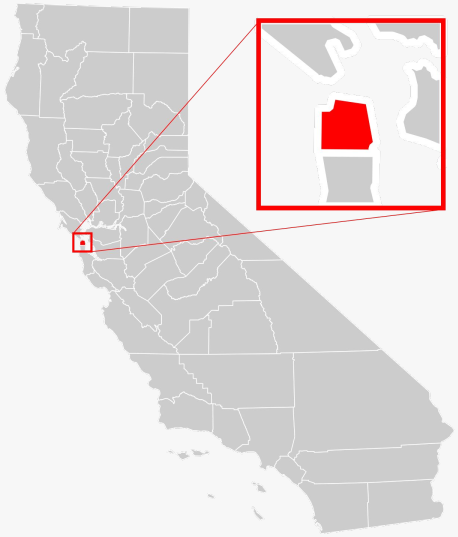
Final Project: CSCI P556

Lending Club Loan Defaulters' Prediction

Company Introduction



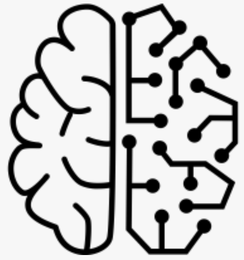
Company Profile:

1. World's largest peer-to-peer lending company based in San Francisco, USA

2. First peer-to-peer lender to register its offerings with Securities and Exchange Commission (SEC)

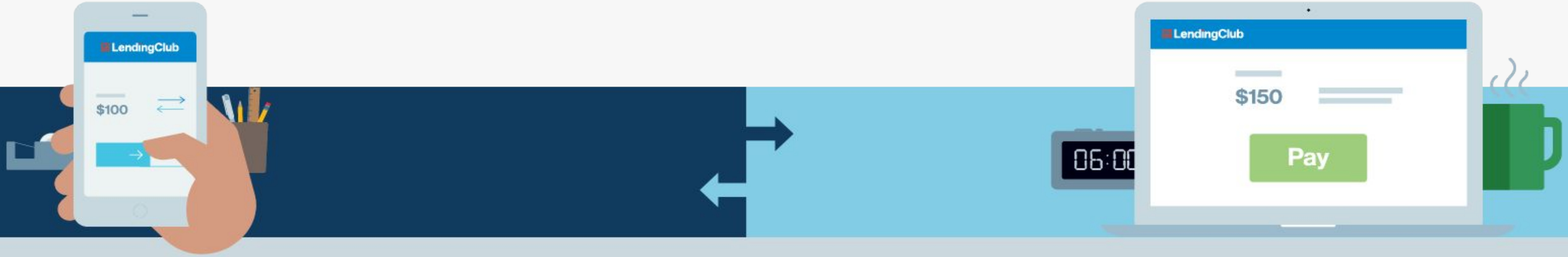
Scope of the Project

Risk Analytics in banking & financial services



Applying Machine Learning to real-world business problems

Business Understanding & Objectives



Investors

In exchange for competitive returns, investors purchase Notes, which correspond to fractions of loans.

LendingClub

LendingClub screens borrowers, facilitates the transaction, and services the loans.

Borrowers

Borrowers use loans to consolidate debt, improve their homes, finance major purchases, and more.

Loans are issued via WebBank, member FDIC

Business Objectives

Minimize Credit Loss

Understand Driving Factors Behind Loan Default

Maximize Profit

Data Description

Columns	Description
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.
term	The number of payments on the loan. Values are in months and can be either 36 or 60.
int_rate	Interest Rate on the loan
installment	The monthly payment owed by the borrower if the loan is approved.
home_ownership	The home ownership status provided by the Borrower during registration or obtained from the credit report. Values used are: Rent, Own, Mortgage and Other
annual_inc	The self-reported annual income provided by the borrower.
loan_status	Current status of the loan
dti	A ratio calculated using the borrower’s total monthly debt payments on the total debt obligations, excluding mortgage and the requested Lending Club loan, divided by the borrower’s self-reported monthly income.
open_acc	The number of open credit lines in the borrower's credit file.
pub_rec	Number of derogatory public records
total_acc	The total number of credit lines currently in the borrower's credit file
mort_acc	Number of mortgage accounts on the applicant's file.
pub_rec_bankruptcies	Number of public record bankruptcies

EXPLORATORY DATA ANALYSIS

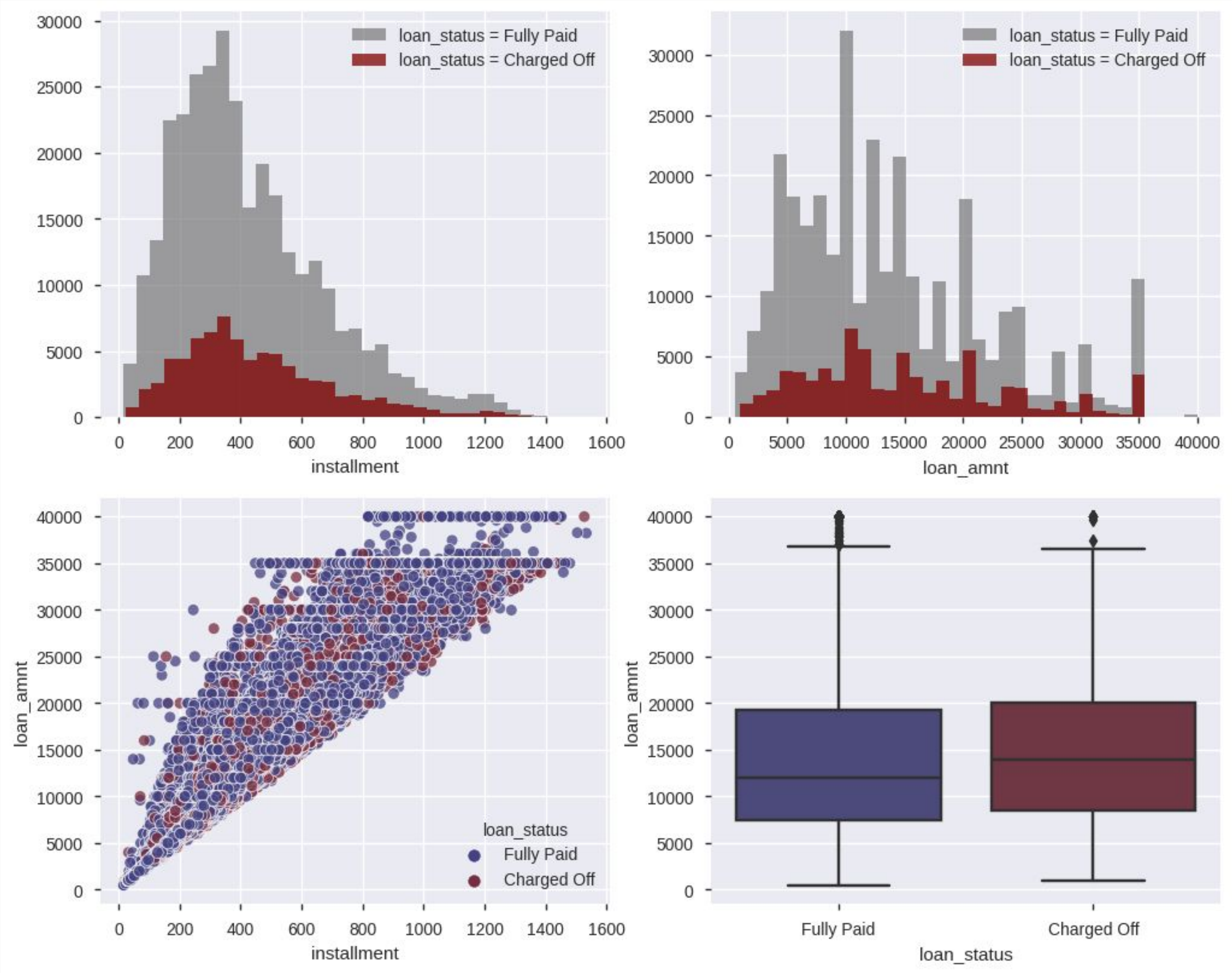
train.info()

```
 #   Column                Non-Null Count   Dtype
---  ------                --------------   -----
 0   loan_amnt             396030 non-null  float64
 1   term                  396030 non-null  object
 2   int_rate              396030 non-null  float64
 3   installment           396030 non-null  float64
 4   grade                 396030 non-null  object
 5   sub_grade             396030 non-null  object
 6   emp_title             373103 non-null  object
 7   emp_length            377729 non-null  object
 8   home_ownership        396030 non-null  object
 9   annual_inc            396030 non-null  float64
 10  verification_status   396030 non-null  object
 11  issue_d               396030 non-null  object
 12  loan_status           396030 non-null  object
 13  purpose               396030 non-null  object
 14  title                 394275 non-null  object
 15  dti                   396030 non-null  float64
 16  earliest_cr_line      396030 non-null  object
 17  open_acc              396030 non-null  float64
 18  pub_rec               396030 non-null  float64
 19  revol_bal             396030 non-null  float64
 20  revol_util            395754 non-null  float64
 21  total_acc             396030 non-null  float64
 22  initial_list_status   396030 non-null  object
 23  application_type      396030 non-null  object
 24  mort_acc              358235 non-null  float64
 25  pub_rec_bankruptcies  395495 non-null  float64
 26  address               396030 non-null  object
dtypes: float64(12), object(15)
```

display(train.isnull().sum().sort_values(ascending=False))

```
mort_acc                37795
emp_title               22927
emp_length              18301
title                    1755
pub_rec_bankruptcies      535
revol_util                276
address                     0
verification_status         0
term                        0
int_rate                    0
installment                 0
grade                       0
sub_grade                   0
home_ownership              0
annual_inc                  0
purpose                     0
issue_d                     0
loan_status                 0
dti                         0
earliest_cr_line            0
open_acc                    0
pub_rec                     0
revol_bal                   0
total_acc                   0
initial_list_status         0
application_type            0
loan_amnt                   0
dtype: int64
```

EXPLORATORY DATA ANALYSIS

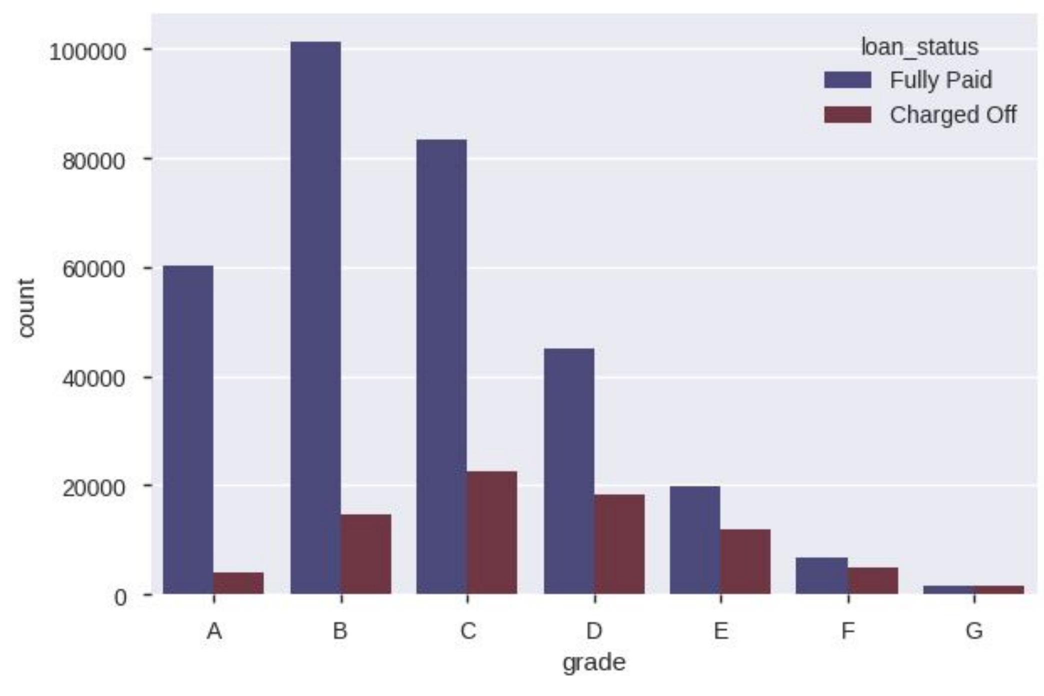


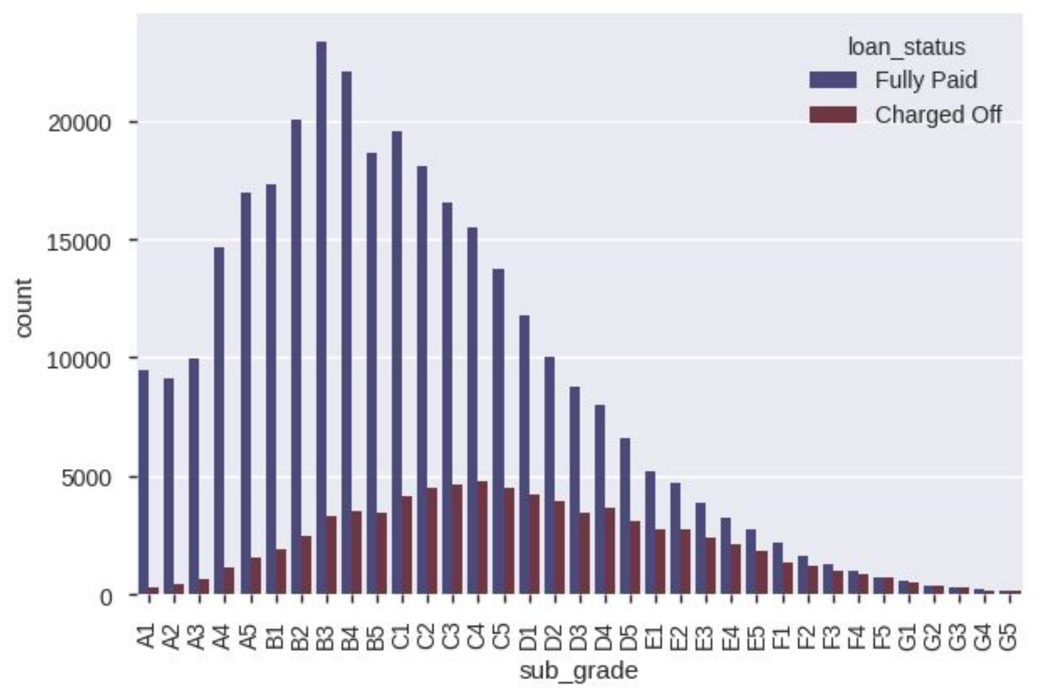
Insights

1. Relative to loan status, both installment & loan amount are fairly distributed amongst the values

2. Median loan amount for defaulters is higher as compared to non-defaulters

Exploratory Data Analysis

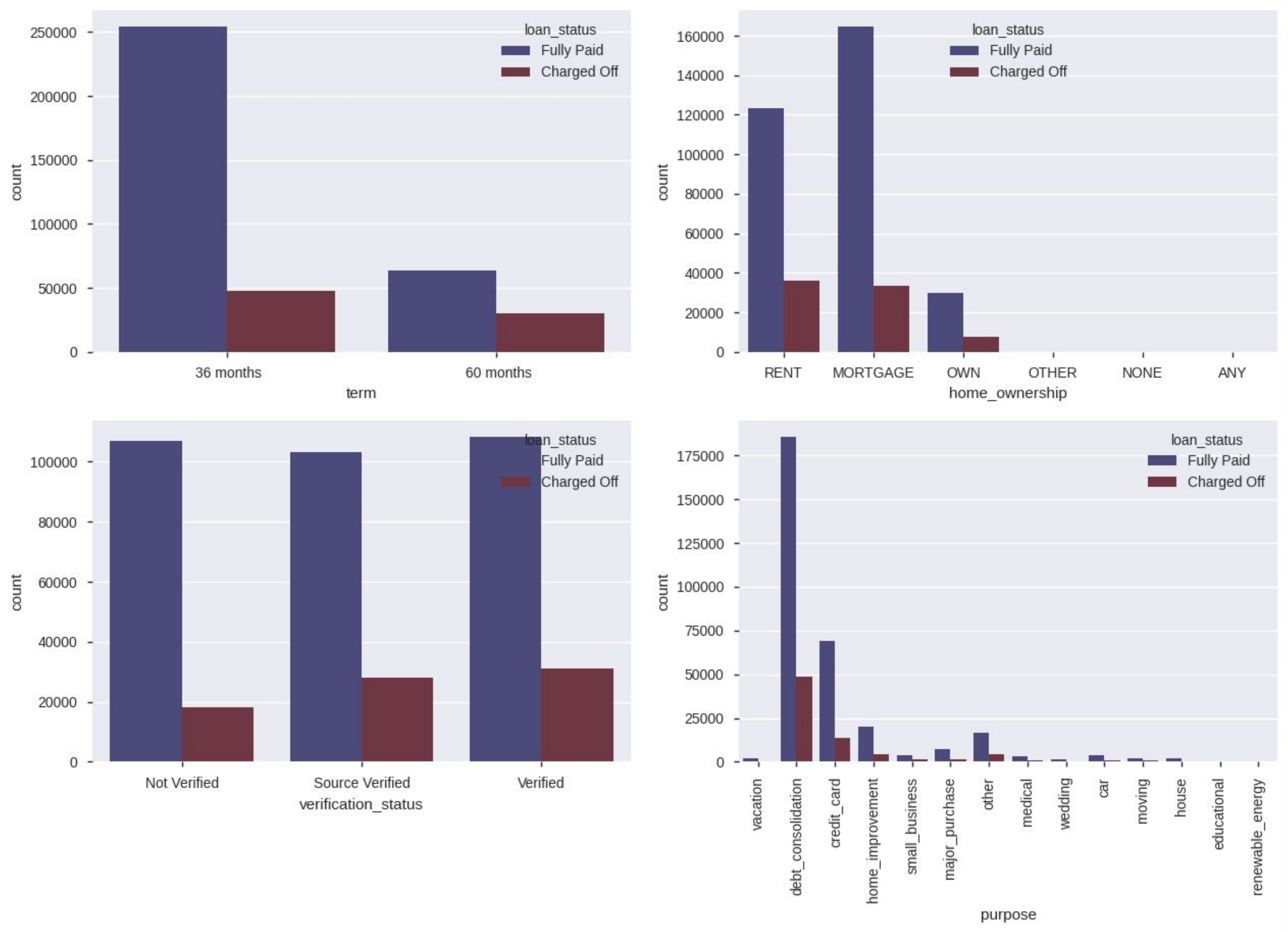




Insights

1. Grades F & G show higher probability of loan default
2. F5 – G5 call for special concern within these grades for Lending Club

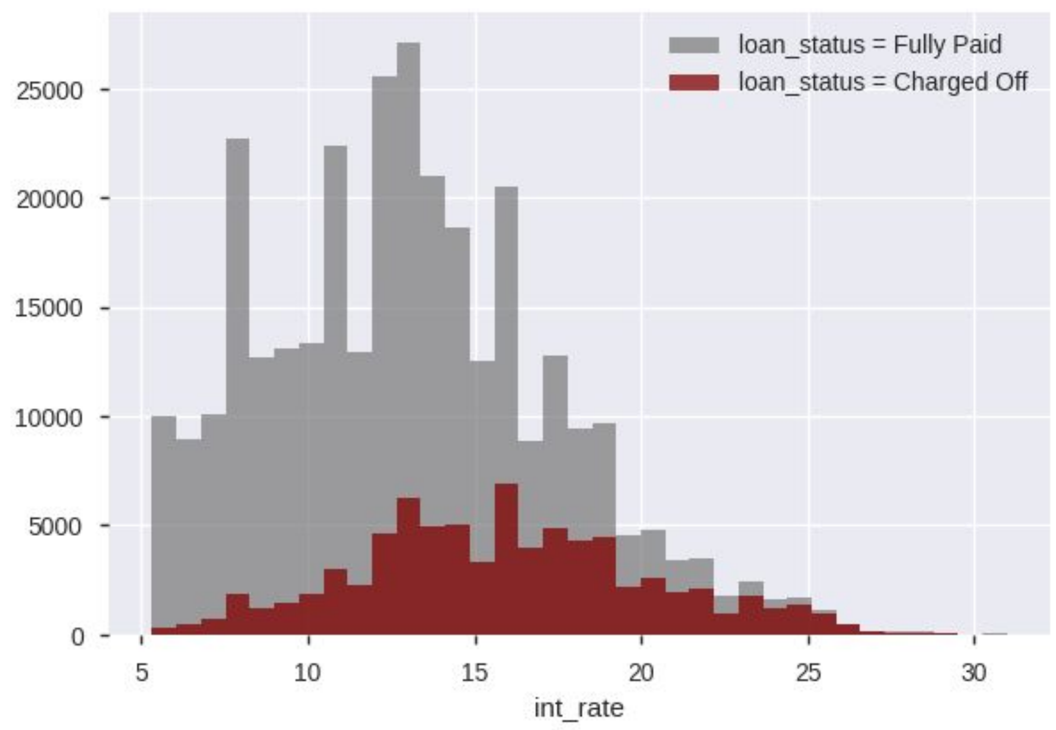
Exploratory Data Analysis

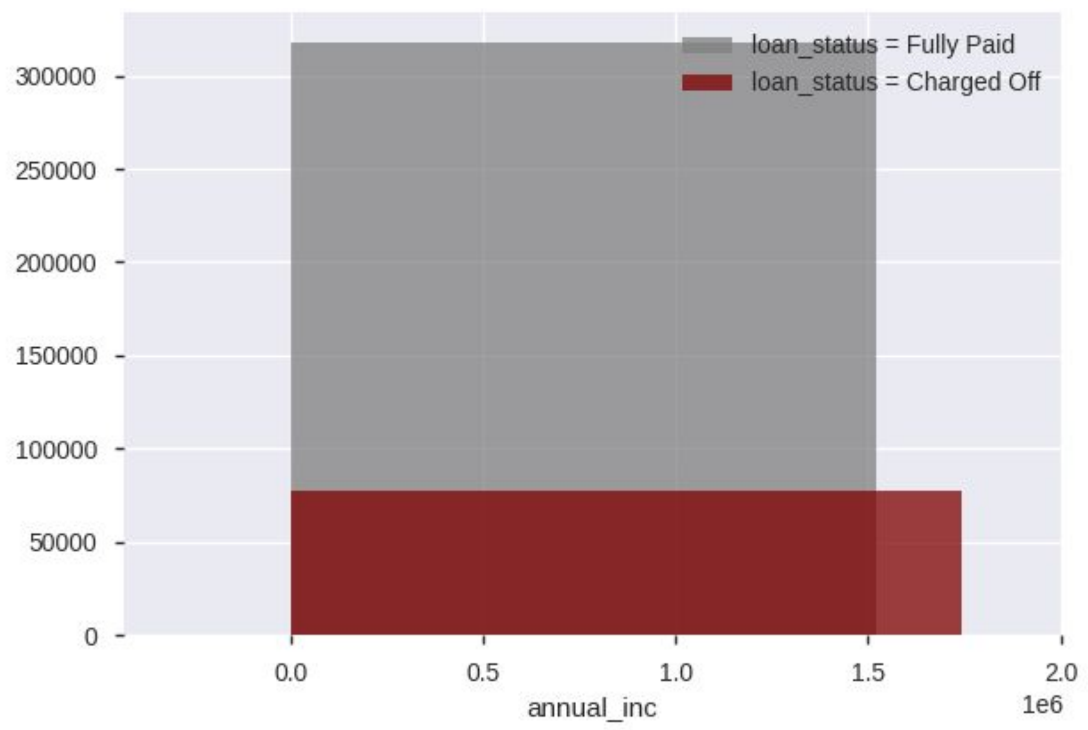


Insights

1. Longer term loan is more likely to default
2. Counter intuitively people with no verification have slightly less likelihood of loan default
3. Debt consolidation seems to be the most common reason for loan availing

Exploratory Data Analysis

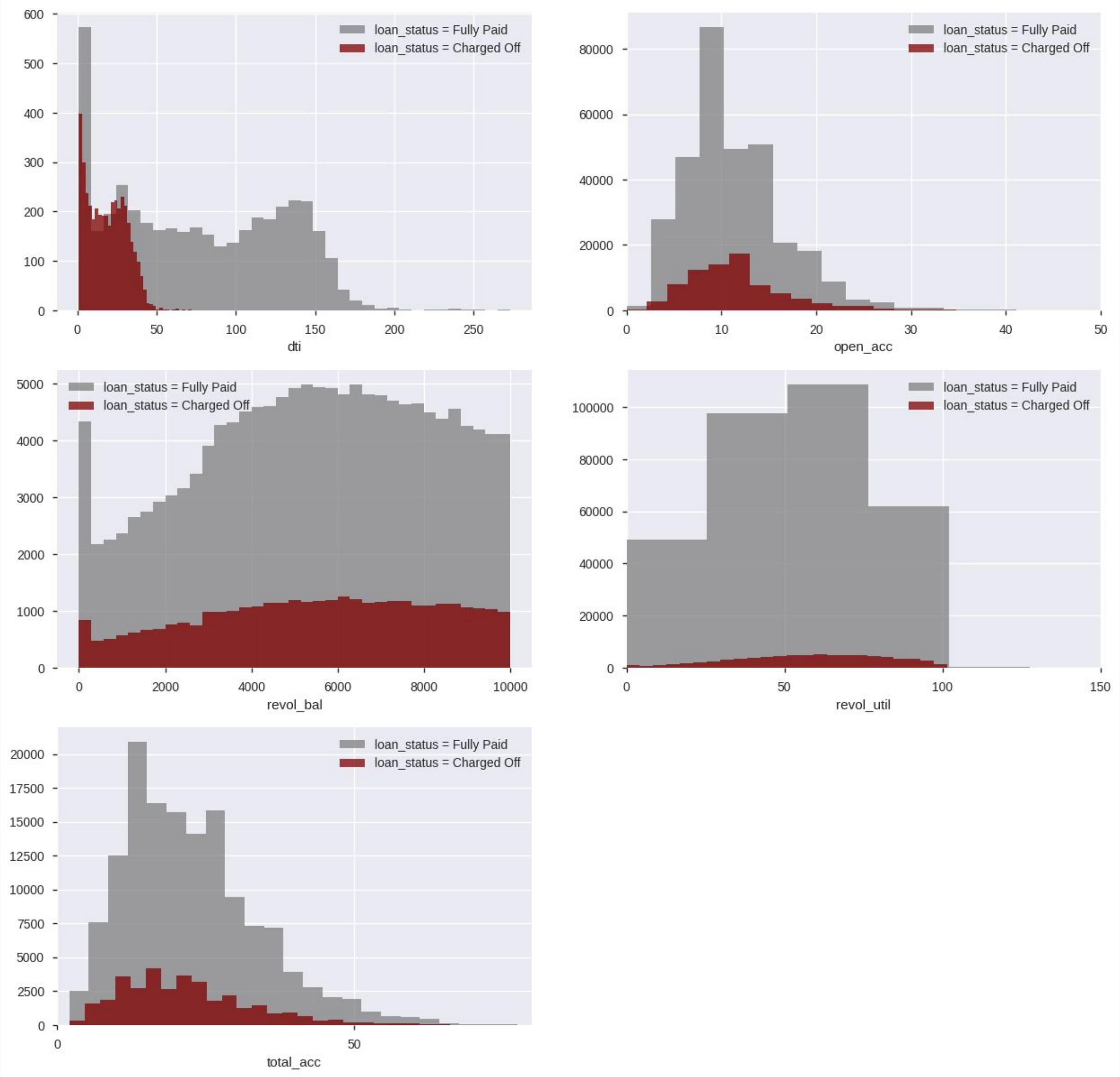




Insights

1. Proportion of default becomes higher as the interest rate increases
2. A continuous distribution is observed for annual income

Exploratory Data Analysis

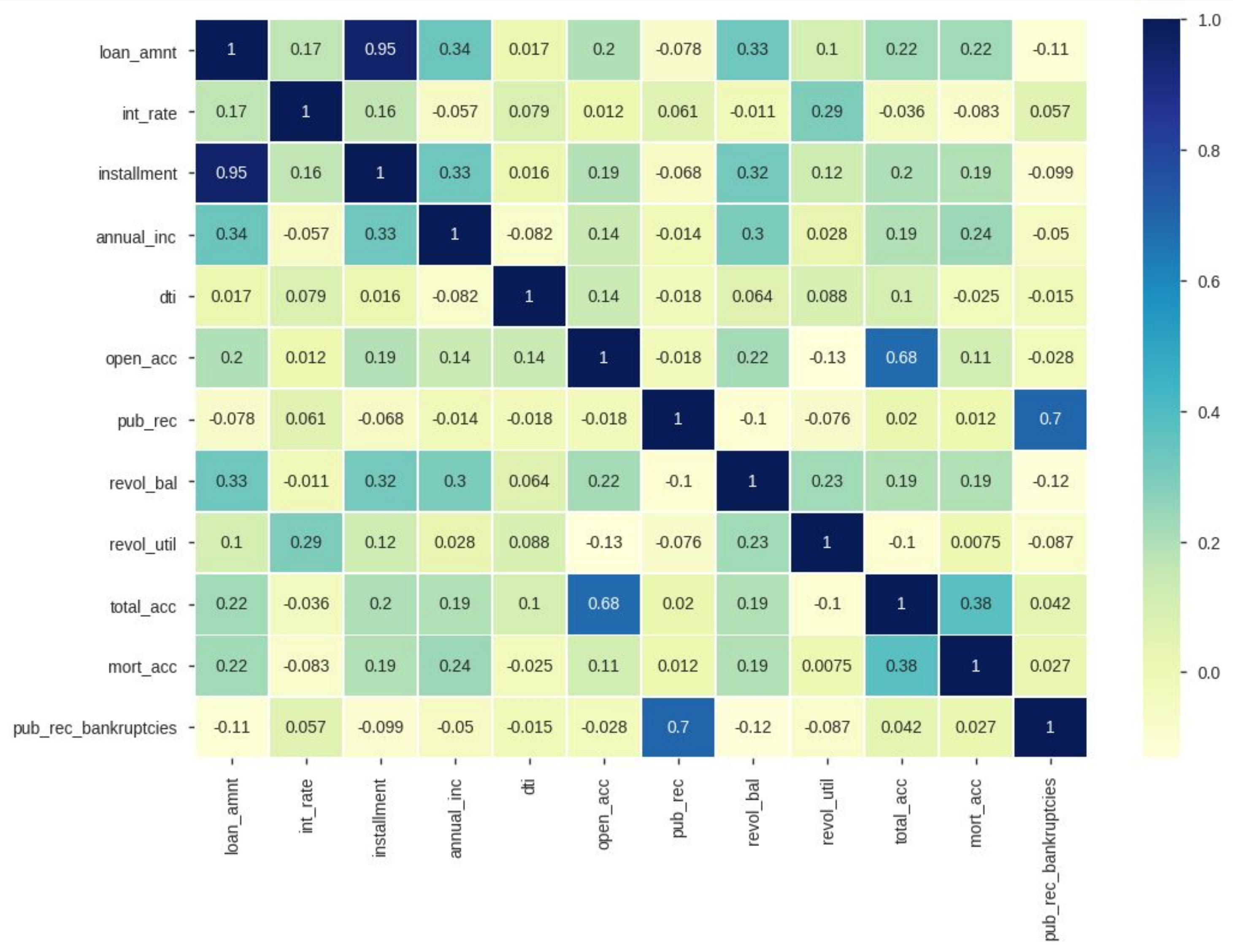


Insights

1. Smaller the dti, more chances of defaulting on the repayment

2. Total accounts follow a normal distribution but the mean for accounts which have defaulted is much lesser than the mean for accounts that have fully paid

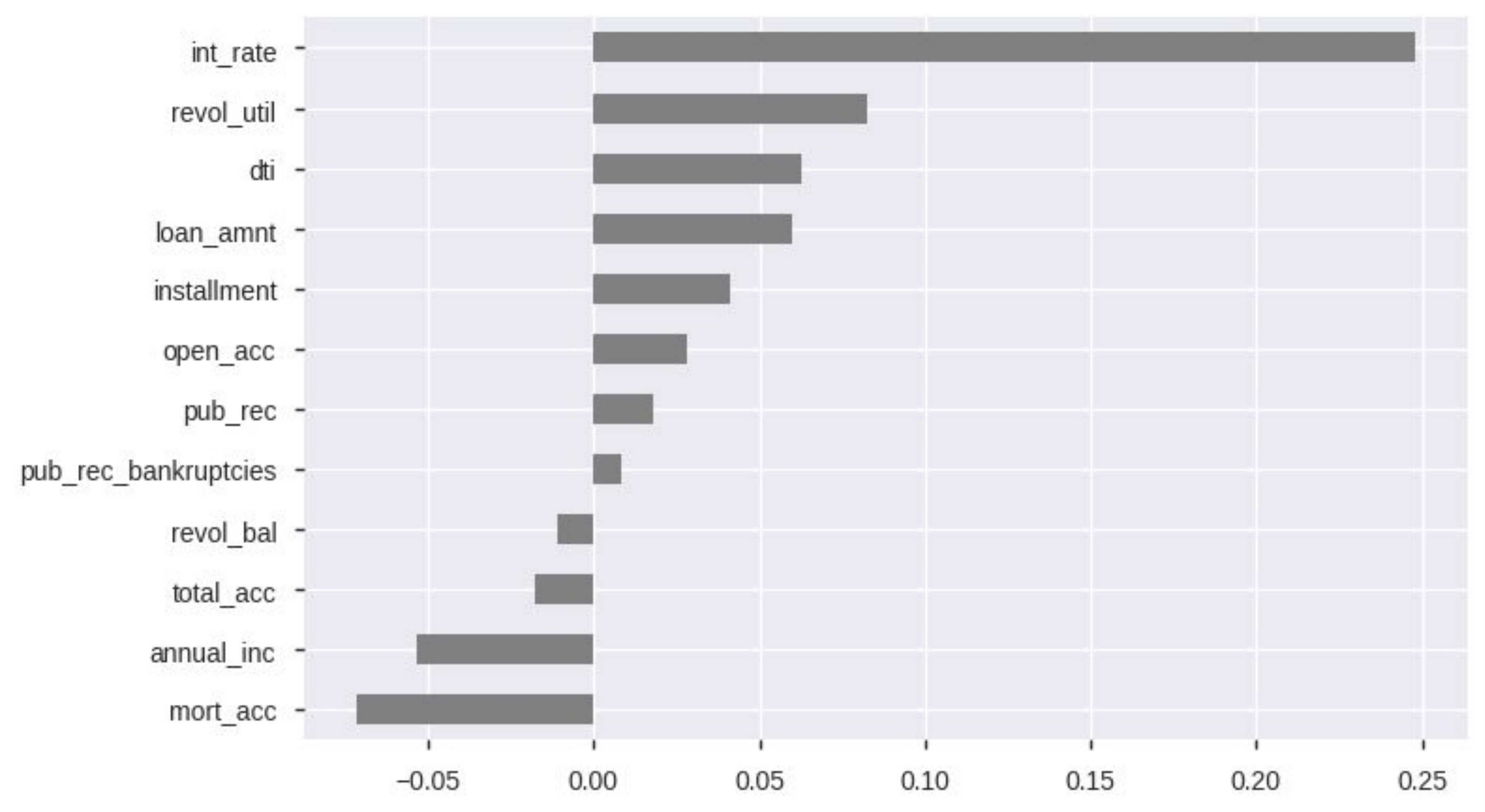
Exploratory Data Analysis



Insights

1. Most of the variables are mutually correlated
2. Strong correlation can be seen between loan amount and installments
3. Moderately strong correlation can be seen between open credit lines and total accounts
4. Slightly weaker correlation can be seen between mortgage accounts and total accounts

Exploratory Data Analysis



Insights

1. Interest rate is the strongest positive correlation with loan status
2. Whereas, mortgage accounts show the strongest negative correlation

Exploratory Data Analysis

```
report = pp.ProfileReport(train)
display(report)
```

HIGH CARDINALITY:

- emp_title has a high cardinality: 173105 distinct values
- issue_d has a high cardinality: 115 distinct values
- title has a high cardinality: 48817 distinct values
- earliest_cr_line has a high cardinality: 684 distinct values
- address has a high cardinality: 393700 distinct values

HIGH CORRELATION:

- loan_amnt is highly correlated with installment
- installment is highly correlated with loan_amnt
- sub_grade is highly correlated with grade
- grade is highly correlated with sub_grade

MISSING VALUES:

- emp_title has 22927 (5.8%) missing values
- emp_length has 18301 (4.6%) missing values
- mort_acc has 37795 (9.5%) missing values

SKEWNESS:

- annual_inc is highly skewed (γ1 = 41.04272475)
- dti is highly skewed (γ1 = 431.0512254)

ZERO VALUES:

- pub_rec has 338272 (85.4%) zeros
- mort_acc has 139777 (35.3%) zeros
- pub_rec_bankruptcies has 350380 (88.5%) zeros

Data Pre-Processing

Below attributes were dropped from the model:

emp_title

```
data.drop('emp_title', axis=1, inplace=True)
```

emp_length

```
data.drop('emp_length', axis=1, inplace=True)
```

title

```
data.drop('title', axis=1, inplace=True)
```

grade

```
data.drop('grade', axis=1, inplace=True)
```

issue_d

```
data.drop('issue_d', axis=1, inplace=True)
```

Data Pre-Processing

Below attributes were re-engineered from the model:

mort_acc

```
total_acc_avg = data.groupby(by='total_acc').mean().mort_acc
```

```
def fill_mort_acc(total_acc, mort_acc):
    if np.isnan(mort_acc):
        return total_acc_avg[total_acc].round()
    else:
        return mort_acc
```

```
data['mort_acc'] = data.apply(lambda x: fill_mort_acc(x['total_acc'], x['mort_acc']), axis=1)
```

revol_util &
pub_rec_bankruptcies

```
for column in data.columns:
    if data[column].isna().sum() != 0:
        missing = data[column].isna().sum()
        portion = (missing / data.shape[0]) * 100
        print(f"'{column}': number of missing values '{missing}' ==> '{portion:.3f}%'")
```

```
'revol_util': number of missing values '276' ==> '0.070%'
'pub_rec_bankruptcies': number of missing values '535' ==> '0.135%'
```

```
data.dropna(inplace=True)
```

```
data.shape
```

address

```
data['zip_code'] = data.address.apply(lambda x: x[-5:])
```

DATA PRE-PROCESSING

Converting to dummy variables:

```
print([column for column in data.columns if data[column].dtype == object])
```

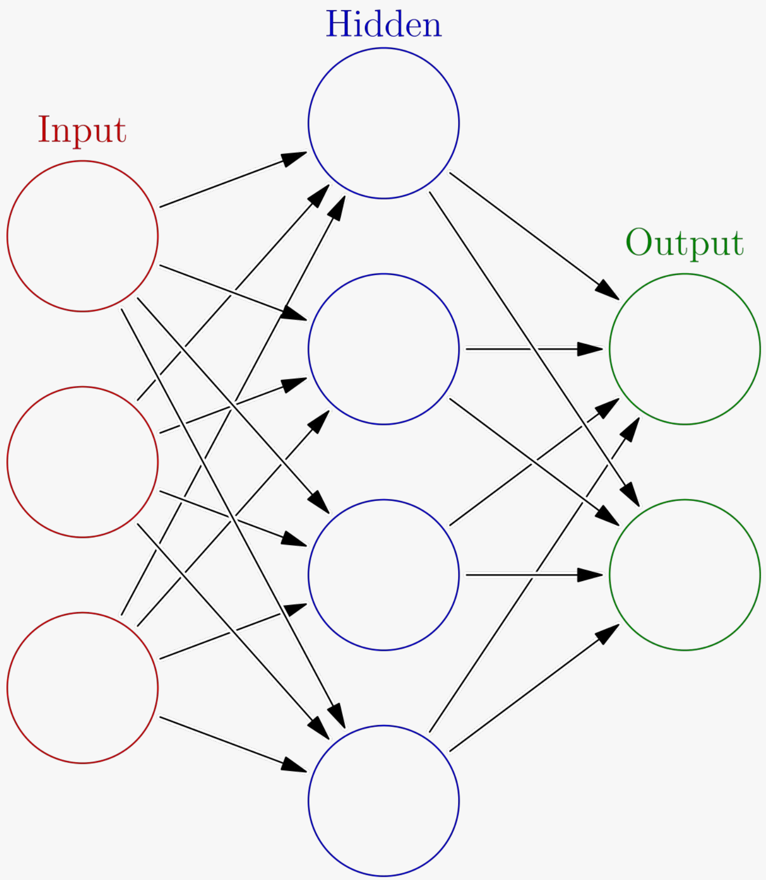
```
['term', 'grade', 'sub_grade', 'home_ownership', 'verification_status', 'issue_d', 'purpose', 'earliest_cr_line', 'initial_list
_status', 'application_type', 'address']
```

```
train.drop('grade', axis=1, inplace=True)
```

```
dummies = ['sub_grade', 'verification_status', 'purpose', 'initial_list_status',
           'application_type', 'home_ownership']
train = pd.get_dummies(train, columns=dummies, drop_first=True)
```

```
train.head()
```

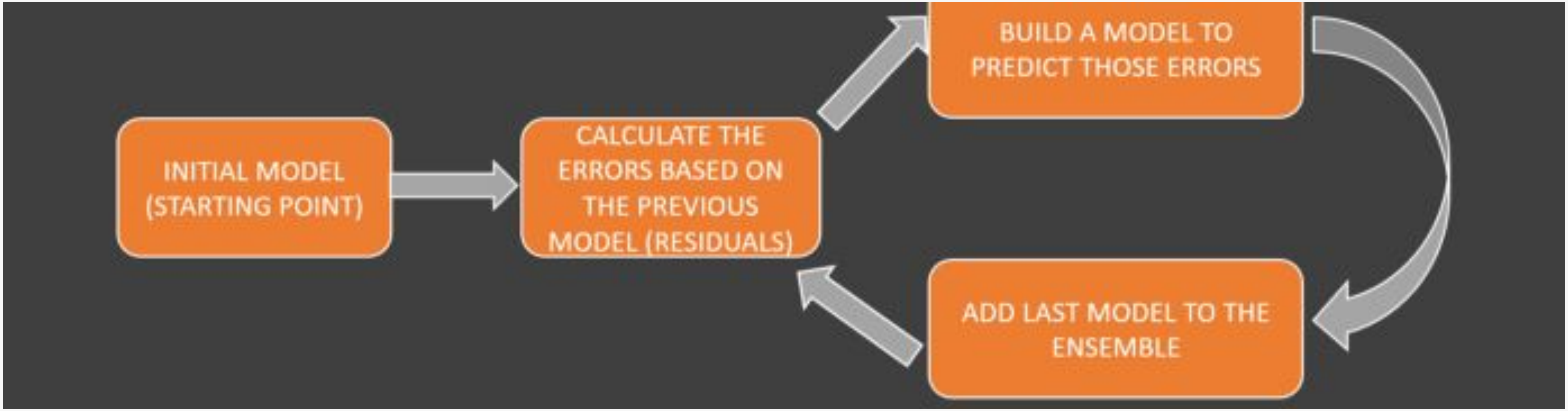
Model Building & Testing




```python
model = Sequential()

model.add(Dense(X_train.shape[1], activation='relu'))
# model.add(Dropout(0.2))

model.add(Dense(128, activation='relu'))
# model.add(Dropout(0.2))

model.add(Dense(56, activation='relu'))
# model.add(Dropout(0.2))

model.add(Dense(28, activation='relu'))
model.add(Dropout(0.2))

model.add(Dense(1, activation='sigmoid'))
```

```python
model.compile(optimizer=tf.keras.optimizers.Adam(0.001),
              loss='binary_crossentropy',
              metrics=['accuracy'])

r = model.fit(
    X_train, y_train,
    validation_data=(X_test, y_test),
    epochs=25,
    batch_size=8,
#     class_weight={0:w_n, 1:w_p}
)
```

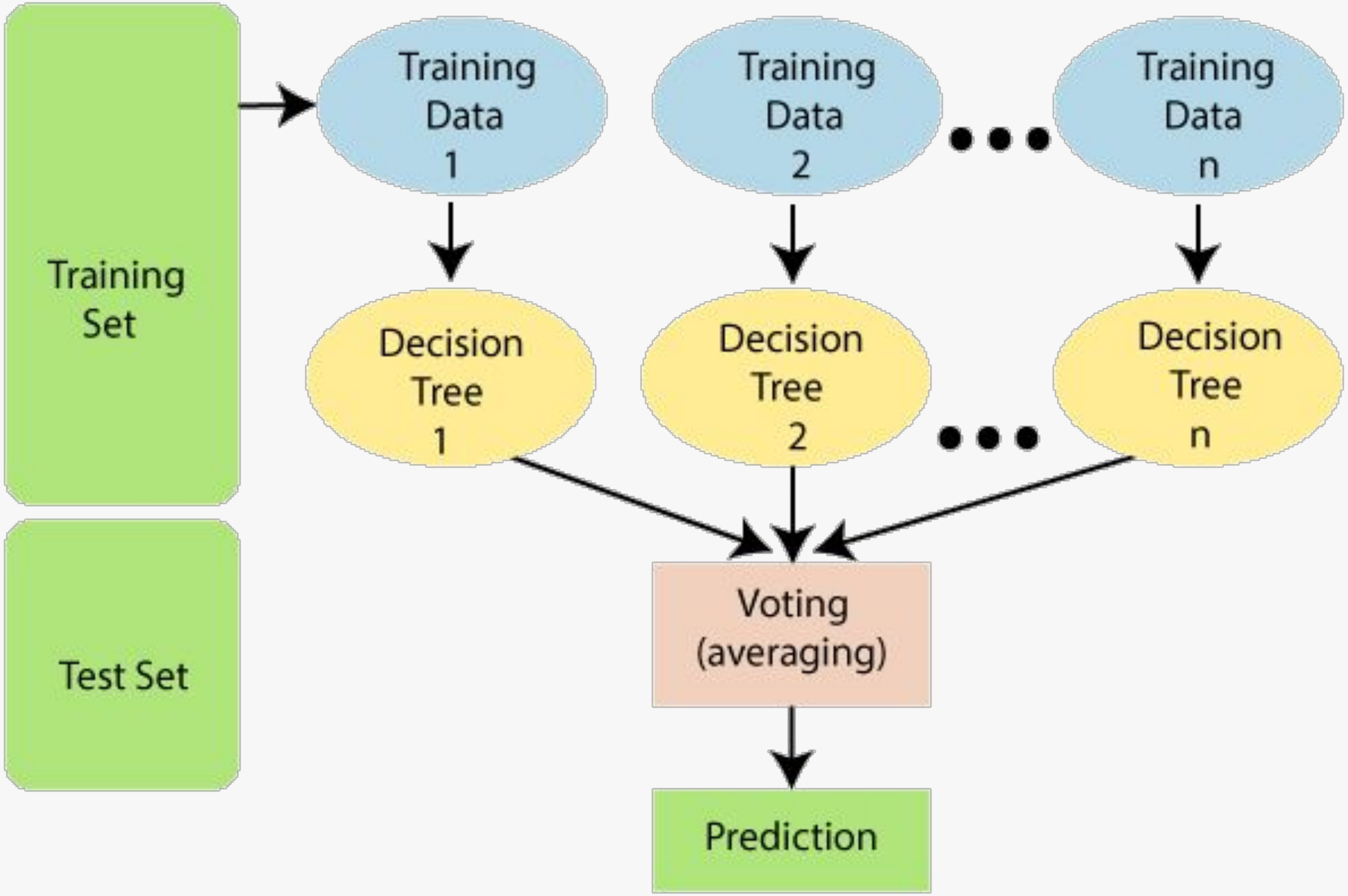
Model Building & Testing



Hyper Parameters

Booster: GBTree
Step Size Shrinkage: 0.3
Alpha Alias: Learning Rate

Model Building & Testing



Comparing Models

ANNs

```
Train Result:
================================================
Accuracy Score: 88.92%
_______________________________________________
CLASSIFICATION REPORT:
                 0.0       1.0  accuracy  macro avg  weighted avg
precision       0.89      0.89      0.89       0.89          0.89
recall          0.99      0.50      0.89       0.74          0.89
f1-score        0.93      0.64      0.89       0.79          0.88
support    222387.00  54266.00      0.89  276653.00     276653.00
_______________________________________________
Confusion Matrix:
 [[219075   3312]
 [ 27339  26927]]

Test Result:
================================================
Accuracy Score: 88.75%
_______________________________________________
```

XGBoost

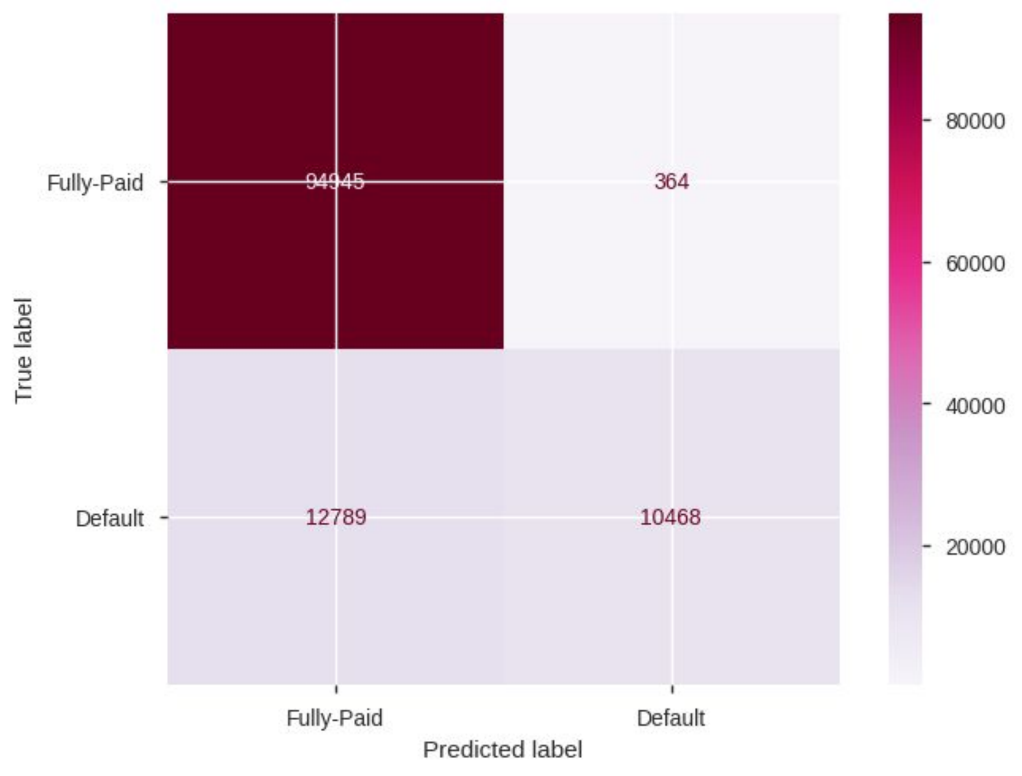
```
Train Result:
================================================
Accuracy Score: 89.58%
_______________________________________________
CLASSIFICATION REPORT:
                 0.0       1.0  accuracy  macro avg  weighted avg
precision       0.89      0.94      0.90       0.92          0.90
recall          0.99      0.50      0.90       0.75          0.90
f1-score        0.94      0.65      0.90       0.80          0.88
support    222387.00  54266.00      0.90  276653.00     276653.00
_______________________________________________
Confusion Matrix:
 [[220726   1661]
 [ 27161  27105]]

Test Result:
================================================
Accuracy Score: 88.92%
```

Random Forest

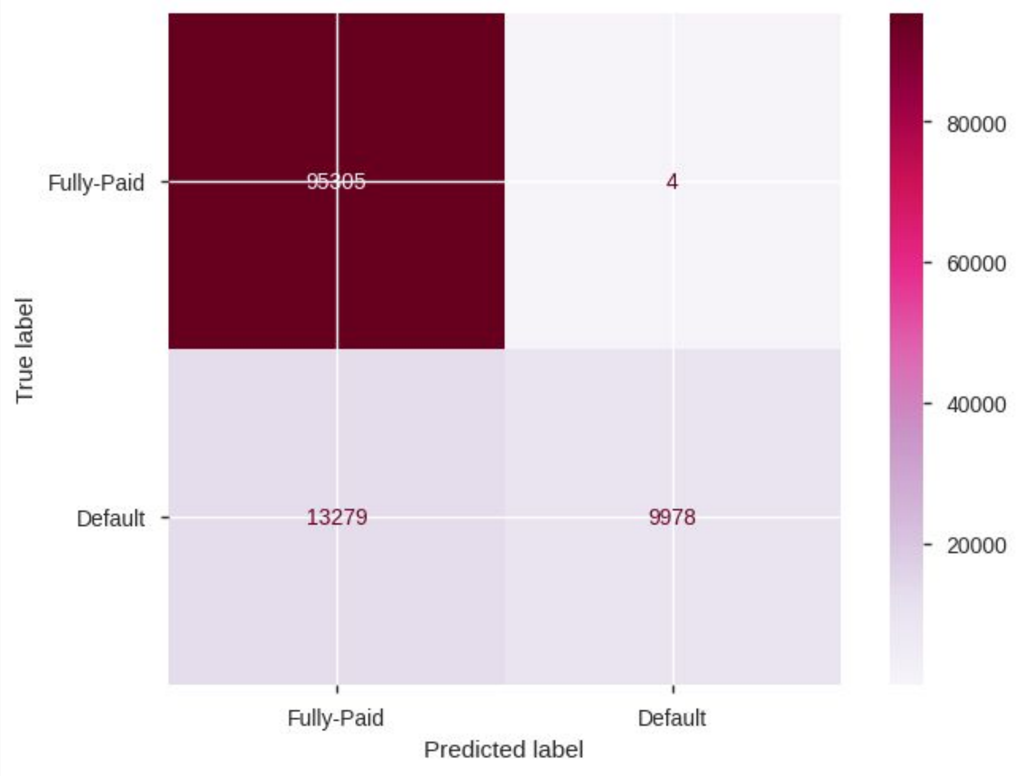
```
Train Result:
================================================
Accuracy Score: 100.00%
_______________________________________________
CLASSIFICATION REPORT:
                 0.0       1.0  accuracy  macro avg  weighted avg
precision       1.00      1.00      1.00       1.00          1.00
recall          1.00      1.00      1.00       1.00          1.00
f1-score        1.00      1.00      1.00       1.00          1.00
support    222387.00  54266.00      1.00  276653.00     276653.00
_______________________________________________
Confusion Matrix:
 [[222387      0]
 [     2  54264]]

Test Result:
================================================
Accuracy Score: 88.88%
```

Comparing Models

Confusion Matrix for Random Forest



Confusion Matrix for XGBoost



Research Papers

1. Predicting Loan Defaults using Machine Learning Techniques | Abhishek Bhagat
 http://scholarworks.csun.edu/bitstream/handle/10211.3/203343/Bhagat-Abhishek-thesis-2018.pdf?sequence=1

Recommendations & Way Forward

1. More data points for loan defaulters (false positives) will help us classify them better

2. Further categorization amongst the loan defaulters will help us minimize the false negative which in turn will help us maximize profit

3. As per the linear measures of correlation between the predictors and the response, the most important variables for predicting loan defaulters are total mortgage accounts, annual income, installments and loan amount

Thank You!