Problem Statement

The main aim of this project is to build a classification model which effectively predicts whether a loan will default or be fully paid.

What is Loan Defaulting

Defaulting on a loan happens when repayments aren't made for a certain period of time. When a loan defaults, it is sent to a debt collection agency whose job is to contact the borrower and receive the unpaid funds. Defaulting will drastically reduce your credit score, impact your ability to receive future credit, and can lead to the seizure of personal property.

Some of the reasons for Loan Defaulting are:

- Job Loss
- Loss in Business
- Frauds

Objective

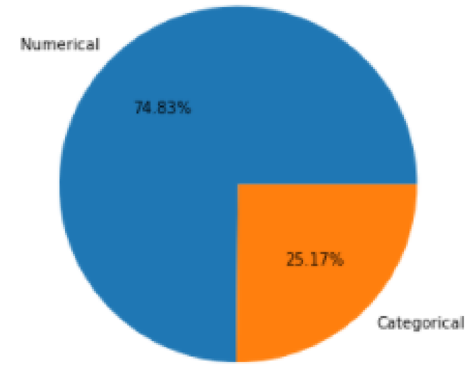
- Within this project, we intend to build a machine learning algorithm for the purpose of correctly identifying if a person, given certain characteristics, has a high likelihood to default on a loan.
- We show an analysis on loan default behavior. Lastly, we show an approach using machine learning to help aid investors/banks in choosing which loan to fund.

DATA SOURCE

https://www.kaggle.com/wordsforthewise/lending-club

```
Shape of the Dataset : (2260701, 151)
Number of Rows : 2260701
Number of Features : 151
```

Types of Variable	Count
Numerical	113
Categorical	38



DATA DICTIONARY

Sl.no	Features	Description	Datatype
0	**loan_amnt**	**The listed amount of the loan applied for by the borrower. If at some point in time, the credit department reduces the loan amount, then it will be reflected in this value.**	**float64**
1	term	The number of payments on the loan. Values are in months and can be either 36 or 60.	object
2	**int_rate**	**Interest Rate on the loan**	**float64**
3	**installment**	**The monthly payment owed by the borrower if the loan originates.**	**float64**
4	**grade**	**LC assigned loan grade**	**object**
5	sub_grade	LC assigned loan subgrade	object
6	emp_title	The job title supplied by the Borrower when applying for the loan.*	object
7	**emp_length**	**Employment length in years. Possible values are between 0 and 10 where 0 means less than one year and 10 means ten or more years.**	**object**

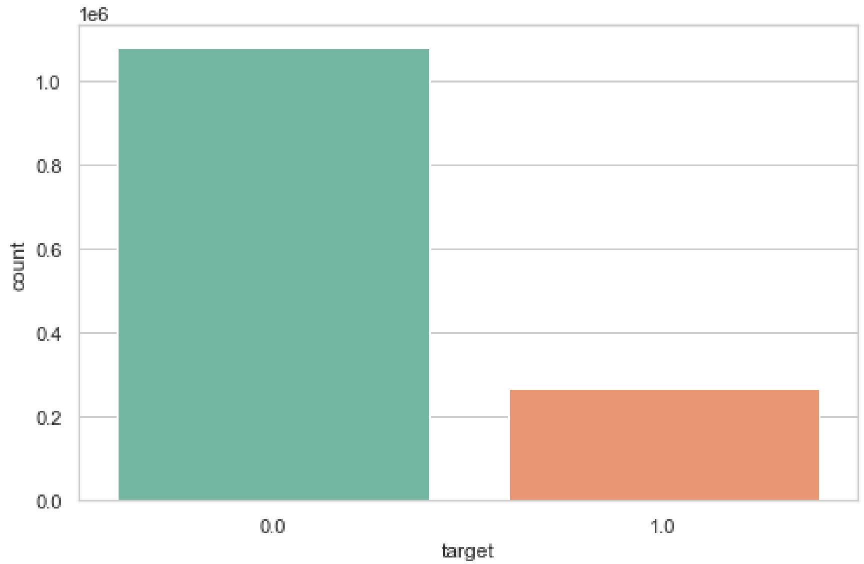
Exploratory Data Analytics

Target Variable

- ✓The **target variable is categorical** in nature.
- ✓It is a binary class (**1 : default, 0: fully paid**).
- ✓This is a **binary classification** problem.

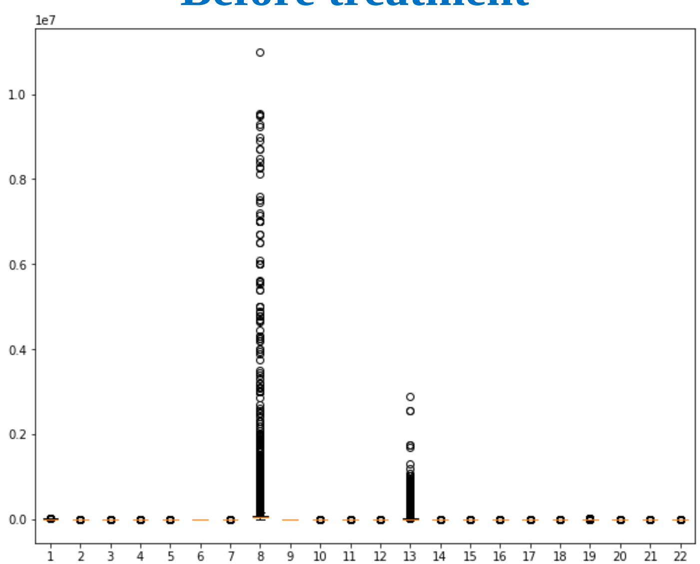
Null Values

- ✓There were many features with 50-100% null values
- ✓We kept a threshold of 50% missing values

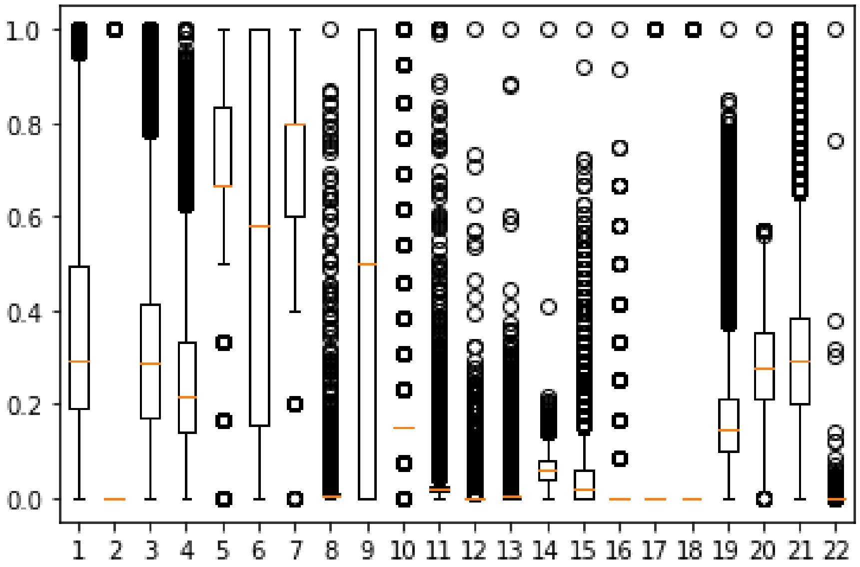


Outliers

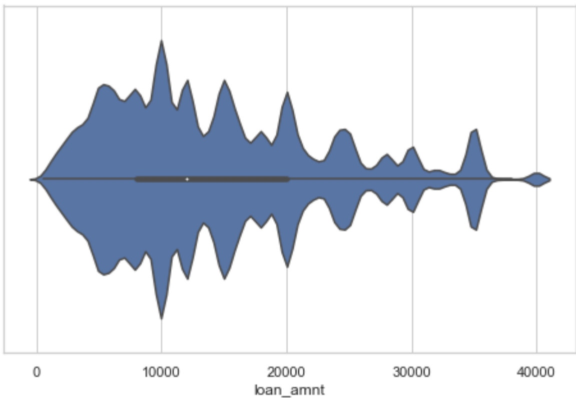
Before treatment

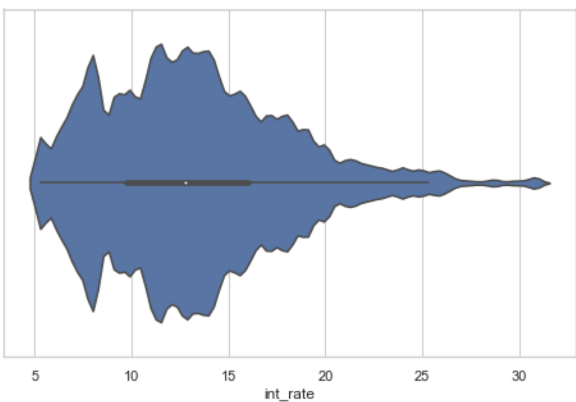


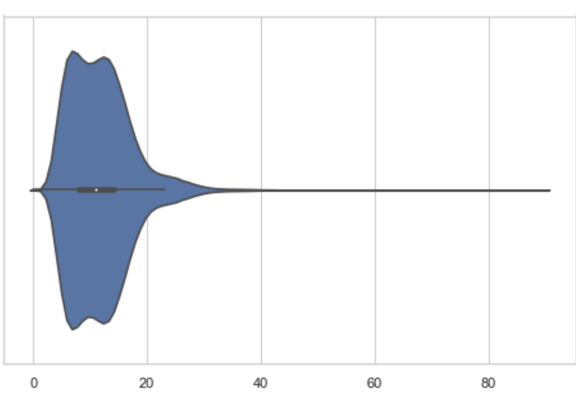
After Treatment



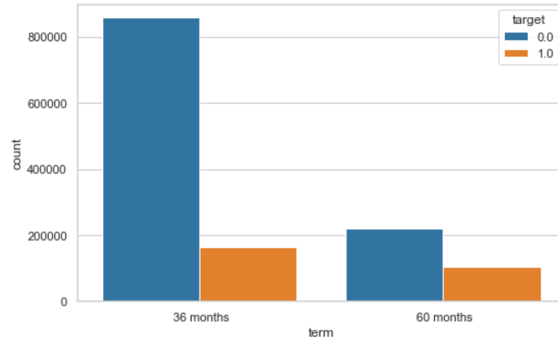
Numerical Data

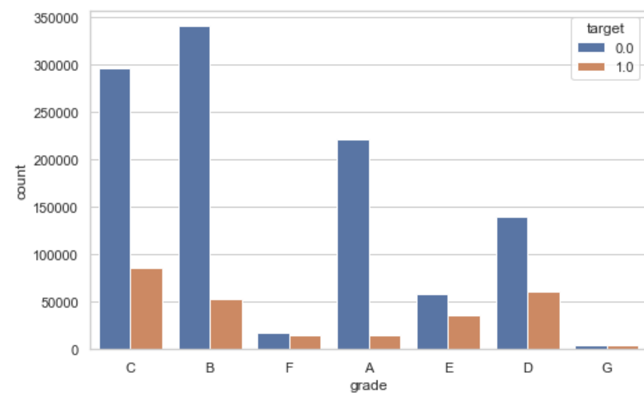


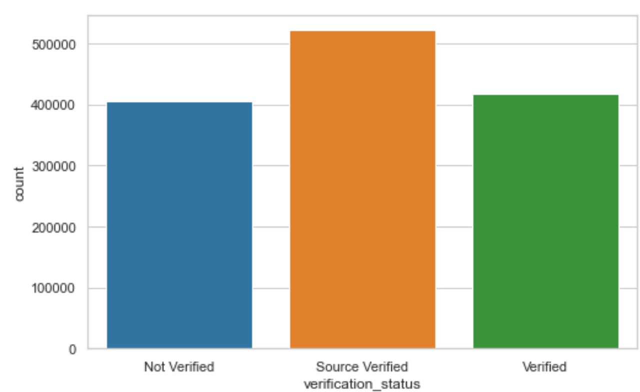




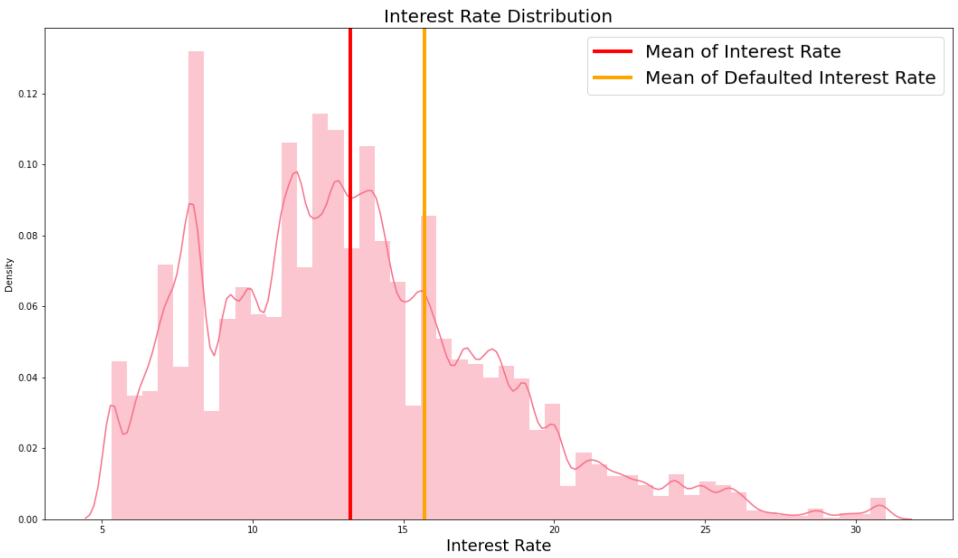
Categorical Data



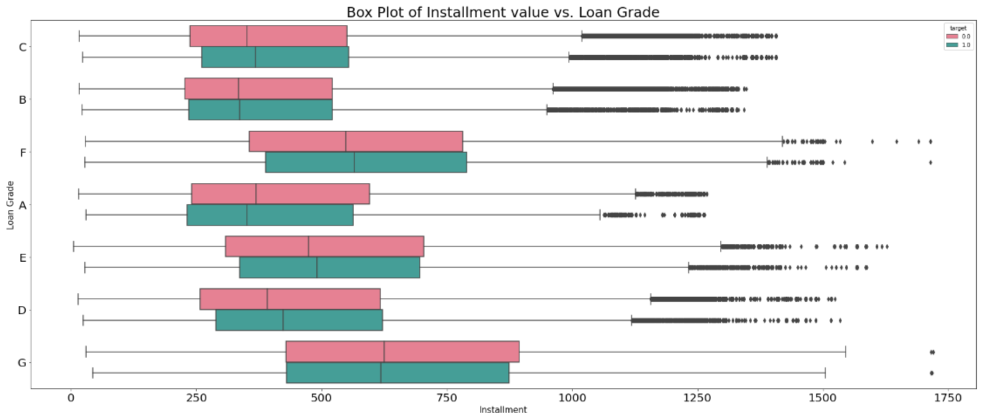




BIVARIATE ANALYSIS

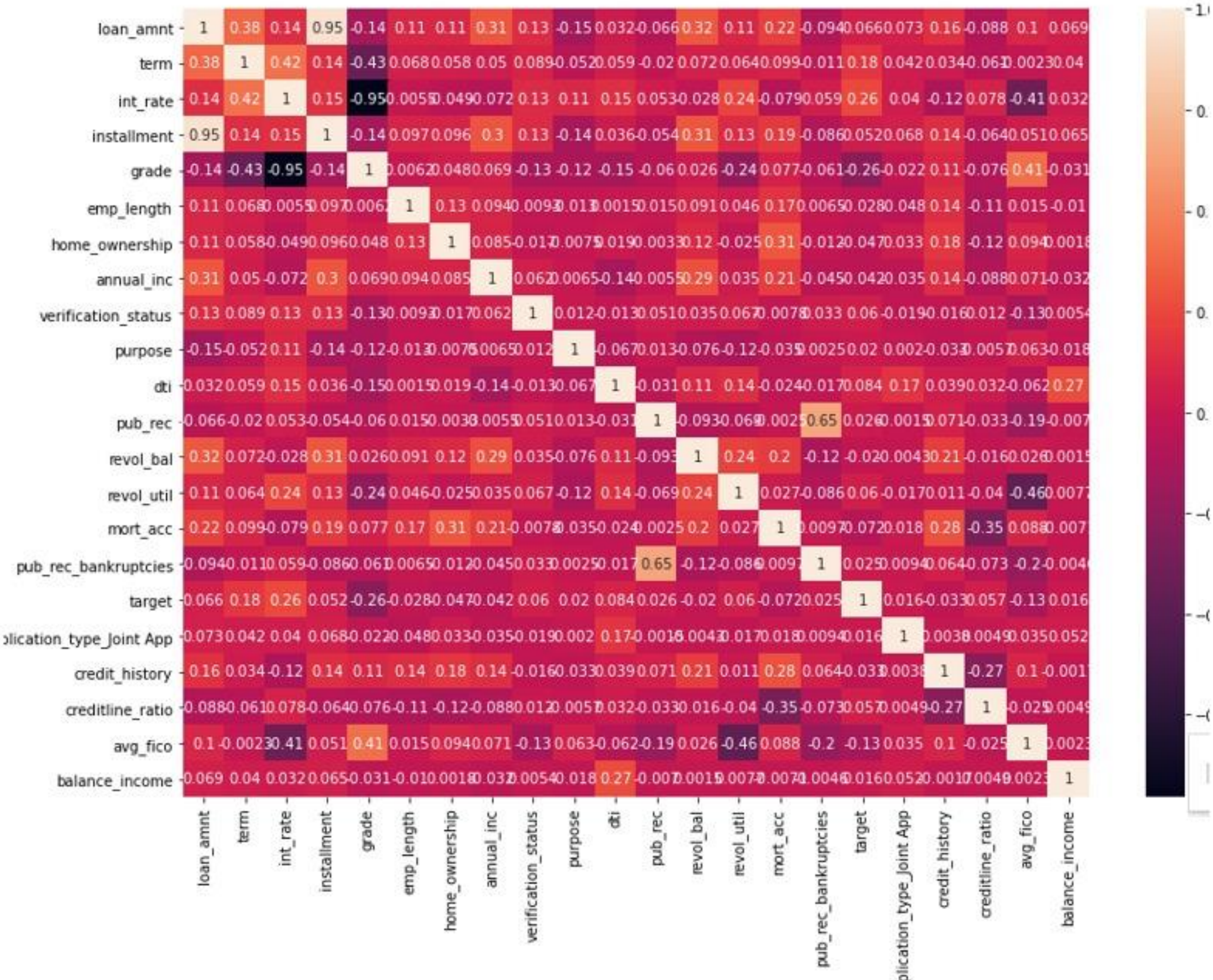


This plot identifies the distributions of loan interest rate records within the dataset. The solid red line represents the mean interest rate for all loans. The solid orange line represents the mean interest rate for loans that have been defaulted. There is a clear 3.95% increase in interest rate between defaulted loans vs. non-defaulted.



The lowest rated loans have the highest average installments. Average installments seem to increase from D-grade moving down to G-grade. The average installment for G-graded loans is around 625 dollars and for A-graded loans, 350 dollars.

Correlation Heatmap:



Statistics

- Did t-test to check if sample is representation of the population.
- Since p-value > 0.05 we fail to reject the null hypothesis i.e. *train and test data is representation of the population*
- Performed feature selection through k-best algorithm to pick features with k-score >1500

```
X = df.drop(['target'], axis=1)
y = df['target']

X_train, X_test, y_train, y_test = train_test_split(X,y, test_size=0.2, stratify=y, random_state
```

```
from scipy.stats import ttest_ind
```

```
ttest_ind(X,X_train)
```

```
Ttest_indResult(statistic=array([-0.24523179, -0.2082384 ,  0.33888499, -0.0689428 , -0.29485498,
        0.09127064, -0.10756269, -0.09013114,  0.32042314,  0.06939033,
        0.01843527,  0.39617408,  0.13851919, -0.00981895,  0.16660083,
        0.10210528, -0.64497115, -0.40716496, -0.72762156, -0.09988012,
       -0.33345793]), pvalue=array([0.80627698, 0.83504284, 0.7346964 , 0.94503515, 0.76810469,
       0.92727756, 0.9143426 , 0.92818301, 0.74864761, 0.94467893,
       0.98529162, 0.69197665, 0.88983012, 0.99216574, 0.86768416,
       0.91867312, 0.51894594, 0.68388687, 0.46684535, 0.9204395 ,
       0.73878867]))
```

```
ttest_ind(X,X_test)
```

```
Ttest_indResult(statistic=array([ 0.60071114,  0.51016896, -0.83008129,  0.16882168,  0.72217605,
       -0.22360291,  0.26358831,  0.22203255, -0.78485131, -0.17002553,
       -0.04595201, -0.9730762 , -0.34009681,  0.02403139, -0.40804161,
       -0.25010373,  1.58503858,  0.99864118,  1.78233342,  0.2446656 ,
        0.93303135]), pvalue=array([0.54803248, 0.60993317, 0.40649295, 0.86593693, 0.47018635,
       0.82306633, 0.7920972 , 0.82428858, 0.43254087, 0.86499008,
       0.9633485 , 0.33051558, 0.73378367, 0.98082757, 0.68324318,
       0.80250716, 0.11295776, 0.31796869, 0.07469506, 0.80671539,
       0.35080392]))
```

```
ttest_ind(y,y_train)
```

```
Ttest_indResult(statistic=0.000431915366753624, pvalue=0.9996553814435957)
```

```
ttest_ind(y,y_test)
```

```
Ttest_indResult(statistic=-0.0010579691451133597, pvalue=0.9991558630415367)
```

Specs	pvalue	Score
grade	0.000000e+00	69327.458896
int_rate	0.000000e+00	67772.737802
term	0.000000e+00	30061.174720
avg_fico	0.000000e+00	16321.616421
dti	0.000000e+00	6963.855903
mort_acc	0.000000e+00	5004.741160
loan_amnt	0.000000e+00	4010.364243
verification_status	0.000000e+00	3428.836546
revol_util	0.000000e+00	3385.844543
creditline_ratio	0.000000e+00	3092.886531
installment	0.000000e+00	2481.616322
home_ownership	0.000000e+00	2156.654738
annual_inc	0.000000e+00	1691.189928
credit_history	1.274056e-225	1028.852008
emp_length	2.876030e-160	727.951905
pub_rec	4.934743e-156	708.463630
pub_rec_bankruptcies	4.412152e-144	653.466228
revol_bal	2.235358e-87	392.693620
purpose	7.293071e-79	353.576049
balance_income	2.403720e-69	309.859909
application_type_Joint App	3.391694e-66	295.398596

Sampling

Treating Imbalance in the data

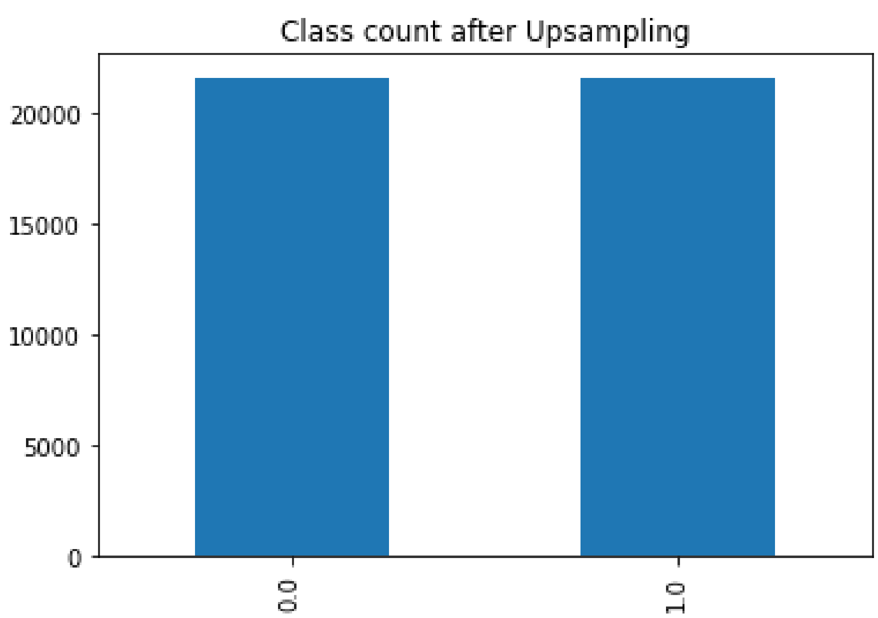
Sampling methods used :

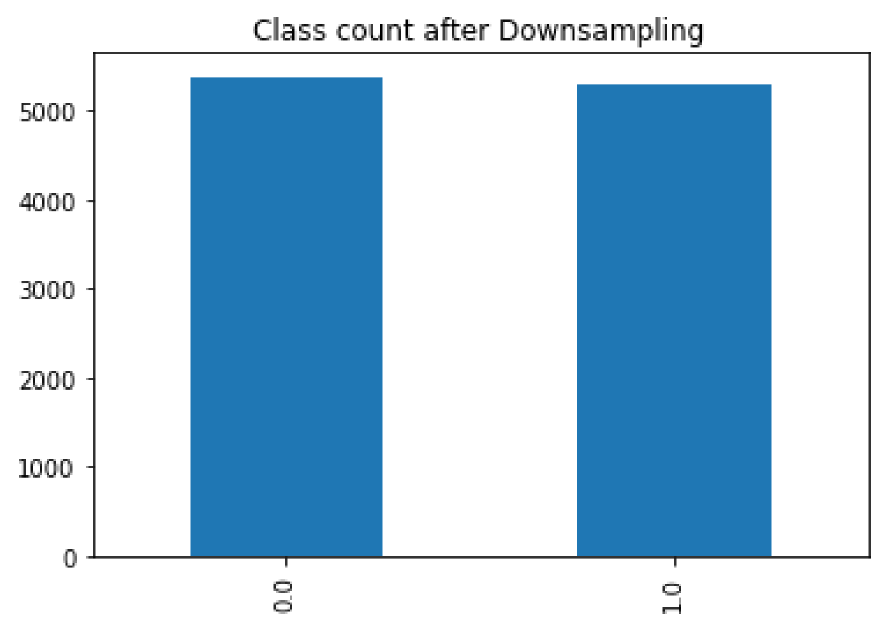
1. Normal Sampling
2. Oversampling
3. Under sampling
4. Smote

```
# Upsample minority class
df_minority_upsampled = resample(df_minority,
                                 replace=True,     # sample with replacement
                                 n_samples = 21522,    # to match majority class
                                 random_state= 42) # reproducible results
# Combine majority class with upsampled minority class
df_upsampled = pd.concat([df_majority, df_minority_upsampled])
# Display new class counts
df_upsampled.target.value_counts()
```

```
# Downsample majority class
df_majority_downsampled = resample(df_majority,
                                 replace=False,    # sample without replacement
                                 n_samples=5378,     # to match minority class
                                 random_state=587) # reproducible results
# Combine minority class with downsampled majority class
df_downsampled = pd.concat([df_majority_downsampled, df_minority])
# Display new class counts
df_downsampled.target.value_counts()
```

```
sm = SMOTE(random_state=40)
X_SMOTE, y_SMOTE = sm.fit_resample(X_train, y_train)
print(len(y_SMOTE))
print(y_SMOTE.sum())
```



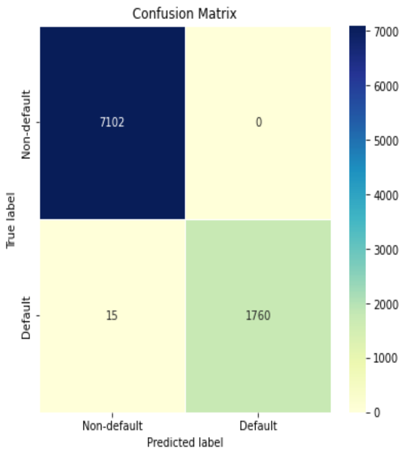


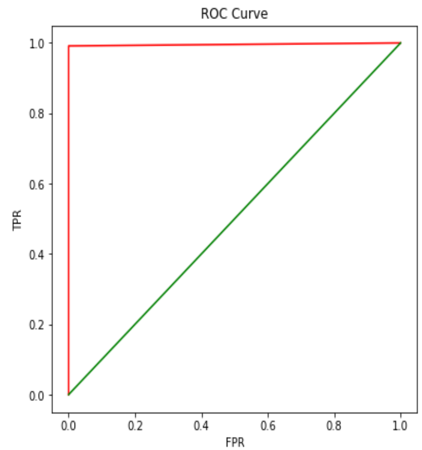
Algorithms considered

- Base model built with Logistic Regression
- Firstly, We have created baseline model using Logistic Regression and see average accuracy using 5 cross validation as 0.523
- Metric used for model evaluation is F1-Score

Model	Data	F1-Score	Accuracy	K-Fold score	AUC-ROC Score
Logistic Regression	Under Sampling	0.68	0.71	0.64	0.71
KNN	Over Sampling	1.0	0.99	0.88	0.99
Random forest	Over Sampling	0.71	0.71	0.71	0.81
Ada Boost	Normal Sampling	0.8	0.8	0.51	0.69
Gradient Boost	Over Sampling	0.81	0.8	0.76	0.88
Decision Tree	Over Sampling	0.84	0.84	0.77	0.94
XGBoost	Over Sampling	1.0	0.99	0.931	0.99
LightGBM	Over Sampling	1.0	0.99	0.922	0.99
CATBoost	Over Sampling	0.95	0.94	0.859	0.98

KNN

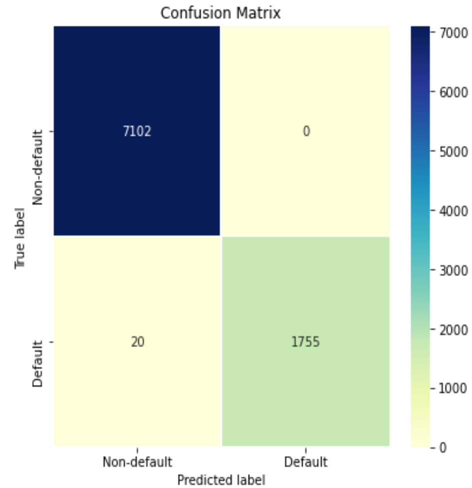


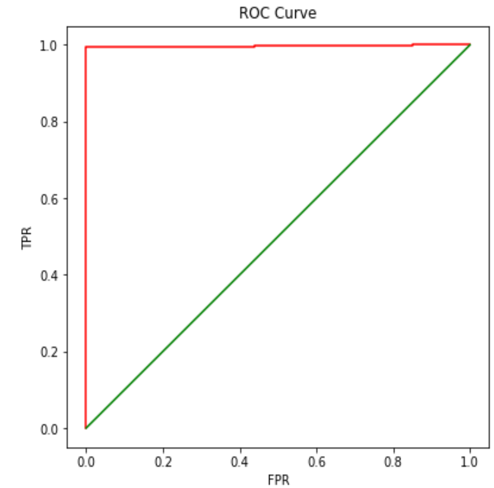



```
Classification Report of Test
              precision    recall  f1-score   support

         0.0       1.00      1.00      1.00      7102
         1.0       1.00      0.99      1.00      1775

    accuracy                           1.00      8877
   macro avg       1.00      1.00      1.00      8877
weighted avg       1.00      1.00      1.00      8877

K-Fold scores: 0.883 (+/- 0.00001)
```

```
Tuned KNN Parameters: {'n_neighbors': 1, 'p': 1} for Over Sampling
```

Light GBM





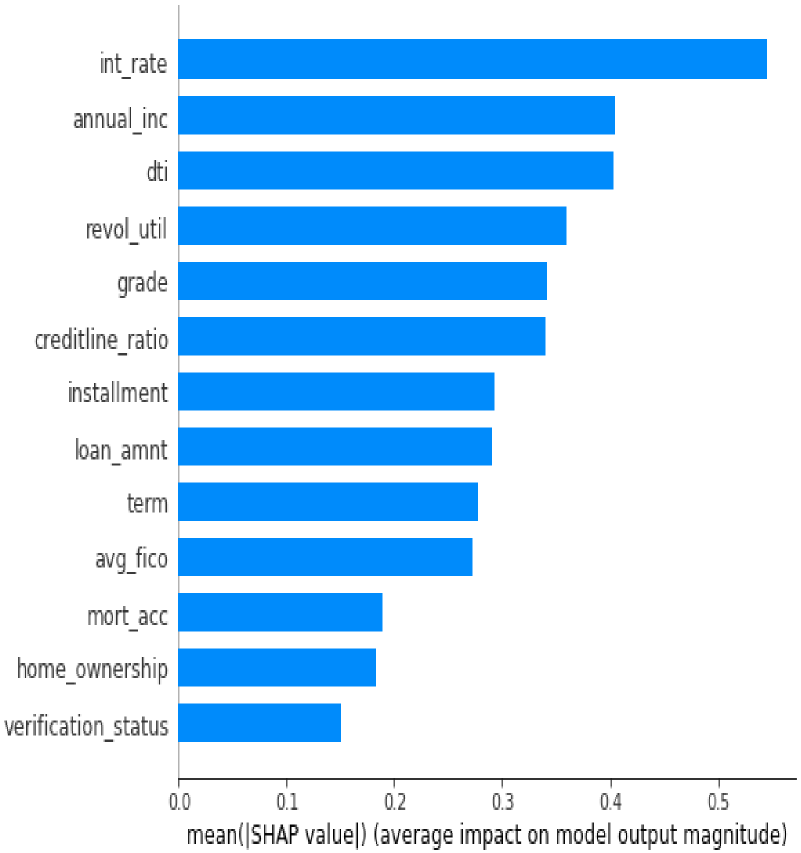

```
Classification Report of Test
              precision    recall  f1-score   support

         0.0       1.00      1.00      1.00      7102
         1.0       1.00      0.99      0.99      1775

    accuracy                           1.00      8877
   macro avg       1.00      0.99      1.00      8877
weighted avg       1.00      1.00      1.00      8877
```

```
K-Fold scores: 0.922 (+/- 0.00001)
```

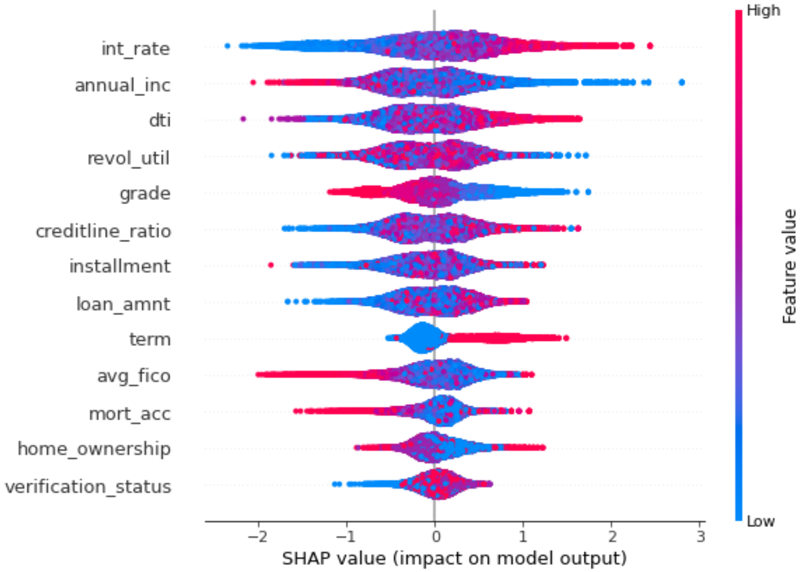
```
Tuned Light Boost Parameters: {'learning_rate': 0.75, 'max_depth': 17, 'min_data_in_leaf': 10, 'n_estimators': 116, 'num_leave
s': 215} for Over Sampling
```

Feature Importance

Average impact of SHAP value



SHAP value (impact on model output)



Business Interpretation

This project is a typical binary classification problem, which leverages the loan and personal information to predict whether the customer will default the loan. The goal is to use the model as a tool to help make decisions on issuing the loans.

From this we can deduce that based on our assumptions:

- If a person has high interest rate, he/she is more likely to default
- If a person has low annual income, he/she is likely to default
- If the debt-income ratio is high he/she will be a default risk

- As machine learning models are trained to provide more accurate results they do so at the cost of interpretability. This is what we have tried to mitigate by using SHAP values. These are part of an explainable AI which helps us to decode feature interactions and how the solution in our hands work.
- Feature importance is plotted w.r.t. the average impact of a feature on the model output.
- It doesn't give us an idea about causality

Thank You