LendingClub Analysis

by Charla Gaddy

About LendingClub

- They were established in 2007 helping borrows take control of their debt, grow small businesses and invest in their future.

- LendingClub is a US peer-to-peer lending company, that brings borrowers and investors together transforming the way people access credit.

Questions Asked

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

- How well can a supervised model predict good loan applications?

Unsupervised algorithm will cluster the top 10 features from Random Forest supervised model.

- Can KMeans and Mean Shift models 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?

Questions Asked cont

LendingClub provides data on their website for both approved and rejected loan. Both datasets will be used for this project.

First, supervised and unsupervised learning will model the approved loans. Supervised learning will predict the good versus bad loans.

Then, how many clusters can KMeans and Mean Shift predict on the top 10 features of Random Forest. Also, are they able to cluster the good loans?

Next the rejected loan data will forecast the number of loan rejections for the next 2 months(8 weeks). LSTM model is created to predict the loans

Data

All data was downloaded from the LendingClub website: https://www.lendingclub.com/info/download-data.action. There are links for approved and rejected loan applications. I will be using both datasets.

Approved Loans

This dataset has 610,917 rows and 144 columns. Each row represents a loan and the 144 columns are the various attributes used to determine if you are eligible for a loan. After data cleaning the data, such as removing missing values and then applying label encoding to the categorical data, the final dataset has 87,932 rows and 42 columns. For this dataset Random Forest and XGBoost will predict the number of loans approved. The accuracy, ROC and other metrics will be evaluated to see which model is better at predictions. Kmeans and Mean Shift will then try to cluster the top 10 features of the Random Forest model to see if there are any unknown features that can help determine loan status.

Rejected Loans

This dataset has 19,158,655 rows and 2 columns. Each row represents a loan and the two columns are 1.) date of the loan application 2.) the number of loans for that day. This dataset was converted in to weeks. The final dataset now has 117 rows. SARIMA will predict 8 weeks of loans as well as forecast 12 weeks out. LSTM will also predict 8 weeks of loans

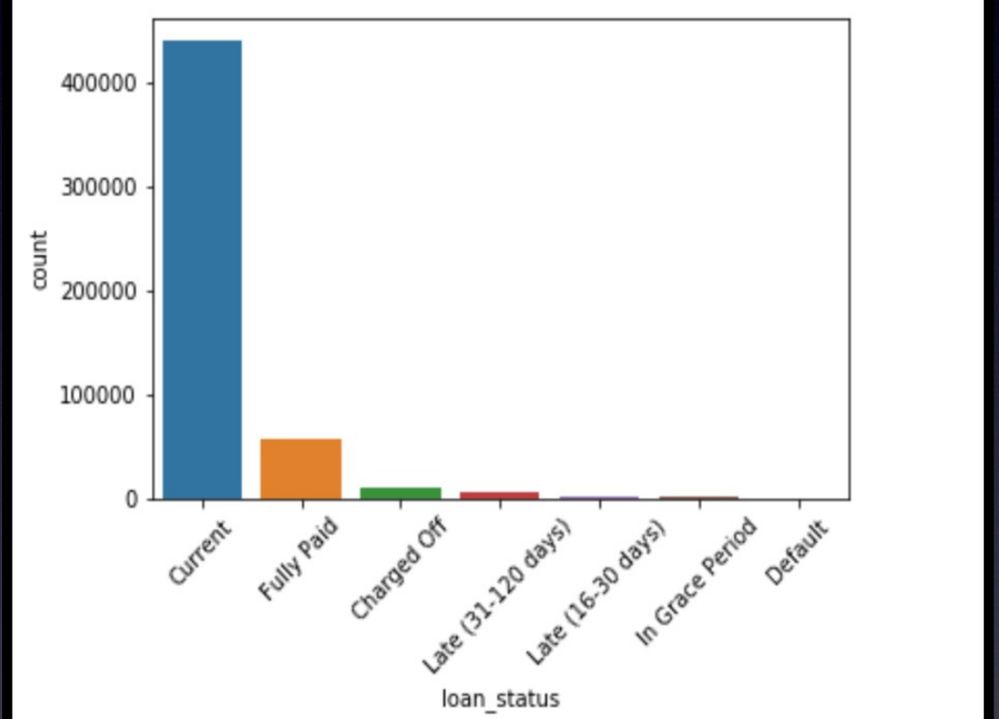
```
DatetimeIndex(['2017-01-02', '2017-01-09', '2017-01-16', '2017-01-23',
               '2017-01-30', '2017-02-06', '2017-02-13', '2017-02-20',
               '2017-02-27', '2017-03-06',
               ...
               '2019-01-21', '2019-01-28', '2019-02-04', '2019-02-11',
               '2019-02-18', '2019-02-25', '2019-03-04', '2019-03-11',
               '2019-03-18', '2019-03-25'],
              dtype='datetime64[ns]', name='Application_Date', length=117, freq=None)
```

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) and 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
```

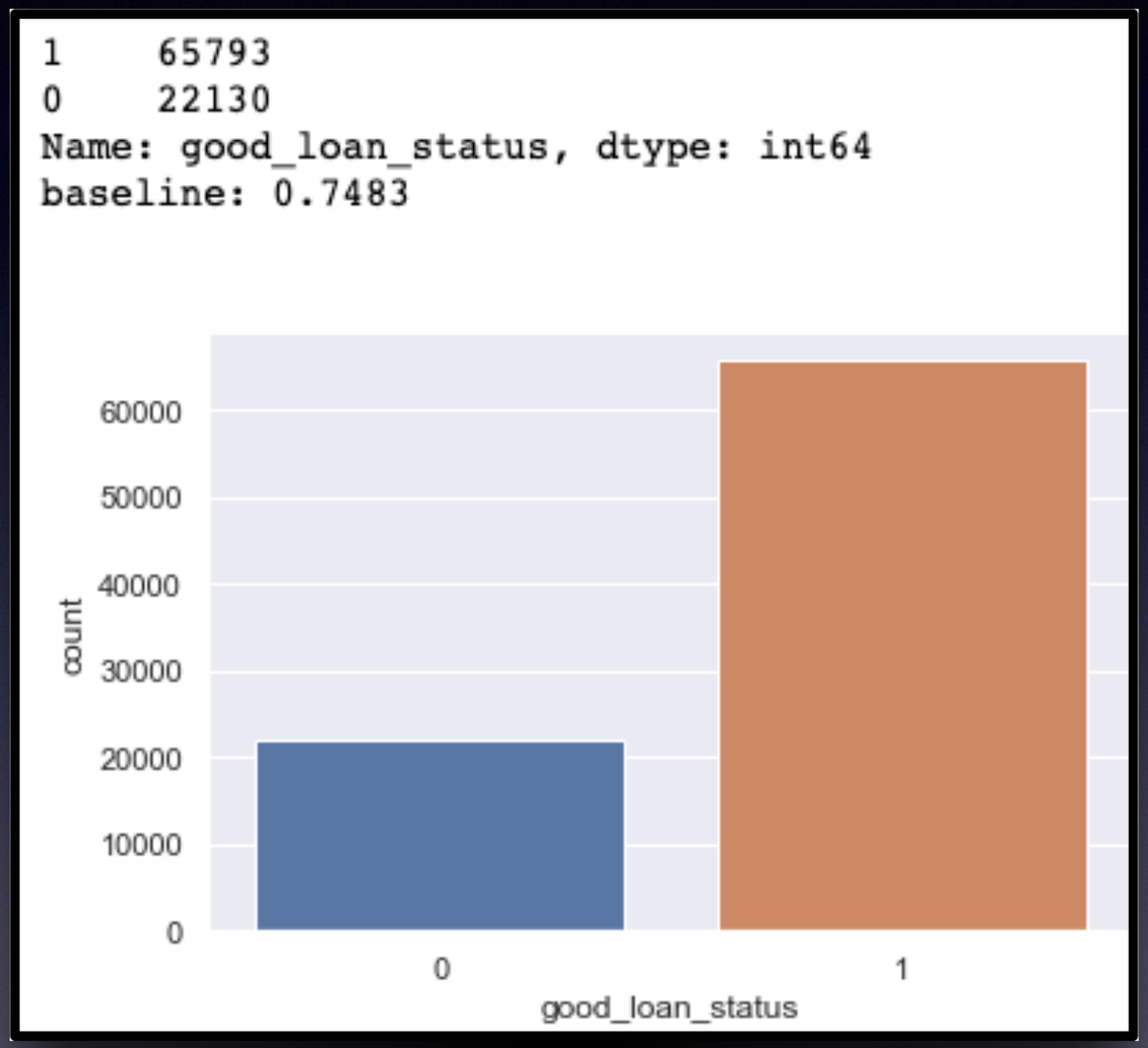



```
1 df = df[df['loan_status'] != 'Current']
2 df = df[df['loan_status'] != 'In Grace Period']
3 df = df[df['loan_status'] != 'Late (16-30 days)']
```

Data

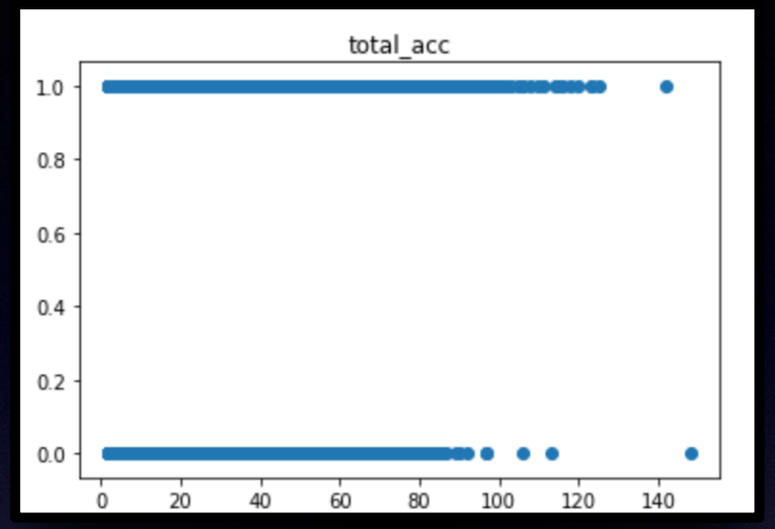
- The baseline has accuracy of 74.83%
- The ratio for good and bad loans is very large almost 3 to 1.

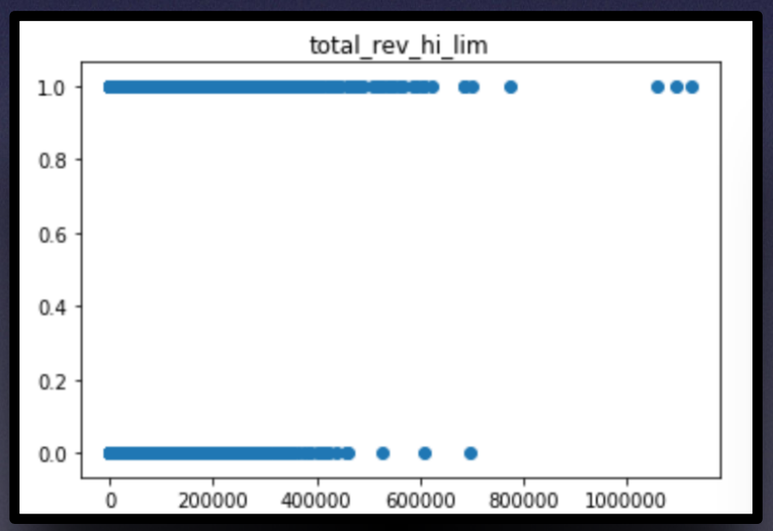
```
1    65793
0    22130
Name: good_loan_status, dtype: int64
baseline: 0.7483
```

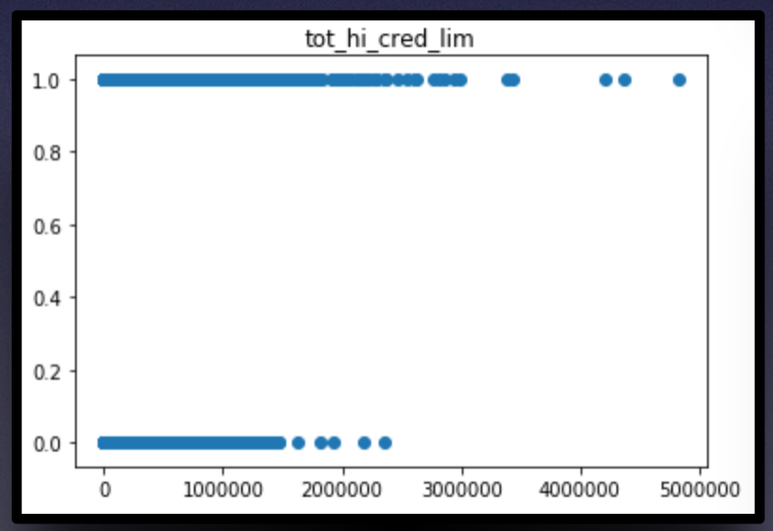


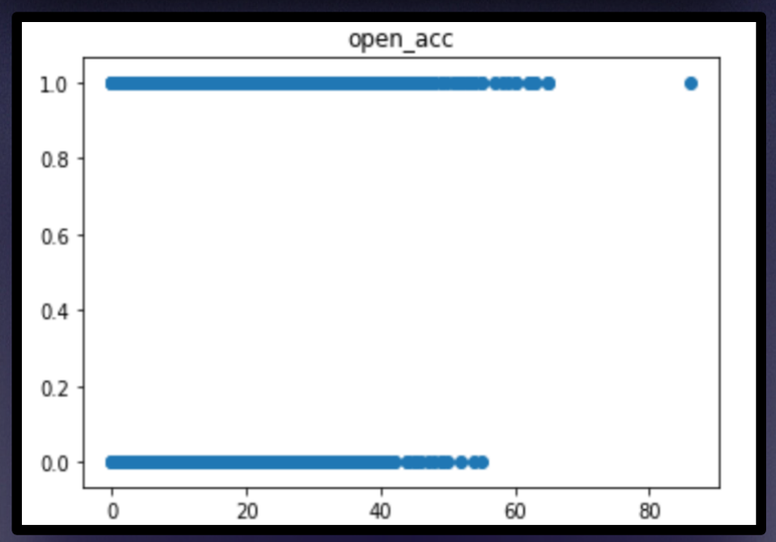
A few feature plots of good vs bad loan status

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

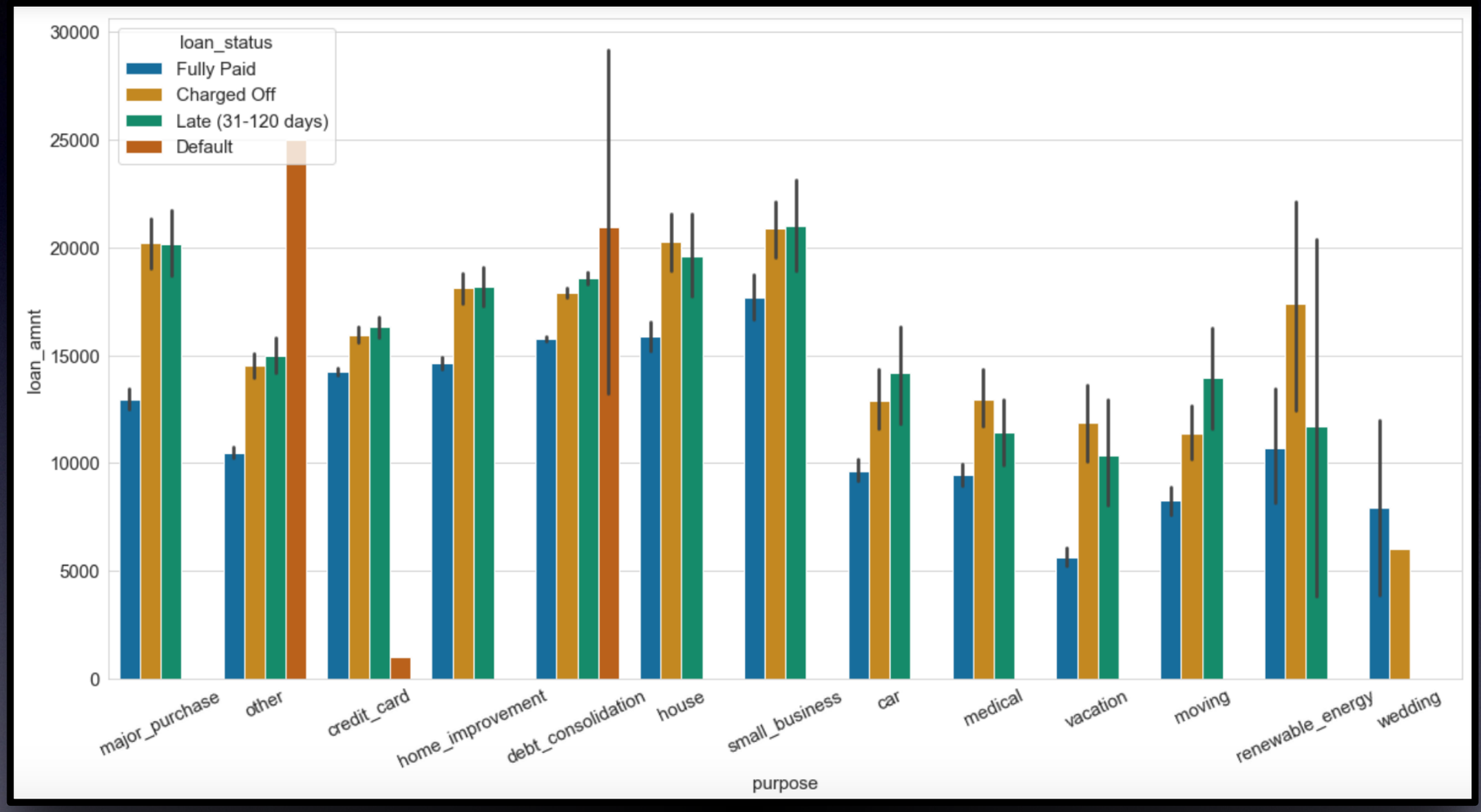




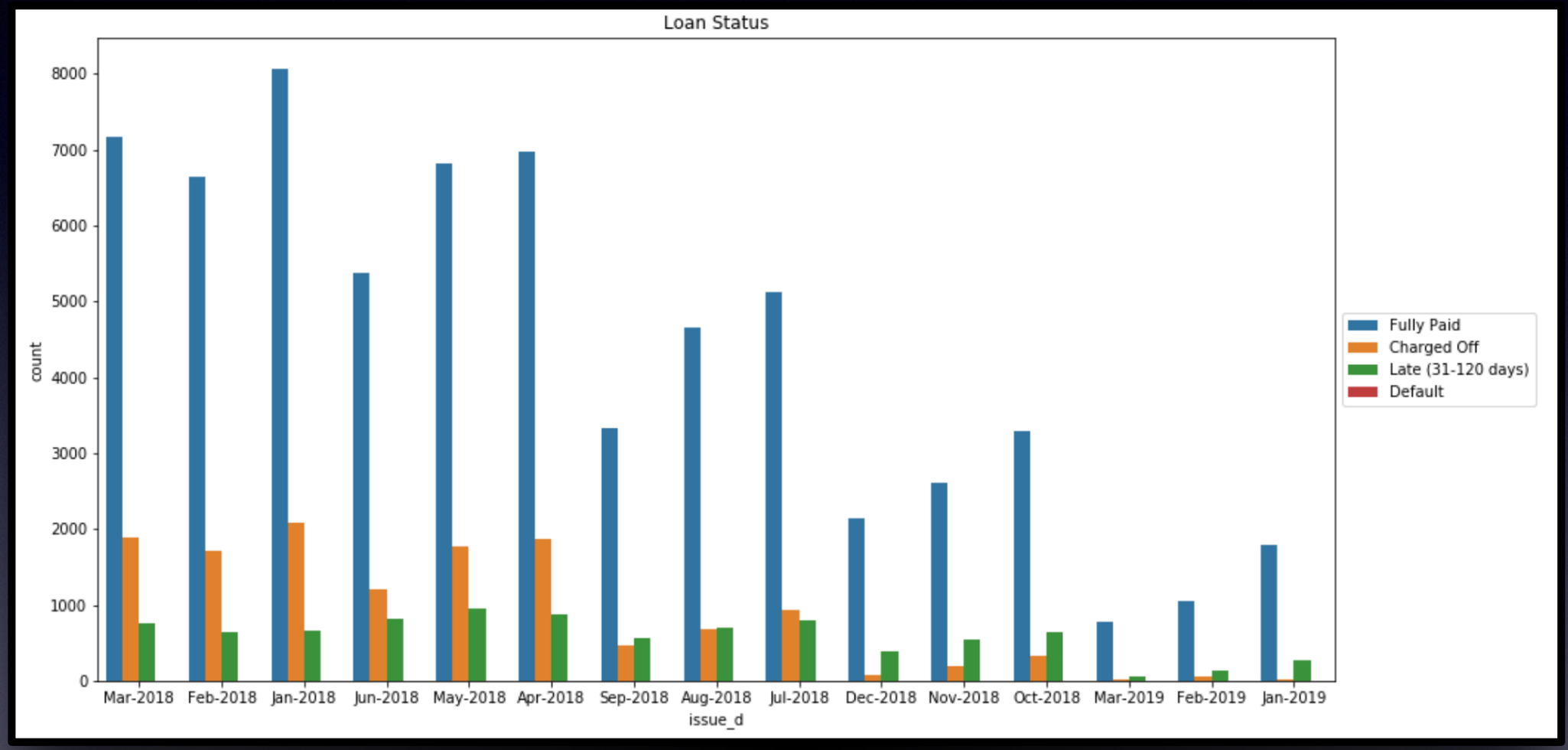




- debt consolidation loans go into default more often than not
- all loans except for wedding loan are more likely to be late, in default or charged off



- 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, ADABoost Random Forest and XGBoost are the algorithms used to determine the percentage of good loans that will be issued. Grid-search was used to find the best hyper-parameters for each model. Below is the result.

```
rf_final = RandomForestClassifier(max_depth=8,
min_samples_leaf=10,
min_samples_split=60,
max_features=50,
n_estimators=20)
```

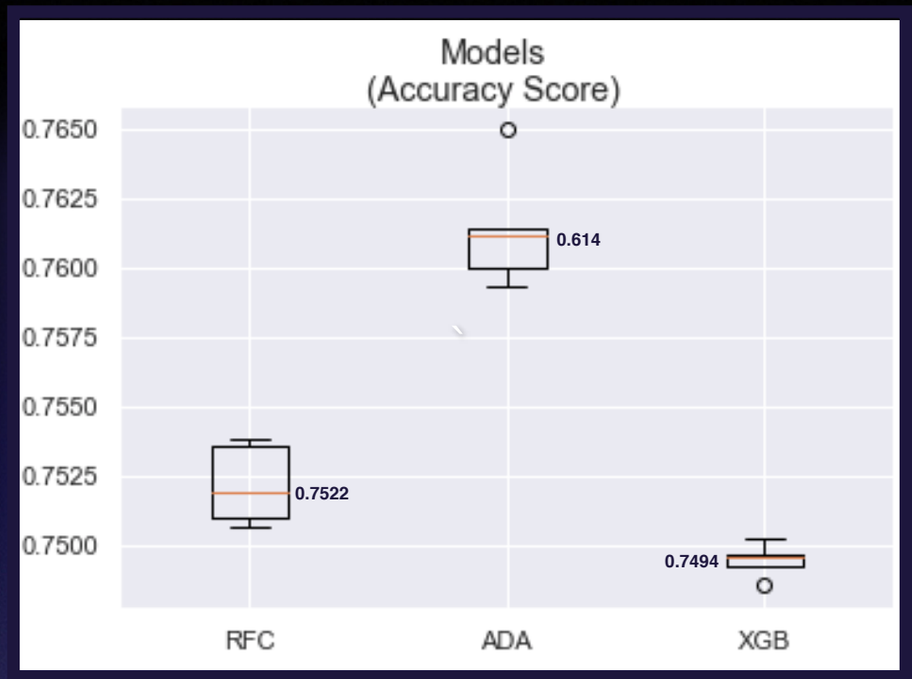
```
ada_final = AdaBoostClassifier(base_estimator=rf_final,
n_estimators=14)
```

```
xgb_final = XGBClassifier(max_depth=6,
n_estimators=8,
learning_rate=0.0125,
subsample=0.03,
colsample_bytree=0.05,
colsample_bylevel=0.005)
```

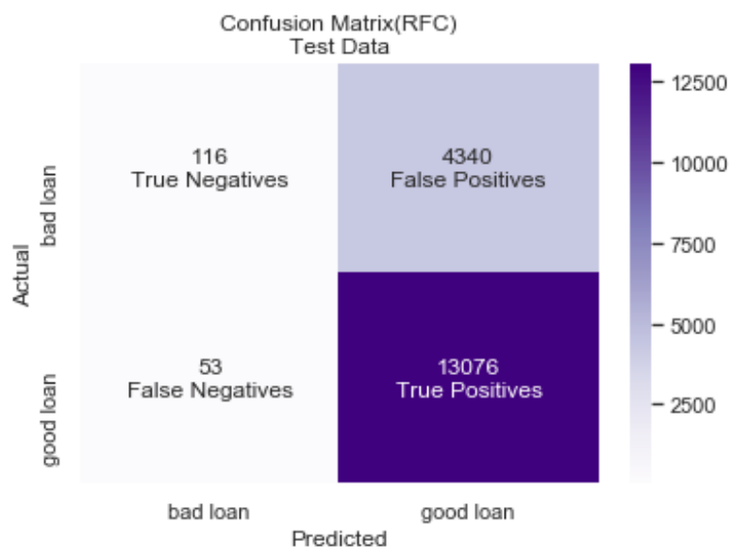
ADA Boosted Random Forest had the best score

- recision is the ratio of correctly predicted positive observations to the total predicted positive observations. The question that this metric answers is of all loans that are labeled as good, how many actually were good? The RFC model has a ~75% precision, which is pretty good.

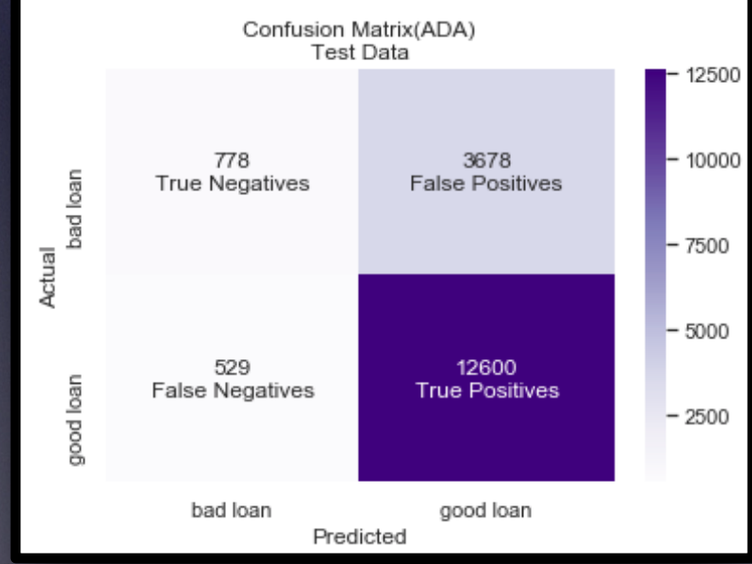
- Recall is the ratio of correctly predicted positive observations to the all observations. The question recall answers is: Of all the loans that are truly good, how many did the model label correctly? The recall for all the models is very high.



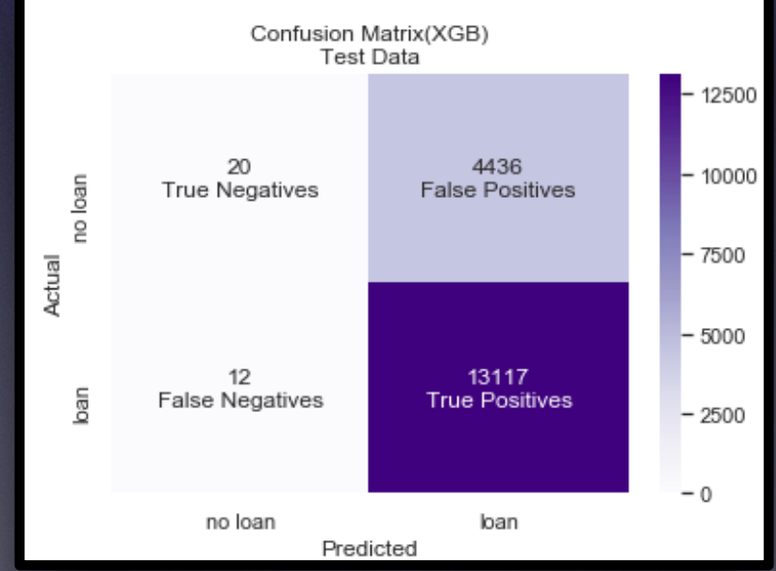

```
Precision:      0.7508
Recall:         0.9960
F1 Score:       0.8562
```




```
Precision:      0.7741
Recall:         0.9597
F1 Score:       0.8569
```

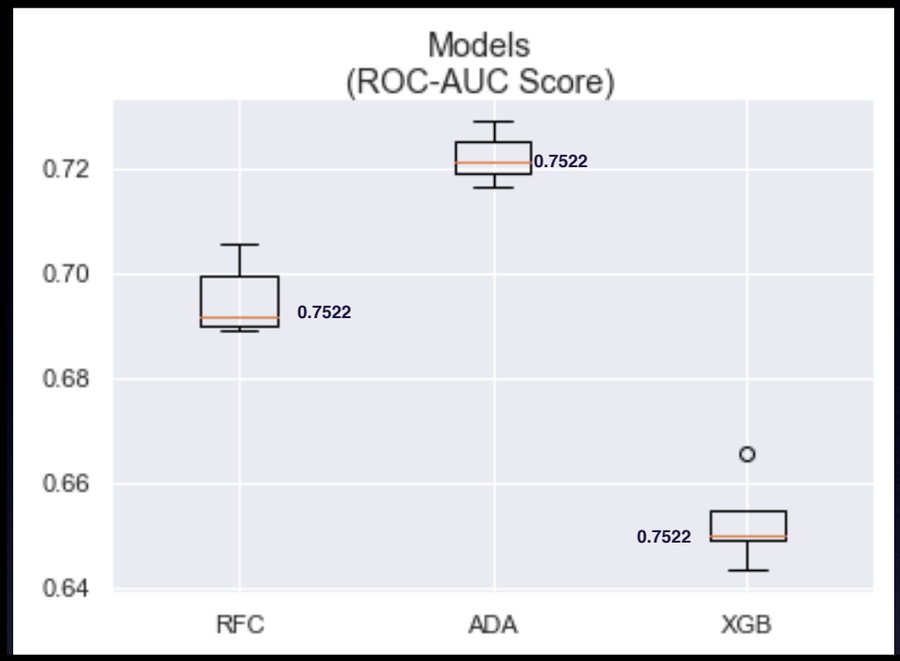


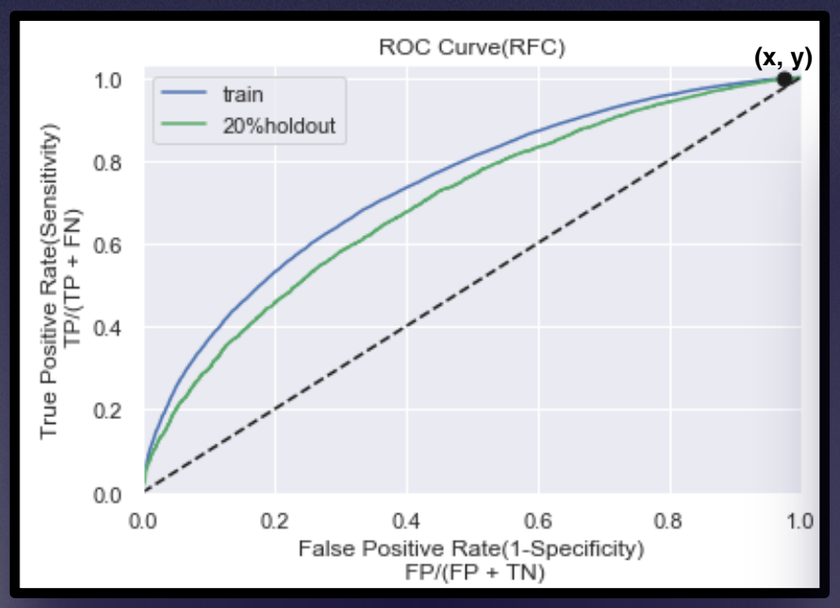

```
Precision:      0.7473
Recall:         0.9991
F1 Score:       0.8550
```

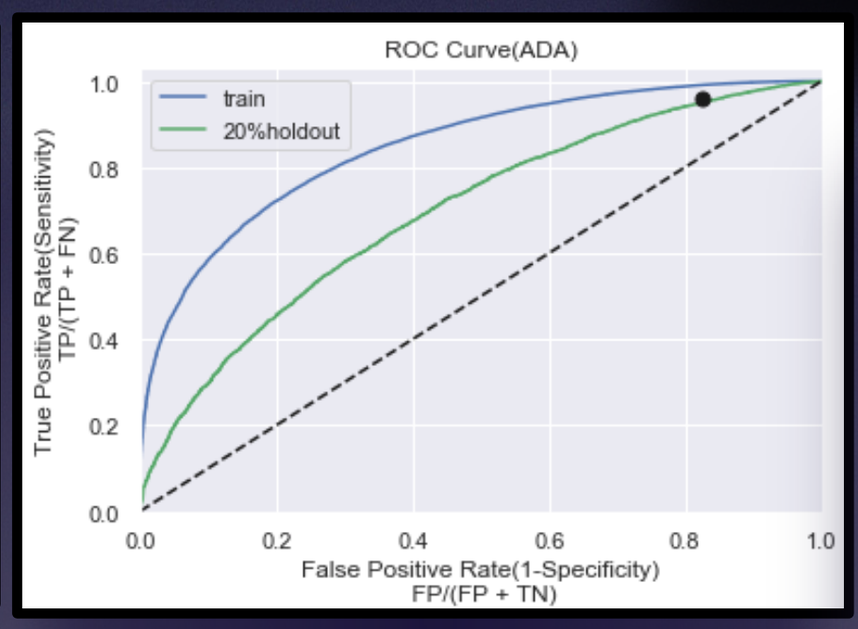


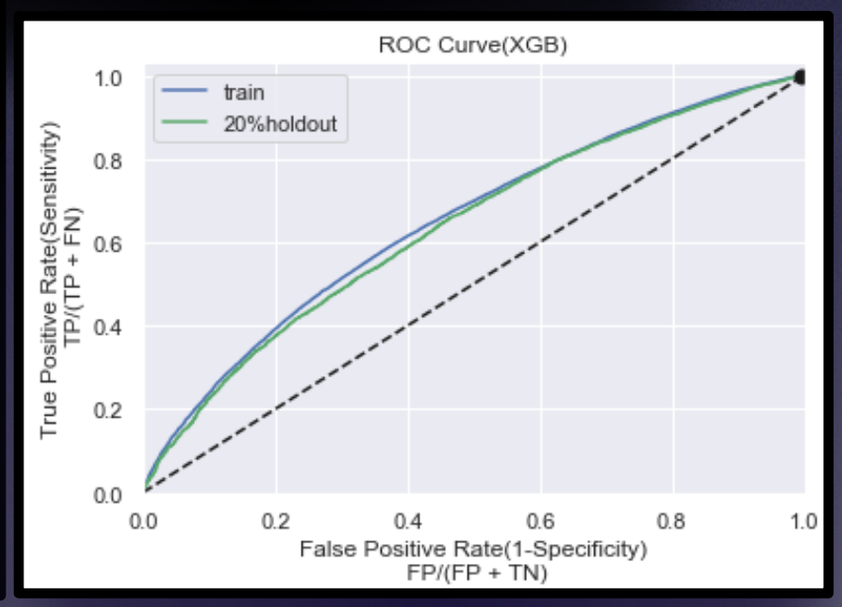
ROC/AUC scores

- 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.
- The best ROC is of the ADA boost model, the true positives are increasing faster than false positive rate.
- x = fp/(fp+tn) y = tp/(tp+fn)



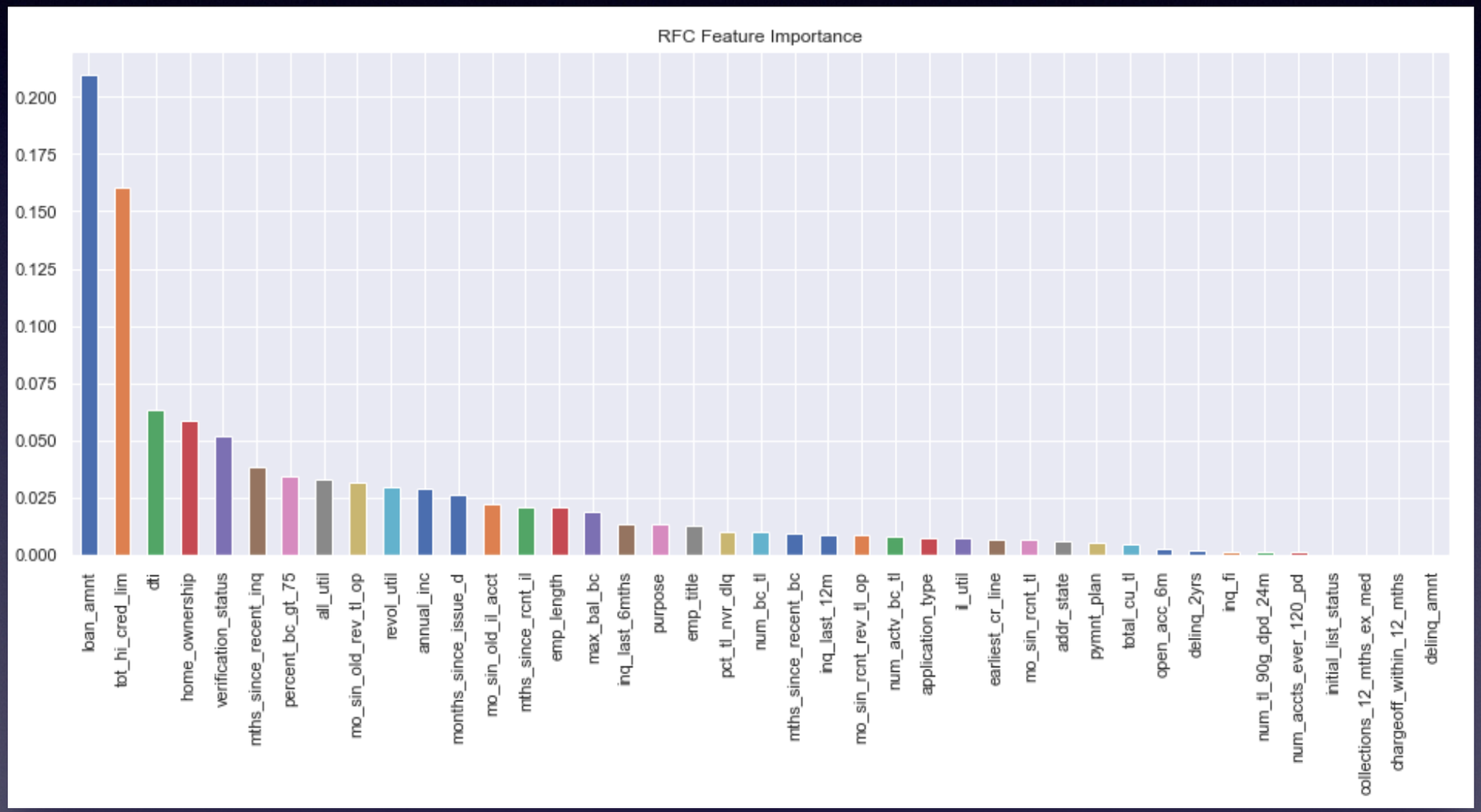






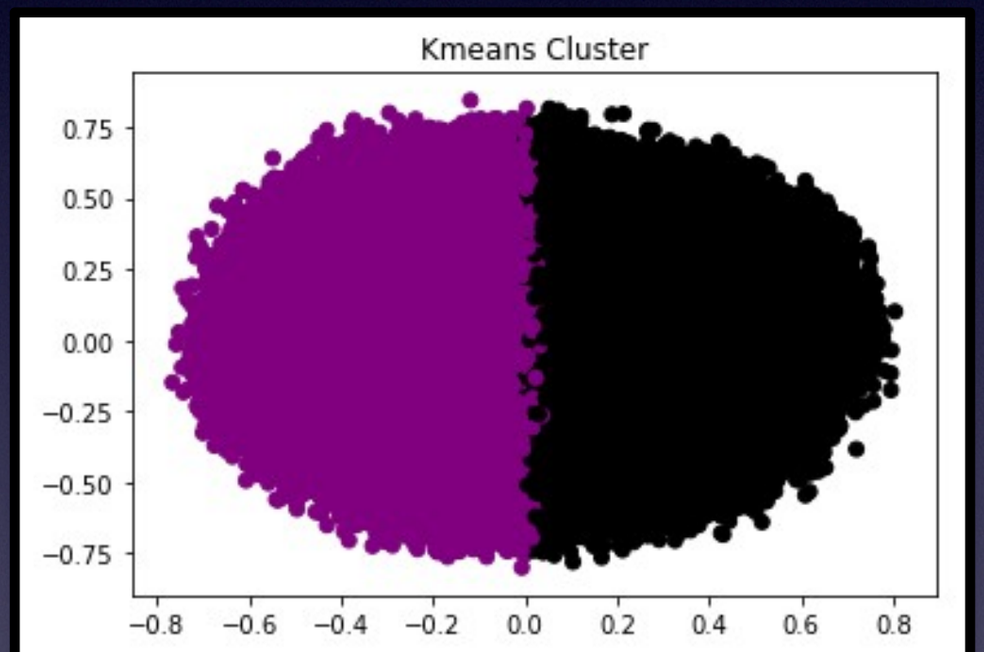
Feature Importance

The top 10 features will be used for Kmeans & Mean Shift clustering.



KMeans & KMeans Minibatch

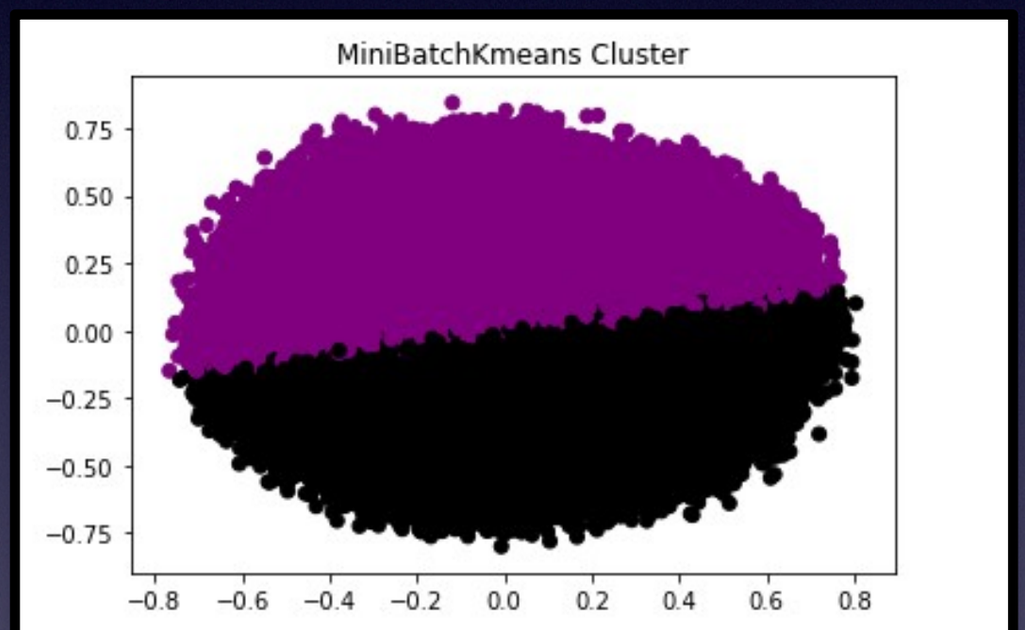
44,298 (33717+10581) correctly classified
43,625 (32076+11549) mis-classified



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 3 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	loan_amnt	0
1	verification_status	0
2	all_util	0
3	annual_inc	0
4	tot_hi_cred_lim	1
5	home_ownership	2
6	dti	2
7	percent_bc_gt_75	2
8	mths_since_recent_inq	2
9	mo_sin_old_rev_tl_op	3

Kmeans Clusters

Time Series

Rejected Loans (weekly)

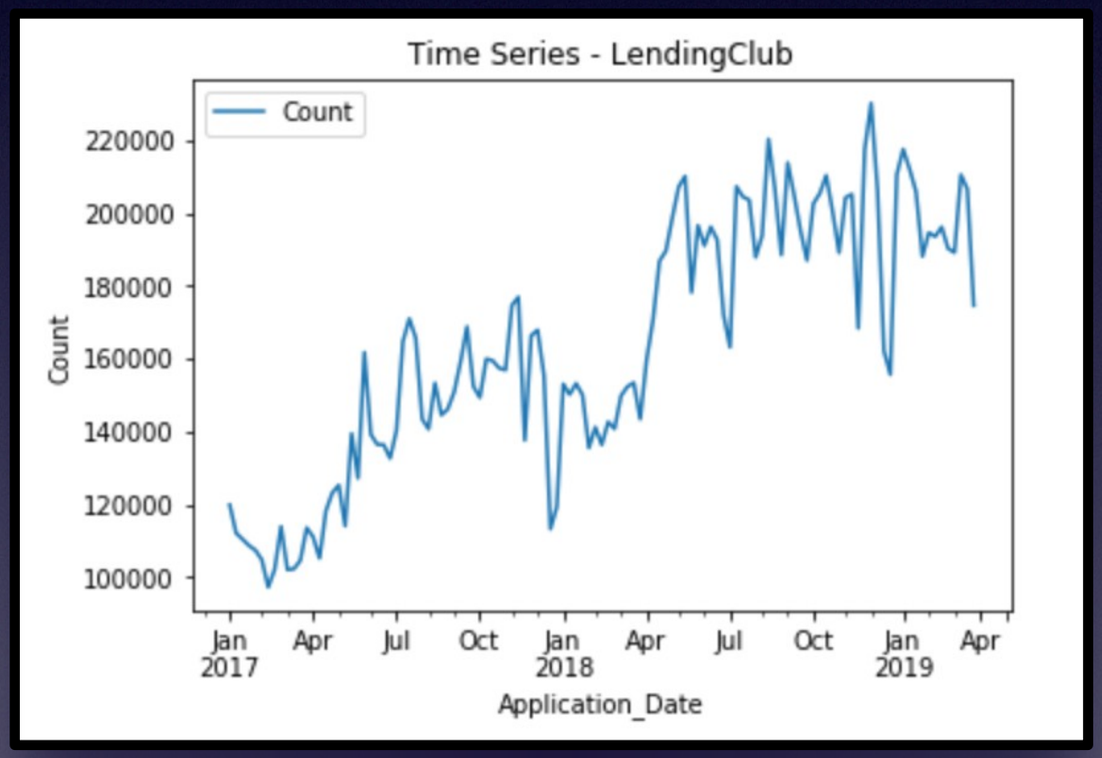
Application_Date	Count
2017-01-02	119903
2017-01-09	112116
2017-01-16	110457
2017-01-23	108717
2017-01-30	107435

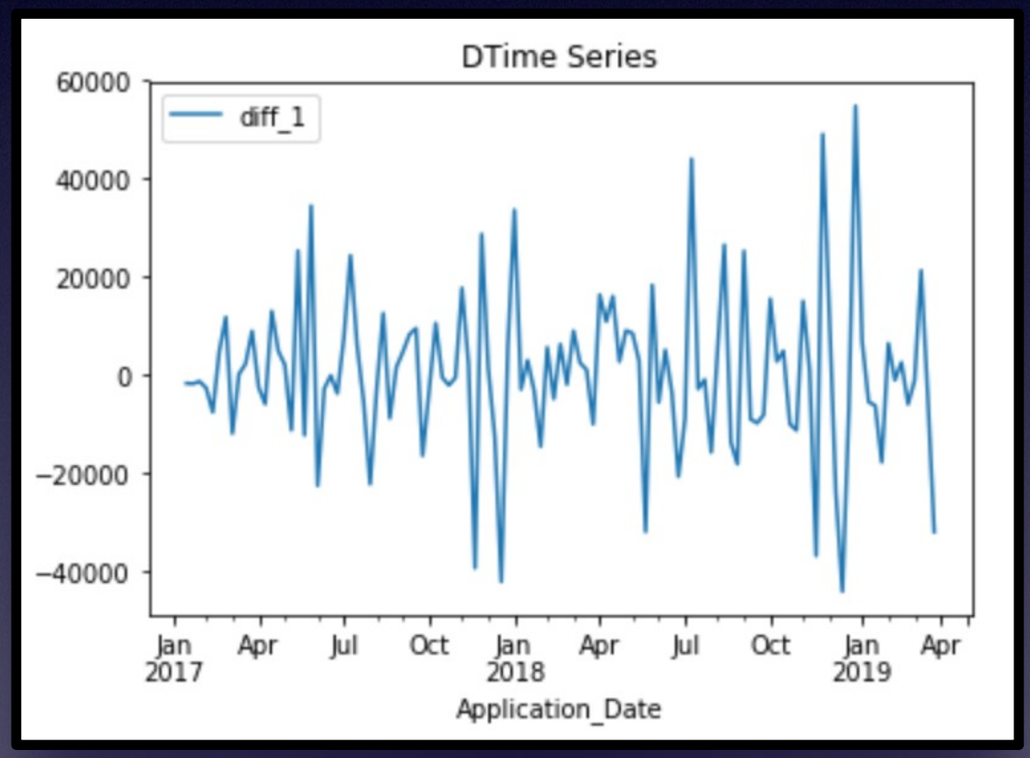
- For time series we are using the rejected loan application dataset. The original dataset has 19158655 rows but after converting the date to weeks, it is reduced to 117 rows.

- SARIMA will predict 8 weeks of rejections as well as forecast rejection 12 weeks out. LSTM will also predict 8 weeks of loans using the same data.

Original and Differenced time series

- The data is not stationary therefore a difference of one is done to make it stationary.

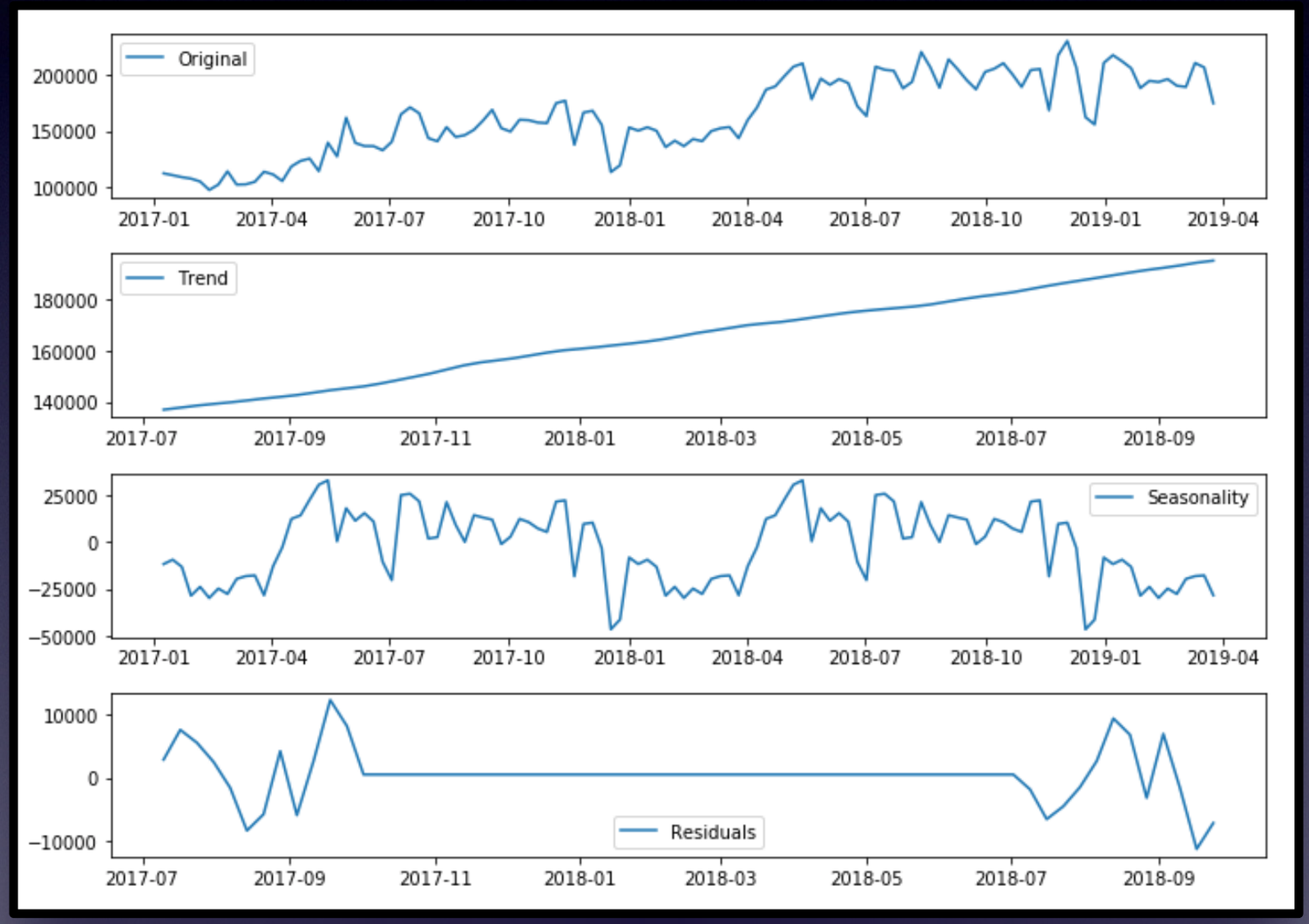




Decompose Seasonality

A statistical task that deconstructs a time series into several components.

Here we can see that the trend and seasonality are separated out from data and we can model the residuals.



ACDF test

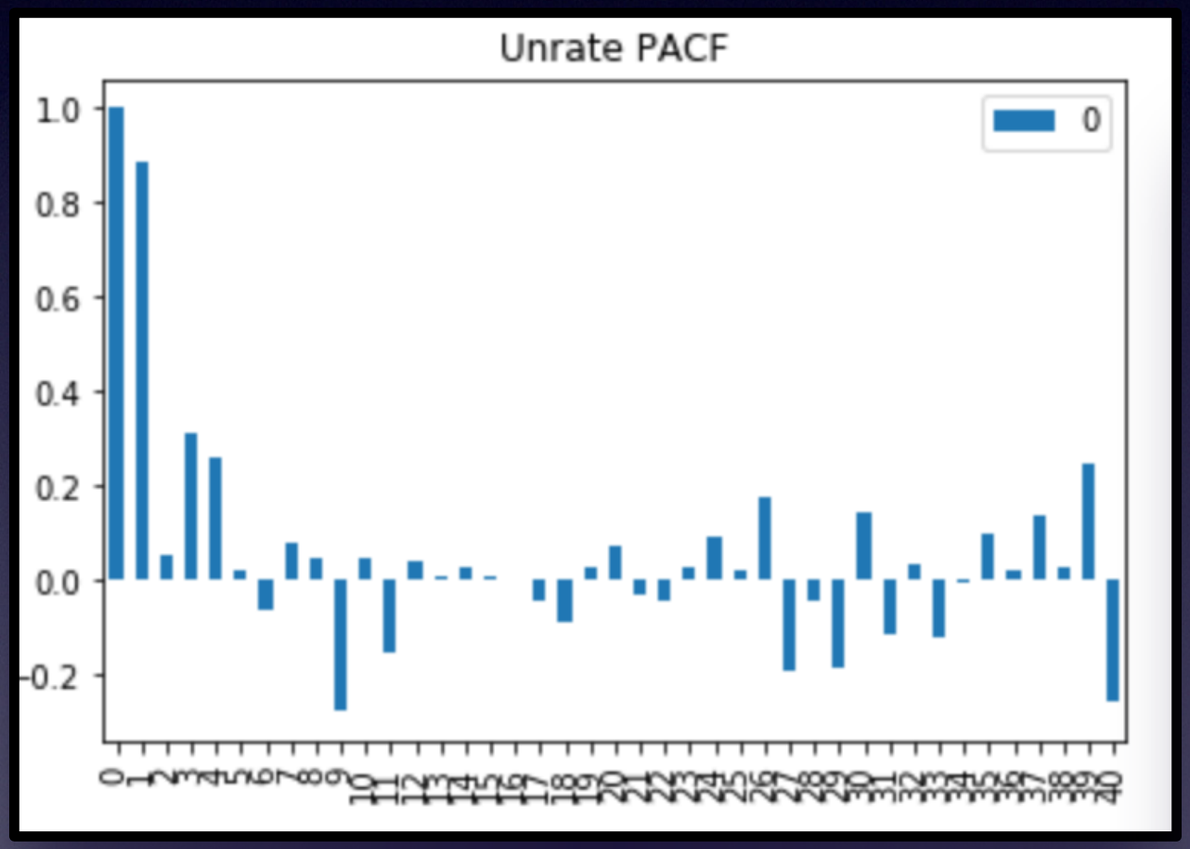
The p-value is greater than 0.05, this is an indication that the model is not stable.

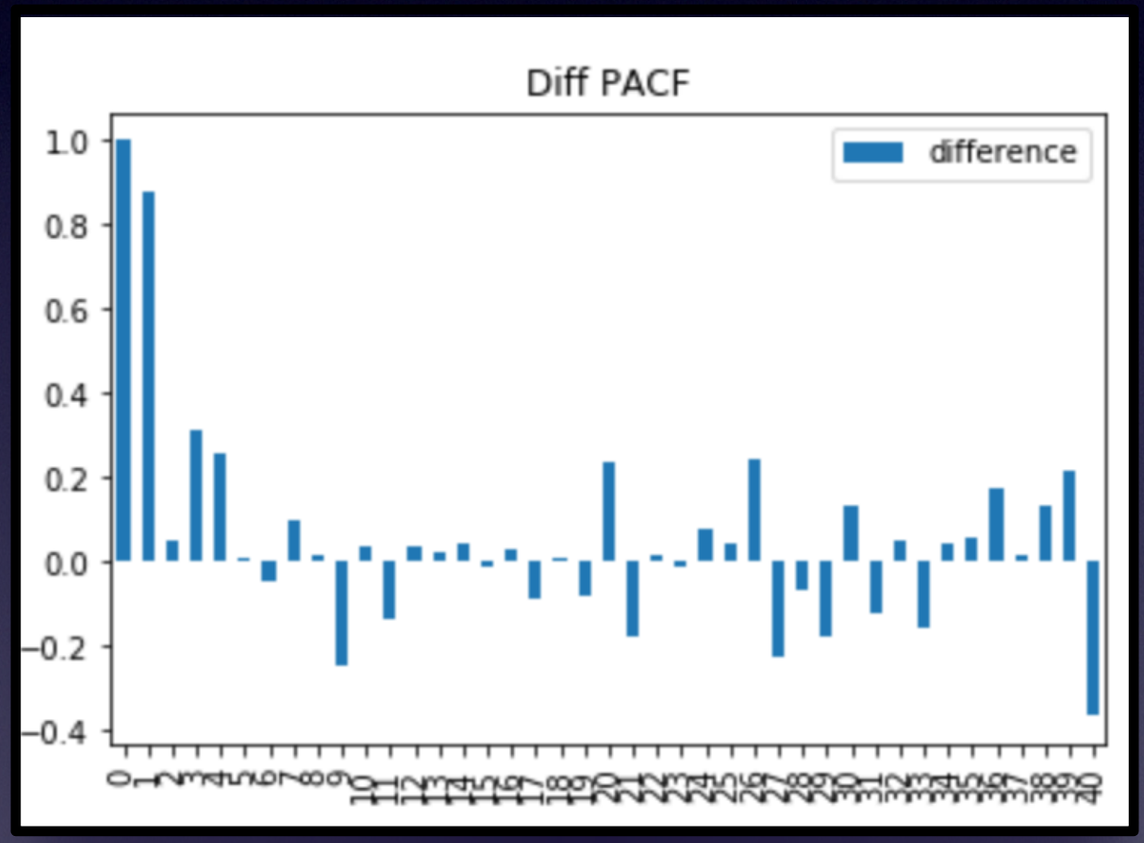
```
1  from statsmodels.tsa.stattools import adfuller
2
3  # raw data
4  acdf_test = adfuller(dfIndex['Count'], autolag='AIC')
5  df_output = pd.Series(acdf_test[0:4], index=[
6                        'Test Statistic',
7                        'p-value',
8                        '#lags used',
9                        '#nobs used'])
10 print('raw data\n', df_output)
11 for k, v in acdf_test[4].items():
12     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
```

PACF original and differenced time series

The series has a strong correlation at 1 and a strong negative at 9.
Auto-arima will be used to find the best model.





SARIMA Time Series

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

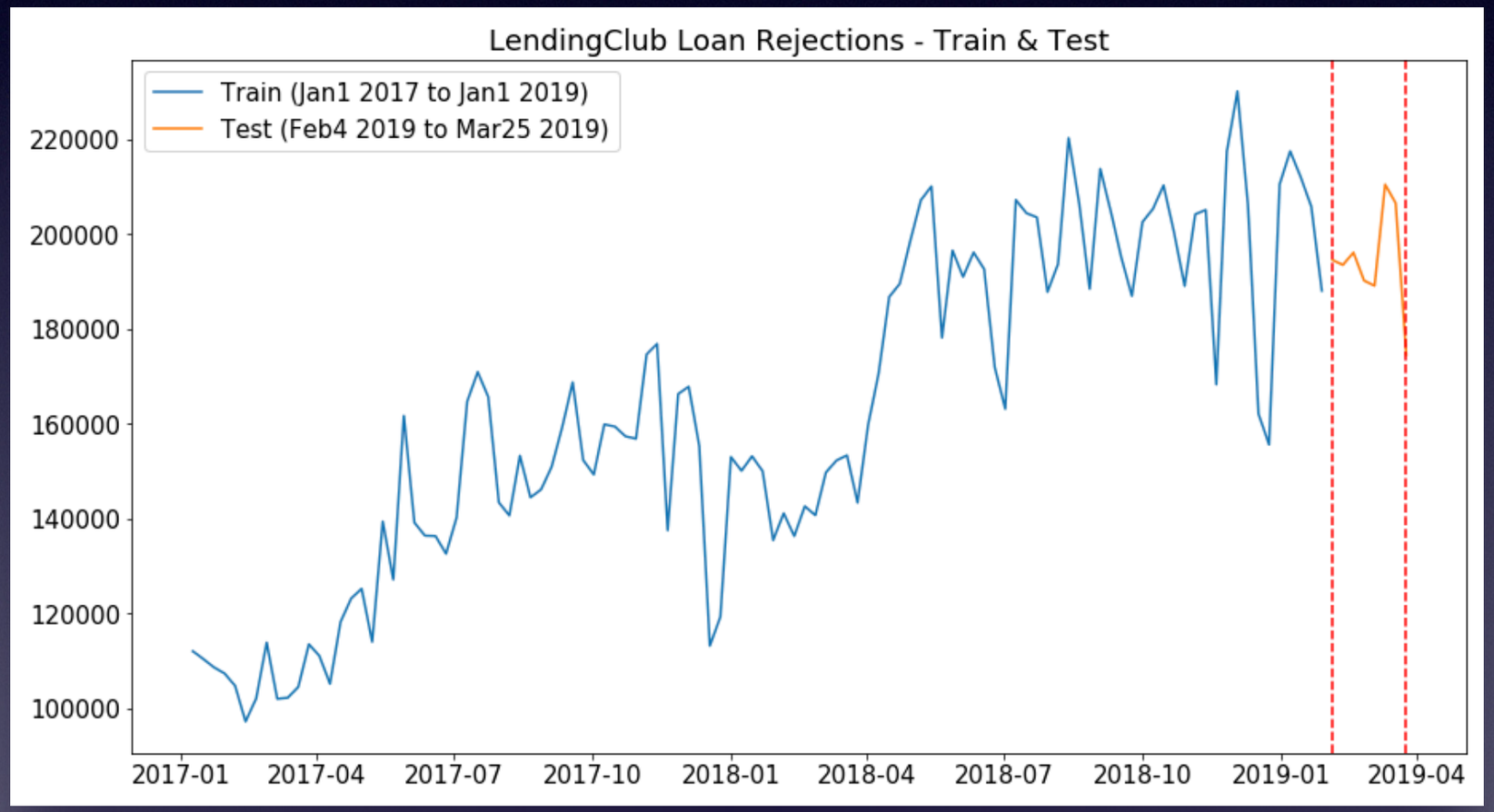
SARIMA Time Series (p,d,q) results

```
stepwise_model = auto_arima(dfIndex['Rejected_Loan_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_model.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:	Sun, 14 Jul 2019	**AIC**	2281.713			
Time:	12:19:48	**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			

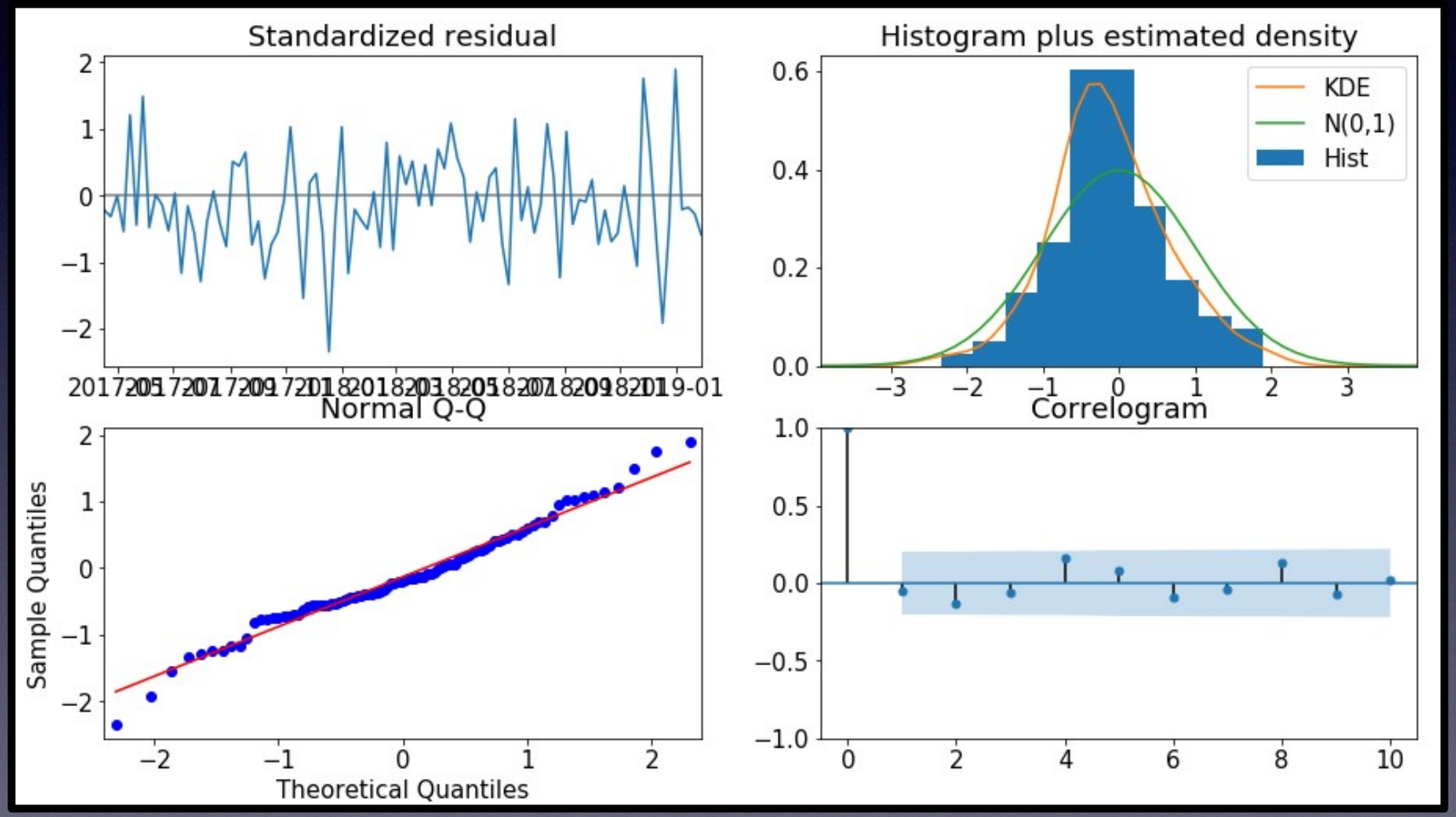
Forecast 8 weeks Loan Rejections

Loan dataset will have 116 rows.
train, test = dfIndex[:108], dfIndex[108:]
Train: 108 rows weekly loan rejections from Jan1 2017 to Jan1 2019
Test: 8 rows of weekly loan rejections from Feb4 2019 to Mar25 2019

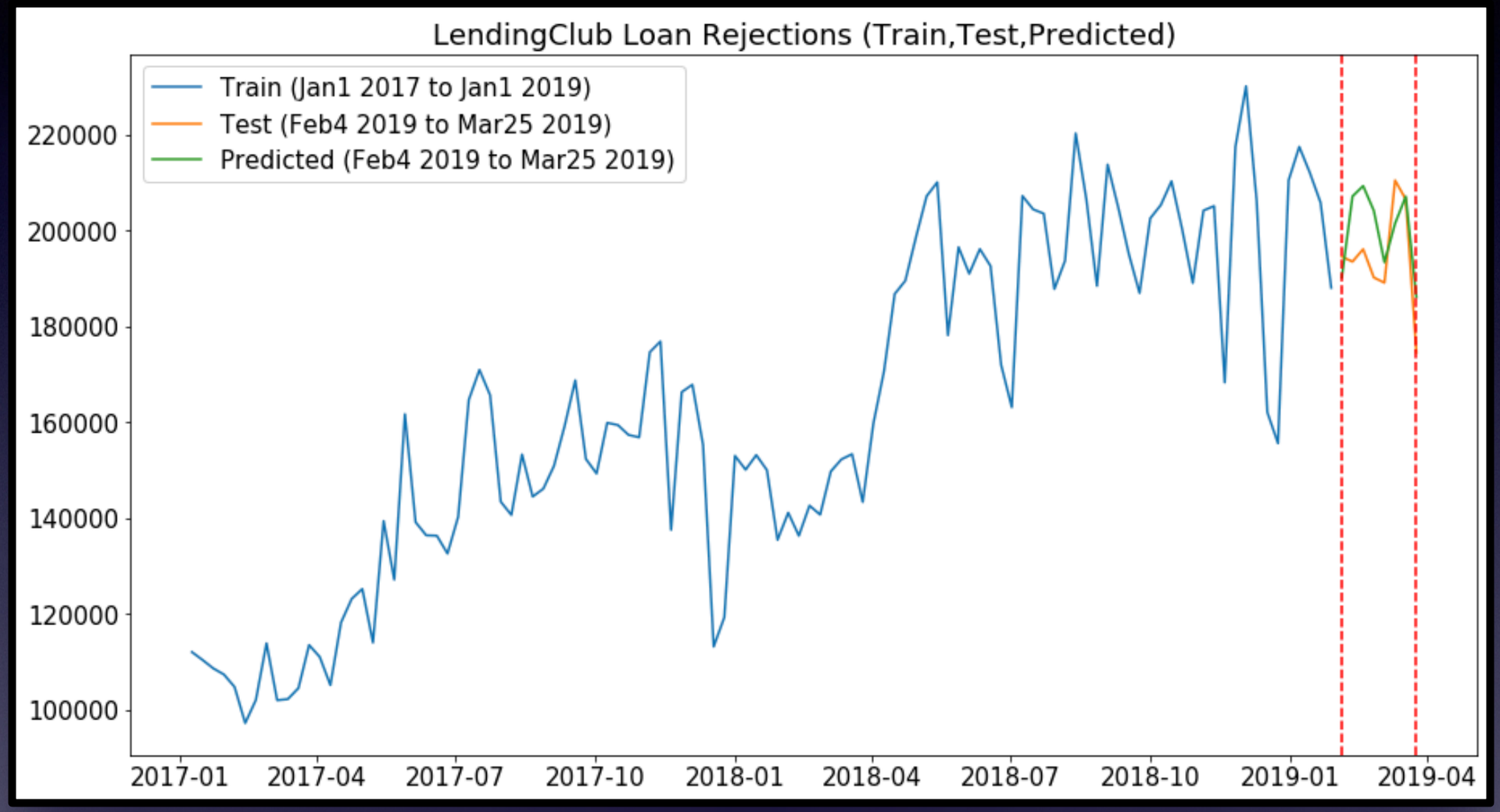


Results

- The residuals look random and normally distributed
- The distribution has a Gaussian look
- The Q-Q plot also shows a normal distribution but with a few outliers.
- There isn't any obvious autocorrelation trend in the plot



SARIMA Forecast



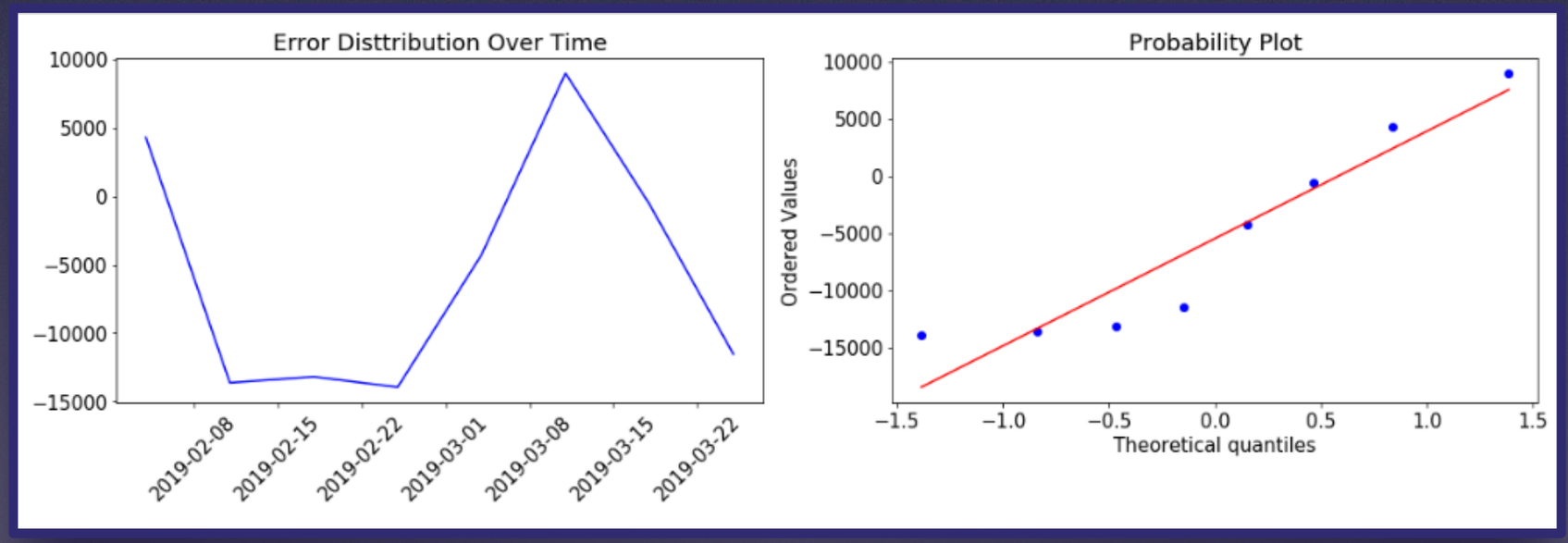
SARIMA Forecast Errors

Application_Date	Loan_Count	Predicted_Count	Error
2019-02-04	194570	190282.514309	4287.485691
2019-02-11	193545	207165.140204	-13620.140204
2019-02-18	196149	209330.691465	-13181.691465
2019-02-25	190241	204185.997558	-13944.997558
2019-03-04	189141	193419.028761	-4278.028761
2019-03-11	210487	201513.062827	8973.937173
2019-03-18	206549	207129.721024	-580.721024
2019-03-25	174618	186142.098573	-11524.098573

Mean of Number of Loans: 194,412.50

Root Mean Square Error: 10,025.50
Mean Squared Error: 100,510,732.85
Percent Mean Error: 5,483.53

The model is off by 5.16%.
The with the mean number of loan = 194,412.50
and the root mean square error of 10,025.50



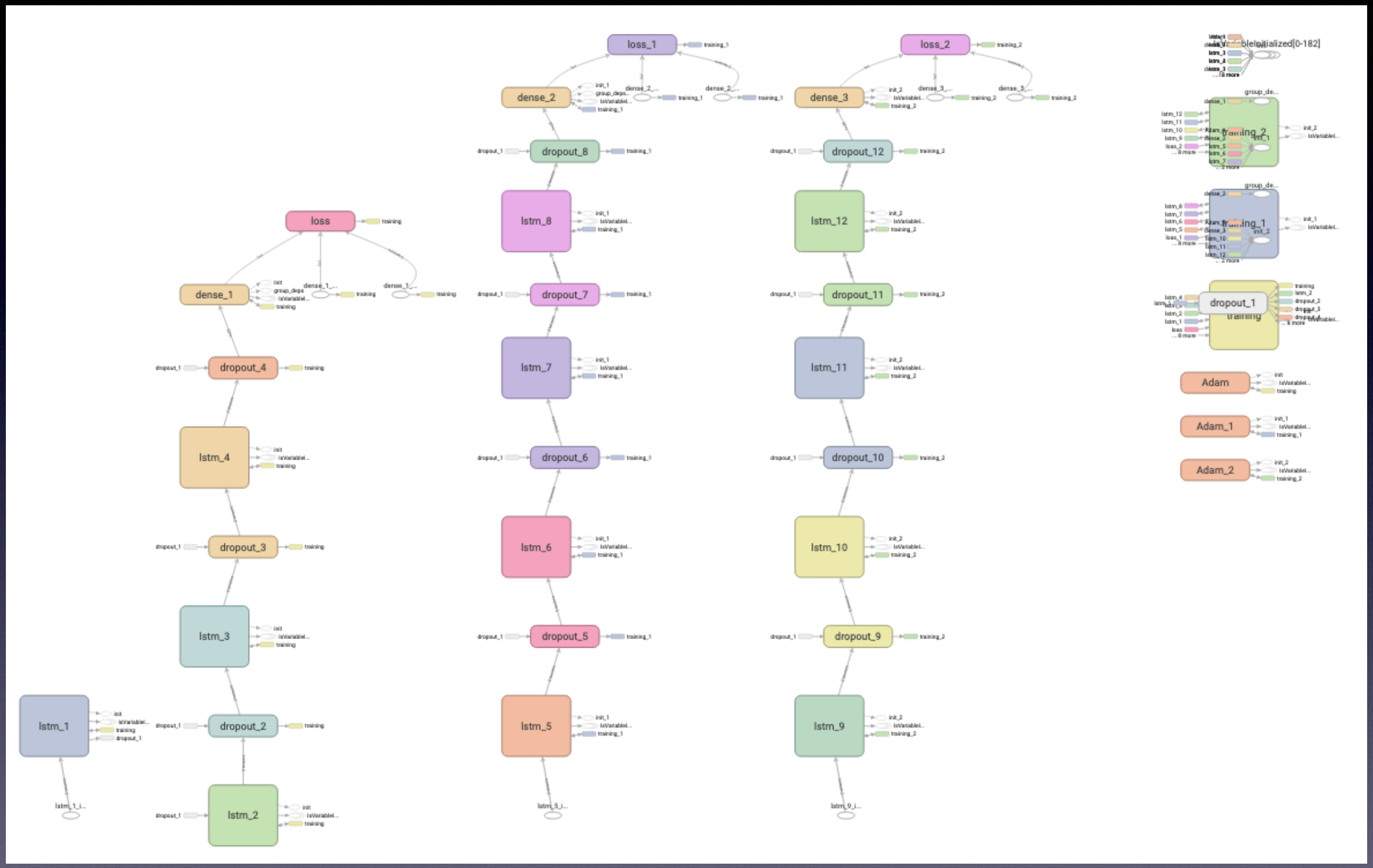
LSTM Time Series

- 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 = 110
  - dropout = 0.3
  - epochs = 900
  - batch size = 1

LSTM Sequential Model & Summary

```
lstm_model = Sequential()

lstm_model.add(LSTM(units=100,
                    return_sequences=True,
                    input_shape=(X_train.shape[1], 1)))
lstm_model.add(Dropout(0.2))

lstm_model.add(LSTM(units=100,
return_sequences=True))
lstm_model.add(Dropout(0.2))

lstm_model.add(LSTM(units=100, return_sequences=True))
lstm_model.add(Dropout(0.2))

lstm_model.add(LSTM(units=100))
lstm_model.add(Dropout(0.2))

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,
                         callbacks=[tensorboard])
```

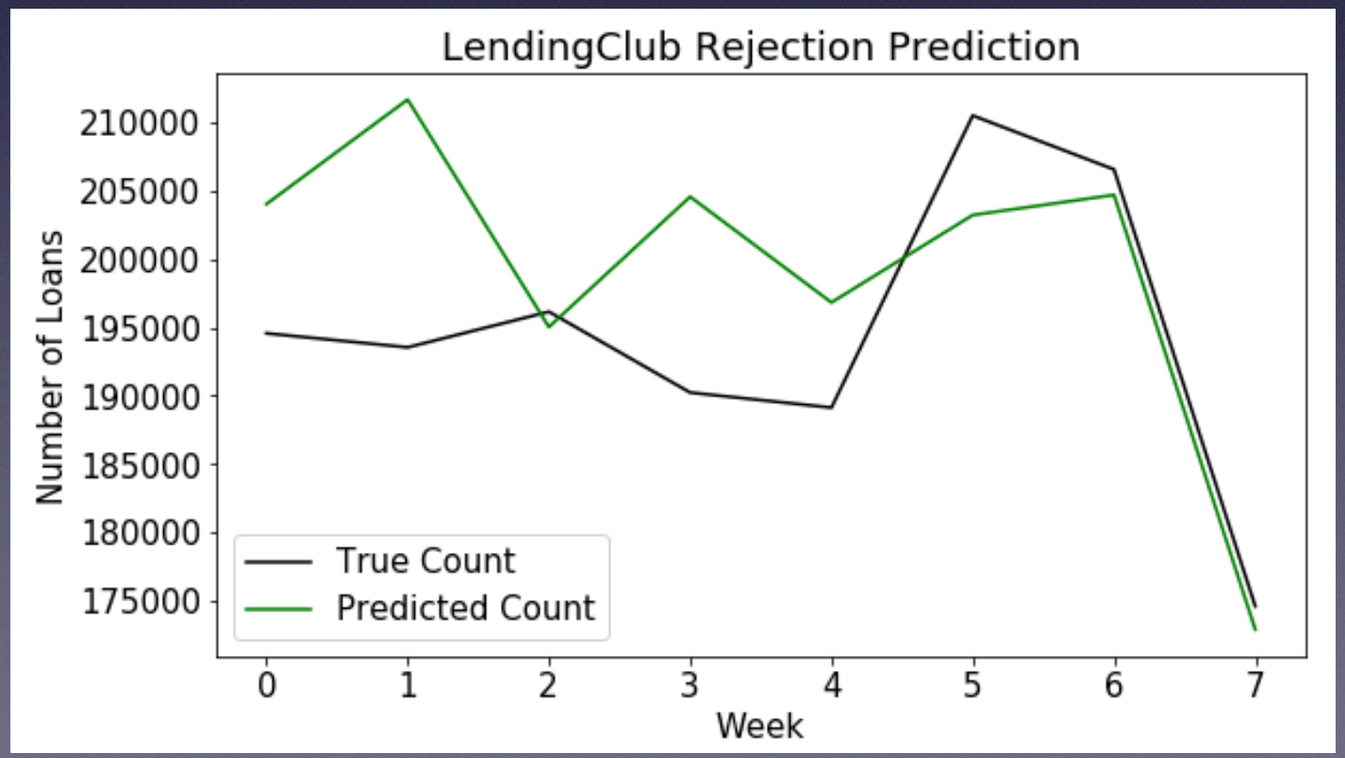
Layer(type)	Output Shape	Param#
lstm_29 (LSTM)	(None,8 (Dense)	40800
dropout_29 (Dropout)	(None,8 (Dense)	0
lstm_30 (LSTM)	(None,8 (Dense)	80400
dropout_30 (Dropout)	(None,8 (Dense)	0
lstm_31 (LSTM)	(None,8 (Dense)	80400
dropout_31 (Dropout)	(None,8 (Dense)	0
lstm_32 (LSTM)	(None,100)	80400
dropout_32 (Dropout)	(None,100)	0
dense_8 (Dense)	(None,1)	101
Total params: 282,101		
Traina ble params: 282,101		
Non-tr ainable params: 0		



LSTM TensorBoard Graph

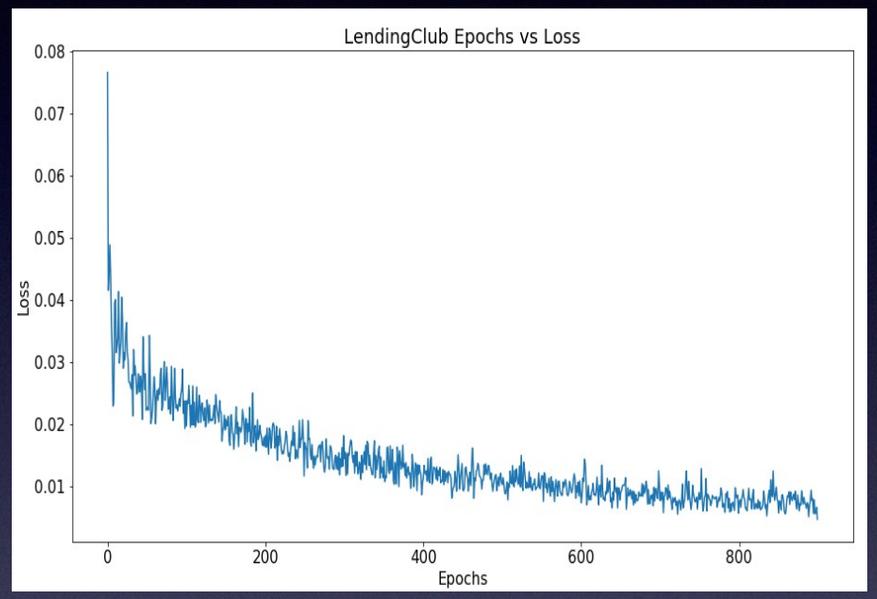
LSTM Prediction

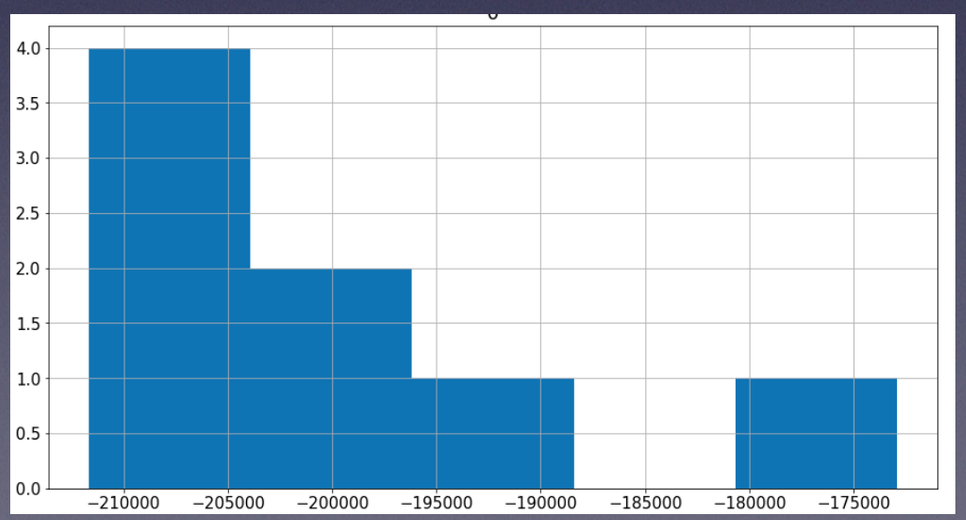
- Well the model didn't do good at all.
- The predictions were not that great.
- I want to research more on how to tune the model better.
- The Train and Test RMSE are not close at all.
- RMSE Train Score: 3717.63
- MSE Train Score: 13820803.0497
- RMSE Test Score: 9627.7526
- MSE Test Score: 92693620.0280

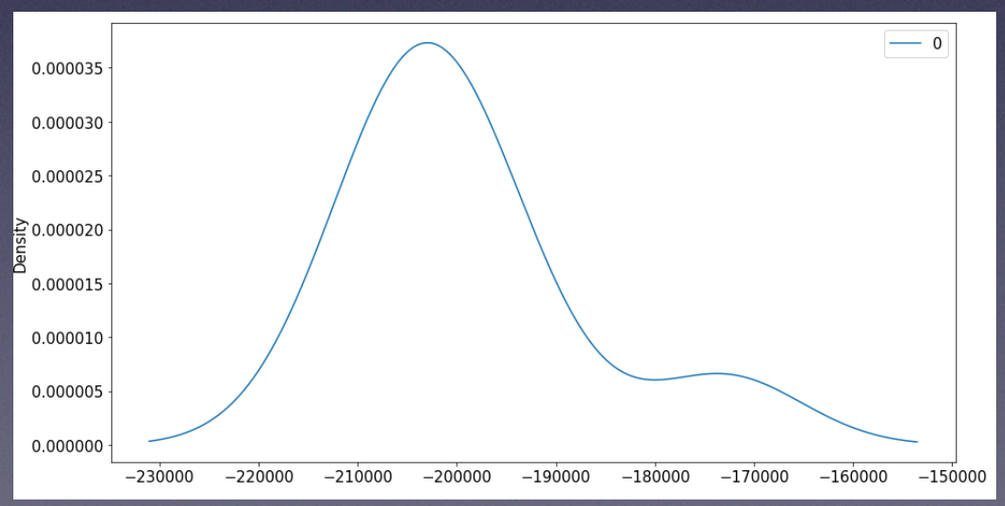


Loss, Residuals and Density

- There is a sharp drop off for the loss at the start and then it tries to level out
- For the residual plot, the distribution does have a Gaussian with right skewness (the mean is to the right of the median)







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

- 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 difference of ~ 1000.

- Lastly, it would be interesting to use my own information to see if my loan will fall into the good or bad status.

Questions & Suggestion?

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