LendingClub Analysis

by Charla Gaddy

about LendingClub

- LendingClub is a US peer-to-peer lending company.
- They bring borrowers and investors together transforming the way people access credit.
- They were established in 2007 helping borrows take control of their debt, grow small business and investing in their future.

Questions & Predictions

The supervised model will act as an underwriter. An underwriter is responsible for deciding if someone is eligible for a loan based on many factors provided.

- Can a Supervised model predict good loans?

The unsupervised model will cluster the top 10 features from the best supervised model.

- Can KMeans model predict unknown clusters?
- How well does KMeans and Mean Shift cluster good loans?

The rejected loan application dataset is used to forecast the number of applications rejected.

- Can SARIAM forecast the number of rejected loans per week?
- LSTM predict the number of rejected loans per week?

Data

https://www.lendingclub.com/info/download-data.action

The data was downloaded from their website. I used both sets of data.

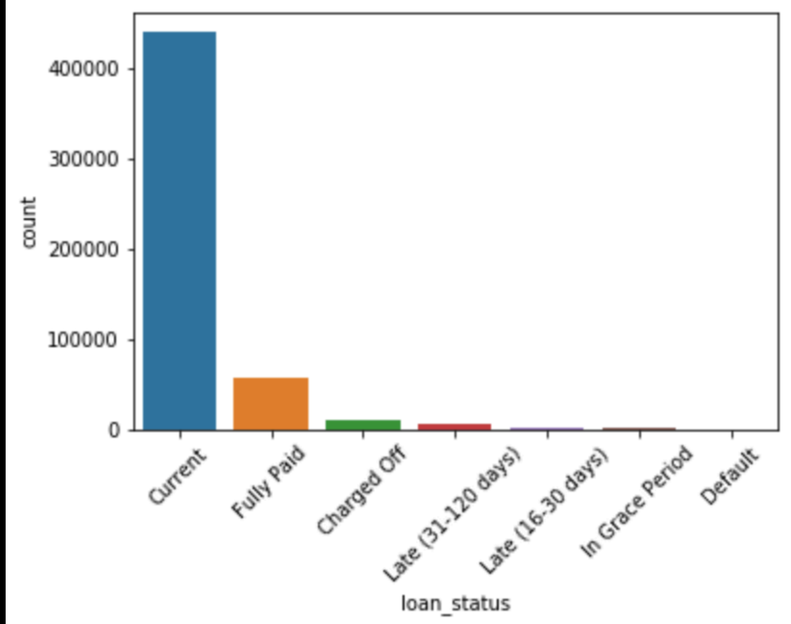
- Loan application data - This file includes all loans issued in 2018 and Q1 of 2019. Initially there are 610917 rows and 144 columns. After cleaning and label encoding the data the final dataset has 73318 rows and 38 columns.
    - Supervised and Unsupervised learning.
- Declined loan application data - All loan applications that did not meet LendingClub's credit underwriting policy issued 2017, 2018 and Q1 of 2019. The dataset initially had 19,158,655 rows and 2 columns. After dropping null values and using the Grouper function to group into weeks, the data set is now 117 rows and 2 columns.
    - Time Series forecast - Arima & LSTM

Data

- Feature ‘good_loan_status’ was created from the categorical feature ‘loan_status’.
- A good loan application is considered: Fully Paid
- A bad loan application is considered: Charged Off, Late(31-120 days) and Default.
- The other values will be discarded: Current, Late(16-30 days), In Grace Period

```
1 loan_stat = df["loan_status"]
2 sns.countplot(loan_stat)
3 stat_temp = df.loan_status.value_counts().sum()
4 plt.xticks(rotation=45)
```

```
(array([0, 1, 2, 3, 4, 5, 6]), <a list of 7 Text xtickla
```

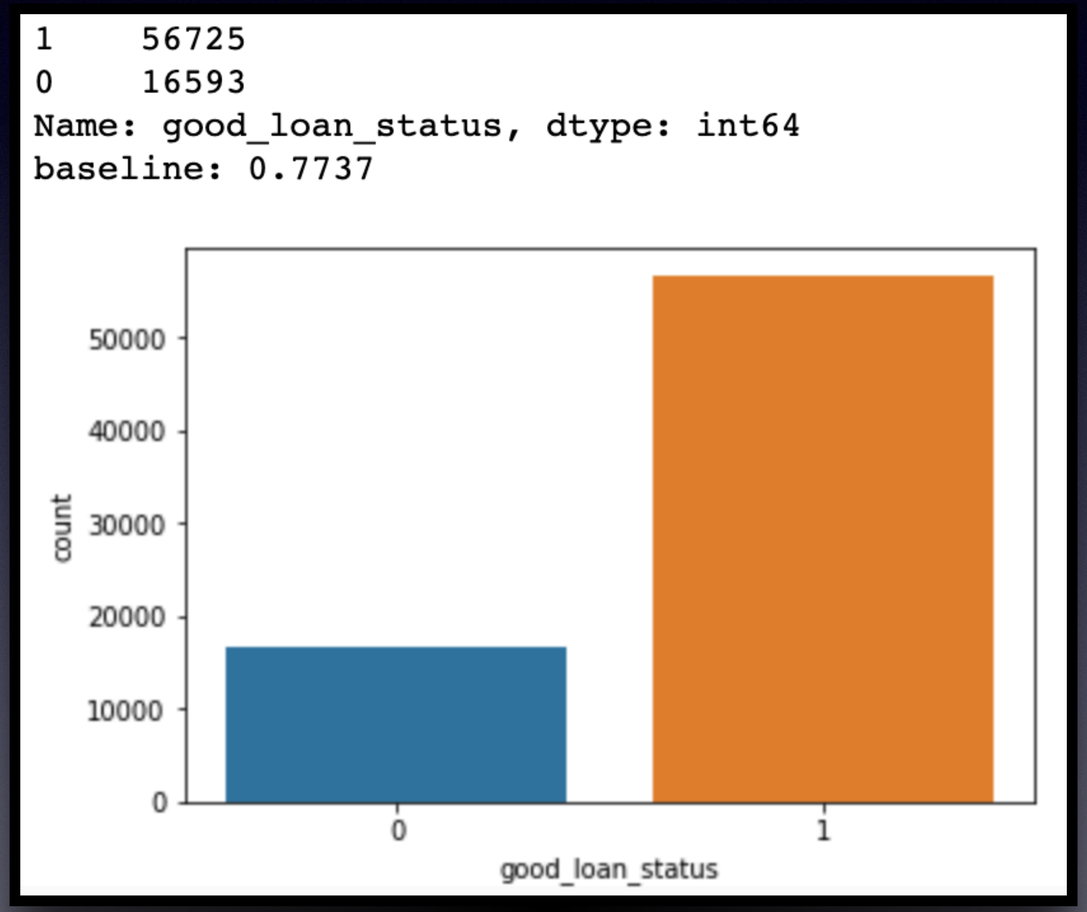



```
1 df = df[df['loan_status'] != 'Current']
2 df = df[df['loan_status'] != 'In Grace Period']
3 df = df[df['loan_status'] != 'Late (16-30 days)']
```

Data

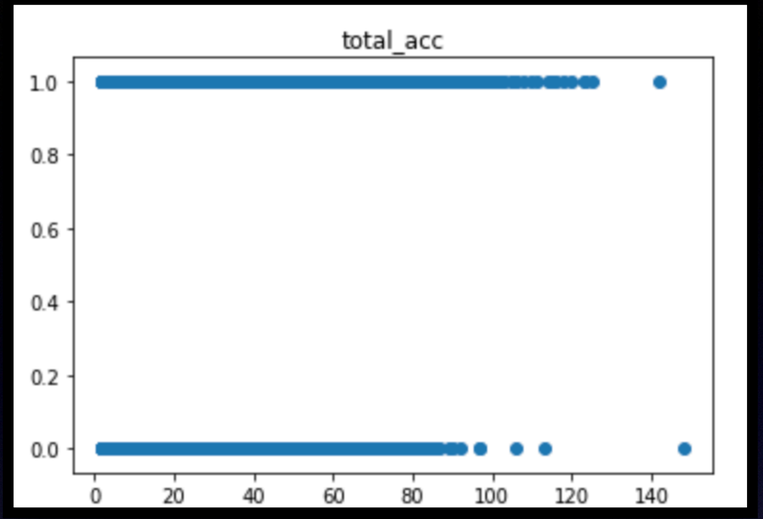
- The baseline has accuracy of 77.37%
- The ratio for good and bad loans is very large almost 3 to 1.

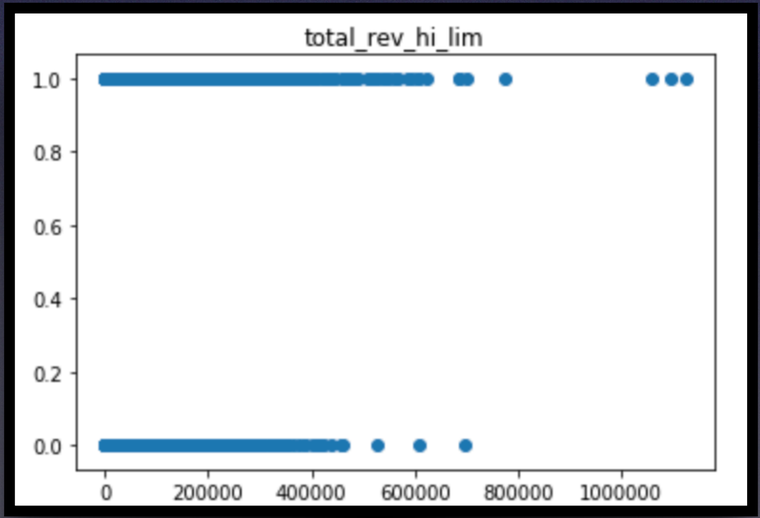
```
1     56725
0     16593
Name: good_loan_status, dtype: int64
baseline: 0.7737
```

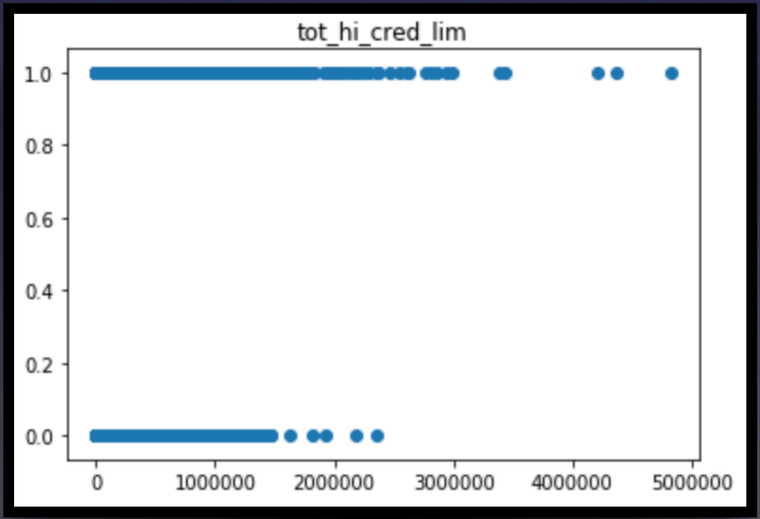


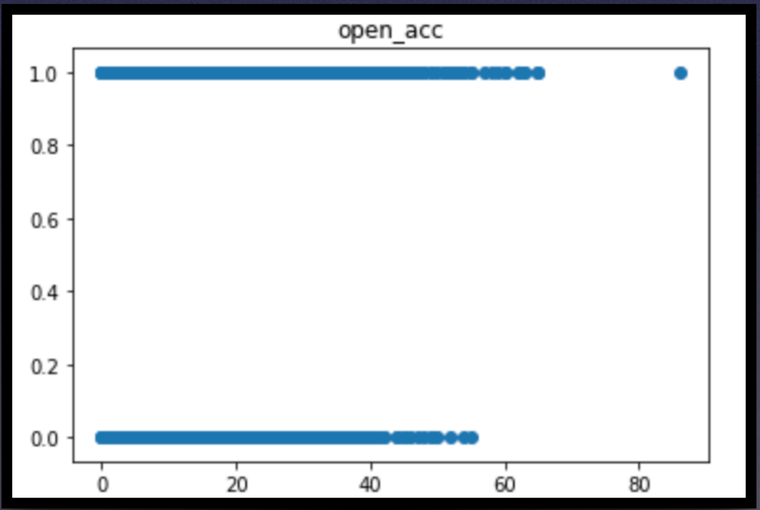
A few feature plots of good vs bad loan status

People who have a good loan status (1) tend to have more accounts (open or total) and a higher credit limit than those who have

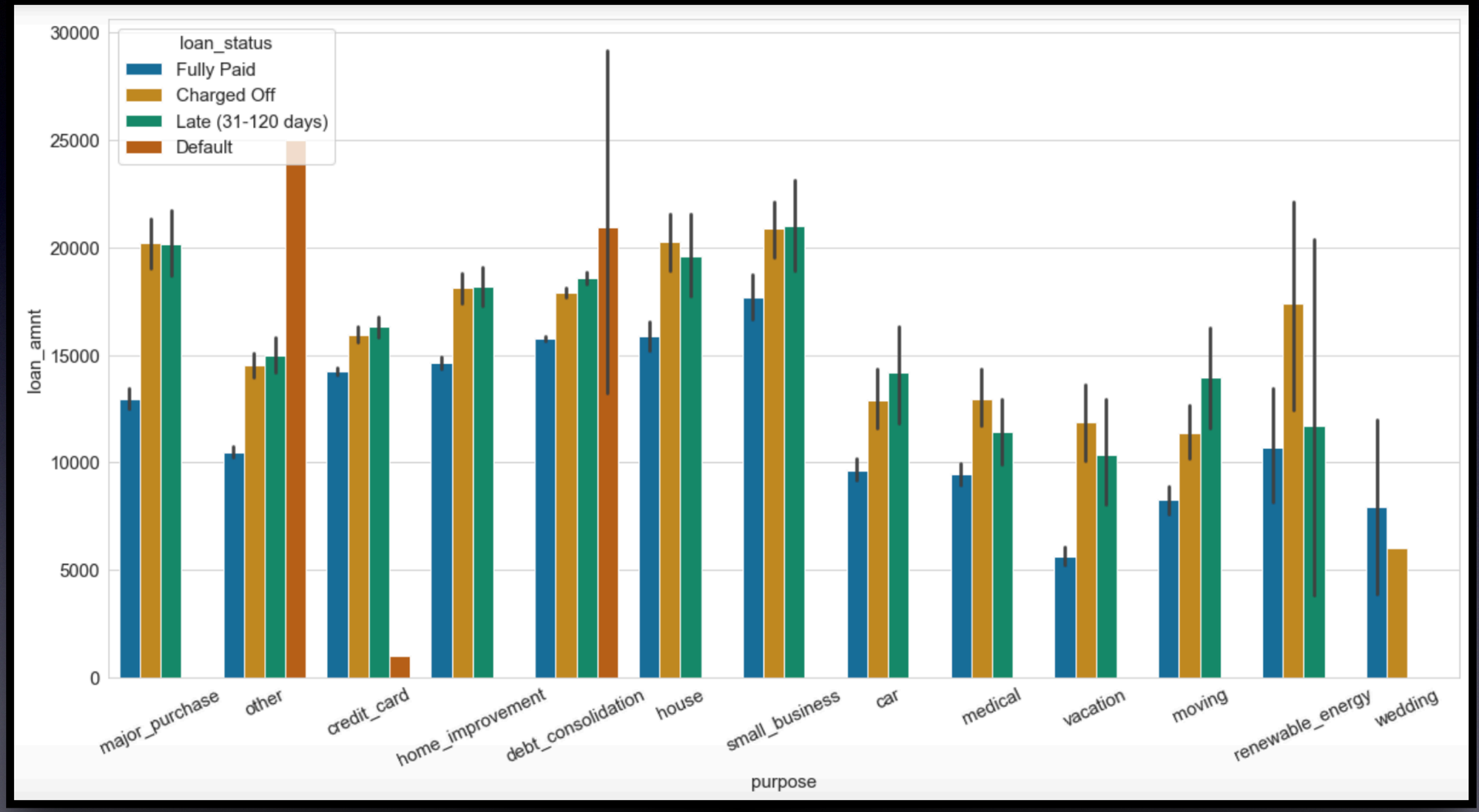




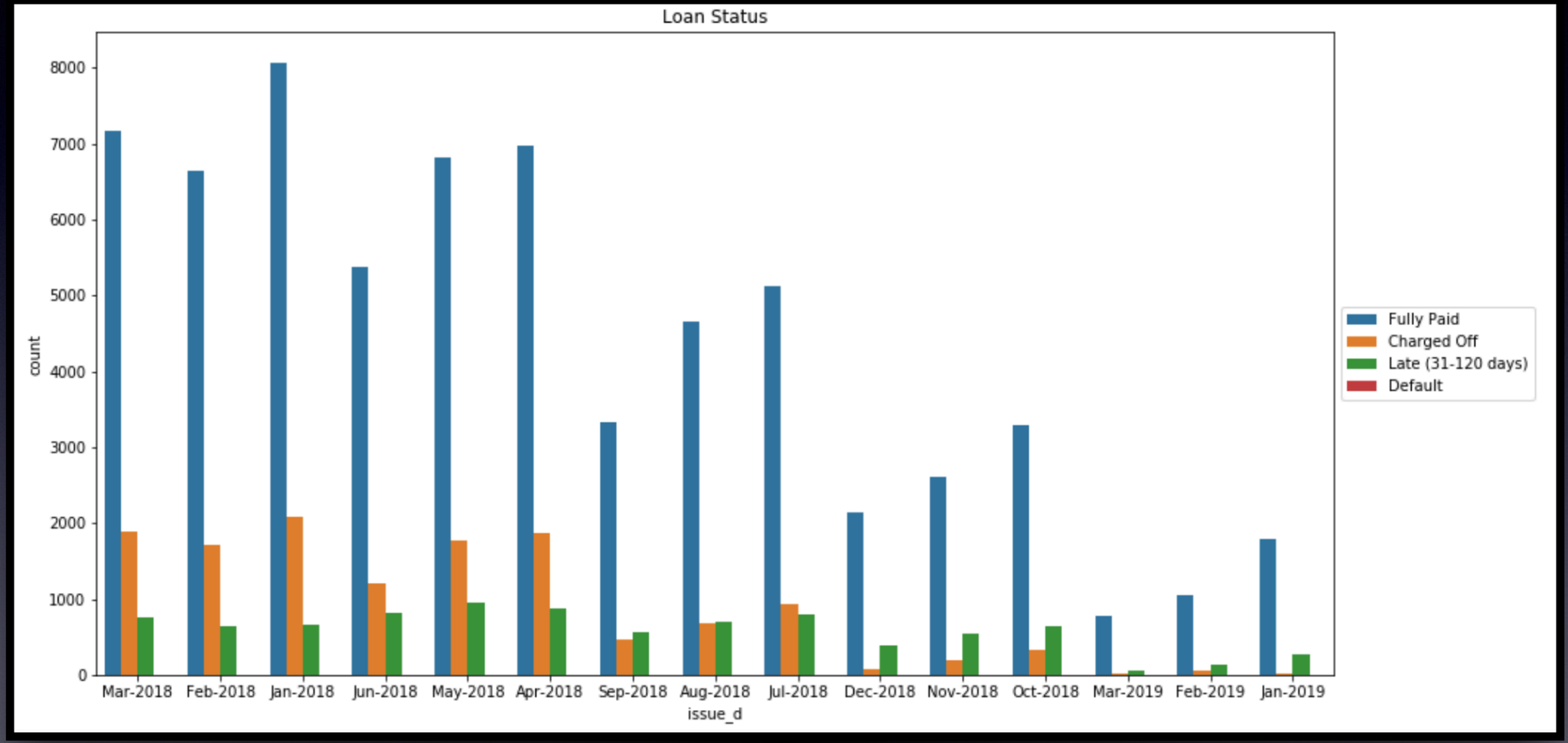




- debt consolidation loans go into default more often than not
- all loans except for wedding loan are more likely to be late, in default or charged off



- Most loans are paid in full the months of January thru May
- There are more loans charged offs during this same time



What percentage of loans are good?

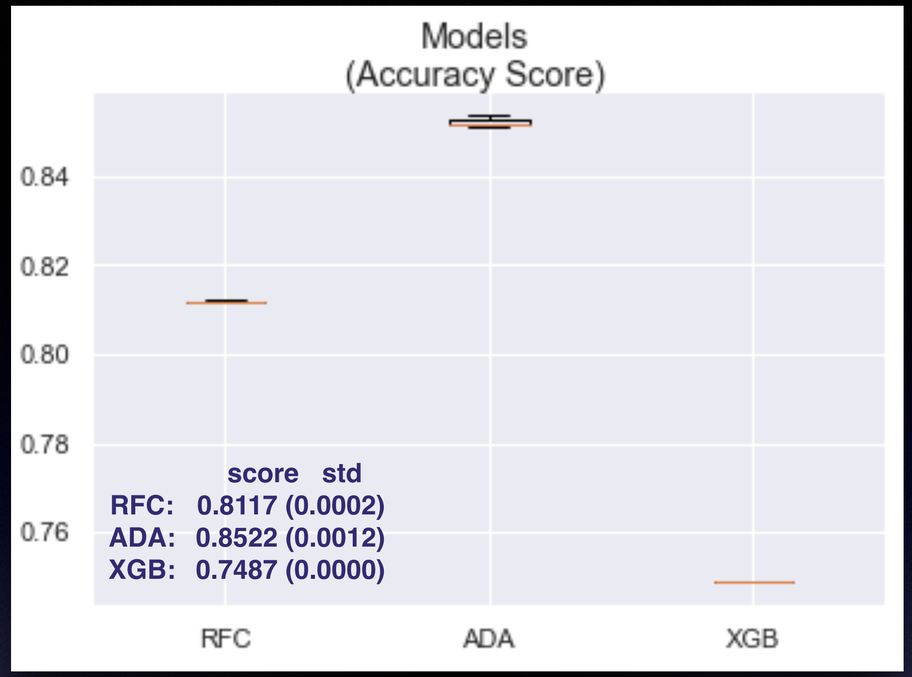
Random Forest ,Boosted Random Forest and Extreme Gradient Boost are the algorithms used to determine the percentage of good loans that will be issued.

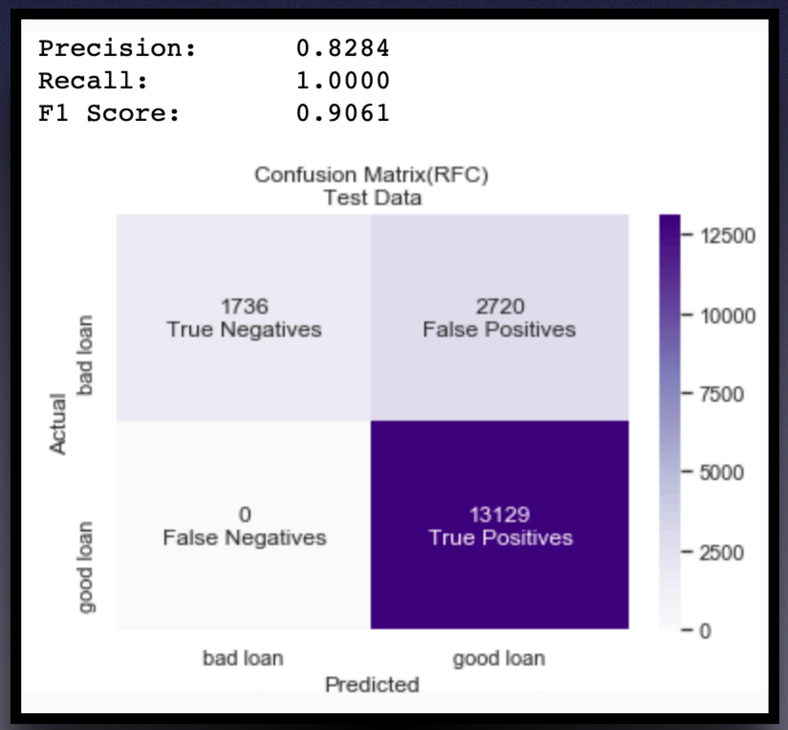
```
rf_final = RandomForestClassifier(max_depth=6,
                                  min_samples_leaf=50,
                                  min_samples_split=60,
                                  max_features=14,
                                  n_estimators=80)

ada_final = AdaBoostClassifier(base_estimator=rf_final,
                               n_estimators=27)

xgb_final = XGBClassifier(max_depth=3,
                          n_estimators=7,
                          learning_rate=0.1,
                          subsample=0.01,
                          colsample_bytree=0.001,
                          colsample_bylevel=0.0001)
```

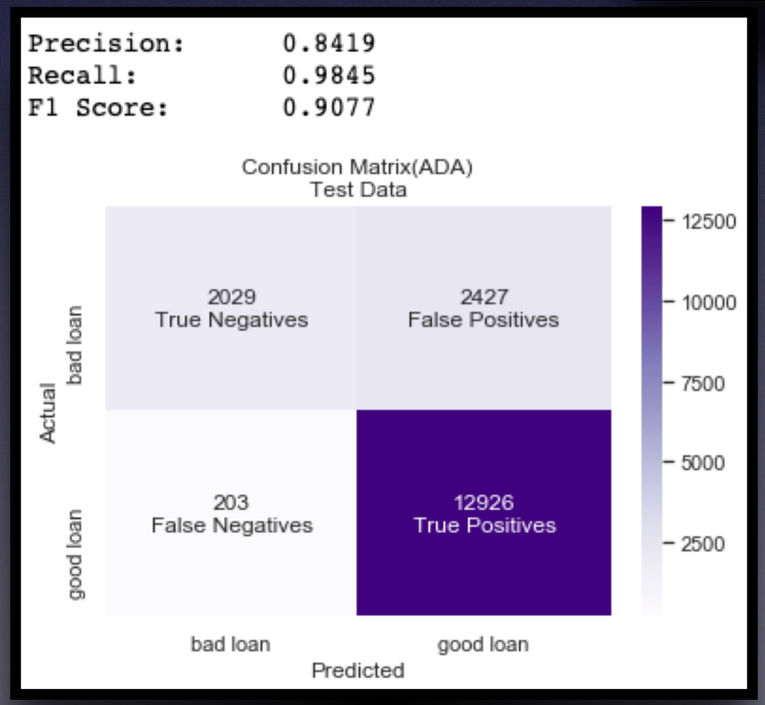
ADA Boosted Random Forest had the best score

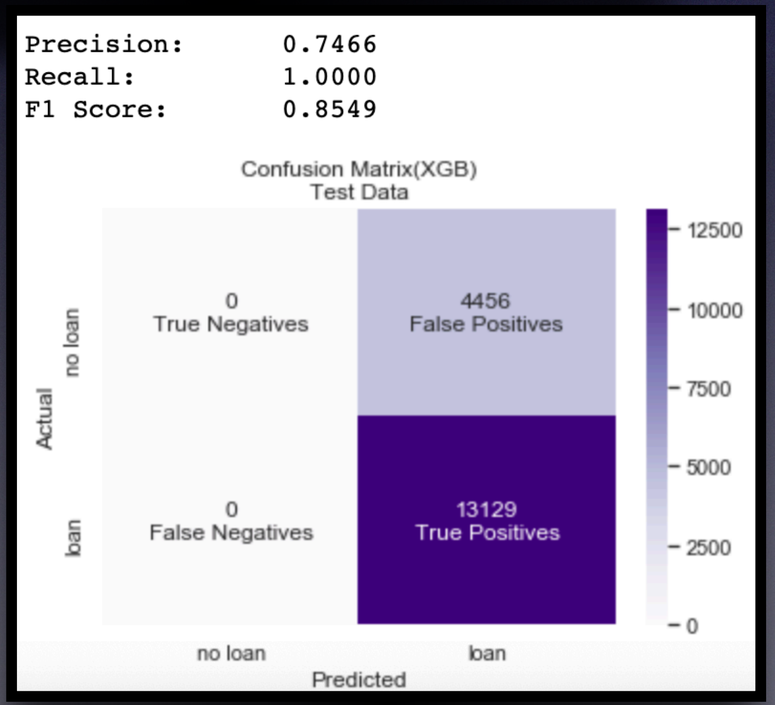
High precision means that an algorithm returned substantially more relevant results than irrelevant ones

High recall means that an algorithm returned most of the relevant results.





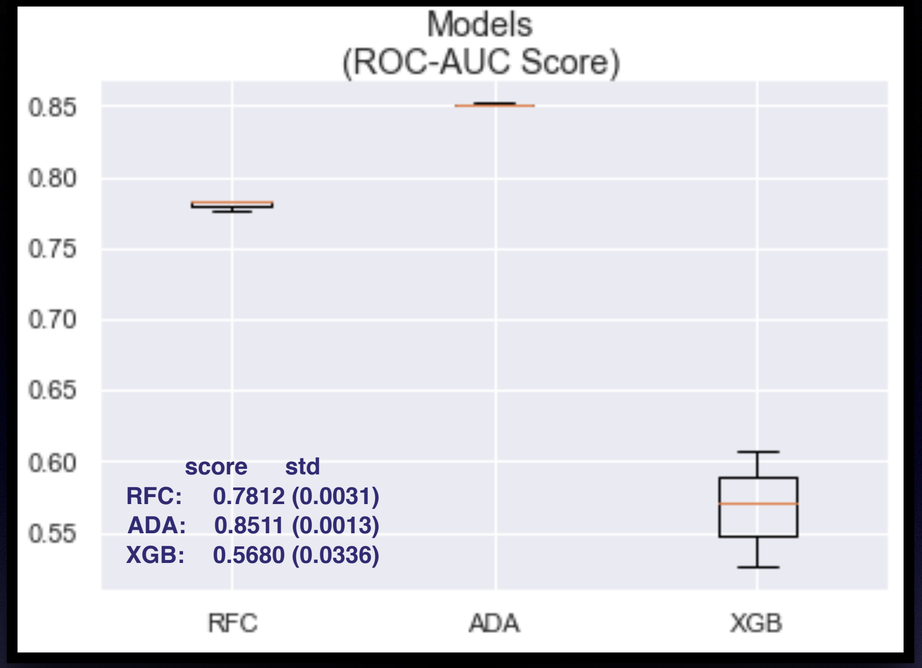


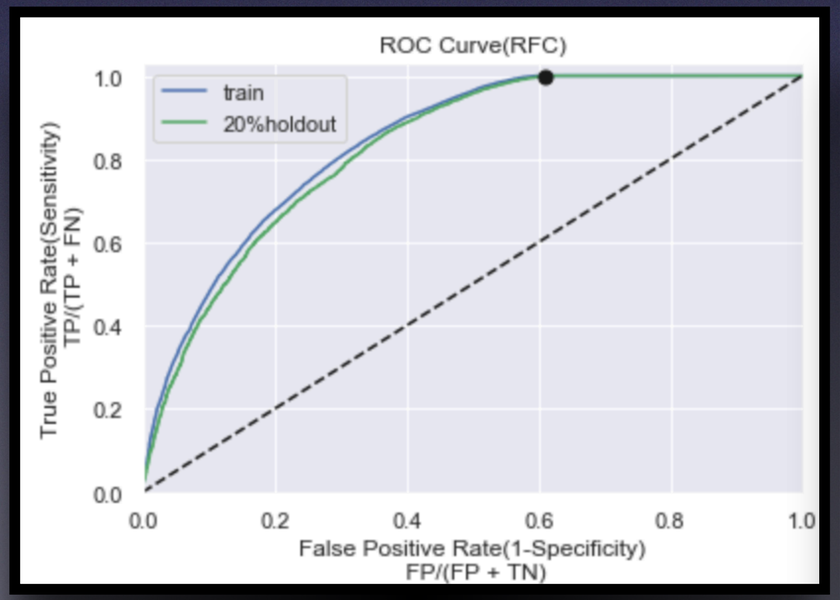


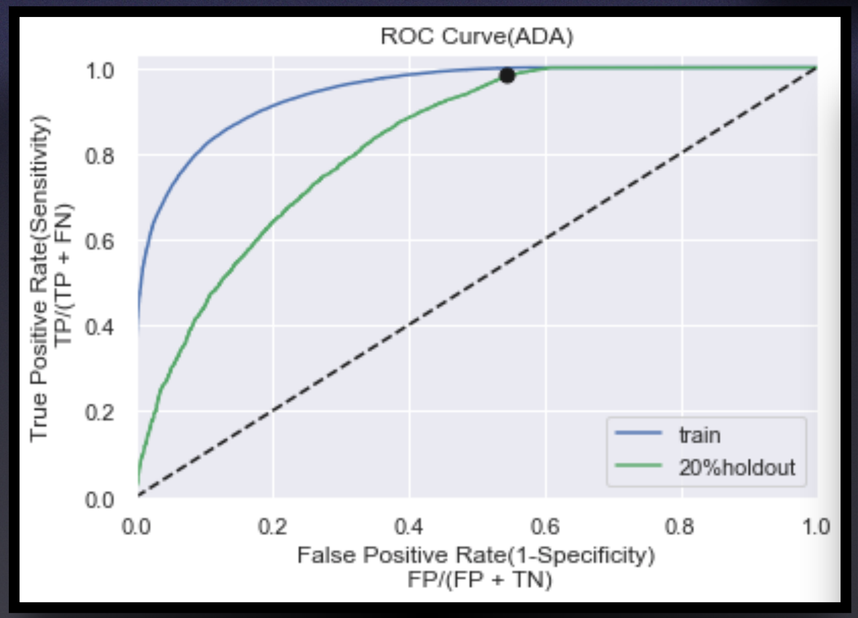
ROC/AUC scores

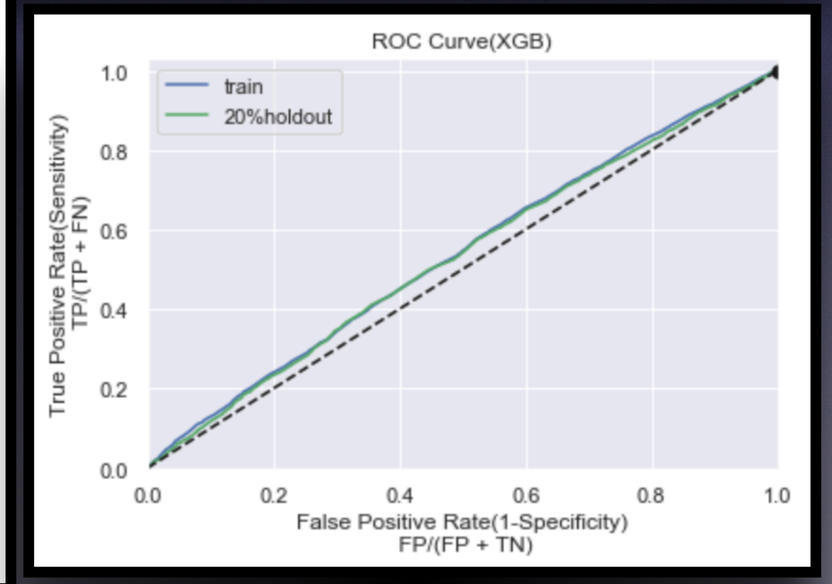
ADA Boost model compared to the Random Forest you can see it increases slower(closer to the y-axis).

As we saw with the XGB confusion matrix and score, it was worse than the baseline score. And we can see the ROC is almost linear.



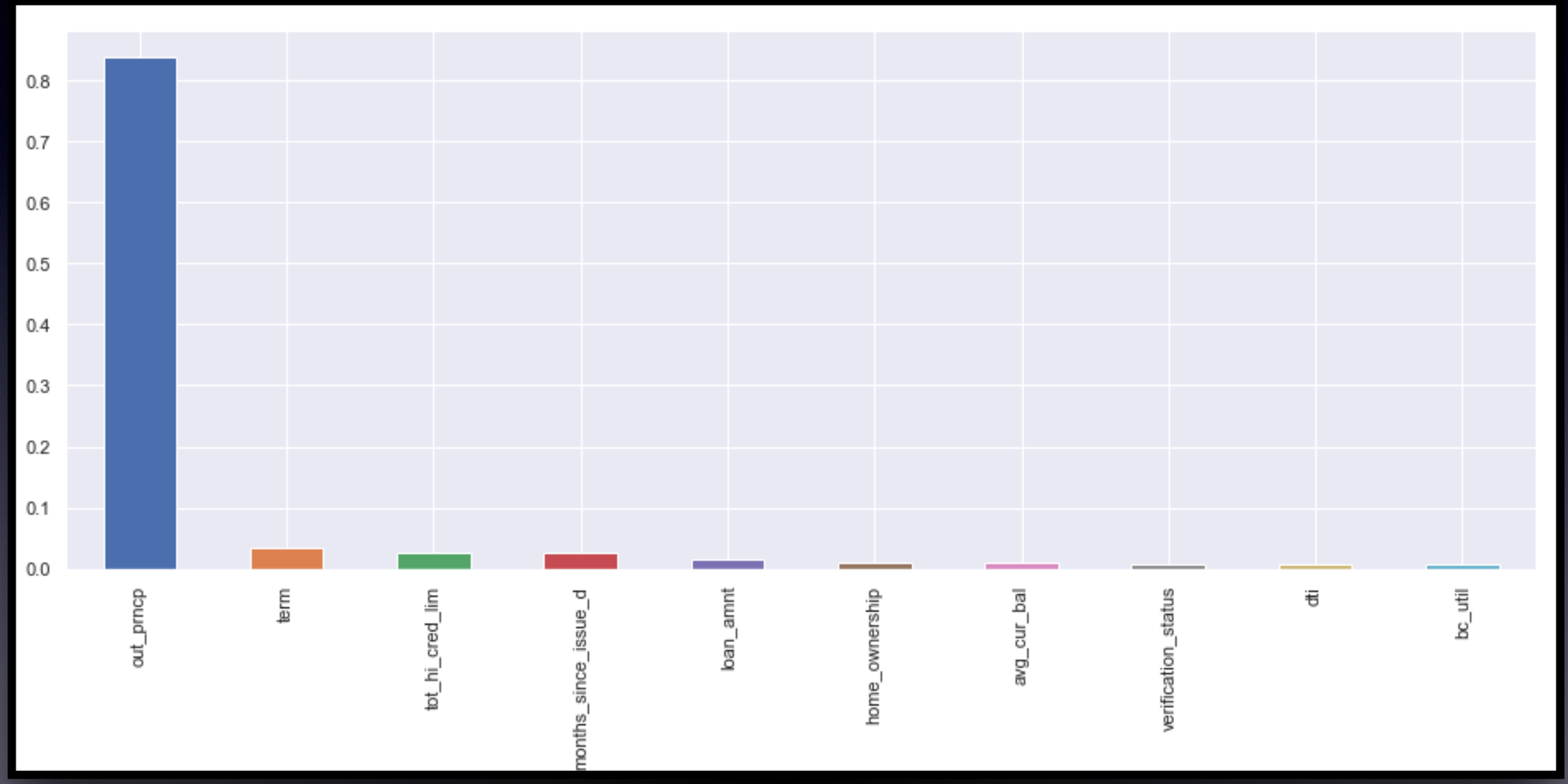






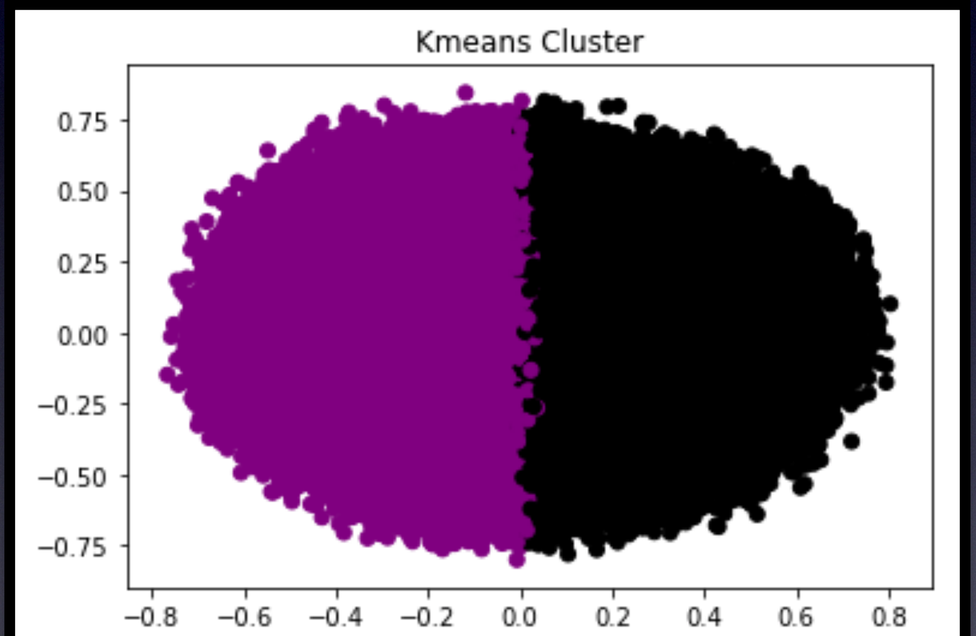
Feature Importance

Top 10 features of RFC model



KMeans & KMeans Minibatch

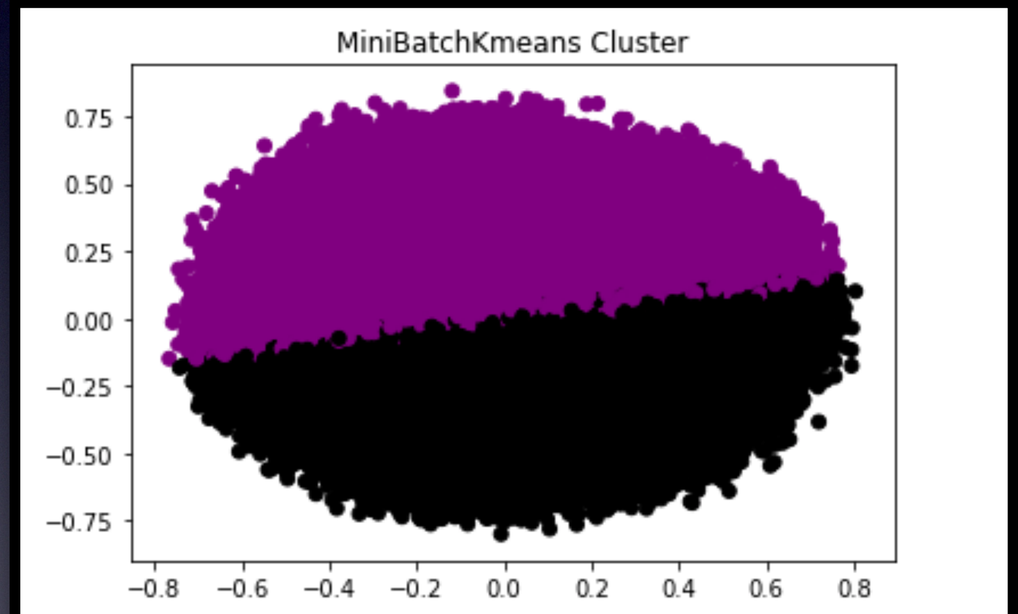
44,298 (33717+10581) correctly classified
43,625 (32076+11549) mis-classified



Comparing k-means clusters against the data:

good_loan_status	0	1
row_0		
0	10581	32076
1	11549	33717

43,207 (32263+10944) correctly classified
44,716 (11186+33530) mis-classified



Comparing k-means and mini batch k-means solutions:

good_loan_status	0	1
row_0		
0	10944	33530
1	11186	32263

From the top 10 Random Forest features, Kmeans found 2 new clusters.
Cluster 2 : total high credit limit and average current balance into a cluster.
Cluster 0: Had all other features

Kmeans Clusters

	Features	Cluster
0	out_prncp	0
1	term	0
2	tot_hi_cred_lim	2
3	months_since_issue_d	0
4	loan_amnt	0
5	home_ownership	0
6	avg_cur_bal	2
7	verification_status	0
8	dti	0
9	bc_util	0

Time Series

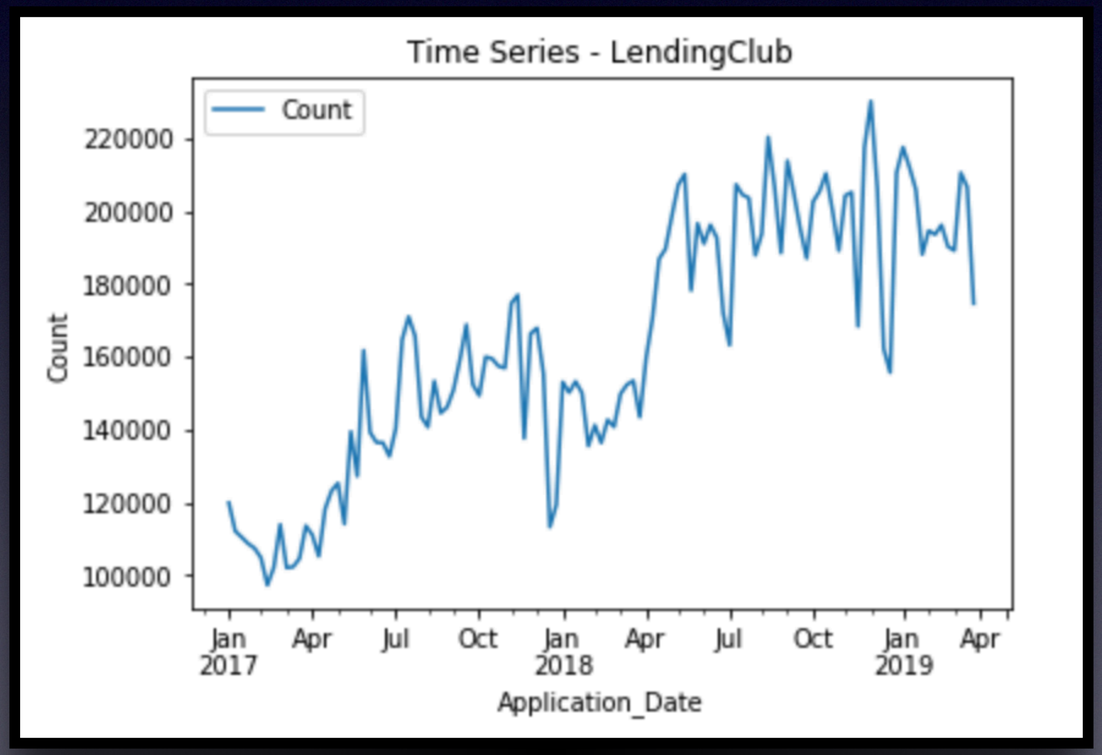
Rejected Loans (weekly)

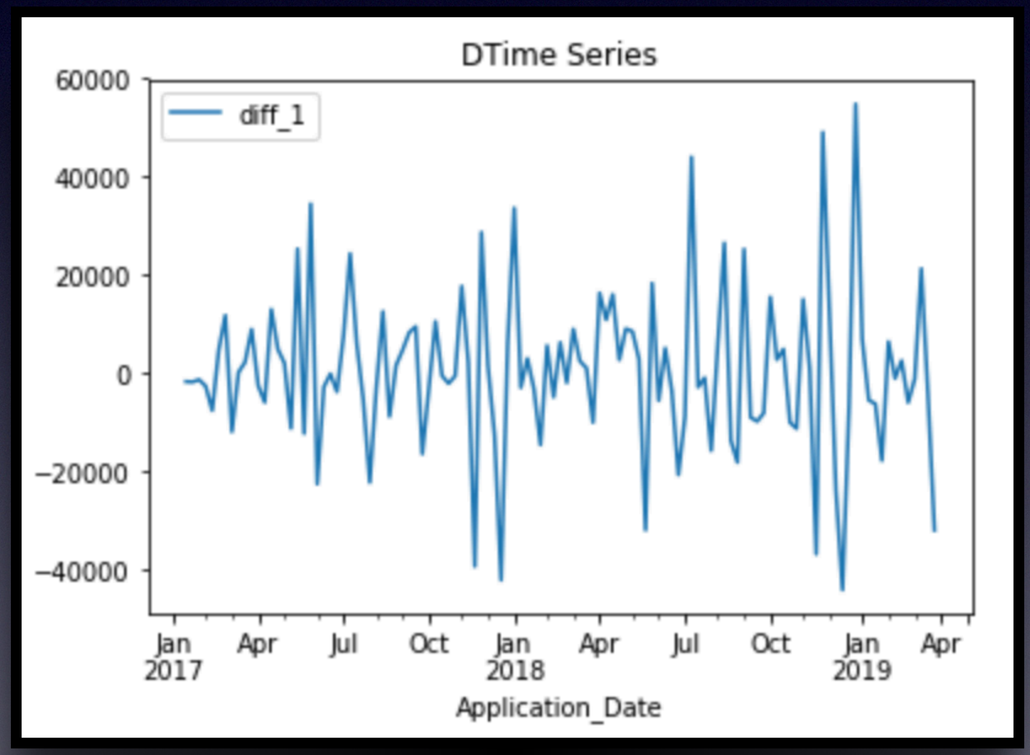
Application_Date	Count
2017-01-02	119903
2017-01-09	112116
2017-01-16	110457
2017-01-23	108717
2017-01-30	107435

- For time series we are using the rejected loan application dataset. The original dataset has 19158655 rows but after converting the date to weeks, it is reduced to 117 rows.
- SARIMA and LTSM will forecast the number of loans rejected weekly with this data.

Original and Differenced time series

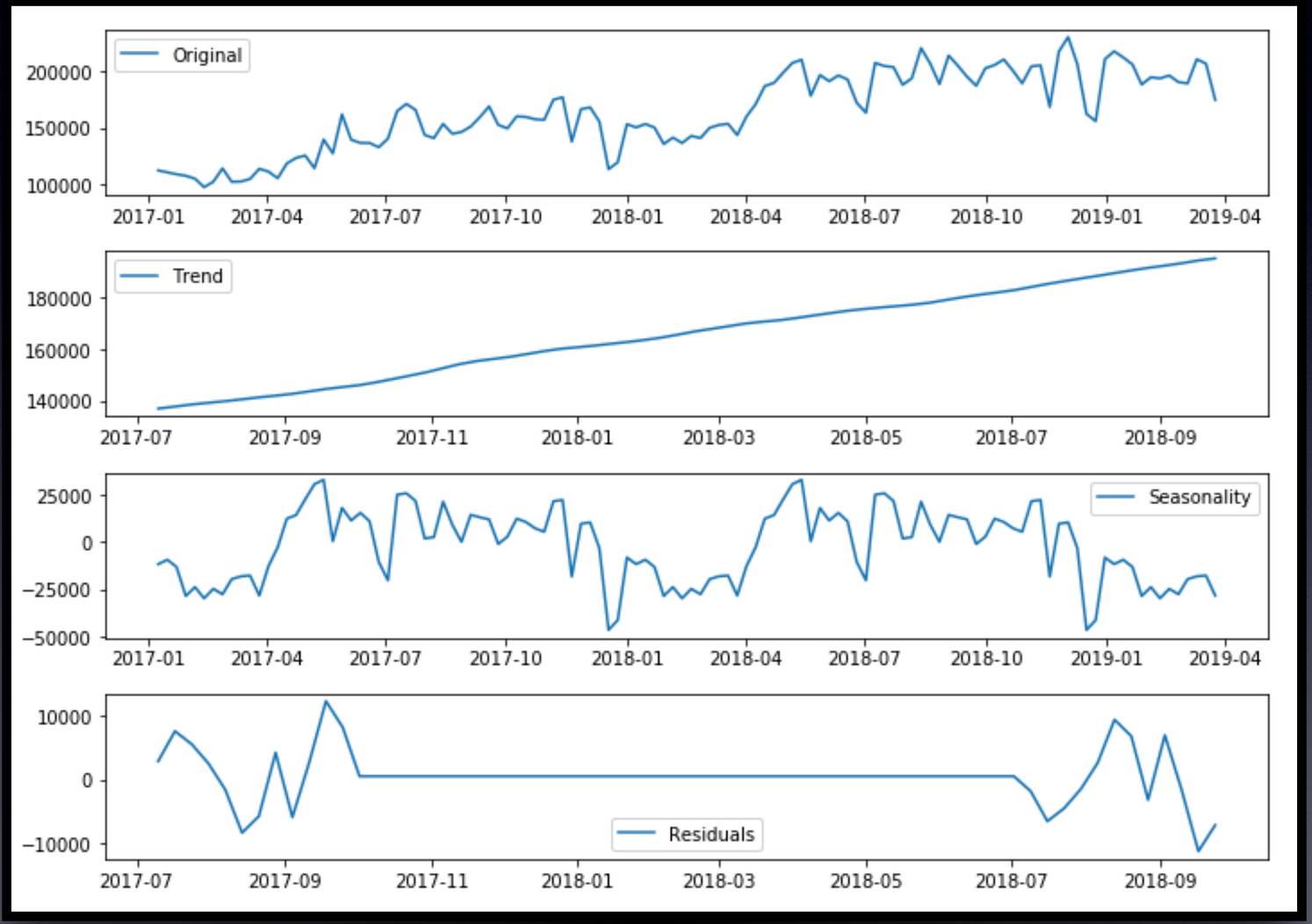
- We have non-stationary data. A difference of one is done to make it stationary.





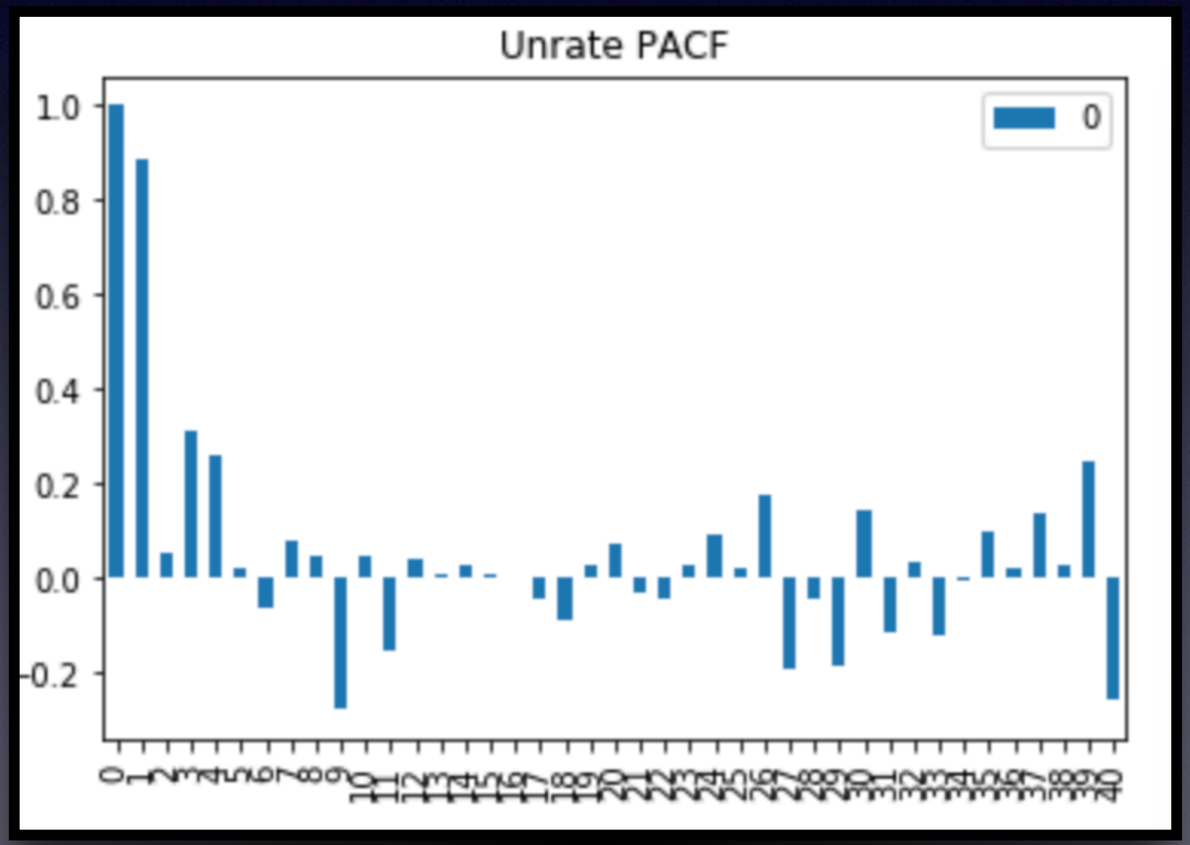
Decompose Seasonality

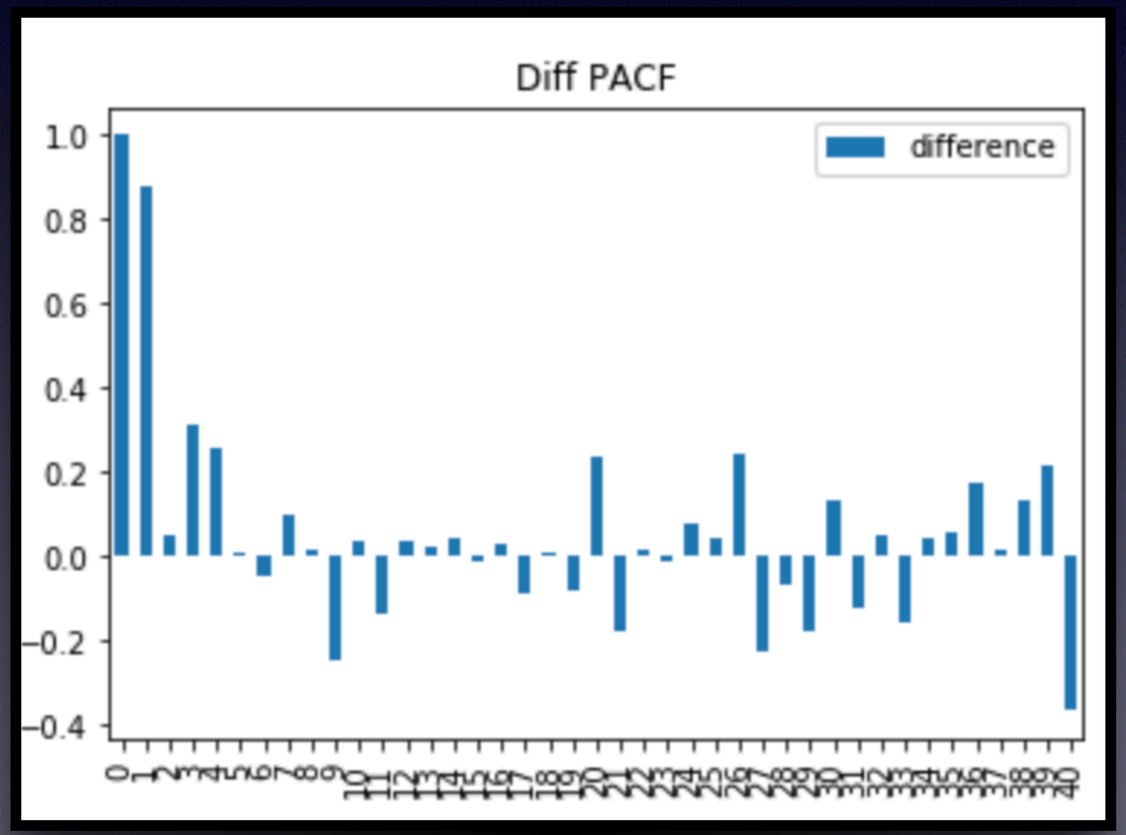
A statistical task that deconstructs a time series into several components.
There is definitely a seaonal trend



PACF original and differenced time series

- The series has 1 strong auto-correlation, therefore the p will be set to 1.





ACDF test

The p-value is greater than 0.05, this is an indication that the model is not stable.

```python
from statsmodels.tsa.stattools import adfuller

# raw data
acdf_test = adfuller(dfIndex['Count'], autolag='AIC')
df_output = pd.Series(acdf_test[0:4], index=[
                        'Test Statistic',
                        'p-value',
                        '#lags used',
                        '#nobs used'])
print('raw data\n', df_output)
for k, v in acdf_test[4].items():
    print(k, v)
```

```
raw data
 Test Statistic     -1.653998
p-value              0.454956
#lags used           3.000000
#nobs used         112.000000
dtype: float64
1% -3.4901313156261384
5% -2.8877122815688776
10% -2.5807296460459184
```

SARIMA Time Series

- Auto arima used to get the best fit.
- (p, d, q) = (0,1,2)x(1,1,1,13)

```
stepwise_fit = auto_arima(dfIndex['Count'],
                          start_p=0, start_q=0,
                          max_p=13, max_q=13, m=13,
                          start_P=0, start_Q=0,
                          seasonal=True,
                          d=1, D=1, trace=True,
                          error_action='ignore',
                          suppress_warnings=True,
                          random_state=42,
                          n_fits=3,
                          stepwise=True)
stepwise_fit.summary()
```

Statespace Model Results

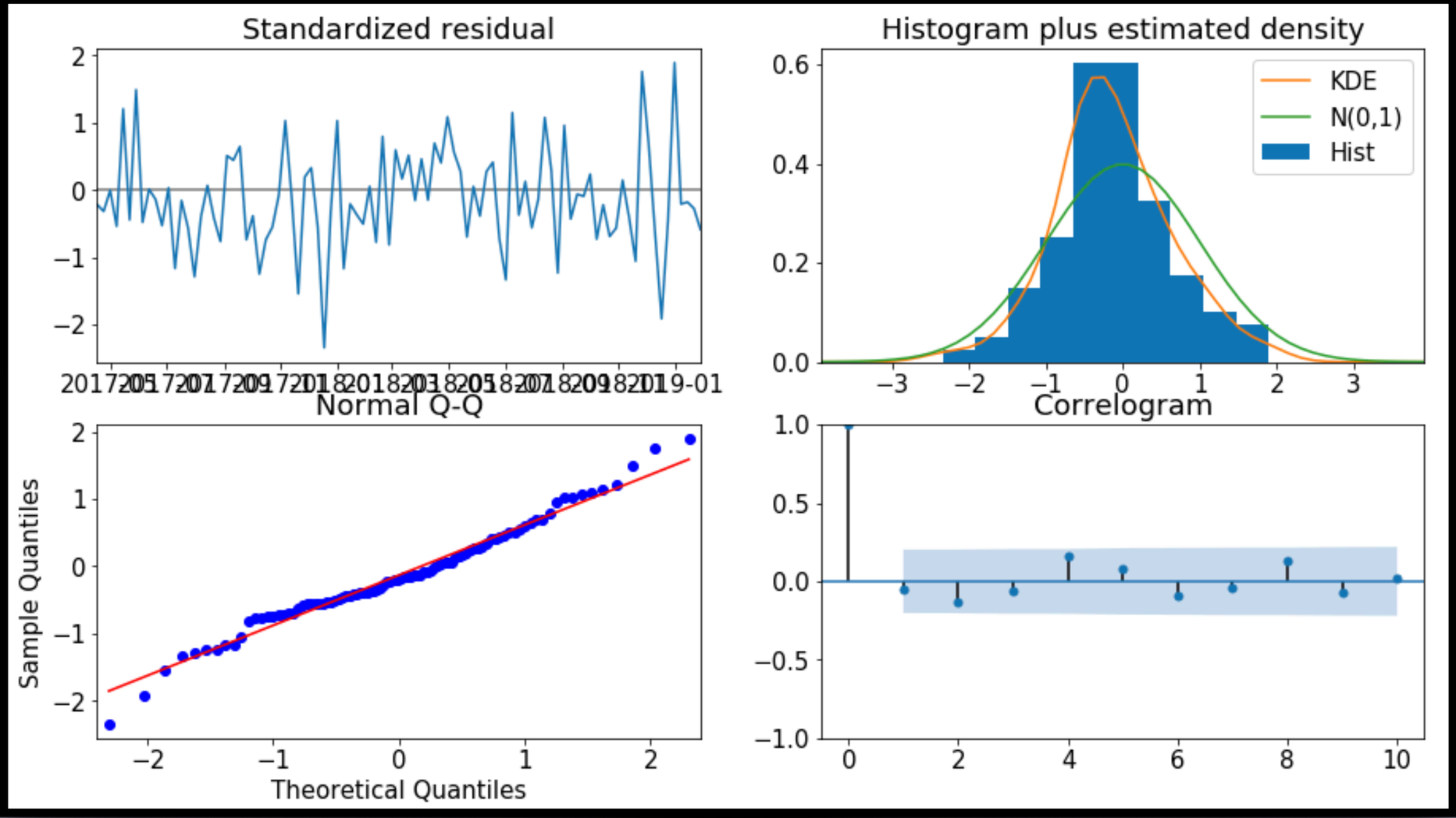
Dep. Variable:	y	**No. Observations:**	116
Model:	SARIMAX(0, 1, 2)x(1, 1, 1, 13)	**Log Likelihood**	-1134.857
Date:	Fri, 28 Jun 2019	**AIC**	2281.713
Time:	16:08:46	**BIC**	2297.463
Sample:	0	**HQIC**	2288.091
	- 116		
Covariance Type:	opg		

	coef	std err	z	P>|z|	[0.025	0.975]
intercept	-315.9325	725.235	-0.436	0.663	-1737.367	1105.502
ma.L1	-0.3026	0.238	-1.272	0.204	-0.769	0.164
ma.L2	-0.2153	0.198	-1.086	0.278	-0.604	0.173
ar.S.L13	-0.2300	0.210	-1.096	0.273	-0.641	0.181
ma.S.L13	-0.6329	0.261	-2.428	0.015	-1.144	-0.122
sigma2	4.291e+08	0.001	7.23e+11	0.000	4.29e+08	4.29e+08

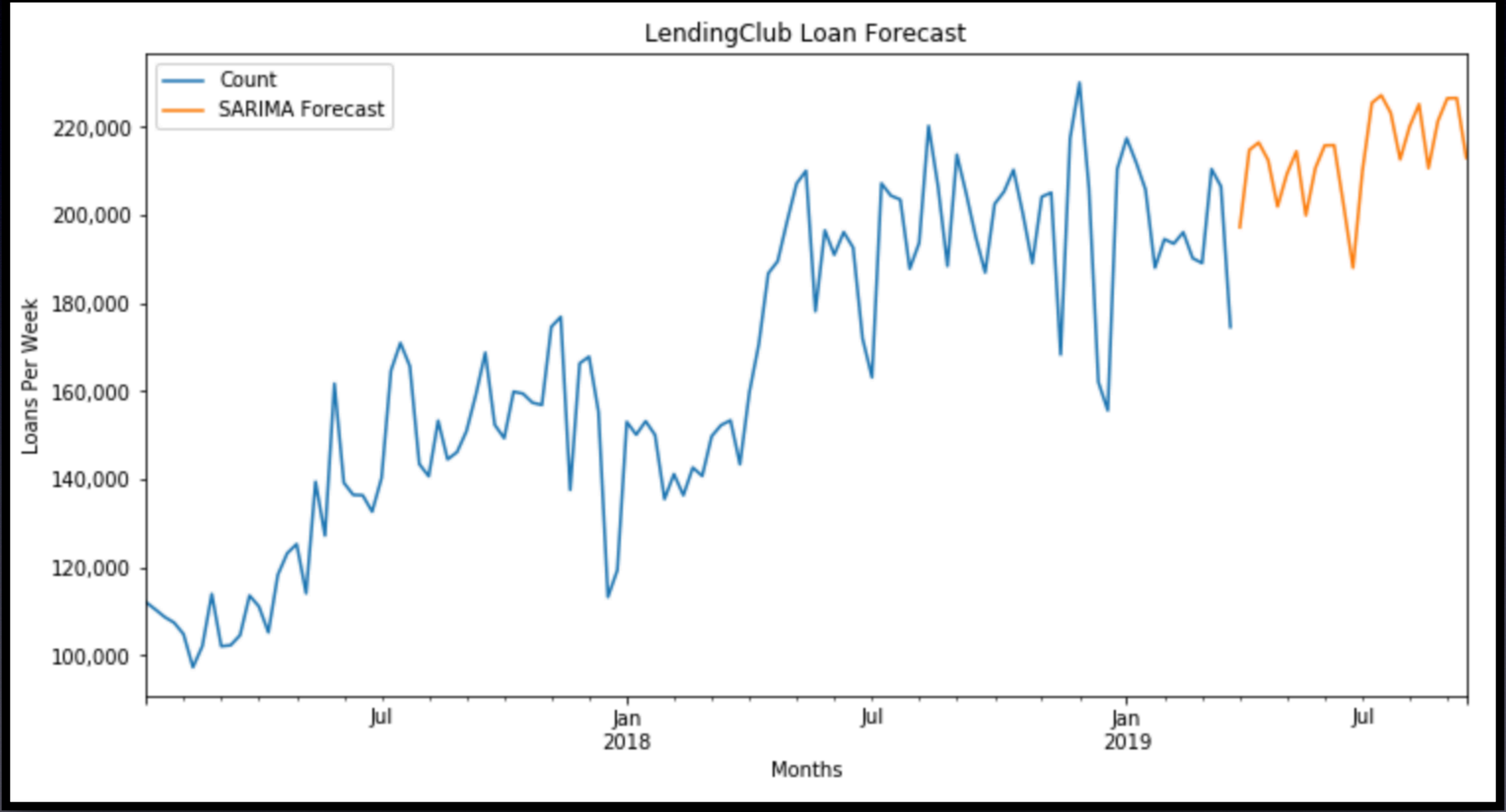
Ljung-Box (Q):	38.70	**Jarque-Bera (JB):**	0.85
Prob(Q):	0.53	**Prob(JB):**	0.65
Heteroskedasticity (H):	1.30	**Skew:**	0.14
Prob(H) (two-sided):	0.45	**Kurtosis:**	3.34

Decompose Seasonality

This is a statistical task that deconstructs a time series into several components.
There is definitely a seasonal trend



SARIMA Forecast



LSTM Time Series

- LSTM is capable of learning long-term dependencies by remembering information for long periods of time.
- It also uses a sigmod function which helps determine what to keep(>0) or discard(<0).
- LSTM will look at the previous 8 weeks(time steps) to predict the 9th week.
- After many different combinations of the parameters, this yielded the best result
  - units = 90
  - dropout = 0.7
  - epochs = 900
  - batch size = 1

```
lstm_model = Sequential()

lstm_model.add(LSTM(units = 90,
                    return_sequences = True,
                    input_shape = (X_train.shape[1], 1)))
lstm_model.add(Dropout(0.7))

lstm_model.add(LSTM(units = 90,return_sequences = True))
lstm_model.add(Dropout(0.7))

lstm_model.add(LSTM(units = 90,return_sequences = True))
lstm_model.add(Dropout(0.7))

lstm_model.add(LSTM(units = 90))
lstm_model.add(Dropout(0.7))

lstm_model.add(Dense(units = 1))


lstm_model.compile(optimizer = 'adam',
                   loss = 'mean_squared_error')

history=lstm_model.fit(X_train,
               y_train,
               epochs = 900,
               batch_size = 1)
```

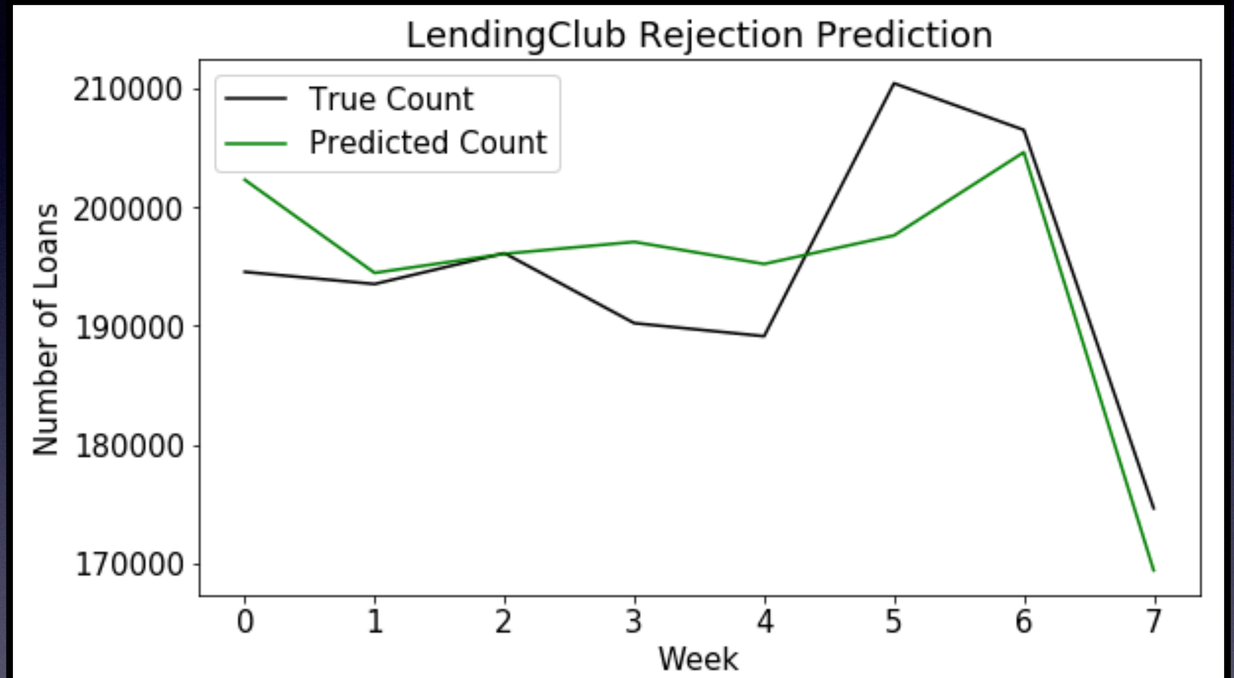
LSTM Prediction

Well the model didn't do good at all.
The first, second, sixth and seventh weeks predicted great.

The rest of the weeks predictions were not great.

I want to research more on how to tune the model better.

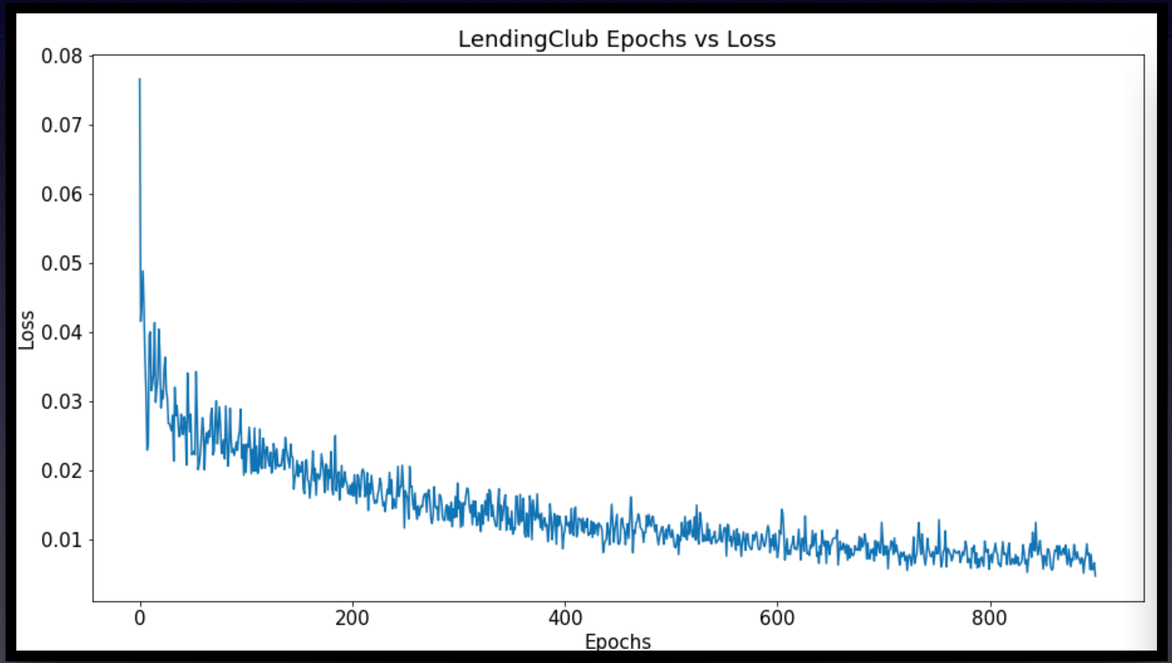
The Train and Test RMSE are not close at all, they are off by ~1000.

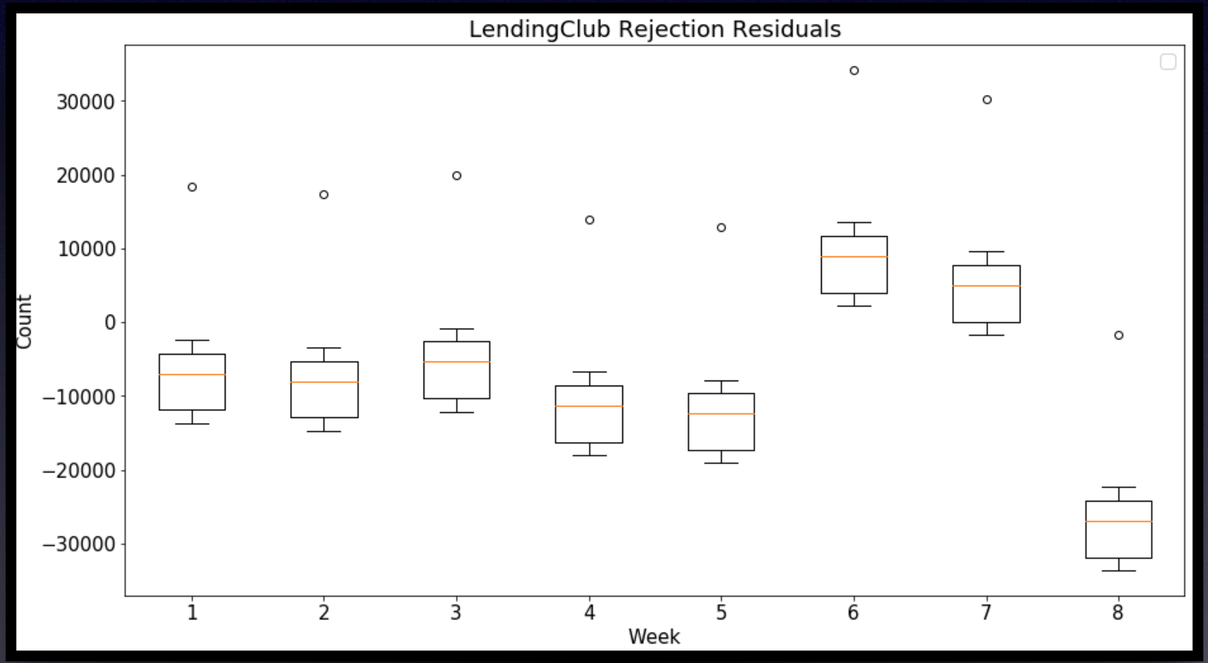



```
RMSE Train Score: 5762.00
MSE Train Score: 33200682.4385

RMSE Test Score: 6534.2960
MSE Test Score: 42697024.6312
```

Loss and Residuals

There is a sharp drop off for the loss then it trys to level out





Practical Use

- As an investor, knowing the projected good loans per week is a good indication on how hard you need to work to hit your target.
- Lending club can get new investors when they show them the projected number of loans per month.
- Knowing the number of rejected loans
- They bring borrowers and investors together transforming the way people access credit.
- They were established in 2007 helping borrows take control of their debt, grow small business and investing in their future.

Conclusion & Future Work

- As an underwriter, the supervised model can predict a good loans 80% of the time. All models had a hard time with predicting False Positives(applications which were predicted as good but they were actually bad loans.)
- Even though the supervised learning accuracy score improves 4% with ADA Boost, it is worth testing the model using different sampling methods(up/down sampling) to balance the target variable.
- Of the 87,923 loan applications, KMeans and MiniBatch Kmeans were able to find ~38,000 good loan applications which is not that great.
- When trying to find unknown clusters with KMeans and Mean Shift in the data, Kmeans found 2 clusters and Mean Shift didn’t find any clusters.

Conclusion & Future Work

- SARIMA forecasted 8 weeks of rejected loan applications. To test the model, I would like to get the July loan data to see if the forecast was accurate.

- LSTM needs to be tuned better. Them root mean square for the train and test had a differnce of ~ 1000.

Questions & Suggestion?

https://www.linkedin.com/in/charla-gaddy/