LendingClub

Defaulter Prediction

LendingClub

Business Understanding

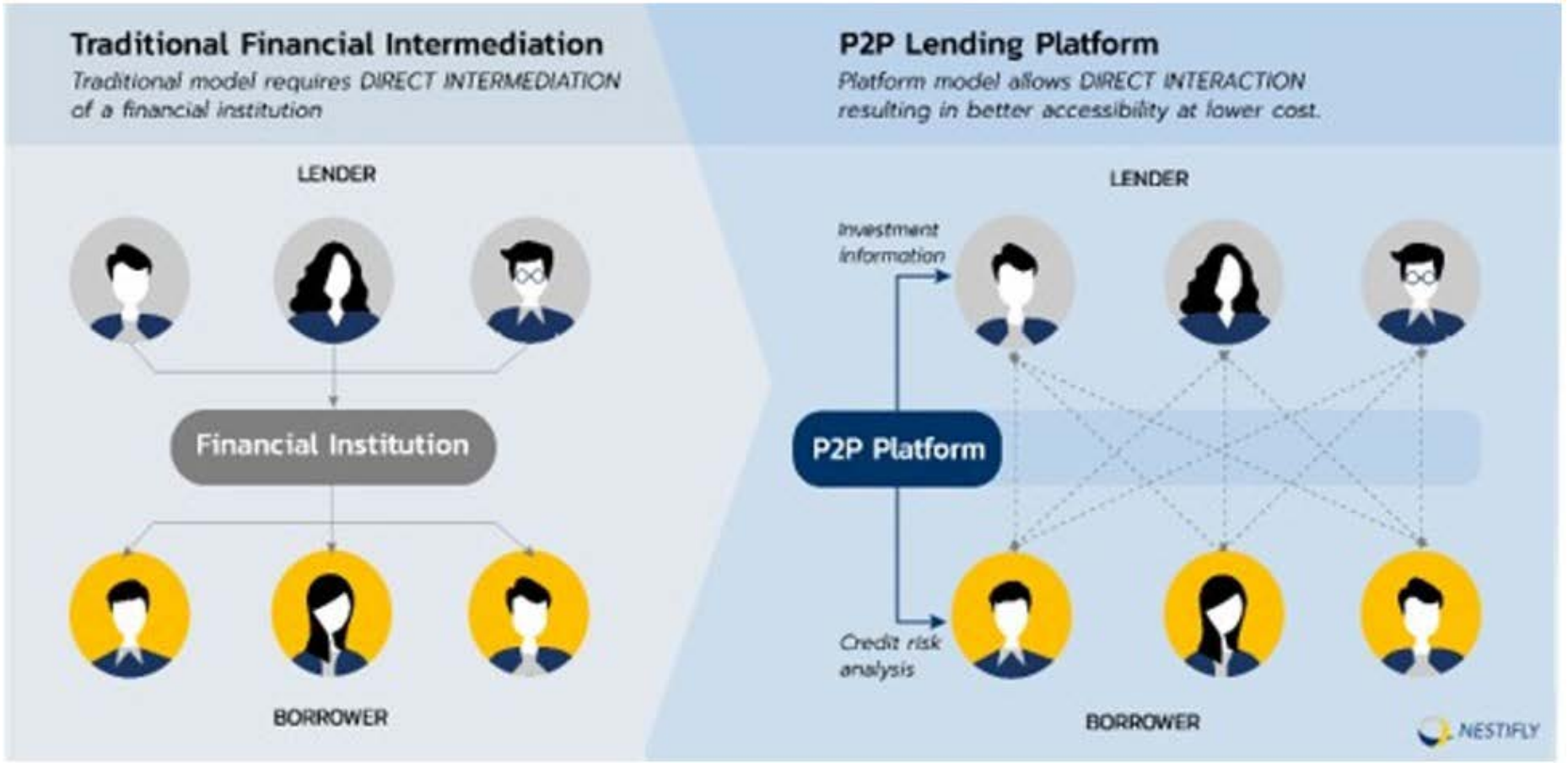
- เป็นบริษัท Fintech ขนาดใหญ่
  ให้บริการเกี่ยวกับสินเชื่อ / กู้ยืมออนไลน์
- ไม่ผ่านตัวกลางอย่างสถาบันการเงิน
- เป็นการกู้ลักษณะ Peer-to-Peer Lending (P2P)
- มีการประเมินความเสี่ยงของผู้กู้
- มีรายได้จากค่าธรรมเนียมในการทำธุรกรรมแต่ละครั้ง

Source : [Fair Finance Thailand | Facebook]

LendingClub

Source : [Peer-to-Peer Lending (P2P) : ทางเลือกการขอสินเชื่อและการลงทุนรูปแบบใหม่ที่ สะดวก รวดเร็ว ไม่ต้องใช้สลิปเงินเดือน โดย Share Loan by NestiFly Bualuang Knowledge Sharing]

LendingClub



NestiFly (blockdit.com)

LendingClub

Project Objectives

- ต้องการฝึกฝนและพัฒนาการทำความเข้าใจและสำรวจข้อมูลด้วยมุมมองต่าง ๆ
- ต้องการเข้าใจวิธีการเตรียมข้อมูลในรูปแบบต่าง ๆ และวิธีการ Coding
- ต้องการฝึกฝน Coding ในการทำ Machine Learning และรู้จักกับ Model ใหม่ๆ ที่ไม่เคยศึกษา
- ต้องการ Case Study ที่เกี่ยวกับมุมมองการเงิน

Business Objectives

- ต้องการทำความเข้าใจตัวแปรหรือปัจจัยที่มีผลต่อการผิดนัดชำระหนี้ด้วยการ Exploratory Data Analysis
- การให้กู้ยืมเงินเป็นการลงทุนที่มีความเสี่ยง ดังนั้นเมื่อเกิดการผิดนัดชำระหนี้ จะส่งผลเสียกับผู้ให้กู้หรือนักลงทุน หมายความว่าหากระบุ Defaulter ได้จะส่งผลให้ลดความสูญเสียในรูปตัวเงินจากการผิดนัดชำระหนี้ได้ (Change-off / Default)
- ต้องการโมเดลที่ช่วยกรองผู้ที่มี “โอกาสผิดนัดชำระหนี้” เบื้องต้น เพื่อให้นักลงทุนใช้ในการพิจารณาเลือกผู้กู้ที่เหมาะสมแก่การลงทุนและยังช่วยลดความเสี่ยงได้ระดับนึง

LendingClub

PROJECT PROCESS

1

Exploratory Data Analysis

(Data Understanding)

2

Data Preparation

3

Modeling

4

Evaluation

LendingClub

PROJECT PROCESS

1

2

Data Preparation

1. Handling Numerical Variable
2. Handling Categorical Variable
3. Handling Missing Values
4. Drop Variables
5. Check for duplicate rows
6. Train Test Split
7. Outlier Handling for Train data
8. X_train y_train X_test y_test Split
9. Data Normalization (X_train, X_test)
10. Features Selection by LightGBM
11. Encoding (One Hot Encoding)

3

4

I

Exploratory Data Analysis

(Data Understanding)

*** for target variable and features selected by LightGBM ***

LendingClub

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 396030 entries, 0 to 396029
Data columns (total 27 columns):
 #   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)
memory usage: 81.6+ MB
```

Number of data is 396,030 rows

1 row = 1 transaction & information of borrower

Data types

Categorical (Object) – 15 features

Numerical (Float64) – 12 features

Total 27 features

LendingClub

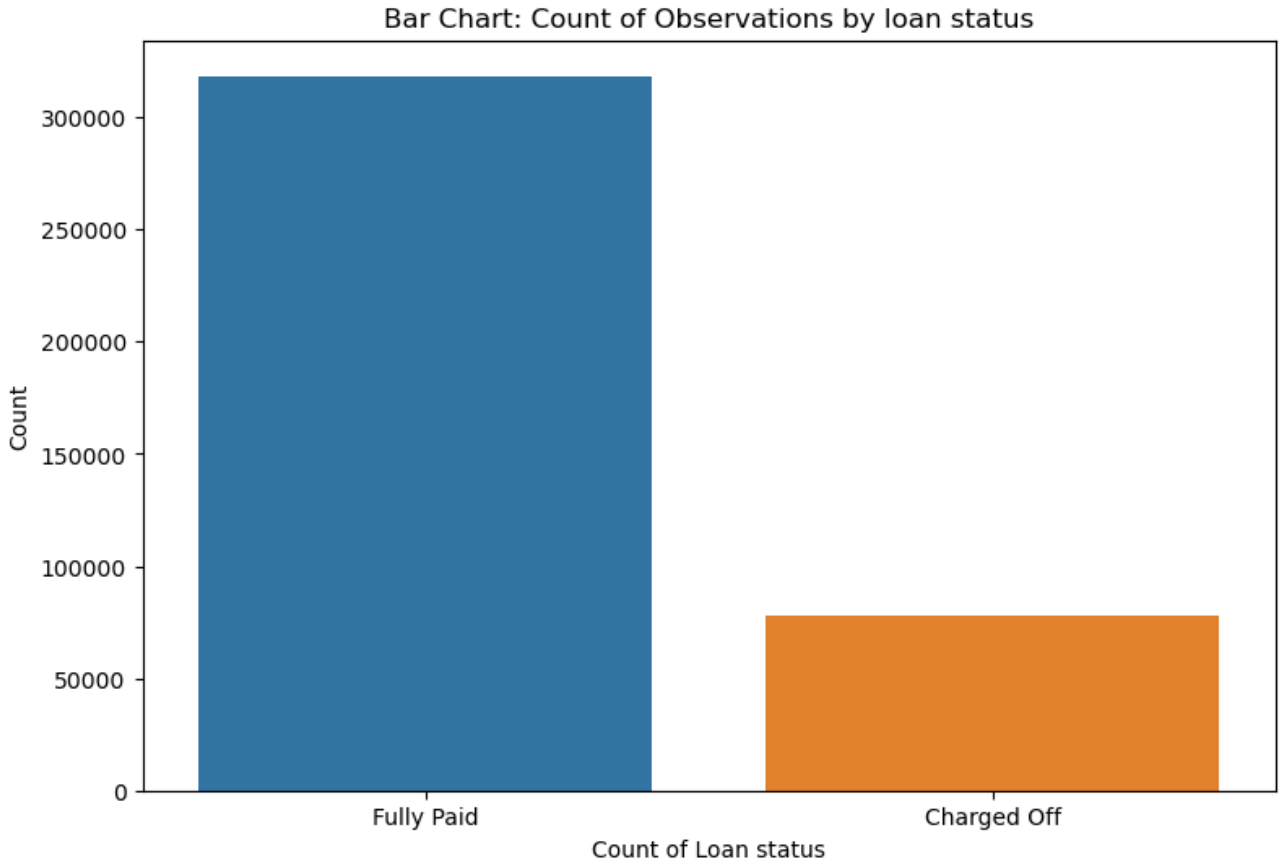
Target variable for supervised learning

loan_status

Loan status คือ สถานะของเงินกู้

- Fully Paid : 0 คือ จ่ายครบแล้ว
- Charge Off : 1 คือ ผิดนัดชำระหนี้ (อย่างน้อย 1 ครั้ง)

```
Fully Paid     318357
Charged Off     77673
Name: loan_status, dtype: int64
Fully Paid    0.80
Charged Off   0.20
Name: loan_status, dtype: float64
```



LendingClub

Features selected by LightGBM

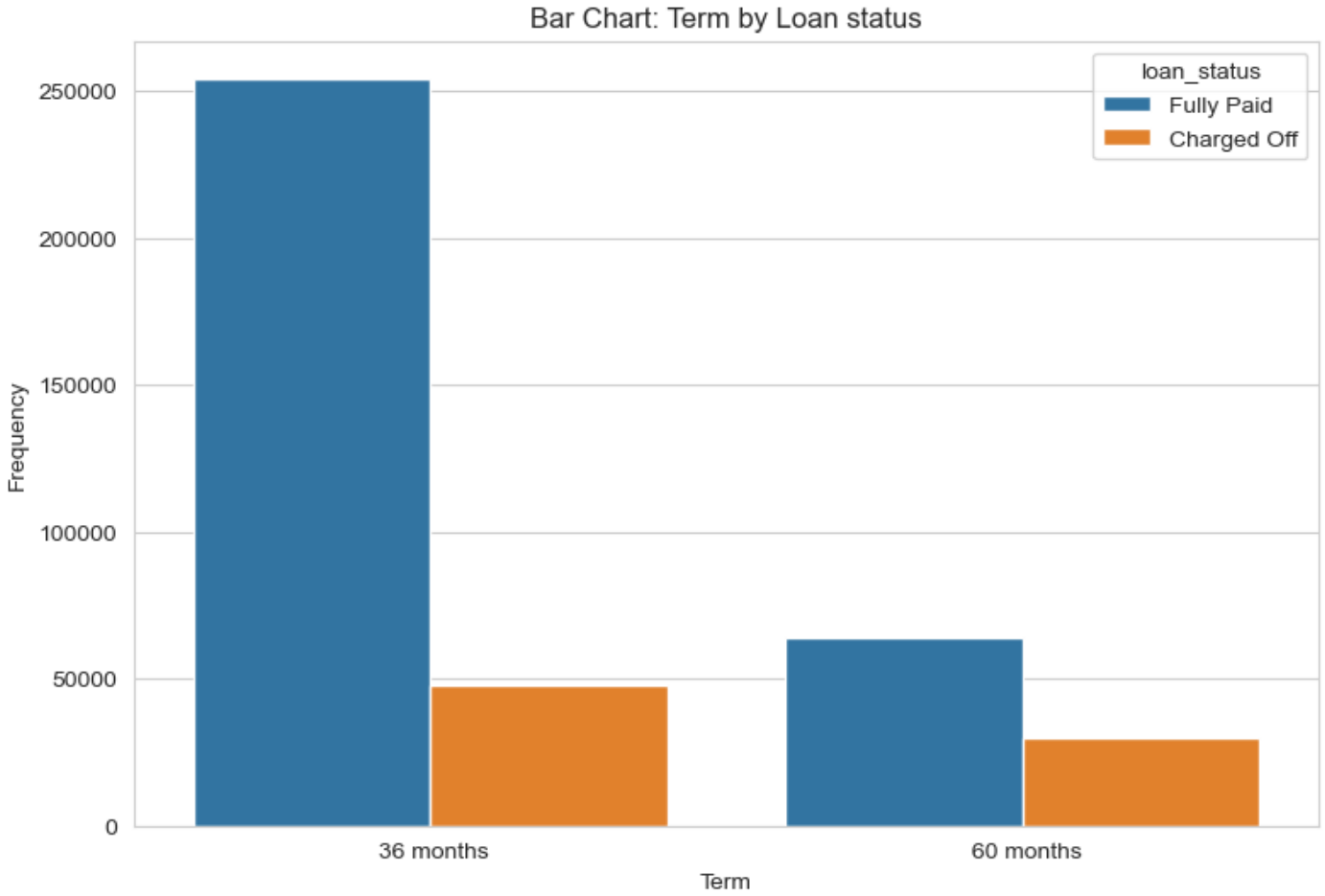
1. term
2. sub_grade
3. annual_inc
4. dti
5. mort_acc
6. home_ownership
7. int_rate
8. zip_code
9. installment

term

Term of loan คือ อายุของเงินกู้ มีอยู่ 2 ค่า ได้แก่

- 36 months (3 years) : 36
- 60 months (5 years) : 60

```
term         loan_status
 36 months   Fully Paid     0.84
             Charged Off    0.16
 60 months   Fully Paid     0.68
             Charged Off    0.32
Name: loan_status, dtype: float64
```

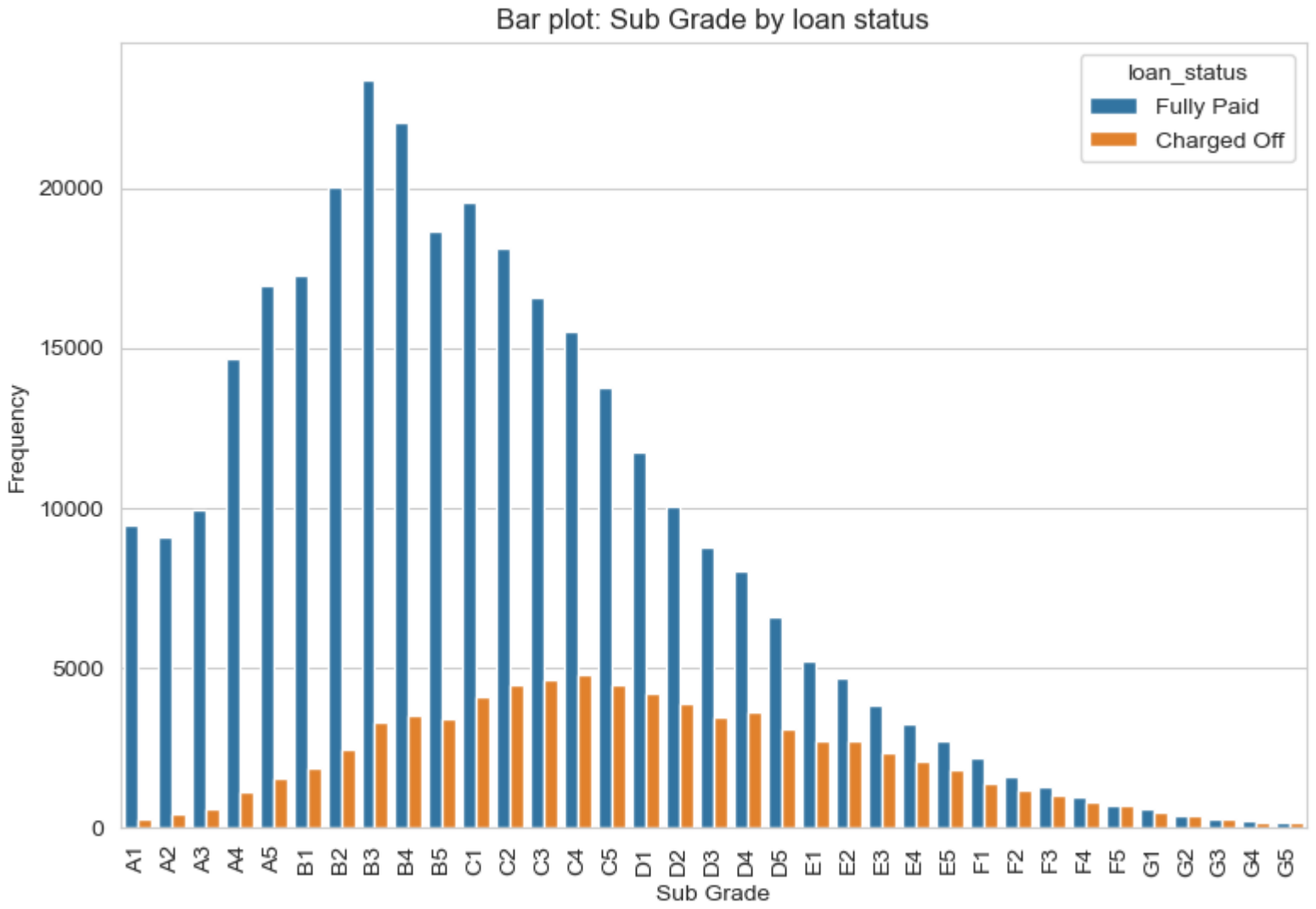


LendingClub

sub_grade

Sub Grade คือ เกรดสินเชื่อระดับย่อย
ที่ Lending Club เป็นคนกำหนด
เช่น A1, A2, A3, ..., G4, G5
(G5 เป็นเกรดระดับต่ำที่สุดและ
โอกาสการผิดนัดชำระหนี้สูงที่สุด)

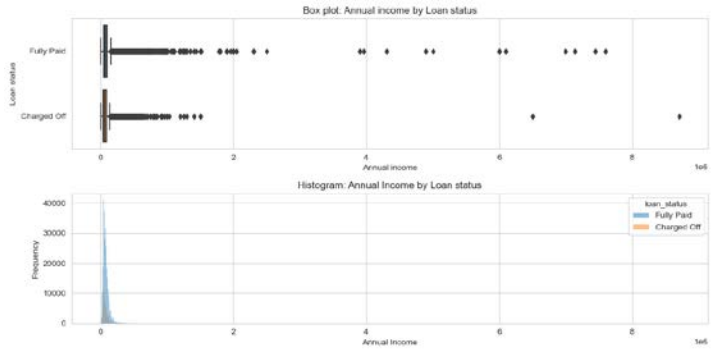
```
sub_grade  loan_status
A1         Fully Paid     0.97
           Charged Off    0.03
A2         Fully Paid     0.95
           Charged Off    0.05
A3         Fully Paid     0.94
                          ...
G3         Fully Paid     0.49
G4         Fully Paid     0.55
           Charged Off    0.45
G5         Charged Off    0.50
           Fully Paid     0.50
Name: loan_status, Length: 70, dtype: float64
```

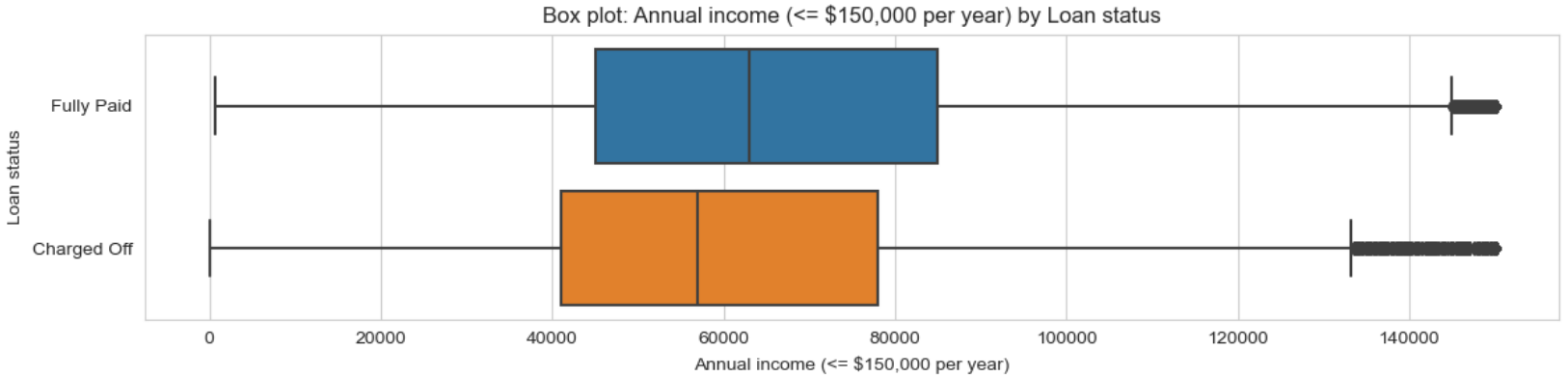


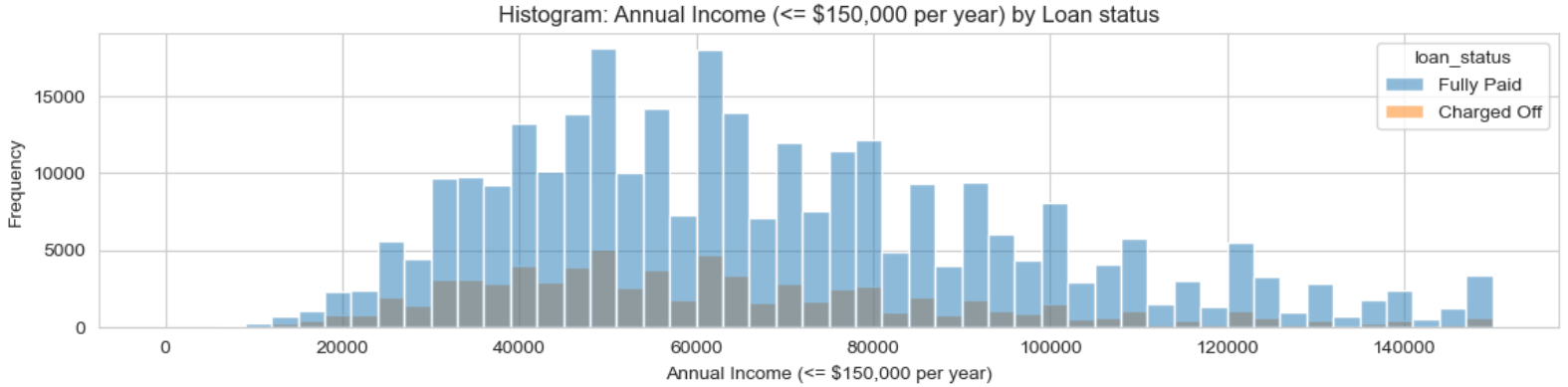
LendingClub

annual_income

Annual Income คือ
รายได้ประจำปีของผู้กู้







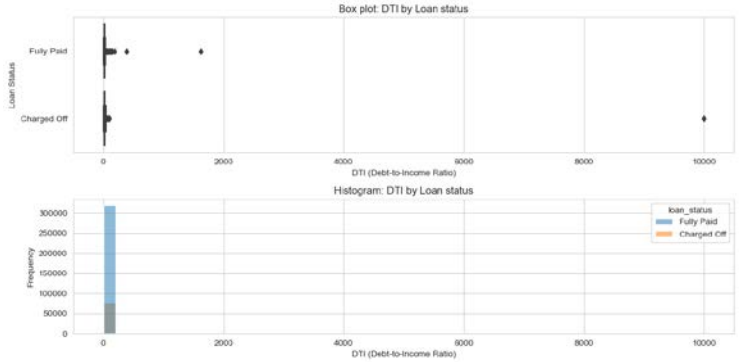
Annual Income by Loan Status

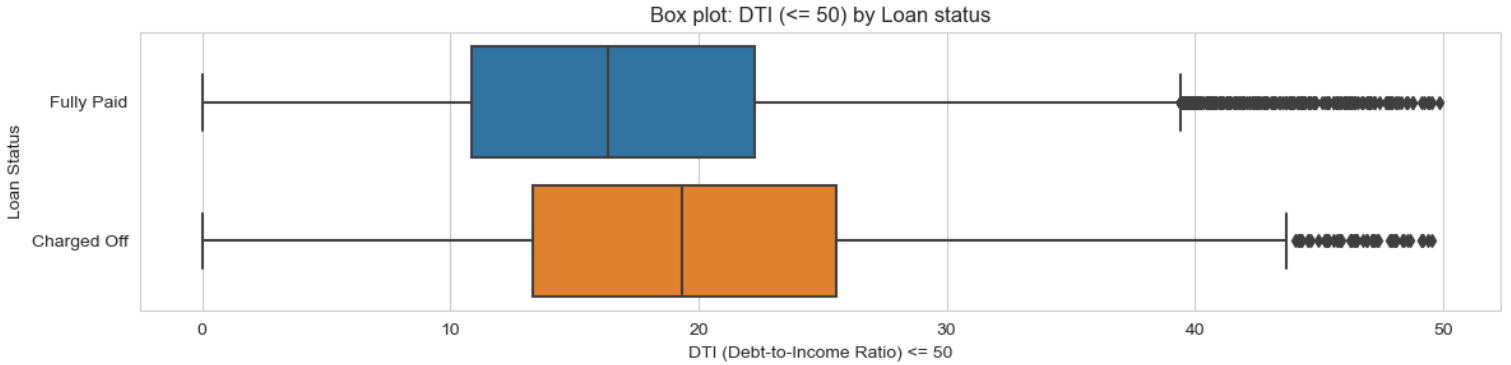
loan_status	count	mean	std	min	10%	25%	50%	75%	90%	95%	max
Charged Off	77673.00	67535.54	58303.46	0.00	31000.00	42000.00	59000.00	80000.00	110000.00	137440.00	8706582.00
Fully Paid	318357.00	75829.95	62315.99	600.00	35000.00	46050.53	65000.00	90000.00	125000.00	150000.00	7600000.00

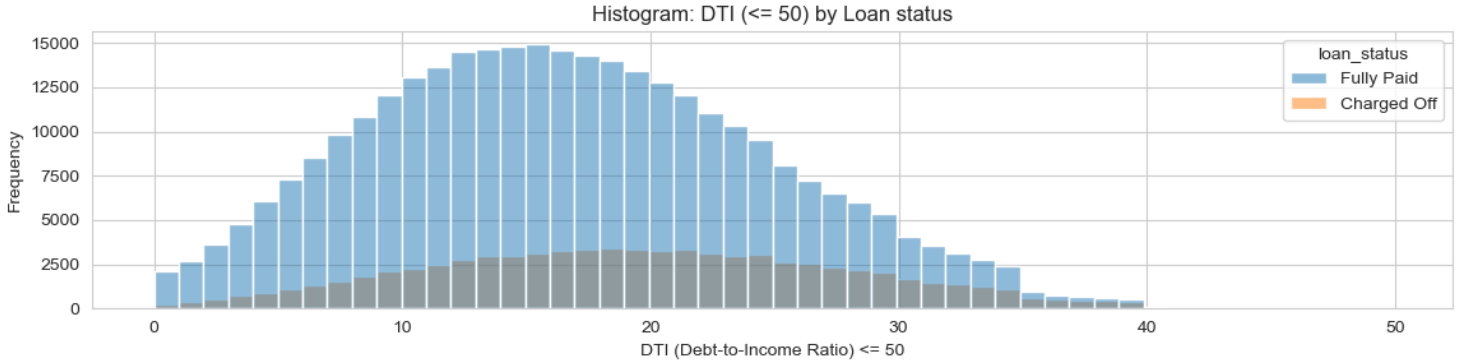
LendingClub

dti

Debt to Income Ratio คือ
หนี้สินรวมต่อเดือนหารายได้ต่อเดือน







Debt to Income by Loan Status

loan_status	count	mean	std	min	10%	25%	50%	75%	90%	95%	max
Charged Off	77673.00	19.66	36.78	0.00	8.49	13.33	19.34	25.55	30.86	33.66	9999.00
Fully Paid	318357.00	16.82	8.50	0.00	6.61	10.87	16.34	22.29	27.77	30.82	1622.00

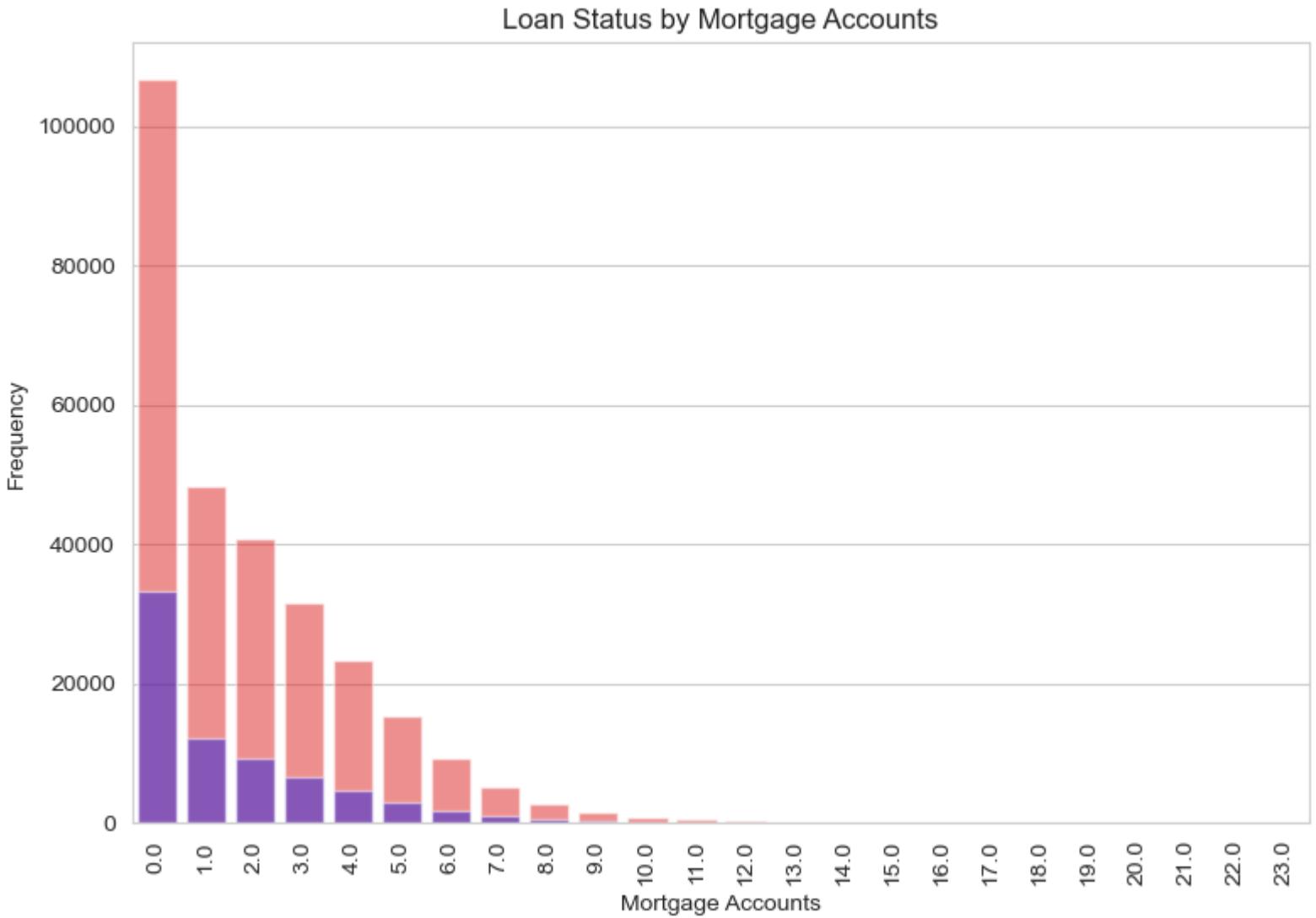
LendingClub

mort_acc

Mortgage Account คือ
จำนวนบัญชีที่ใช้อสังหาริมทรัพย์
ในการจำนองเพื่อขอกู้

```
data.groupby("mort_acc")["loan_status"].value_counts(normalize=True).head(16)
```

```
mort_acc  loan_status
0.00      Fully Paid     0.76
          Charged Off    0.24
1.00      Fully Paid     0.80
          Charged Off    0.20
2.00      Fully Paid     0.81
          Charged Off    0.19
3.00      Fully Paid     0.83
          Charged Off    0.17
4.00      Fully Paid     0.83
          Charged Off    0.17
5.00      Fully Paid     0.84
          Charged Off    0.16
6.00      Fully Paid     0.84
          Charged Off    0.16
7.00      Fully Paid     0.85
          Charged Off    0.15
Name: loan_status, dtype: float64
```



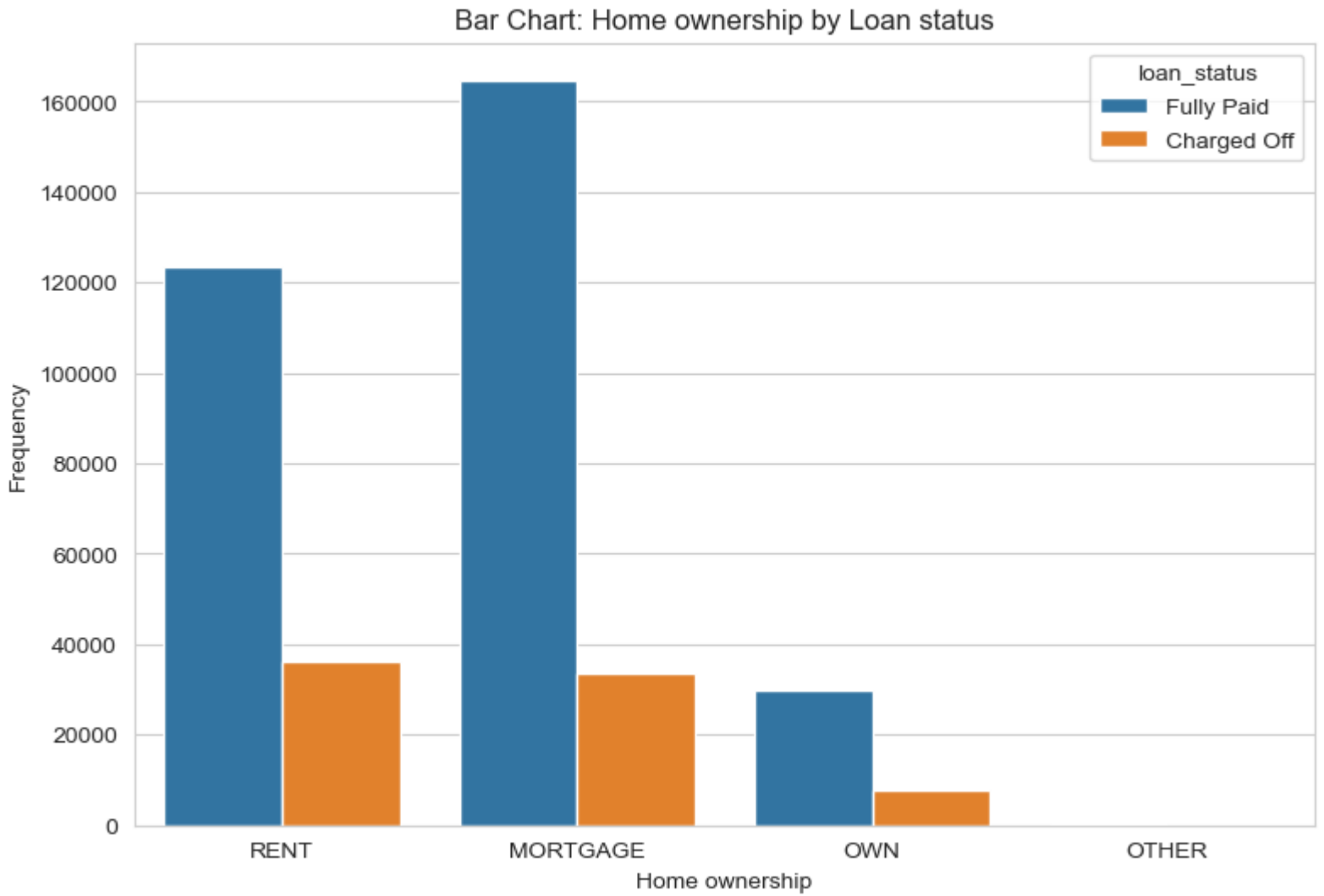
LendingClub

home_ownership

Home Ownership คือ สถานะความเป็นเจ้าของบ้านที่ผู้กู้ระบุระหว่างการลงทะเบียนหรือได้รับจากรายงานเครดิต

- เช่า (RENT)
- เป็นเจ้าของ (OWN)
- จำนอง (MORTGAGE)
- อื่นๆ (OTHER)

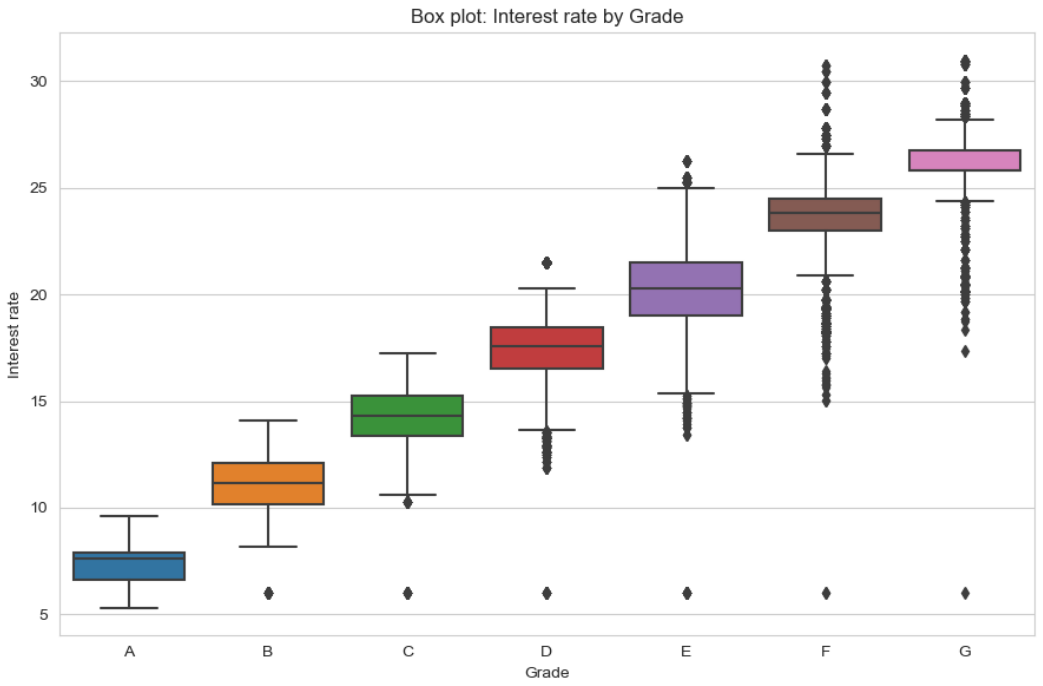
```
home_ownership  loan_status
MORTGAGE        Fully Paid     0.83
                Charged Off    0.17
OTHER           Fully Paid     0.84
                Charged Off    0.16
OWN             Fully Paid     0.79
                Charged Off    0.21
RENT            Fully Paid     0.77
                Charged Off    0.23
Name: loan_status, dtype: float64
```

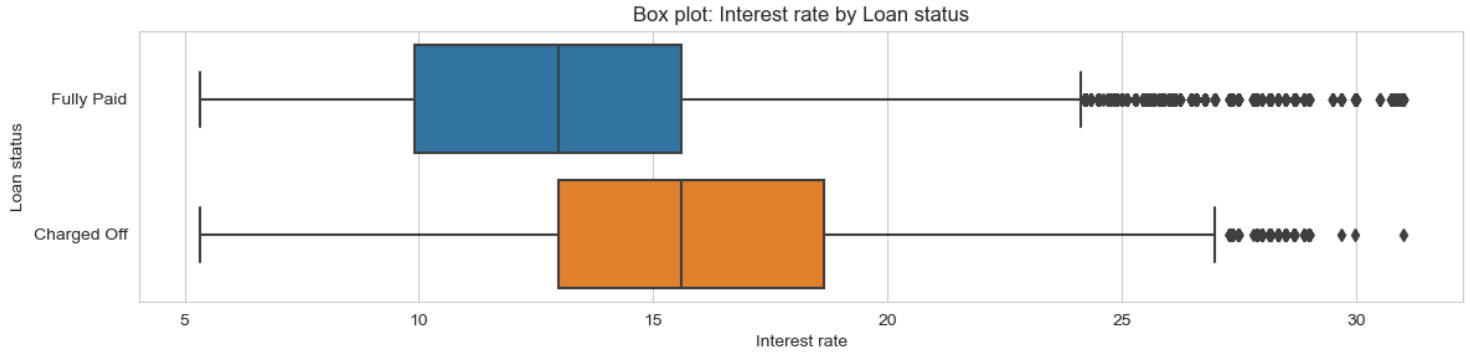


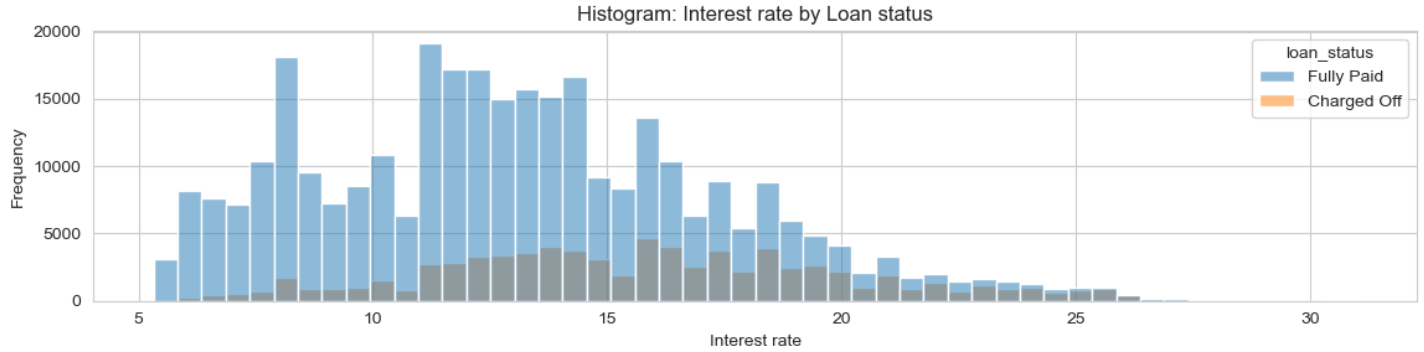
LendingClub

int_rate

Interest Rate คือ
อัตราดอกเบี้ยเงินกู้







Interest rate by Loan Status

loan_status	count	mean	std	min	10%	25%	50%	75%	90%	95%	max
Charged Off	77673.00	15.88	4.39	5.32	10.25	12.99	15.61	18.64	21.99	23.99	30.99
Fully Paid	318357.00	13.09	4.32	5.32	7.62	9.91	12.99	15.61	18.85	20.99	30.99

LendingClub

zip_code

Zip code คือ รหัสไปรษณีย์ที่ผู้กู้ระบุในทะเบียนบ้าน (เป็นตัวแปรที่ generate ขึ้นมาใหม่จาก address)

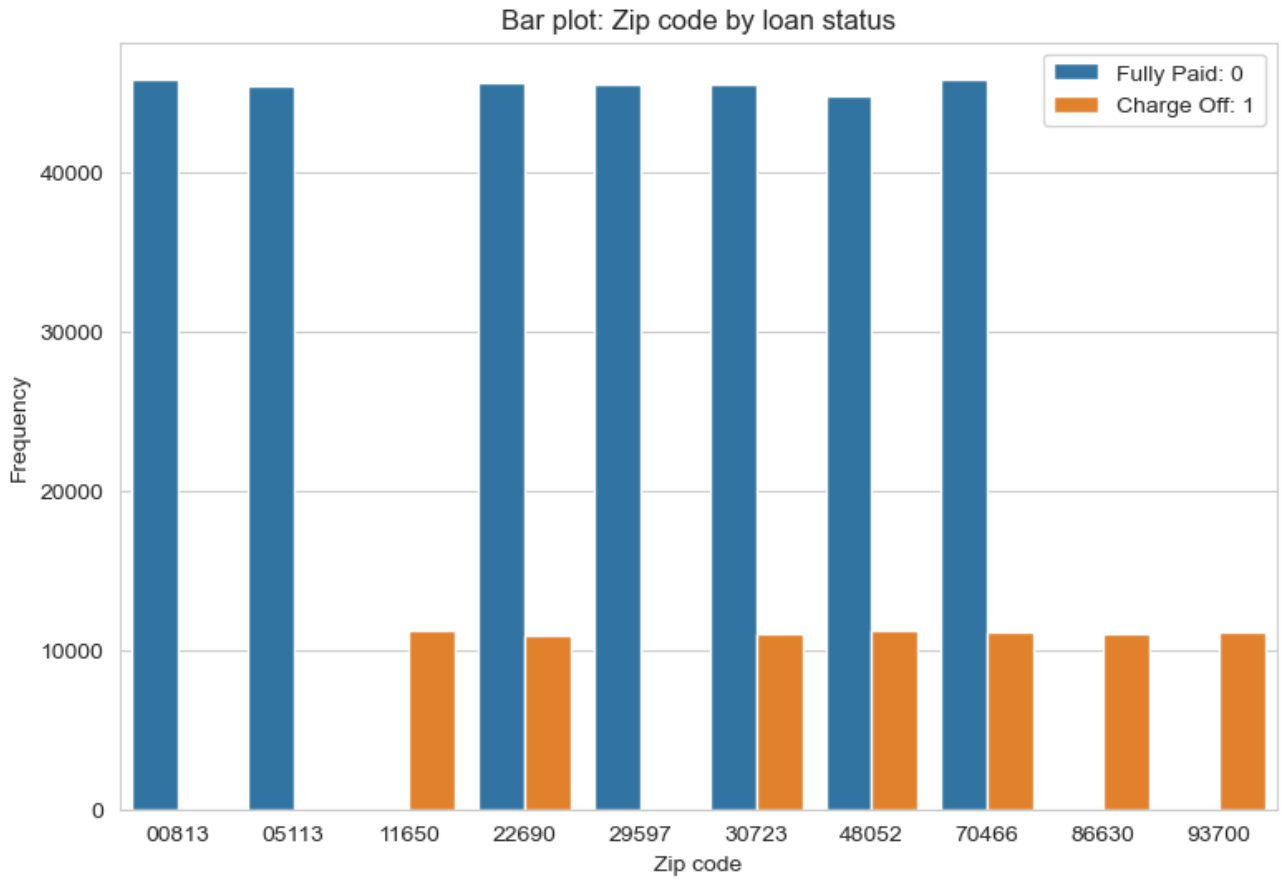
```
data.address.head()
```

```
0       0174 Michelle Gateway\r\nMendozaberg, OK 22690
1    1076 Carney Fort Apt. 347\r\nLoganmouth, SD 05113
2    87025 Mark Dale Apt. 269\r\nNew Sabrina, WV 05113
3             823 Reid Ford\r\nDelacruzside, MA 00813
4              679 Luna Roads\r\nGreggshire, VA 11650
Name: address, dtype: object
```

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

```
data.groupby("zip_code")["loan_status"].value_counts(normalize=True).sort_index(ascending=True)
```

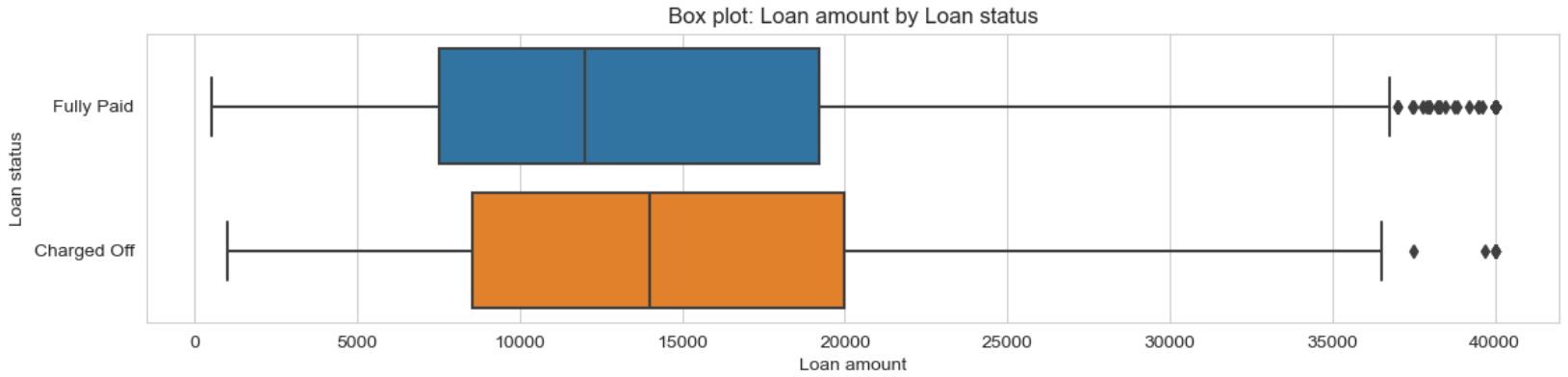
```
zip_code  loan_status
00813     0            1.00
05113     0            1.00
11650     1            1.00
22690     0            0.81
          1            0.19
29597     0            1.00
30723     0            0.81
          1            0.19
48052     0            0.80
          1            0.20
70466     0            0.80
          1            0.20
86630     1            1.00
93700     1            1.00
Name: loan_status, dtype: float64
```

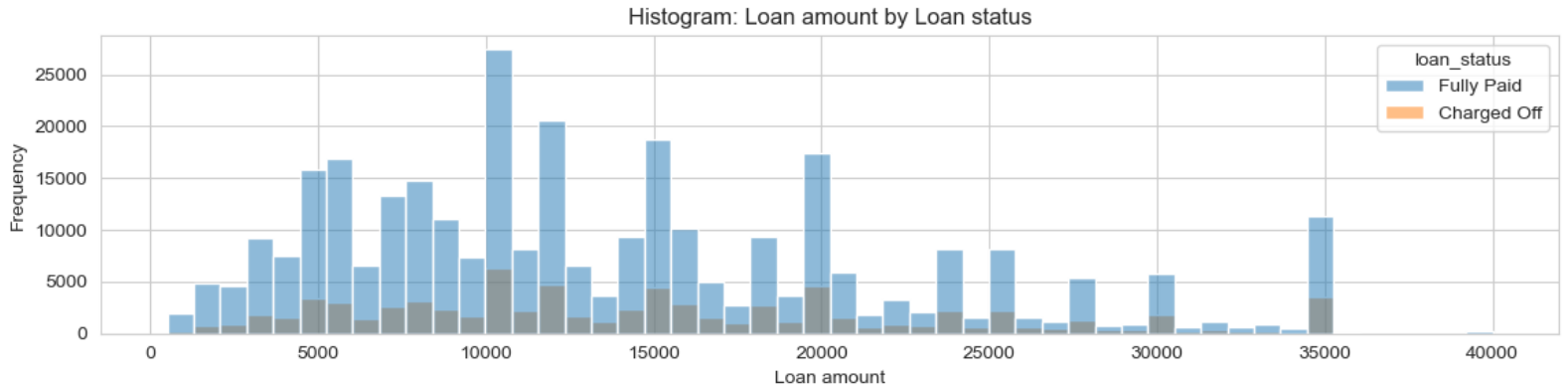


LendingClub

Installment

คือ ค่างวดเงินผ่อนที่ชำระในแต่ละงวด





Installment by Loan Status

loan_status	count	mean	std	min	10%	25%	50%	75%	90%	95%	max
Charged Off	77673.00	452.70	249.10	21.62	172.09	274.86	399.06	585.67	806.61	944.78	1527.00
Fully Paid	318357.00	426.76	250.86	16.08	155.93	244.46	369.51	562.89	780.52	918.55	1533.81

2

Data Preparation

Data Cleansing
Data Transformation

LendingClub

Handling Numerical Variables

```
[70]: def pub_rec(number):
          if number == 0.0:
              return 0
          else:
              return 1

      def mort_acc(number):
          if number == 0.0:
              return 0
          elif number >= 1.0:
              return 1
          else:
              return number

      def pub_rec_bankruptcies(number):
          if number == 0.0:
              return 0
          elif number >= 1.0:
              return 1
          else:
              return number

[71]: data['pub_rec'] = data['pub_rec'].apply(pub_rec)
      data['mort_acc'] = data['mort_acc'].apply(mort_acc)
      data['pub_rec_bankruptcies'] = data['pub_rec_bankruptcies']
```

```
mort_acc  loan_status
0.00      Fully Paid     0.76
          Charged Off    0.24
1.00      Fully Paid     0.82
          Charged Off    0.18
Name: loan_status, dtype: float64
```

LendingClub

Handling Categorical Variables

loan_status

```
[73]: data["loan_status"].unique()

[73]: array(['Fully Paid', 'Charged Off'], dtype=object)

[74]: data["loan_status"] = data["loan_status"].map({"Charged Off":1, "Fully Paid":0})
      data["loan_status"].sample(10)
```

term

```
[75]: data["term"].unique()

[75]: array([' 36 months', ' 60 months'], dtype=object)

[76]: data["term"] = data["term"].map({' 36 months': 36, ' 60 months': 60})
      data["term"].sample(10)
```

address and generate new variable is zip_code

```
[78]: data.address.head()

[78]: 0       0174 Michelle Gateway\r\nMendozaberg, OK 22690
      1    1076 Carney Fort Apt. 347\r\nLoganmouth, SD 05113
      2    87025 Mark Dale Apt. 269\r\nNew Sabrina, WV 05113
      3              823 Reid Ford\r\nDelacruzside, MA 00813
      4               679 Luna Roads\r\nGreggshire, VA 11650
      Name: address, dtype: object

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

LendingClub

Change Data type from objects to category for Feature Selections

```python
def convert_obj_columns_to_category(df):
    for c in df.columns:
        col_type = df[c].dtype
        if col_type == 'object' or col_type.name == 'category':
            df[c] = df[c].astype('category')
    return df

convert_obj_columns_to_category(data)
```

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 396030 entries, 0 to 396029
Data columns (total 22 columns):
 #   Column                Non-Null Count   Dtype
---  ------                --------------   -----
 0   loan_amnt             396030 non-null  float64
 1   term                  396030 non-null  int64
 2   int_rate              396030 non-null  float64
 3   installment           396030 non-null  float64
 4   sub_grade             396030 non-null  category
 5   home_ownership        396030 non-null  category
 6   annual_inc            396030 non-null  float64
 7   verification_status   396030 non-null  category
 8   loan_status           396030 non-null  int64
 9   purpose               396030 non-null  category
 10  dti                   396030 non-null  float64
 11  earliest_cr_line      396030 non-null  int64
 12  open_acc              396030 non-null  float64
 13  pub_rec               396030 non-null  int64
 14  revol_bal             396030 non-null  float64
 15  revol_util            396030 non-null  float64
 16  total_acc             396030 non-null  float64
 17  initial_list_status   396030 non-null  category
 18  application_type      396030 non-null  category
 19  mort_acc              396030 non-null  float64
 20  pub_rec_bankruptcies  396030 non-null  float64
 21  zip_code              396030 non-null  category
dtypes: category(7), float64(11), int64(4)
memory usage: 51.0 MB
```

LendingClub

Handling Missing Value

```
# Calculate the fractions of missing values
missing_sumations = pd.DataFrame(data.isnull().sum(), columns=["Missing Sum"]) \
                    .sort_values(by="Missing Sum",ascending=False)

# Calculate the fractions of missing values
missing_fractions = pd.DataFrame(data.isnull().sum(), columns=['Missing Percentage']) \
                    .apply(lambda x: round((x/data.shape[0])*100, 4), axis=1) \
                    .sort_values(by="Missing Percentage" ,ascending=False)

missing_tb = pd.merge(missing_sumations, missing_fractions, left_index=True, right_index=True)
missing_tb.head(10)
```

	Missing Sum	Missing Percentage
mort_acc	37795	9.54
emp_title	22927	5.79
emp_length	18301	4.62
title	1755	0.44
pub_rec_bankruptcies	535	0.14
revol_util	276	0.07

```
[86]: data["mort_acc"] = data["mort_acc"].fillna(data["mort_acc"].median())
      data["revol_util"] = data["revol_util"].fillna(data["revol_util"].median())
      data["pub_rec_bankruptcies"] = data["pub_rec_bankruptcies"].fillna(data["pub_rec_bankruptcies"].median())
```

LendingClub

Drop Variables

ก่อนเลือก features อีกครั้ง

```
data[['term', 'grade', 'sub_grade', 'emp_title',
      'emp_length', 'home_ownership', 'verification_status',
      'purpose', 'title', 'initial_list_status', 'application_type',
      'address']].nunique().sort_values(ascending=False)
```

```
address              393700
emp_title            173105
title                 41327
sub_grade                35
purpose                  14
emp_length               11
```

```
drop_list = ["grade", "emp_title", "title", "address", "issue_d", "emp_length"]
data.drop(drop_list, axis=1, inplace=True)
```

```
emp_length  loan_status
1 year      0              0.80
            1              0.20
10+ years   0              0.82
            1              0.18
2 years     0              0.81
            1              0.19
3 years     0              0.80
            1              0.20
4 years     0              0.81
            1              0.19
5 years     0              0.81
            1              0.19
6 years     0              0.81
            1              0.19
7 years     0              0.81
            1              0.19
8 years     0              0.80
            1              0.20
9 years     0              0.80
            1              0.20
< 1 year    0              0.79
            1              0.21
Name: loan_status, dtype: float64
```

เหตุผลที่ drop variables ดังกล่าว

- **grade** - ซ้ำกับตัวแปรอย่าง sub_grade จึงเลือกตัวที่แสดงรายละเอียดได้มากกว่าอย่าง sub_grade
- **emp_title** – เนื่องจากเป็นตัวแปรที่มีค่าเป็นชื่ออาชีพต่าง ๆ จึงมี unique value ถึง 173,105 ตัว
- **title** - แสดงรายละเอียดเกี่ยวกับว่ากู้เงินไปทำอะไร จึงมี unique value ถึง 41,327 ตัว และมีตัวแปรอย่าง purpose ที่แยกเป็นประเภทเหตุผลการกู้เงินชัดเจน สามารถใช้แทนกันได้
- **address** –เกี่ยวกับที่อยู่ของผู้กู้ จึงมี unique value ถึง 393,700 ตัว
- **issue_d** – เกี่ยวกับวัน/เดือน/ปี ที่กู้เงินของ transaction ดังกล่าว *คิดว่าไม่ควรนำมาใช้ทำนาย
- **emp_length** – เกี่ยวกับระยะเวลาการทำงานของผู้กู้ ทุกระยะเวลามีค่าแทบจะเหมือนกันจึงไม่สำคัญ

LendingClub

Train Test Split

[95]:
```
# Splitting data into train and test sets
train, test = train_test_split(data, test_size=0.33, random_state=42)
print("Train shape is", train.shape)
print("Test shape is", test.shape)
```

```
Train shape is (265340, 22)
Test shape is (130690, 22)
```

Drop Outlier แค่ train data เพราะ test data จะได้คงข้อมูลจริงไว้

Outlier Handling (Outliers Detection and Removing)

[96]:
```
outlier_detec = ["loan_amnt", "installment",
                 "int_rate", "annual_inc",
                 "dti",
                 "total_acc", "open_acc",
                 "revol_bal","revol_util"]

print("Shape of Data Remaining is", train.shape)
for i in outlier_detec:
    print("berfore removing outlier of", i, "is", train.shape)
    q1 = train[i].quantile(0.25)
    q3 = train[i].quantile(0.75)
    iqr = q3 - q1
    train = train[(train[i] < (train[i].mean() + 3*train[i].std())) &
                  (train[i] > (train[i].mean() - 3*train[i].std()))]
    print("after removing outlier of", i, "is", train.shape)
    print("============================================================\n")
```

result

```
Shape of Data Remaining is (265340, 22)
berfore removing outlier of loan_amnt is (265340, 22)
after removing outlier of loan_amnt is (265212, 22)
============================================================

berfore removing outlier of installment is (265212, 22)
after removing outlier of installment is (261868, 22)
============================================================

berfore removing outlier of int_rate is (261868, 22)
after removing outlier of int_rate is (261362, 22)
============================================================

berfore removing outlier of annual_inc is (261362, 22)
after removing outlier of annual_inc is (258960, 22)
============================================================

berfore removing outlier of dti is (258960, 22)
after removing outlier of dti is (258954, 22)
============================================================

berfore removing outlier of total_acc is (258954, 22)
after removing outlier of total_acc is (256482, 22)
============================================================

berfore removing outlier of open_acc is (256482, 22)
after removing outlier of open_acc is (253486, 22)
============================================================

berfore removing outlier of revol_bal is (253486, 22)
after removing outlier of revol_bal is (250126, 22)
============================================================

berfore removing outlier of revol_util is (250126, 22)
after removing outlier of revol_util is (250119, 22)
============================================================
```

LendingClub

X_tran y_train X_test y_test Split

[97]:
```
X_train, y_train = train.drop('loan_status', axis=1), train.loan_status
X_test, y_test = test.drop('loan_status', axis=1), test.loan_status
```

Data Normalization

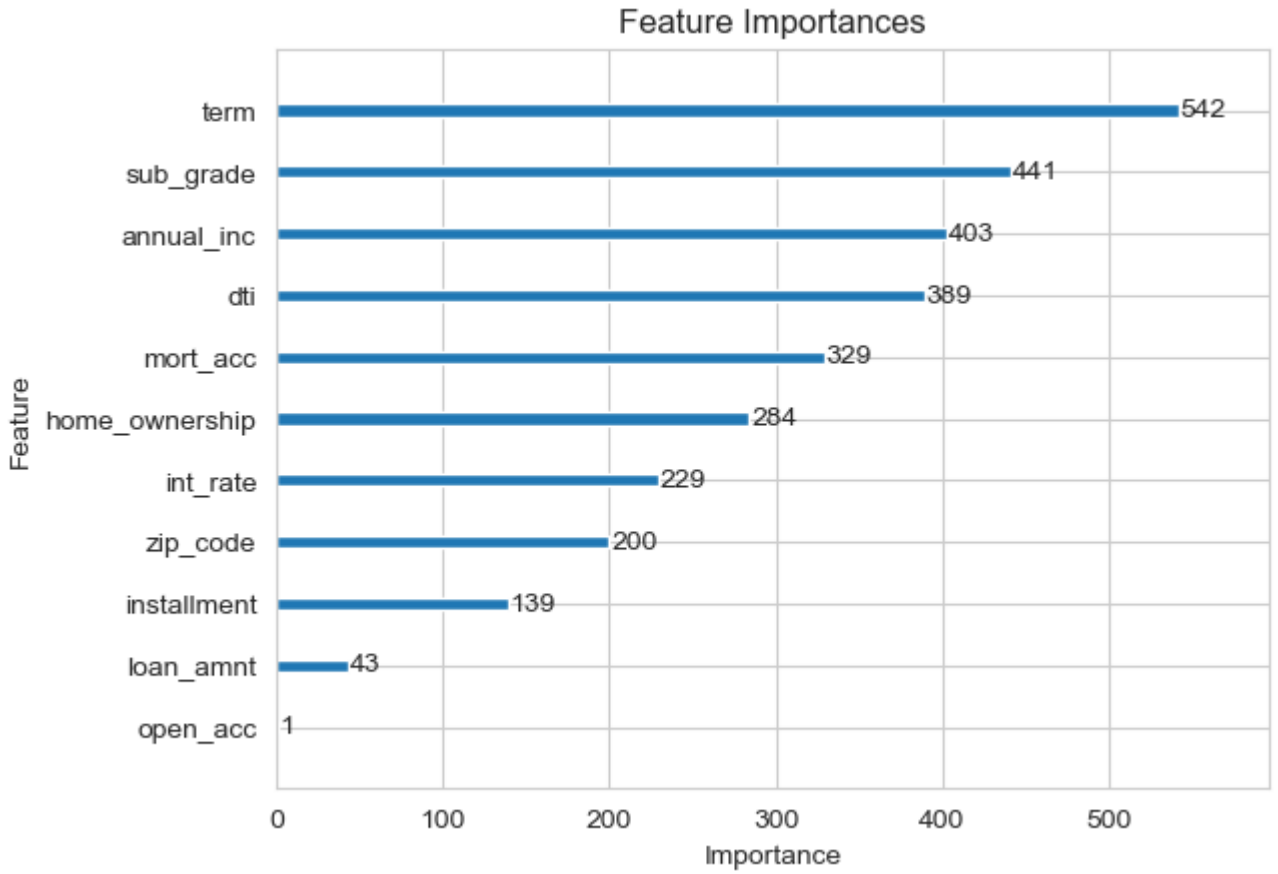
[98]:
```
scaler = MinMaxScaler()

scale_col = list([column for column in X_train.columns if
                  (data[column].dtype == "int32") |
                  (data[column].dtype == "int64") |
                  (data[column].dtype == "float64")])

X_train[scale_col] = scaler.fit_transform(X_train[scale_col])
X_test[scale_col] = scaler.transform(X_test[scale_col])
```

LendingClub

Features Selection (Features Importance)

```python
params = {
    'objective': 'binary',
    'metric': 'binary_logloss',
    'boosting_type': 'gbdt',
    'num_leaves': 31,
    'learning_rate': 0.001,
    'verbose': -1
}

# Train the model using lgb.LGBMClassifier() or lgb.train
model = lgb.LGBMClassifier(**params, random_state = 42)  # Or use lgb.train with lgb.Dataset and train_data

# Train the model
model.fit(X_train, y_train)

# Get feature importances from the model
feature_importances = model.feature_importances_

# Create a DataFrame with feature names and their corresponding importances
importance_lgb = pd.DataFrame({'feature': X_train.columns, 'importance': feature_importances})

# Sort features by importance in descending order
importance_lgb = importance_lgb.sort_values(by='importance', ascending=False)

# Plotting the feature importances using scikit-learn's plot_importance function
plt.figure(figsize=(10, 6))
lgb.plot_importance(model, max_num_features=30)  # Plotting the top 10 most important features
plt.xlabel('Importance')
plt.ylabel('Feature')
plt.title('Feature Importances')
plt.show()
```

```python
[100]: importance_lgb_list = importance_lgb["feature"][importance_lgb["importance"] > 100]
       X_train = X_train[importance_lgb_list]
       X_test = X_test[importance_lgb_list]
```



LendingClub

Encoding by One Hot Encoding (After Features Selection)

[102]:
```
# dummies = list([column for column in data.columns if (data[column].dtype == "object")])
# data = pd.get_dummies(data, columns=dummies, drop_first=True)

dummies = list([column for column in X_train.columns if (X_train[column].dtype == "category")])
print(dummies)
X_train = pd.get_dummies(X_train, columns=dummies, drop_first=True)
X_test = pd.get_dummies(X_test, columns=dummies, drop_first=True)
```

```
['sub_grade', 'home_ownership', 'zip_code']
```

```
6   sub_grade_A2          250119 non-null  uint8
7   sub_grade_A3          250119 non-null  uint8
8   sub_grade_A4          250119 non-null  uint8
9   sub_grade_A5          250119 non-null  uint8
10  sub_grade_B1          250119 non-null  uint8
11  sub_grade_B2          250119 non-null  uint8
12  sub_grade_B3          250119 non-null  uint8
13  sub_grade_B4          250119 non-null  uint8
14  sub_grade_B5          250119 non-null  uint8
15  sub_grade_C1          250119 non-null  uint8
16  sub_grade_C2          250119 non-null  uint8
17  sub_grade_C3          250119 non-null  uint8
18  sub_grade_C4          250119 non-null  uint8
19  sub_grade_C5          250119 non-null  uint8
20  sub_grade_D1          250119 non-null  uint8
21  sub_grade_D2          250119 non-null  uint8
22  sub_grade_D3          250119 non-null  uint8
23  sub_grade_D4          250119 non-null  uint8
24  sub_grade_D5          250119 non-null  uint8
25  sub_grade_E1          250119 non-null  uint8
26  sub_grade_E2          250119 non-null  uint8
27  sub_grade_E3          250119 non-null  uint8
28  sub_grade_E4          250119 non-null  uint8
29  sub_grade_E5          250119 non-null  uint8
30  sub_grade_F1          250119 non-null  uint8
31  sub_grade_F2          250119 non-null  uint8
32  sub_grade_F3          250119 non-null  uint8
33  sub_grade_F4          250119 non-null  uint8
34  sub_grade_F5          250119 non-null  uint8
35  sub_grade_G1          250119 non-null  uint8
36  sub_grade_G2          250119 non-null  uint8
37  sub_grade_G3          250119 non-null  uint8
38  sub_grade_G4          250119 non-null  uint8
39  sub_grade_G5          250119 non-null  uint8
```

```
40  home_ownership_OTHER  250119 non-null  uint8
41  home_ownership_OWN    250119 non-null  uint8
42  home_ownership_RENT   250119 non-null  uint8
43  zip_code_05113        250119 non-null  uint8
44  zip_code_11650        250119 non-null  uint8
45  zip_code_22690        250119 non-null  uint8
46  zip_code_29597        250119 non-null  uint8
47  zip_code_30723        250119 non-null  uint8
48  zip_code_48052        250119 non-null  uint8
49  zip_code_70466        250119 non-null  uint8
50  zip_code_86630        250119 non-null  uint8
51  zip_code_93700        250119 non-null  uint8
```

LendingClub

“การ Imbalance data จะทำพร้อมกับ train model ในคำสั่ง pipeline”

ตัวอย่างของจำนวนข้อมูลหลังจาก Imbalance data

Imbalance (in modeling pipeline)

```
[139] from imblearn.over_sampling import SMOTE
      from collections import Counter
      print('Original dataset shape %s' % Counter(y_train))
      rus = SMOTE(random_state=0, sampling_strategy=0.5)
      X_resampled, y_resampled = rus.fit_resample(X_train, y_train)
      print('Resampled dataset shape %s' % Counter(y_resampled))
```

```
Original dataset shape Counter({0: 201174, 1: 48945})
Resampled dataset shape Counter({0: 201174, 1: 100587})
```

```
[140] from imblearn.under_sampling import RandomUnderSampler
      from collections import Counter
      print('Original dataset shape %s' % Counter(y_train))
      rus = RandomUnderSampler(random_state=0, sampling_strategy=0.5)
      X_resampled, y_resampled = rus.fit_resample(X_train, y_train)
      print('Resampled dataset shape %s' % Counter(y_resampled))
```

```
Original dataset shape Counter({0: 201174, 1: 48945})
Resampled dataset shape Counter({0: 97890, 1: 48945})
```

3

Modeling

by Classification Algorithm of Supervised Learning

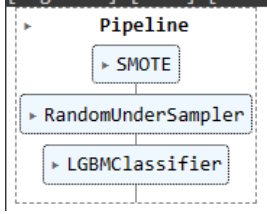
1. **LightGBM – Classifier**
2. **XGboost – Classifier**
3. **Random Forest - Classifier**

LendingClub

Light Gradient Boosting Machine-Classifier (LightGBM)

```
# Create the LightGBM classifier
lgb_model = lgb.LGBMClassifier(objective='binary',
                               class_weight='balanced')

# Define the hyperparameters for the LightGBM model
# param_lgb = {
#     'lightGBM__n_estimators': [100, 200, 300],
#     'lightGBM__learning_rate': [0.01, 0.05, 0.1],
#     'lightGBM__max_depth': [5, 10, -1],
#     'lightGBM__num_leaves': [31, 63, 127]
# }

# Create the pipeline with SMOTE(Over-Sampling), RandomOverSampler(Under-Sampling), and LightGBM
pipeline_lgb = Pipeline([
    # ('smote', SMOTE(random_state=42, sampling_strategy=0.5)),
    ('under_sampler', RandomUnderSampler(random_state=42, sampling_strategy=0.3)),
    ('lightGBM', lgb_model)
])

# # Create the GridSearchCV object
# grid_lgb = GridSearchCV(pipeline_lgb, param_lgb, cv=10, scoring='roc_auc', n_jobs=-1)

# Train the GridSearchCV model with Imbalance data on X_train and y_train
# grid_lgb.fit(X_train, y_train)
pipeline_lgb.fit(X_train, y_train)
```

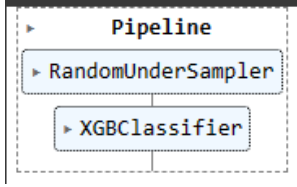
```
[LightGBM] [Info] Number of positive: 100587, number of negative: 100587
[LightGBM] [Warning] Auto-choosing row-wise multi-threading, the overhead of testing was 0.004487 seconds.
You can set `force_row_wise=true` to remove the overhead.
And if memory is not enough, you can set `force_col_wise=true`.
[LightGBM] [Info] Total Bins 1620
[LightGBM] [Info] Number of data points in the train set: 201174, number of used features: 52
[LightGBM] [Info] [binary:BoostFromScore]: pavg=0.500000 -> initscore=0.000000
```



LendingClub

Extreme Gradient Boosting-Classifier (XGboost)

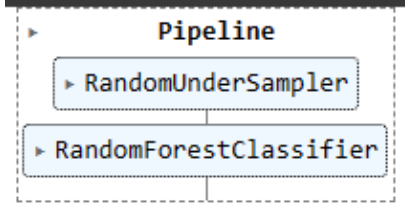
```
# Create the XGboost classifier
xgb_model = xgb.XGBClassifier(objective='binary:logistic', random_state=42)

# Define the hyperparameters for the XGboost model
# param_xgb = {
#     'xgb__n_estimators': [100, 200, 300],
#     'xgb__learning_rate': [0.01, 0.05, 0.1],
#     'xgb__max_depth': [5, 10, -1],
# }

# Create the pipeline with SMOTE(Over-Sampling), RandomOverSampler(Under-Sampling), and XGboost
pipeline_xgb = Pipeline([
    # ('smote', SMOTE(random_state=42, sampling_strategy=0.5)),
    ('under_sampler', RandomUnderSampler(random_state=42, sampling_strategy=1)),
    ('xgb', xgb_model),
])

# Create the GridSearchCV object
# grid_xgb = GridSearchCV(pipeline_xgb, param_xgb, cv=10, scoring='roc_auc', n_jobs=-1)

# Train the GridSearchCV model with Imbalance data on X_train and y_train
# grid_xgb.fit(X_train, y_train)
pipeline_xgb.fit(X_train, y_train)
```



LendingClub

Random Forest

```
# Create the Random Forest model
rf_model = RandomForestClassifier(random_state=42)

# Create the pipeline with SMOTE(Over-Sampling), RandomOverSampler(Under-Sampling), and Random Forest
pipeline_rf = Pipeline([
    # ('smote', SMOTE(random_state=42, sampling_strategy=0.5)),
    ('under_sampler', RandomUnderSampler(random_state=42, sampling_strategy=1)),
    ('rf', rf_model),
])

# Train the GridSearchCV model with Imbalance data on X_train and y_train
pipeline_rf.fit(X_train, y_train)
```



4

Evaluation

Evaluation Metrics

1. Confusion Matrix
   - Accuracy
   - Precision
   - F1-score
   - Recall

2. ROC – AUC Curve

LendingClub

Main Metrics

จากวัตถุประสงค์ที่ต้องการโมเดลที่ช่วยกรองผู้ที่มี “โอกาสผิดนัดชำระหนี้” เบื้องต้น
เพื่อให้นักลงทุนใช้ในการพิจารณาเลือกผู้กู้ที่เหมาะสมแก่การลงทุนและยังช่วยลดความเสี่ยงได้

- **Recall** : เนื่องจากต้องการหาผู้มีโอกาสเป็น Defaulter ให้ได้มากที่สุด
- **AUC score** : ช่วยประเมินความสามารถของโมเดลในการแยกแยะคลาสของ True Positive และ True Negative และเหมาะสมกับการใช้วัดความสามารถกับข้อมูลที่มีลักษณะ Imbalance data

	Predicted 0	Predicted 1
Actual 0	TN	FP
Actual 1	FN	TP

LendingClub

Comparison table of 3 classification models and comparison by imbalance method

Imbalance Method of **LightGBM** class_weight = "None"	0 - Fully Paid			1 - Charge off			Accuracy	AUC
	precision	recall	f1-score	precision	recall	f1-score		
Non-Imbalance	0.88	0.99	0.93	0.94	0.46	0.62	0.89	0.9048
Over-Sampling (SMOTE) (100%)	0.92	0.87	0.89	0.56	0.70	0.62	0.83	0.8952
Over-Sampling (SMOTE) (90%)	0.92	0.88	0.90	0.58	0.68	0.63	0.84	0.8958
Over-Sampling (SMOTE) (70%)	0.91	0.91	0.91	0.64	0.64	0.64	0.86	0.8982
Over-Sampling (SMOTE) (50%)	0.90	0.95	0.92	0.73	0.58	0.65	0.88	0.9012
Over-Sampling (SMOTE) (30%)	0.89	0.99	0.93	0.89	0.49	0.63	0.89	0.9043
Under-Sampling (RandomUnderSampler) (100%)	0.94	0.80	0.87	0.50	0.80	0.61	0.80	0.9040
Under-Sampling (RandomUnderSampler) (90%)	0.94	0.83	0.52	0.77	0.62	0.82	0.82	0.9044
Under-Sampling (RandomUnderSampler) (70%)	0.92	0.88	0.90	0.60	0.70	0.65	0.85	0.9044
Under-Sampling (RandomUnderSampler) (50%)	0.91	0.94	0.92	0.71	0.60	0.65	0.87	0.9046
Under-Sampling (RandomUnderSampler) (30%)	0.89	0.98	0.93	0.89	0.49	0.63	0.89	0.9046
Over (50%) and then Under (100%) Sampling	0.93	0.83	0.88	0.53	0.76	0.62	0.82	0.9016

Imbalance Method of **XGBoost**	0 - Fully Paid			1 - Charge off			Accuracy	AUC	Shape of Y_train
	precision	recall	f1-score	precision	recall	f1-score			
Non-Imbalance	0.88	0.99	0.93	0.92	0.47	0.62	0.89	0.9044	({0: 201174, 1: 48945})
Over-Sampling (SMOTE) (100%)	0.91	0.91	0.91	0.64	0.62	0.63	0.85	0.8924	({0: 201174, 1: 201174})
Over-Sampling (SMOTE) (90%)	0.91	0.92	0.91	0.65	0.61	0.63	0.86	0.8924	({0: 201174, 1: 181056})
Over-Sampling (SMOTE) (70%)	0.90	0.94	0.92	0.69	0.58	0.63	0.87	0.8954	({0: 201174, 1: 140821})
Over-Sampling (SMOTE) (50%)	0.90	0.96	0.93	0.75	0.55	0.64	0.88	0.8980	({0: 201174, 1: 140821})
Over-Sampling (SMOTE) (30%)	0.89	0.98	0.93	0.87	0.49	0.63	0.89	0.9035	({0: 201174, 1: 140821})
Under-Sampling (RandomUnderSampler) (100%)	0.94	0.80	0.86	0.49	0.80	0.61	0.80	0.9026	({0: 48945, 1: 48945})
Under-Sampling (RandomUnderSampler) (90%)	0.94	0.82	0.88	0.52	0.77	0.62	0.81	0.9030	({0: 54383, 1: 48945})
Under-Sampling (RandomUnderSampler) (70%)	0.92	0.88	0.90	0.59	0.71	0.64	0.84	0.9034	({0: 69921, 1: 48945})
Under-Sampling (RandomUnderSampler) (50%)	0.91	0.93	0.92	0.70	0.61	0.65	0.87	0.9038	({0: 97890, 1: 48945})
Under-Sampling (RandomUnderSampler) (30%)	0.89	0.98	0.93	0.86	0.50	0.64	0.89	0.9039	({0: 163150, 1: 48945})
Over (50%) and then Under (100%) Sampling	0.93	0.86	0.89	0.55	0.72	0.63	0.83	0.8987	({0: 100587, 1: 100587})

Imbalance Method of **LightGBM** class_weight = "balance"	0 - Fully Paid			1 - Charge off			Accuracy	AUC
	precision	recall	f1-score	precision	recall	f1-score		
Non-Imbalance	0.94	0.80	0.87	0.50	0.80	0.61	0.80	0.9049
Over-Sampling (SMOTE) (100%)	0.92	0.87	0.89	0.56	0.70	0.62	0.83	0.8952
Over-Sampling (SMOTE) (90%)	0.92	0.86	0.89	0.56	0.71	0.62	0.83	0.8960
Over-Sampling (SMOTE) (70%)	0.93	0.85	0.89	0.54	0.73	0.62	0.83	0.8986
Over-Sampling (SMOTE) (50%)	0.93	0.83	0.88	0.52	0.76	0.62	0.82	0.9014
Over-Sampling (SMOTE) (30%)	0.94	0.81	0.87	0.50	0.79	0.61	0.80	0.9042
Under-Sampling (RandomUnderSampler) (100%)	0.94	0.80	0.87	0.50	0.80	0.61	0.80	0.9040
Under-Sampling (RandomUnderSampler) (90%)	0.94	0.80	0.87	0.50	0.80	0.61	0.80	0.9043
Under-Sampling (RandomUnderSampler) (70%)	0.94	0.80	0.87	0.50	0.80	0.61	0.80	0.9046
Under-Sampling (RandomUnderSampler) (50%)	0.94	0.80	0.87	0.50	0.80	0.61	0.80	0.9048
Under-Sampling (RandomUnderSampler) (30%)	0.94	0.80	0.87	0.50	0.80	0.61	0.80	0.9050
Over (50%) and then Under (100%) Sampling	0.93	0.83	0.88	0.53	0.76	0.62	0.82	0.9016

Imbalance Method of **Random Forest**	0 - Fully Paid			1 - Charge off			Accuracy	AUC	Shape of Y_train
	precision	recall	f1-score	precision	recall	f1-score			
Non-Imbalance	0.89	0.98	0.93	0.87	0.48	0.62	0.88	0.8837	({0: 201174, 1: 48945})
Over-Sampling (SMOTE) (100%)	0.91	0.91	0.91	0.62	0.62	0.62	0.85	0.8843	({0: 201174, 1: 201174})
Over-Sampling (SMOTE) (90%)	0.91	0.92	0.91	0.64	0.61	0.62	0.86	0.8841	({0: 201174, 1: 181056})
Over-Sampling (SMOTE) (70%)	0.90	0.93	0.92	0.68	0.58	0.63	0.86	0.8853	({0: 201174, 1: 140821})
Over-Sampling (SMOTE) (50%)	0.90	0.95	0.92	0.74	0.55	0.63	0.87	0.8855	({0: 201174, 1: 140821})
Over-Sampling (SMOTE) (30%)	0.89	0.98	0.93	0.84	0.50	0.63	0.88	0.8850	({0: 201174, 1: 140821})
Under-Sampling (RandomUnderSampler) (100%)	0.93	0.80	0.86	0.49	0.77	0.60	0.80	0.8890	({0: 48945, 1: 48945})
Under-Sampling (RandomUnderSampler) (90%)	0.93	0.83	0.87	0.51	0.75	0.61	0.81	0.8896	({0: 54383, 1: 48945})
Under-Sampling (RandomUnderSampler) (70%)	0.92	0.87	0.90	0.57	0.69	0.62	0.84	0.8889	({0: 69921, 1: 48945})
Under-Sampling (RandomUnderSampler) (50%)	0.91	0.93	0.92	0.67	0.61	0.64	0.86	0.8884	({0: 97890, 1: 48945})
Under-Sampling (RandomUnderSampler) (30%)	0.89	0.97	0.93	0.82	0.51	0.63	0.88	0.8854	({0: 163150, 1: 48945})
Over (50%) and then Under (100%) Sampling	0.92	0.86	0.89	0.56	0.69	0.62	0.83	0.8874	({0: 100587, 1: 100587})

เมื่อ Precision เพิ่ม Recall จะลดลงในทุกๆ กรณีดังนั้นจึงเลือก Main Metrics เป็น Recall

LendingClub

เลือก Model และ Imbalance Method ที่ให้ค่า Recall และ AUC สูงที่สุด

Model	Imbalance Method	0 - Fully Paid			1 - Charge off			Accuracy	AUC	Shape of Y_train
		precision	recall	f1-score	precision	recall	f1-score			
LightGBM - classweight = "None"	Under-Sampling (RandomUnderSampler) (100%)	0.94	0.80	0.87	0.50	0.80	0.61	0.80	0.9040	({0: 48945, 1: 48945})
LightGBM - classweight = "balanced"	Under-Sampling (RandomUnderSampler) (30%)	0.94	0.80	0.87	0.50	0.80	0.61	0.80	0.9050	({0: 163150, 1: 48945})
XGboost	Under-Sampling (RandomUnderSampler) (100%)	0.94	0.80	0.86	0.49	0.80	0.61	0.80	0.9026	({0: 48945, 1: 48945})
Random forest	Under-Sampling (RandomUnderSampler) (100%)	0.93	0.80	0.86	0.49	0.77	0.60	0.80	0.8890	({0: 48945, 1: 48945})

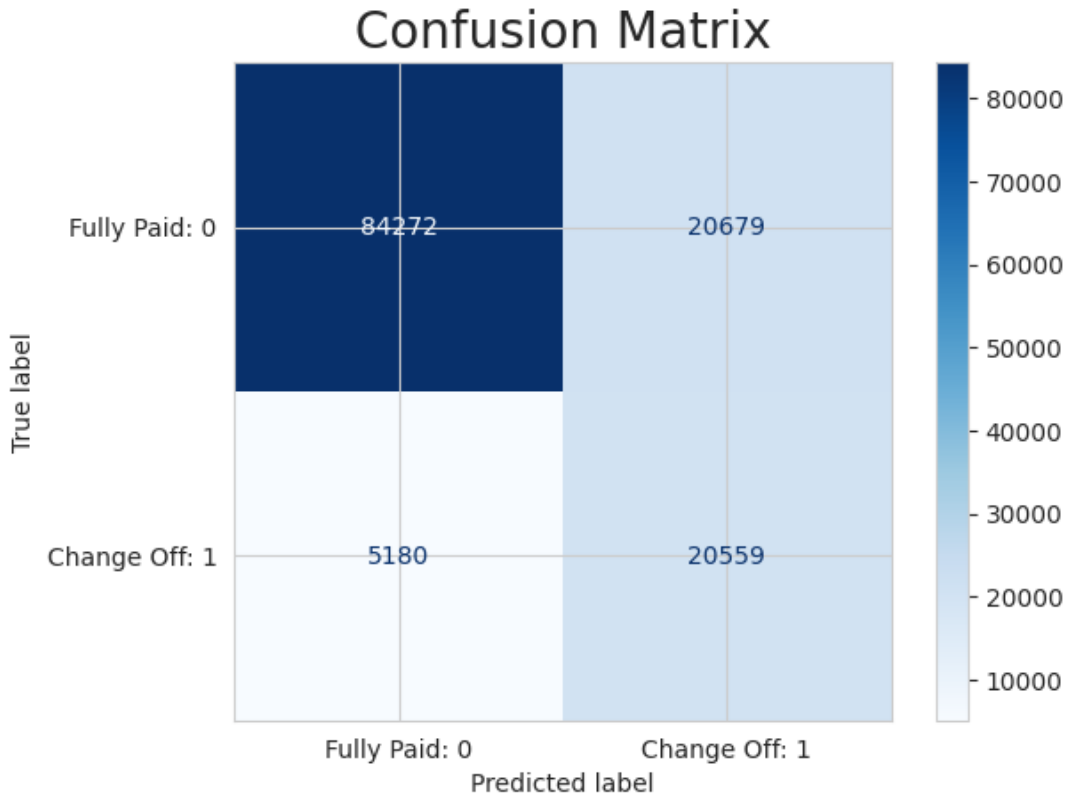
Best Model

Model : LightGBM - class_weight = “balanced”
Imbalance Method : RandomUnderSampler - sampling_strategy = 0.3

Charge-Off Recall = 0.80
AUC score = 0.9050

LendingClub

Best Model Evaluation Metrics



Classification Report of Train Data:

	precision	recall	f1-score	support
0	0.95	0.81	0.87	201174
1	0.51	0.82	0.63	48945
accuracy			0.81	250119
macro avg	0.73	0.81	0.75	250119
weighted avg	0.86	0.81	0.82	250119

Classification Report of Test Data:

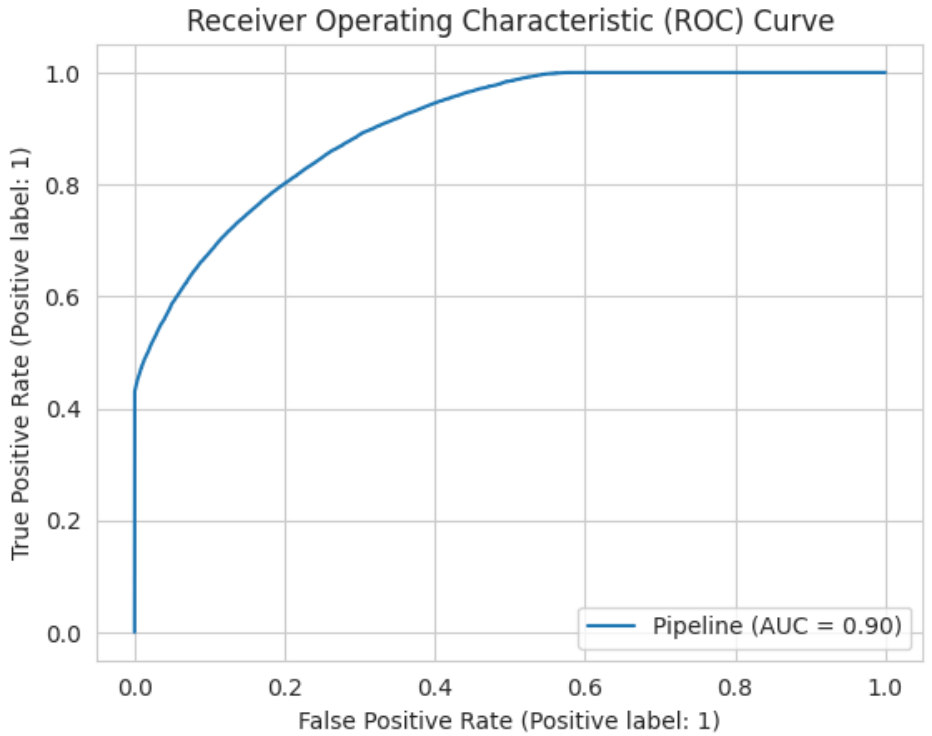
	precision	recall	f1-score	support
0	0.94	0.80	0.87	104951
1	0.50	0.80	0.61	25739
accuracy			0.80	130690
macro avg	0.72	0.80	0.74	130690
weighted avg	0.85	0.80	0.82	130690

LendingClub

Best Model Evaluation Metrics

```
# auc_score_lgb = roc_auc_score(y_test, best_model_lgb.predict_proba(X_test)[:, 1])
# print(f"AUC Score: {auc_score_lgb:.4f}")

auc_score_lgb = roc_auc_score(y_test, pipeline_lgb.predict_proba(X_test)[:, 1])
print(f"AUC Score: {auc_score_lgb:.4f}")
```

```
AUC Score: 0.9050
```



LendingClub

Conclusion

Features Importance
9 features

1. Term
2. Sub grade
3. Annual income
4. Debt to income
5. Mortgage Account
6. Home ownership
7. Interest rate
8. Zip code
9. Installment

หลังจากการ EDA ข้อมูลก็เห็นได้ชัดว่าส่งผลต่อการผิดนัดชำระหนี้

ได้ Model ที่ตรงตามวัตถุประสงค์ คือ

Model : Light Gradient Boosting Machine-Classifier

- Charge-Off Recall = 0.80
- AUC score = 0.9050

- เข้าใจเกี่ยวกับลักษณะการกู้เงินแบบ P2P
- ได้เรียนรู้เทคนิคและวิธีการ EDA
- ได้เรียนขั้นตอนและเครื่องมือต่างๆ ที่สำคัญในการเตรียมข้อมูล

THANK YOU