Introduction

For this project, we will be working with a real credit portfolio dataset to gain insights into credit risk and improve our predictive modeling abilities. The first step in our analysis will be to conduct an exploratory data analysis (EDA) to better understand the structure and distribution of the data. Through our EDA, we will gain insights into the characteristics of borrowers who are more likely to default on their loans and identify potential data quality issues that need to be addressed.

Once we have a better understanding of the data, we will apply our credit data knowledge to treat our data properly. This will involve cleaning and preprocessing the data to ensure that it is suitable for modeling. We will also need to perform feature selection and engineering to identify the most important predictors of credit risk and improve the predictive power of our models.

Next, we will compare different models to select the "best" one based on our evaluation metrics and interpretability. We will consider a variety of models, including logistic regression, decision trees, and neural networks, and evaluate their performance using metrics such as accuracy, precision, recall, and F1 score. We will also consider the interpretability of each model and its ability to provide actionable insights into credit risk.

Finally, we will present our results and conclusions as a formal document. This document will include a detailed description of our methodology, a summary of our findings, and recommendations for improving credit risk management based on our analysis. By working with a real credit portfolio dataset and applying our credit data knowledge to the analysis, we will gain valuable insights into credit risk and improve our ability to make data-driven decisions.

Dataset

At first, we started with a dataset that included many columns, around 70, that had information that is not relevant for our analysis. The majority of the columns had no information at all so we decided to eliminate them, other columns had a date in them or url links that are also unnecessary for our analysis so we decided to do a clean up and ended up with around 16 columns that we believe are relevant.

We decide to keep the columns that we believe are more reliable for our analysis, such as 'loan_amnt', 'funded_amnt', 'funded_amnt_inv', 'term', int_rate', 'installment', 'annual_inc', 'total_acc', 'total_pymnt', 'total_pymnt_inv', 'total_rec_prncp', 'total_rec_int', 'last_pymnt_amnt', 'status', 'home_ownership' and 'verification_status'. These columns contain valuable information that we can use to analyze credit data information.

However, some cells in these columns may be null or missing, which can affect the accuracy of our sentiment analysis. To clean up the dataset, we can use the dropna function in Python to remove rows with missing data. This will remove any rows where the review text or review rating are missing but will not affect the outcome of our analysis because we have decided to focus only on the columns that are relevant to our project. By focusing on the most important columns and cleaning up the dataset, we can improve the accuracy of our analysis and gain valuable insights into customer sentiment towards the product.

Bins

The use of bins is a powerful technique in data analysis that helps to simplify complex data and make it more manageable to work with. By grouping similar values into discrete intervals or categories, bins can help to reduce noise and variability in the data, which makes it easier to identify patterns and trends. Additionally, bins can help to deal with the problem of data overfitting by reducing the complexity of the model and improving its ability to generalize to new data. This makes bins an essential tool for data analysts who want to extract meaningful insights from large and complex datasets.

Furthermore, bins can help to improve the interpretability of the data by providing a clear structure and hierarchy to the information. Continuous variables can be difficult to interpret, especially if they have a large range of values. However, by binning the data into categories, it becomes easier to understand and interpret the data. Bins also facilitate comparisons between groups or categories, which can be important for identifying differences and similarities in the data. In conclusion, the use of bins is an important technique in data analysis that can help to simplify complex data, improve model performance, and enhance the interpretability of the data.

EDA

Exploratory data analysis (EDA) is an essential step in the data analysis process, particularly when working with a credit dataset. In this type of dataset, there are usually a large number of variables or features that need to be explored to identify trends and patterns that could inform the modeling process.

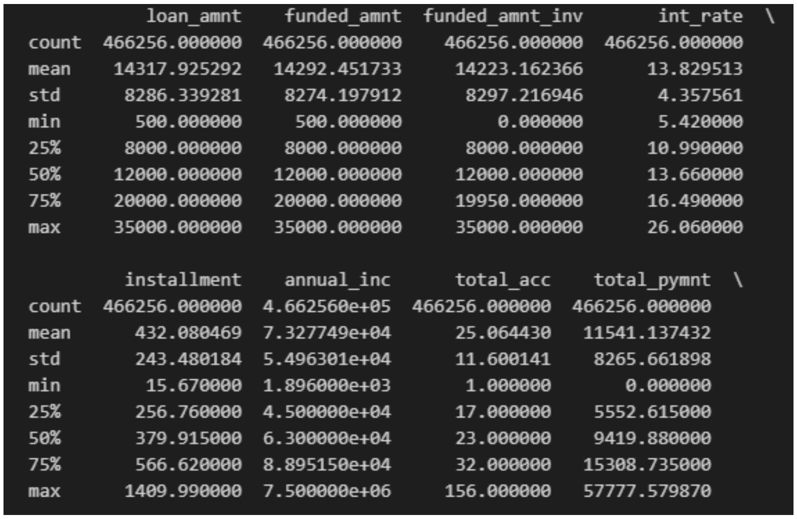
The first step in conducting an EDA on a credit dataset is to get a general understanding of the data. This can be achieved by looking at the summary statistics of each variable, such as mean, median, standard deviation, and the range of values. We can also visualize the distribution of each variable using histograms or density plots to identify any skewness or outliers in the data.

Next, we can explore the relationship between the dependent variable (default status) and the independent variables (predictors) to identify any significant patterns or correlations. We can use scatter plots, heat maps, or correlation matrices to visualize these relationships and identify any variables that are strongly correlated with the default status.

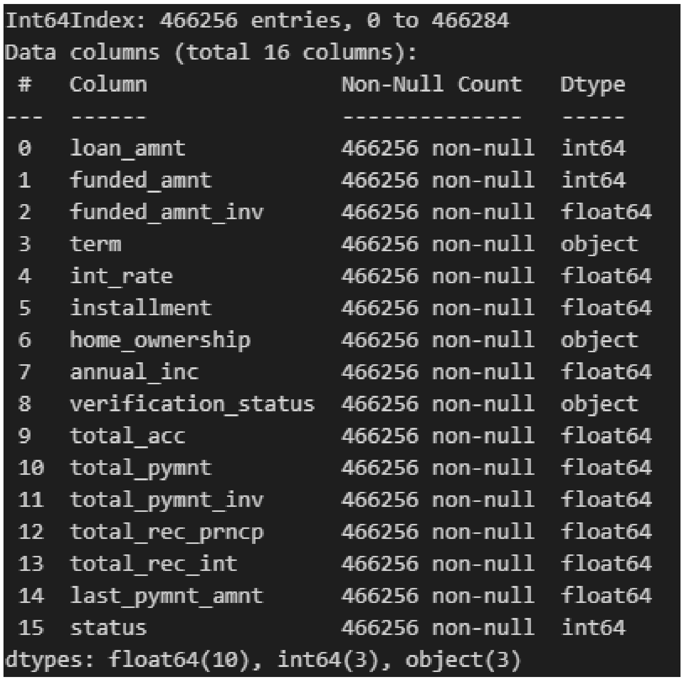
Another important aspect of EDA for credit datasets is identifying any missing data or data quality issues that could affect the modeling process. We can use methods such as heat maps or missing value plots to identify any missing data and determine whether it is necessary to impute these values or exclude them from the analysis. In this case we knew we had missing information and we proceed to do a dropna.

Overall, EDA is a critical step in the analysis of credit datasets, as it provides insights into the structure and distribution of the data, identifies potential issues that need to be addressed, and helps to inform the selection of appropriate modeling techniques to accurately predict credit risk.

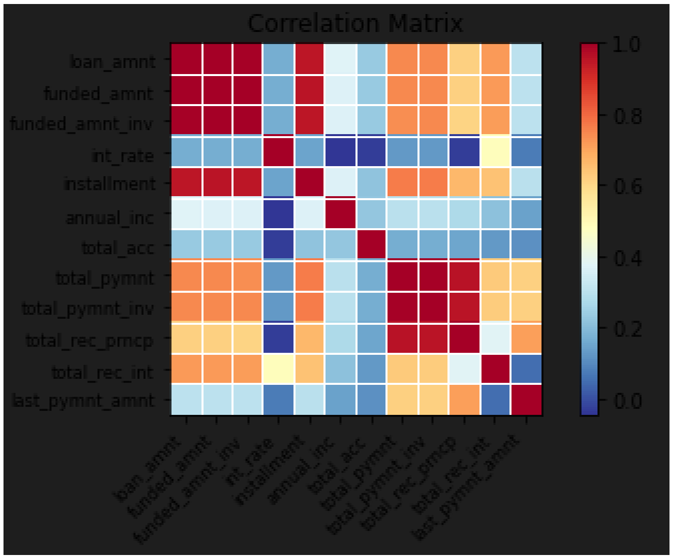
Some of the results in our EDA are the following:

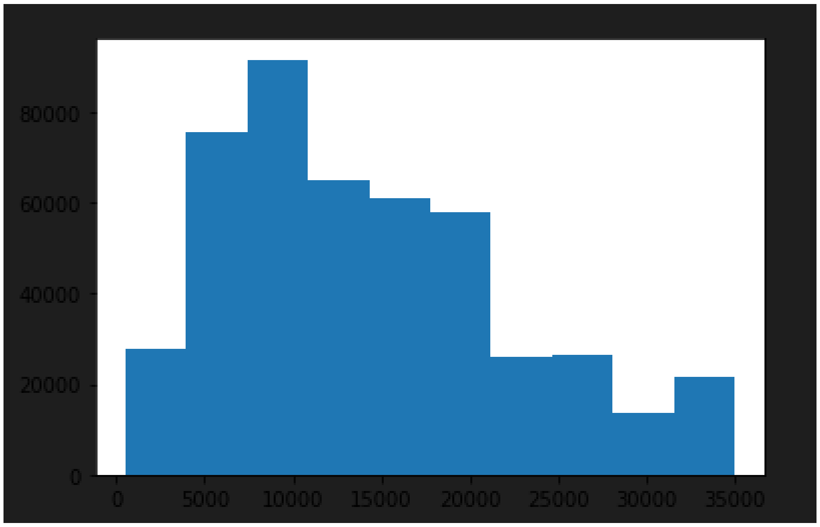


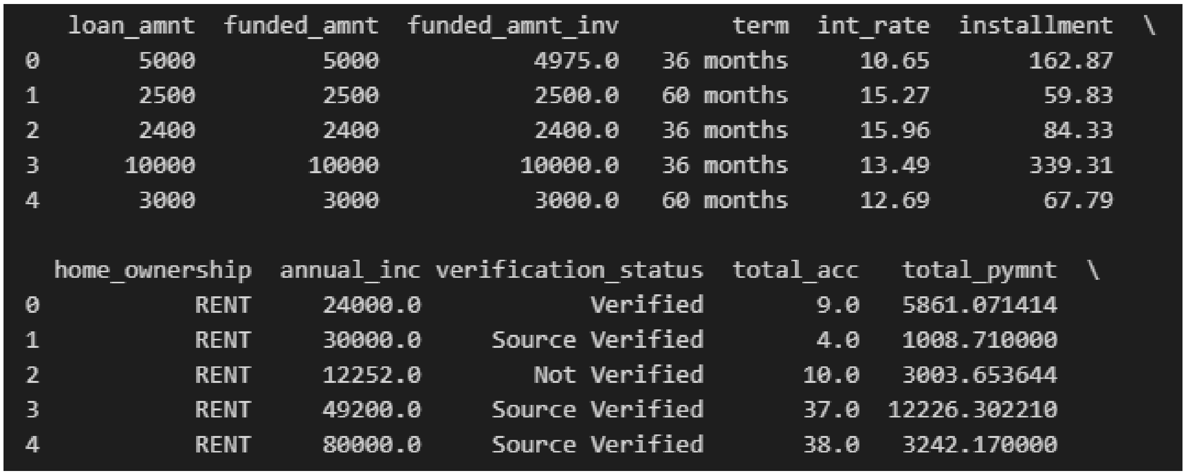
```
           loan_amnt    funded_amnt  funded_amnt_inv       int_rate  \
count  466256.000000  466256.000000    466256.000000  466256.000000
mean    14317.925292   14292.451733     14223.162366      13.829513
std      8286.339281    8274.197912      8297.216946       4.357561
min       500.000000     500.000000         0.000000       5.420000
25%      8000.000000    8000.000000      8000.000000      10.990000
50%     12000.000000   12000.000000     12000.000000      13.660000
75%     20000.000000   20000.000000     19950.000000      16.490000
max     35000.000000   35000.000000     35000.000000      26.060000

         installment    annual_inc      total_acc    total_pymnt  \
count  466256.000000  4.662560e+05  466256.000000  466256.000000
mean      432.080469  7.327749e+04      25.064430   11541.137432
std       243.480184  5.496301e+04      11.600141    8265.661898
min        15.670000  1.896000e+03       1.000000       0.000000
25%       256.760000  4.500000e+04      17.000000    5552.615000
50%       379.915000  6.300000e+04      23.000000    9419.880000
75%       566.620000  8.895150e+04      32.000000   15308.735000
max      1409.990000  7.500000e+06     156.000000   57777.579870
```



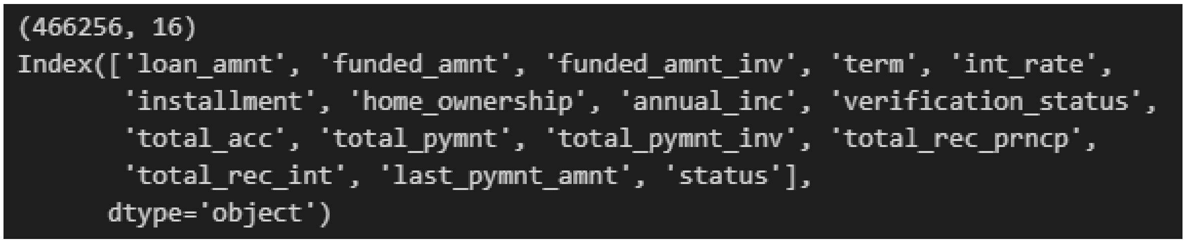
```
Int64Index: 466256 entries, 0 to 466284
Data columns (total 16 columns):
 #   Column               Non-Null Count   Dtype
---  ------               --------------   -----
 0   loan_amnt            466256 non-null  int64
 1   funded_amnt          466256 non-null  int64
 2   funded_amnt_inv      466256 non-null  float64
 3   term                 466256 non-null  object
 4   int_rate             466256 non-null  float64
 5   installment          466256 non-null  float64
 6   home_ownership       466256 non-null  object
 7   annual_inc           466256 non-null  float64
 8   verification_status  466256 non-null  object
 9   total_acc            466256 non-null  float64
 10  total_pymnt          466256 non-null  float64
 11  total_pymnt_inv      466256 non-null  float64
 12  total_rec_prncp      466256 non-null  float64
 13  total_rec_int        466256 non-null  float64
 14  last_pymnt_amnt      466256 non-null  float64
 15  status               466256 non-null  int64
dtypes: float64(10), int64(3), object(3)
```



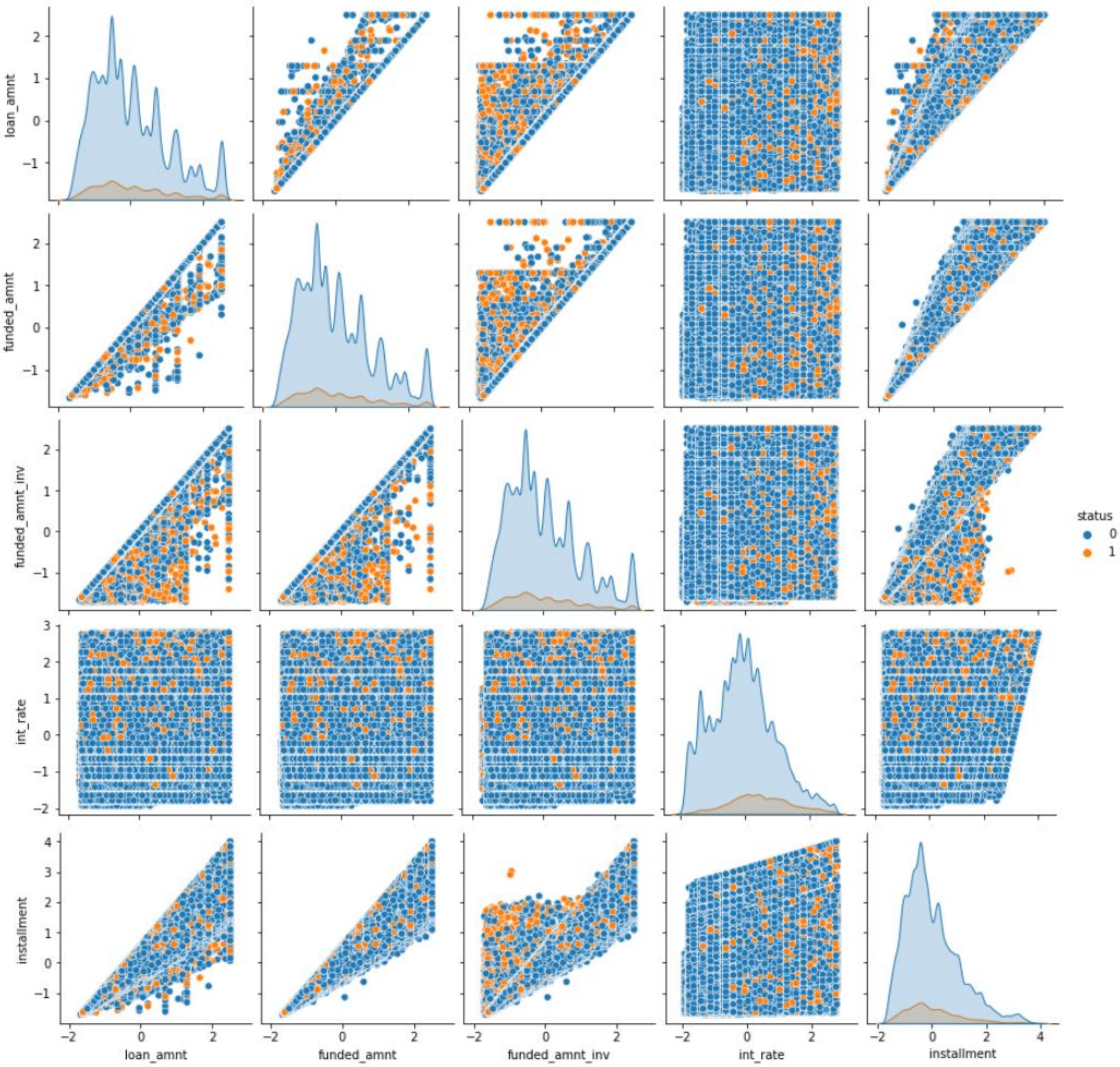




```
   loan_amnt  funded_amnt  funded_amnt_inv       term  int_rate  installment  \
0       5000         5000           4975.0  36 months     10.65       162.87
1       2500         2500           2500.0  60 months     15.27        59.83
2       2400         2400           2400.0  36 months     15.96        84.33
3      10000        10000          10000.0  36 months     13.49       339.31
4       3000         3000           3000.0  60 months     12.69        67.79

  home_ownership  annual_inc verification_status  total_acc   total_pymnt  \
0           RENT     24000.0            Verified        9.0   5861.071414
1           RENT     30000.0     Source Verified        4.0   1008.710000
2           RENT     12252.0        Not Verified       10.0   3003.653644
3           RENT     49200.0     Source Verified       37.0  12226.302210
4           RENT     80000.0     Source Verified       38.0   3242.170000
```



```
(466256, 16)
Index(['loan_amnt', 'funded_amnt', 'funded_amnt_inv', 'term', 'int_rate',
       'installment', 'home_ownership', 'annual_inc', 'verification_status',
       'total_acc', 'total_pymnt', 'total_pymnt_inv', 'total_rec_prncp',
       'total_rec_int', 'last_pymnt_amnt', 'status'],
      dtype='object')
```



Models

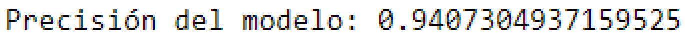
Logistic Regression, Bayesian Ridge, and XGBRegressor are three important algorithms in the field of data analysis. Each algorithm has its own unique strengths, and the choice of which one to use depends on the specific needs and characteristics of the data being analyzed.

Logistic regression is a classification algorithm that is used to predict binary outcomes. It estimates the probability of an event occurring based on one or more predictor variables. Logistic regression is widely used in various applications such as marketing, finance, and healthcare. One of the main advantages of logistic regression is its interpretability. The model outputs coefficients that can be easily interpreted to understand the relationship between the predictor variables and the probability of the event occurring. Additionally, logistic regression can handle both numerical and categorical variables and is relatively fast to train and execute.

Bayesian Ridge is a regression algorithm that is based on Bayesian inference. It is a powerful algorithm that can handle high-dimensional data and is robust to outliers. The algorithm works by estimating a probability distribution over the parameters of the model, which allows for uncertainty to be accounted for in the predictions. One of the main advantages of Bayesian Ridge is its ability to handle multicollinearity, which is when two or more predictor variables are highly correlated. This is a common problem in regression analysis, and Bayesian Ridge can handle it by estimating a probability distribution over the parameters instead of just point estimates. This leads to more accurate and robust predictions.

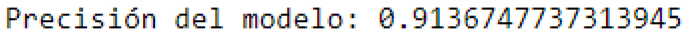
XGBRegressor is a regression algorithm based on gradient boosting, which is an ensemble method that combines multiple weak models to create a strong model. XGBRegressor is a powerful algorithm that can handle large datasets and is highly accurate. One of the main advantages of XGBRegressor is its ability to handle missing data and outliers. It is also able to handle both numerical and categorical variables and can be easily parallelized to improve efficiency. XGBRegressor is also highly customizable, allowing for fine-tuning of the model to fit specific data characteristics.

With Logistic Regression we had an accuracy of:



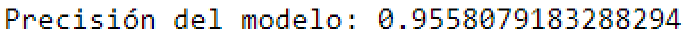
```
Precisión del modelo: 0.9407304937159525
```

With Bayesian Ridge we had an accuracy of:



```
Precisión del modelo: 0.9136747737313945
```

With XGB Regressor we had an accuracy of:



```
Precisión del modelo: 0.9558079183288294
```

With this information we can say that the XGB Regressor had a best accuracy, and it is the one to use as our model.

Conclusion

In conclusion, working with a real credit portfolio dataset requires a systematic and thorough approach to data analysis. Through exploratory data analysis, we were able to gain valuable insights into the structure and distribution of the data, identify potential issues, and inform the selection of appropriate modeling techniques to accurately predict credit risk.

We applied our credit data knowledge to treat the data properly, such as imputing missing values and encoding categorical variables, to ensure the accuracy and completeness of our analysis. We also gained an understanding of the value of each predictor variable and how to improve the predictive power of our models.

We compared different models using evaluation metrics selected the best model based on its interpretability and ability to accurately predict credit risk. We presented our results and conclusions as a formal document, which can be used to inform future credit risk assessment and decision-making processes.

Overall, this project highlights the importance of careful data analysis and model selection when working with credit portfolio datasets, and the potential benefits of using data-driven approaches to inform credit risk assessment and management.