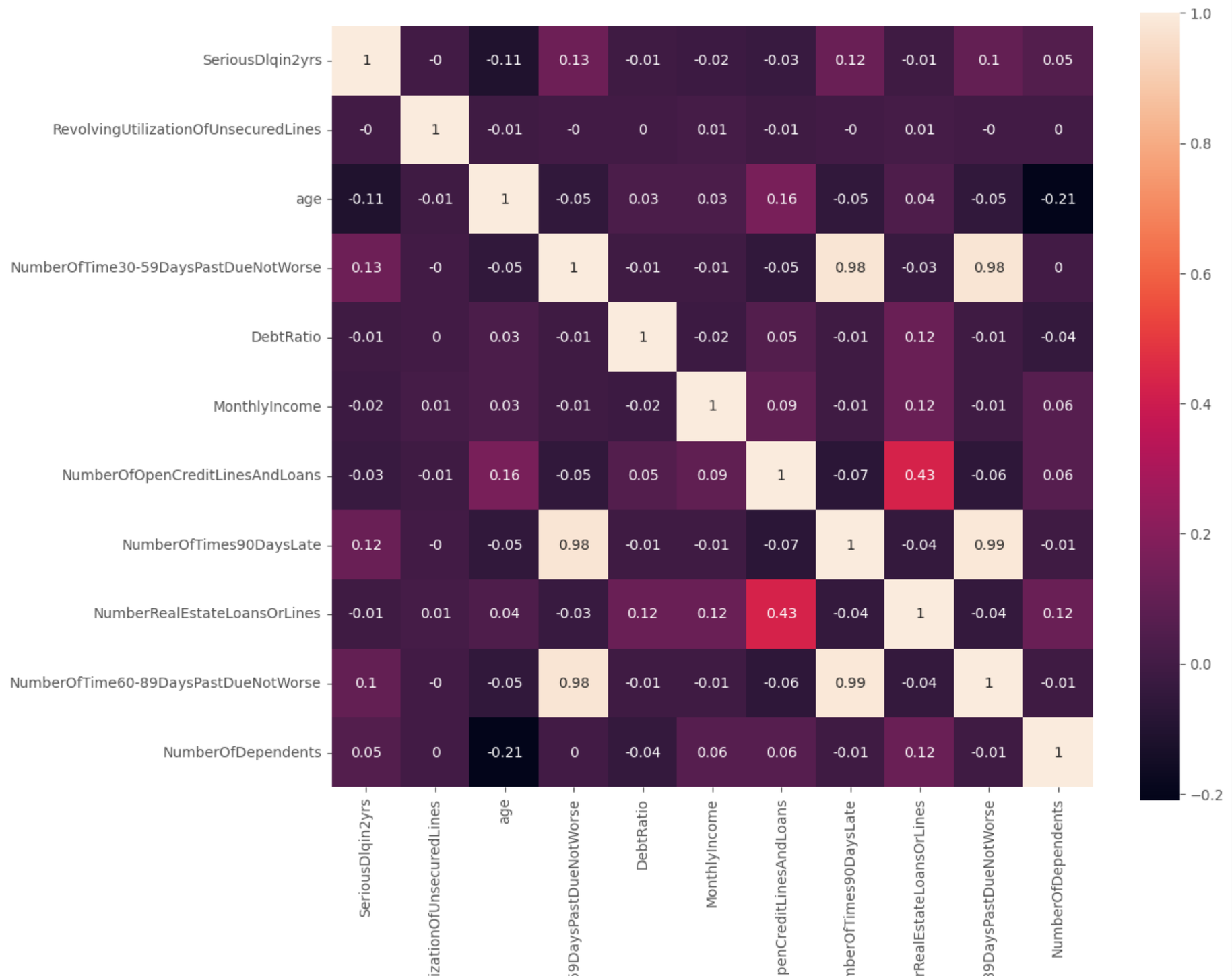
Key Observations

- The dependent variable (target feature) is binary, i.e. taking a value of either 0 - no default, or 1 - default.
- After initial preprocessing (removing missing values and duplicates), the dataset has 145563 records and 11 features (a mix of integers and float data types, no categorical variables).
- The average monthly income is $6,452.70; the median is $5,400; minimum is $0; maximum is $3,008,750.
- The average number of open credit lines and loans is 8.6; median is 8. minimum is 0; maximum is 58.
- The average total balance on credit cards and personal lines as a percentage of the sum of credit limits is 5.9 percent; median is 0.2 percent; minimum is 0 percent; maximum is 50,708%.
- The average customer age is 52 years old.


SeriousDlqin2yrs
140000
120000
100000
80000
60000
40000
20000
0
SeriousDlqin2yrs
0.0
0.2
0.4
0.6
0.8
1.0
SeriousDlqin2yrs


NumberOfOpenCreditLinesAndLoans
40000
35000
30000
25000
20000
15000
10000
5000
0
SeriousDlqin2yrs
0
10
20
30
40
50
60
NumberOfOpenCreditLinesAndLoans


age
20000
17500
15000
12500
10000
7500
5000
2500
0
SeriousDlqin2yrs
0
20
40
60
80
100
age


SeriousDlqin2yrs 1 -0 -0.11 0.13 -0.01 -0.02 -0.03 0.12 -0.01 0.1 0.05
RevolvingUtilizationOfUnsecuredLines -0 1 -0.01 -0 0 0.01 -0.01 -0 0.01 -0 0
age -0.11 -0.01 1 -0.05 0.03 0.03 0.16 -0.05 0.04 -0.05 -0.21
NumberOfTime30-59DaysPastDueNotWorse 0.13 -0 -0.05 1 -0.01 -0.01 -0.05 0.98 -0.03 0.98 0
DebtRatio -0.01 0 0.03 -0.01 1 -0.02 0.05 -0.01 0.12 -0.01 -0.04
MonthlyIncome -0.02 0.01 0.03 -0.01 -0.02 1 0.09 -0.01 0.12 -0.01 0.06
NumberOfOpenCreditLinesAndLoans -0.03 -0.01 0.16 -0.05 0.05 0.09 1 -0.07 0.43 -0.06 0.06
NumberOfTimes90DaysLate 0.12 -0 -0.05 0.98 -0.01 -0.01 -0.07 1 -0.04 0.99 -0.01
NumberRealEstateLoansOrLines -0.01 0.01 0.04 -0.03 0.12 0.12 0.43 -0.04 1 -0.04 0.12
NumberOfTime60-89DaysPastDueNotWorse 0.1 -0 -0.05 0.98 -0.01 -0.01 -0.06 0.99 -0.04 1 -0.01
NumberOfDependents 0.05 0 -0.21 0 -0.04 0.06 0.06 -0.01 0.12 -0.01 1
SeriousDlqin2yrs
age
DebtRatio
MonthlyIncome
NumberOfTimes90DaysLate
NumberRealEstateLoansOrLines
NumberOfDependents
1.0
0.8
0.6
0.4
0.2
0.0
−0.2

Feature Engineering

Normalization of Numerical Features

- In order to mitigate against the model placing greater emphasis on variables with wider ranges, I use scikit learn's StandardScaler function. The StandardScaler function normalises the range of values by removing the mean and scaling to unit variance. (Source: https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html)

```
:
# Create function that normalizes numerical features

def scale_columns(df, columns):
    scaler = StandardScaler()
    for col in columns:
        if col in df.columns:
            df[col] = scaler.fit_transform(df[[col]])
        else:
            print(f"Column {col} not found in DataFrame.")
    return df
```

This custom function applies the StandardScaler to all columns

	SeriousDlqin2yrs	RevolvingUtilizationOfUnsecuredLines	age	NumberOfTime30-59DaysPastDueNotWorse	DebtRatio	MonthlyIncor
0	1	-0.020659	-0.488117	2	-0.171396	0.2038
1	0	-0.019896	-0.831345	0	-0.171745	-0.2944
2	0	-0.021090	-0.968636	1	-0.171764	-0.2606
3	0	-0.022784	-1.517800	0	-0.171789	-0.2409
4	0	-0.020096	-0.213536	1	-0.171795	4.3670
...	...	...	...	...	...	
149995	0	-0.023555	1.502601	0	-0.171692	-0.3326
149996	0	-0.022521	-0.556763	0	-0.171440	-0.0663
149997	0	-0.022735	0.404274	0	1.815639	-0.0804
149998	0	-0.023717	-1.517800	0	-0.171808	-0.0563
149999	0	-0.020323	0.816147	0	-0.171680	0.1303

Snapshot of data after normalization of numerical features

Model Selection and Training

Model Selection

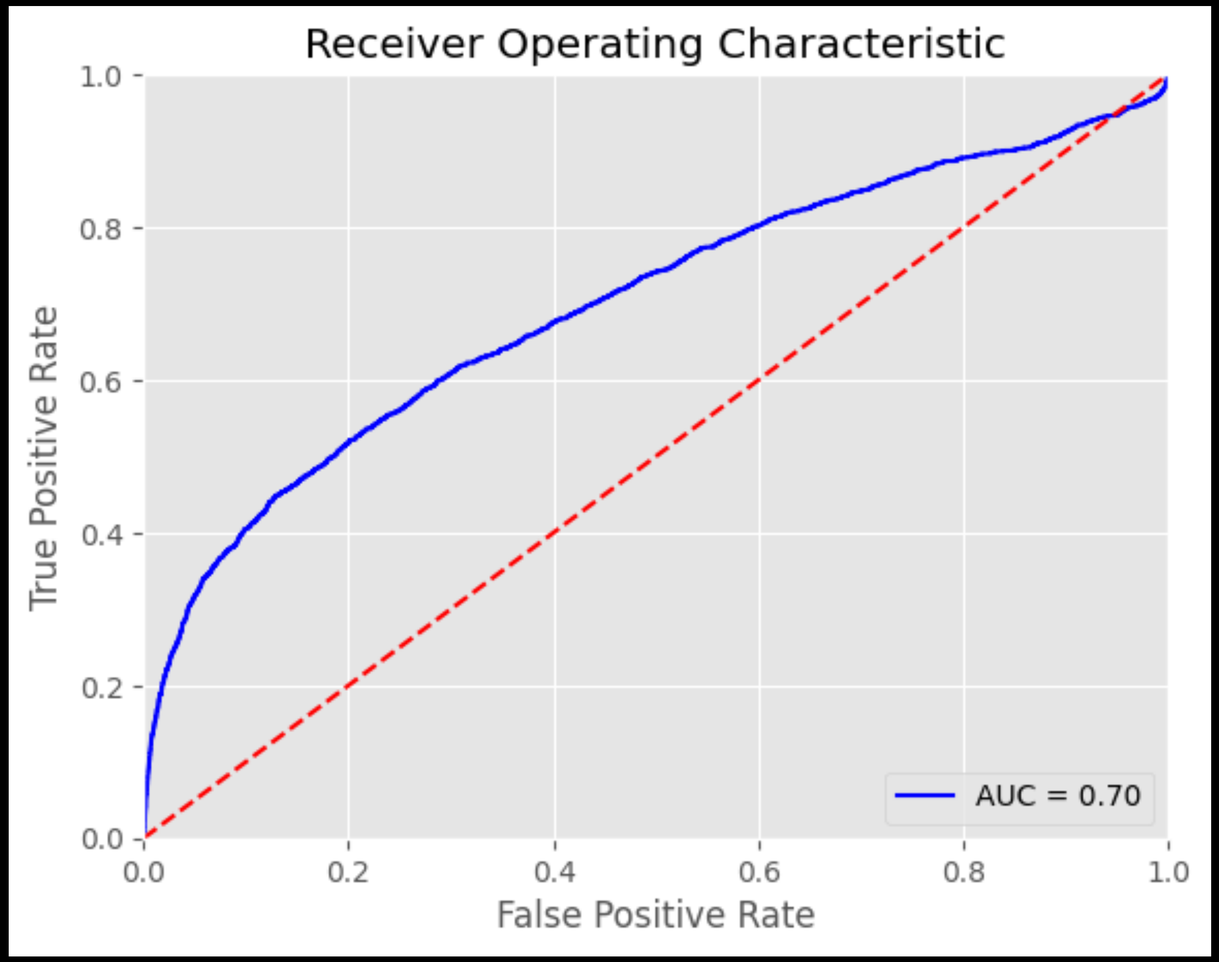
Supervised Problem: Classification Task

- In this project, we have labeled outputs. This fact makes our problem a supervised one. Furthermore, we're predicting a binary categorical output (1 or 0). Taking this into consideration, I used three classification models in order to determine the likelihood of borrower default:
  - Baseline model: Logistic Regression
  - Model #2: Random Forest Classification
  - Model #3: XGBoost Classification
- Logistic Regression is the workhorse of classification, its results are easy to interpret and typically do not have challenges with overfitting. As such, it is a good model with which we can benchmark the performance of the other models. Random Forest and XGBoost are ensemble and boosting methods respectively.

Model Evaluation

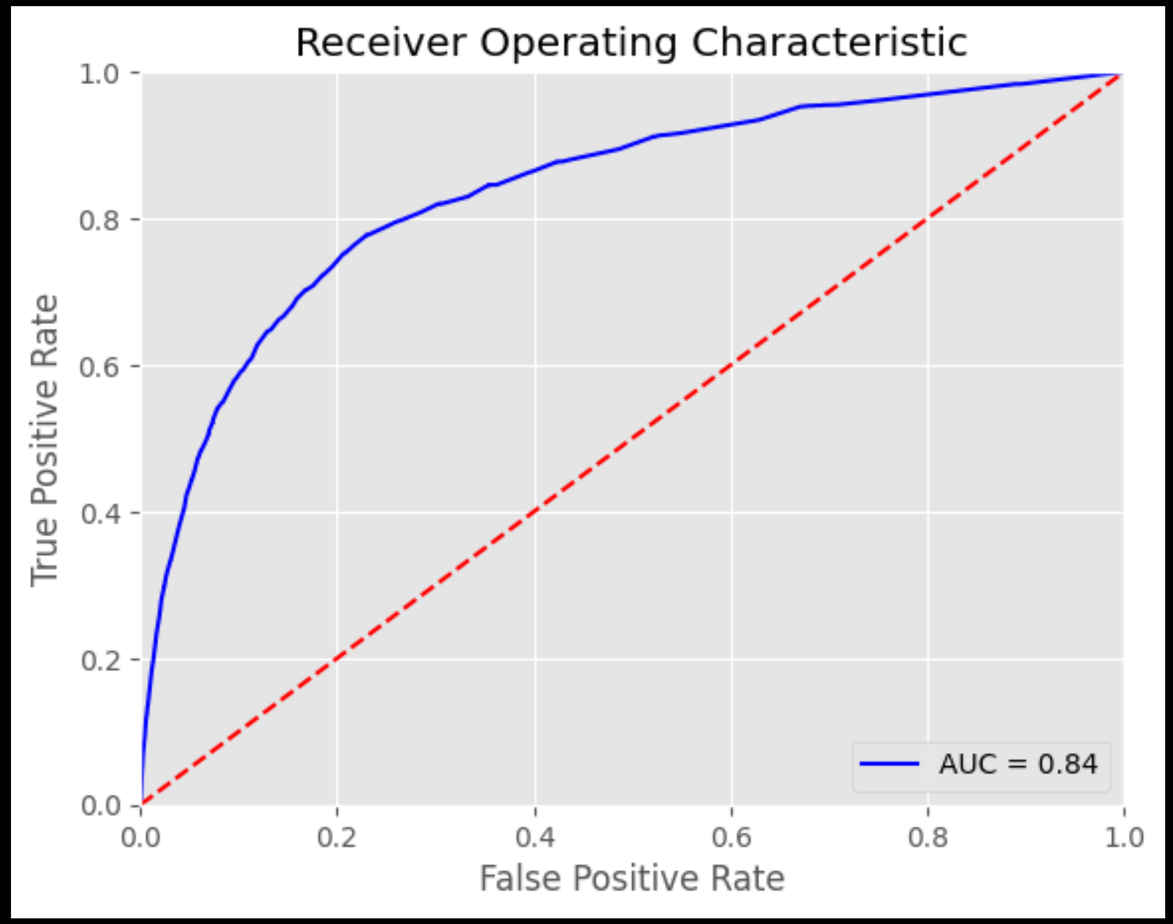
Logistic Regression

ROC curve - AUC score



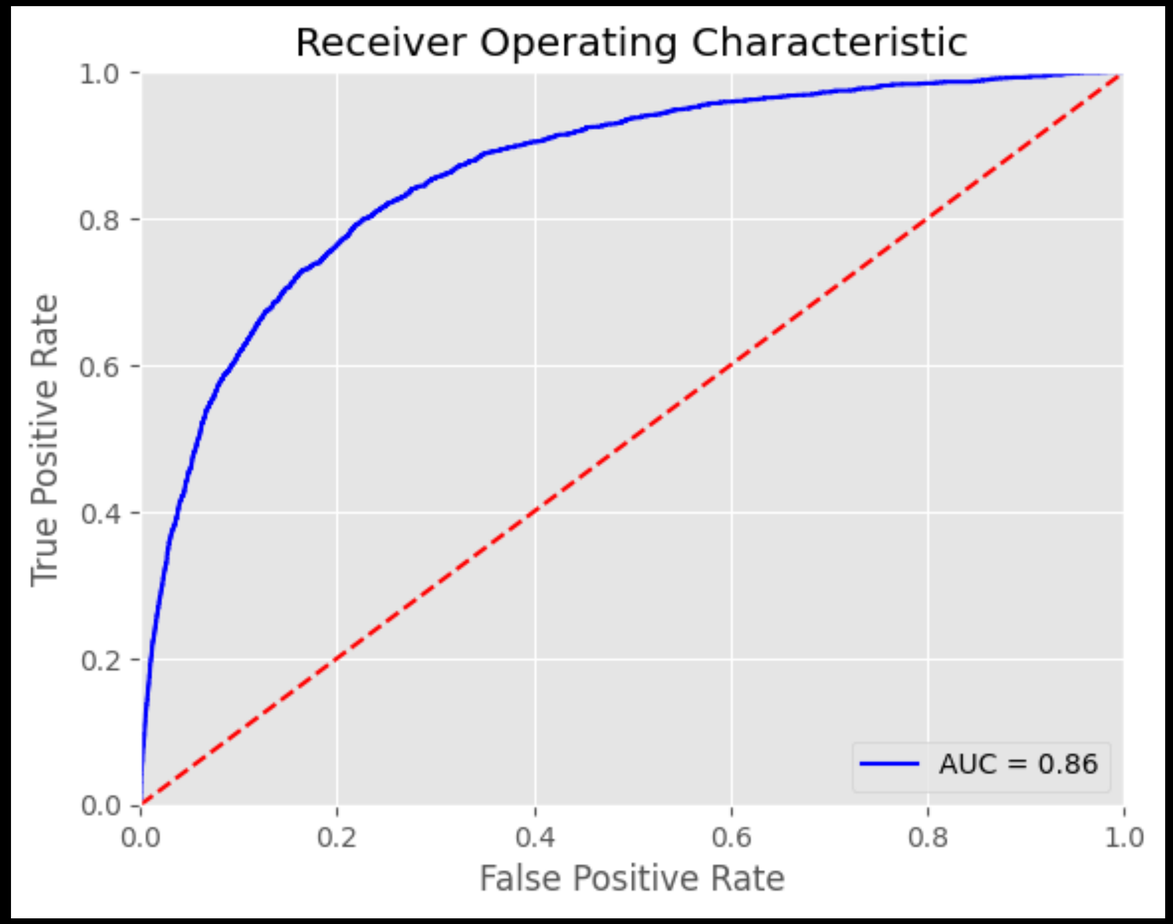
Random Forest Classification

ROC curve - AUC score



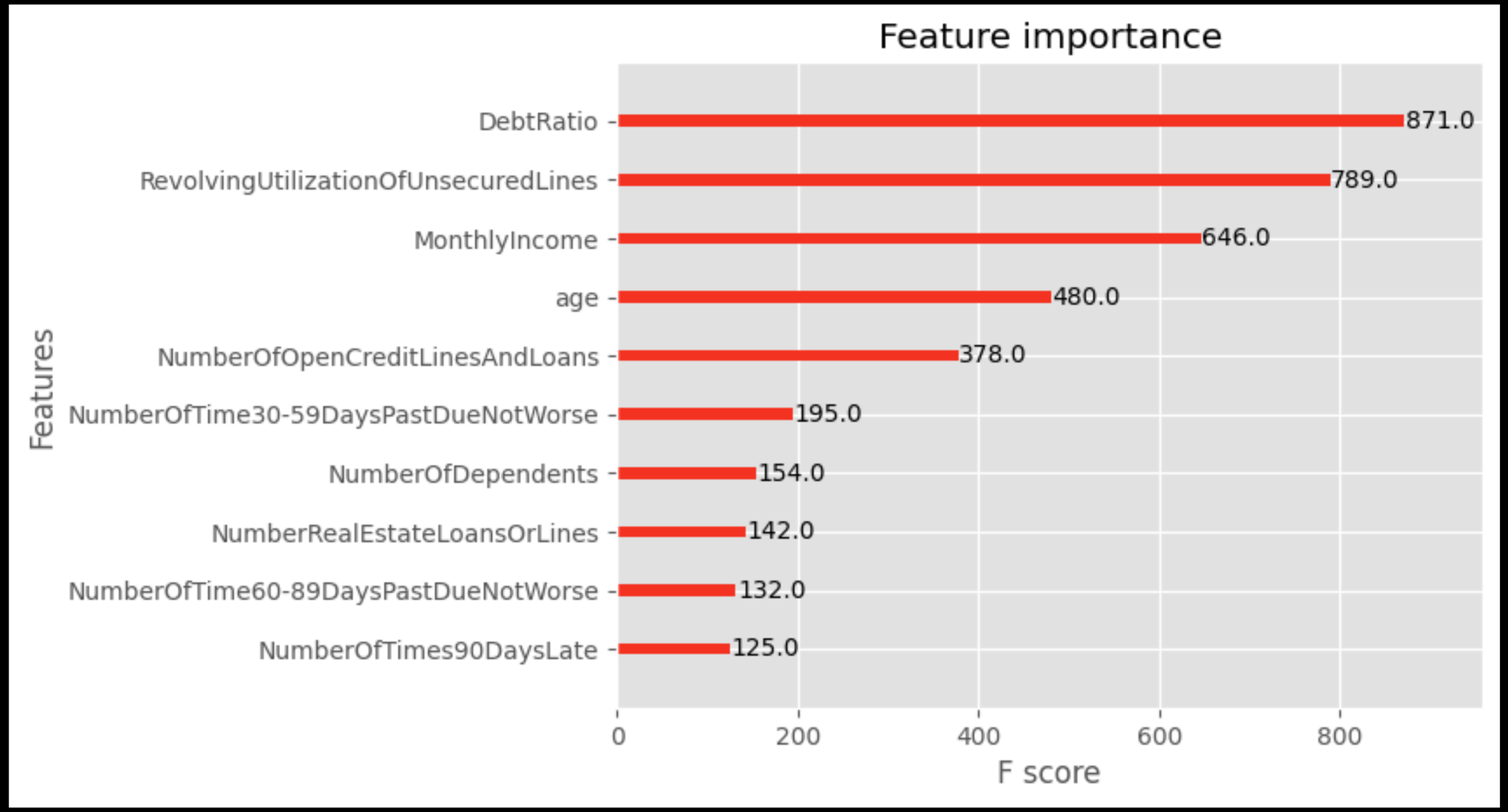
XGBClassifier

ROC curve - AUC score



Feature Importance Plot

DebtRatio, RevolvingUtilization, and Monthly Income explain the largest variation in the model.



Key Takeaways and Implications

- The XGBoost Classifier performed the best among all three models including logistic regression and random forest classification.
- According to XGBoost's feature importance plot: DebtRatio, RevolvingUtilization, and Monthly Income account for the majority of the variation in the model. Secondary factors worth mentioning include age, and the number of open credit lines.
- A key implication that we can draw from this is that DebtRatio is the greatest factor that could lead to the likelihood of borrower default. Therefore, a business decision that could be taken that would help to reduce the number of defaults is to set a limit on the number lines of credit a customer can open based on their existing DebtRatio.

Next Steps

We have options!

- Refactor code from a development environment to a production environment, export the model using pickle, joblib, or similar libraries that it can be further used for integration such as the follows:
  - Expose model to an API endpoint (e.g. Flask or FastAPI) so that the model can be incorporated with other applications.
  - Wrap the model in a frontend framework (e.g. Streamlit, Gradio), deploy to a cloud PAAS (platform as a service) environment such as Heroku or Render.
- In a real world project setting, another useful extension of this model would be to create a dashboard where primary stakeholders can have access to this information at a glance.

Thank you!

My work and thoughts on Machine Learning
can be found on Twitter, LinkedIn and GitHub
@drjodyannjones