Lending Club Case Study

Executive PG Programme in Machine Learning & AI - April 2023
(Batch 4616)

- Manish Kumar Srivastava

PROBLEM STATEMENT

For a consumer finance company which specializes in lending various types of loans to urban customers, **identify driving factors (or driver variables)** behind loan default using historical sample dataset & data dictionary which will indicate if a consumer is likely to default.

1. DATA UNDERSTANDING

Following are the steps taken to understand the data.

- ✓ Opened loan dataset & gone through the available data.
- ✓ Opened data dictionary to understand the meaning of data columns.
- ✓ To do the further analysis, opened jupyter notebook.
- ✓ Loaded all required libraries like pandas, numpy, seaborn, matplotlib etc.
- ✓ Loaded loan data in pandas dataframe

```
df_loan = pd.read_csv("loan.csv",index_col=0)
```

- ✓ To view the shape of the dataset –

```
df_loan.shape
```

- ✓ To view initial few records of the dataset –

```
df_loan.head()
```

- ✓ To view data description like min, max, count etc. –

```
df_loan.describe()
```

- ✓ To view columnwise info –

```
df_loan.info(verbose=True, show_counts=True)
```

2. DATA CLEANING & MANIPULATION (1/2)

Following are the steps taken for data cleaning.

✓ Removed all columns which contained only *null* values

```
df_loan = df_loan.dropna(axis=1, how='all')
```

✓ Checked number of unique values for each column

```
df_loan.nunique()
```

✓ Dropped all columns from the dataset which had same unique value across the column as they cannot be the driving variables

```
df_loan = df_loan.drop(['pymnt_plan', 'initial_list_status', …….. ], axis=1)
```

✓ Dropped all columns which cannot be the driver variables for finding out the default loan pattern like

- Id, member id these are Unique identifiers
- columns containing customer profile details like emp_title, title, zip_code, addr_state etc.

```
df_loan = df_loan.drop(['id', 'member_id', 'sub_grade', 'issue_d',…….. ], axis=1)
```

✓ Checked percentage of missing values and dropped columns which had >50% missing values

```
df_loan.isnull().mean()*100
df_loan = df_loan.drop(['mths_since_last_delinq', 'mths_since_last_record'], axis=1)
```

✓ Removed '%' character from percentage columns to convert the text values into numeric values

```
df_loan['int_rate'] = df_loan['int_rate'].str.replace('%', '')
df_loan['revol_util'] = df_loan['revol_util'].str.replace('%', '')
```

2. DATA CLEANING & MANIPULATION (2/2)

✓ Changed data types after removing '%' character from percentage columns

```
convert_dict = {'int_rate': float,'revol_util': float}
df_loan = df_loan.astype(convert_dict)
```

✓ Used *mode* to handle missing categorial data

```
emp_length_mode = df_loan['emp_length'].mode()[0]
df_loan['emp_length'] = df_loan['emp_length'].fillna(emp_length_mode)
```

✓ Used *median* to handle missing numerical data

```
revol_util_median = df_loan['revol_util'].median()
df_loan['revol_util'] = df_loan['revol_util'].fillna(revol_util_median)
pub_rec_bankruptcies_median = df_loan['pub_rec_bankruptcies'].median()
df_loan['pub_rec_bankruptcies'] = df_loan['pub_rec_bankruptcies'].fillna(pub_rec_bankruptcies_median)
df_loan.info()
```

3. DATA VISUALIZATION & ANALYSIS

✓ Grouped columns into categorial and continuous variables for analysis

```
cat_cols = ['term', 'grade', 'emp_length', 'home_ownership', 'verification_status', 'loan_status', 'purpose']
cont_cols = ['loan_amnt', 'funded_amnt', 'funded_amnt_inv', 'int_rate', 'installment', 'annual_inc', 'dti', 'open_acc', 'pub_rec', 'revol_bal', 'revol_util',
'total_acc', 'out_prncp', 'out_prncp_inv', 'total_pymnt', 'total_pymnt_inv', 'total_rec_prncp', 'total_rec_int', 'total_rec_late_fee', 'last_pymnt_amnt',
'pub_rec_bankruptcies']
```

✓ Data visualization and bivariate analysis through continuous data

```
for i in cont_cols:
    sns.barplot(x=df_loan[i], y=df_loan['loan_status'], ci=None)
    plt.show()
```

✓ Data visualisation and bivariate analysis through categorial data

```
for i in cat_cols:
    sns.histplot(binwidth=0.5, x=df_loan[i], hue="loan_status", data=df_loan, stat="count", multiple="stack")
    plt.show()
```

CONCLUSION

Based on the above analysis, below are the driving factors behind loan default:

- loan_amnt
- funded_amnt
- funded_amnt_inv
- int_rate
- installment
- annual_inc
- dti
- open_acc
- pub_rec
- revol_bal
- revol_util
- total_acc
- total_rec_late_fee
- pub_rec_bankruptcies

Below could be few additional driver variables:

- term
- home_ownership