About

Lending Club (LC) is a peer to peer lending platform that has been part of proliferation of the shadow banking system since the great recession. It takes the main role of bank, connecting people who need money with people who have savings that they wish to invest, without all the offerings and trappings of a modern bank.

LC primarily scrutinizes loan applications and tries to ensure the transfer of funds from the debtor to the investor. LC does little investing in the loans, they tend to fund loans when investors do not fully fund the loan at time of origination.

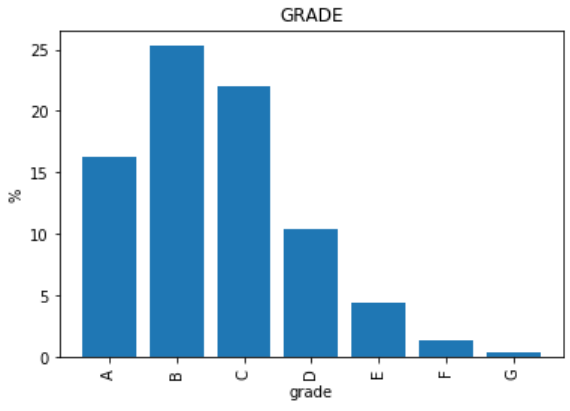
Exploratory Data Analysis

The loans that were examined were issued from 2007 to 2018. The sample size was a little over 2.5 million and had 157 features. You can explore the data in any ipython notebook beginning with "EDA – ". In total about 20% of our loans were charged off which comes close to their data. (look at Fully Paid and Charged Off columns only).

LOAN PERFORMANCE DETAILS

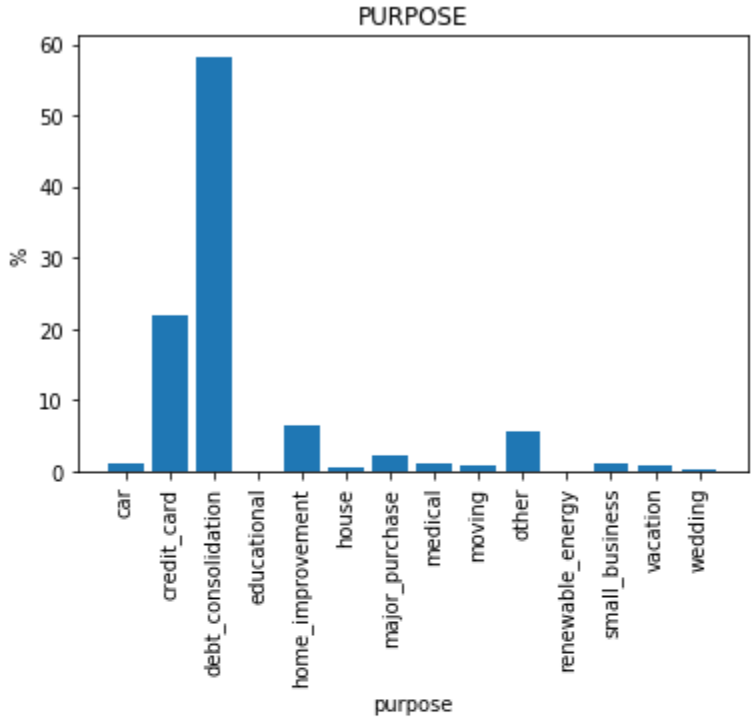
ISSUE DATE START 2007 Q1 ISSUE DATE END 2018 Q4 UNITS Number of loans

	TOTAL ISSUED	FULLY PAID	CURRENT	LATE	CHARGED OFF	AVG. INTEREST RATE	ADJ. NET ANNUALIZED RETURN[1]
A	433,027	220,979	196,171	1,668	14,209	7.07%	4.66%
B	663,557	340,441	264,674	5,788	52,654	10.67%	5.80%
C	650,053	296,518	258,700	9,038	85,797	14.18%	6.11%
D	324,424	140,393	117,037	5,743	61,251	18.18%	5.78%
E	135,639	57,993	38,879	2,574	36,193	21.79%	5.09%
FG	53,968	22,415	11,292	1,045	19,216	26.10%	3.00%
All	2,260,668	1,078,739	886,753	25,856	269,320	13.38%	5.55%



	index	n_samples	percent failed	t_stat	p_value
0	A	434114	5.989210	385.262877	0.0
1	B	731908	13.260137	166.590005	0.0
2	C	711534	22.216507	-47.734153	0.0
3	D	374050	30.109344	-136.592400	0.0
4	E	177466	38.392706	-160.493995	0.0
5	F	61314	45.046808	-125.328639	0.0
6	G	17254	49.913064	-78.939165	0.0

The EDA portion of this project was mostly non-eventful. Most of the statistics that were generated stated the obvious. Lower FICO scores, lower grades, lower [insert "lower is worse" feature here] along with higher debt, high debt to income ratio, higher [insert "higher is worse" feature here] had higher rates of failed loans. Most of the loan's purpose was for refinancing purposes.



	index	n_samples	percent failed	t_stat	p_value
0	car	27082	14.400709	25.607504	6.334079e-143
1	credit_card	551450	16.858827	59.609327	0.000000e+00
2	debt_consolidation	1461114	21.066255	-35.635184	5.268528e-278
3	educational	652	17.177914	1.817053	6.966871e-02
4	home_improvement	161660	17.622170	23.658446	1.569464e-123
5	house	12736	21.278266	-3.899294	9.696873e-05
6	major_purchase	54082	18.165009	10.248270	1.268976e-24
7	medical	28352	21.621050	-7.186320	6.820440e-13
8	moving	17616	23.206176	-10.507353	9.522257e-26
9	other	141124	20.924860	-9.796408	1.186193e-22
10	renewable_energy	1744	23.279817	-3.374322	7.562013e-04
11	small_business	28818	29.308071	-35.220922	4.233367e-266
12	vacation	16622	18.998917	2.843311	4.470197e-03
13	wedding	4588	12.162162	15.959449	7.614718e-56

Running a t-test on any of the features was also yield little new information as the large sample size made almost every feature statistically significant.

There were of course some outliers and anomalies that brought new concepts to me. Applicants whose income was not verified had lower charged off rates than applicants whose income was

verified. LC tends to verify their applicant's income for 2 reasons: random spot checking and if the applicant is deemed suspicious. An example of this is if the applicant's income is higher than average for the job title. You can learn more here.

People who worked for a stated 0 years had no defaults. I wish I was one of them?

I also discovered that joint loans had higher rates of failure. This is despite the cosigner in many cases had more desirable characteristics. The co-signer was probably used to better the applicant's chance of approval and/or to receive a better rate.

The ML Data

The data set used to train the machine learning algorithm was slightly smaller than the EDA data set. There are two reasons for this: first every forward-looking feature was [hopefully] removed and secondly there were some edge cases removed mainly the strange mortgage features.

The main data set had 100 features but, was expanded to 147 to accommodate categorical features. Secondary features for single applicants were filled with their single counterpart. Nans were filled with 0s. During testing we tried a classifier with the training data fill with averages but, the classifier filled with 0s outperformed the classifier trained with the average.

There was a second data set used, with features that only investors see. This data set had 25 features but, was expanded to 72 to accommodate categorical features. It performed similarly to the main data set.

After the training and the decision to use a classifier, the chosen classifier was tested against many slices of our data to figure out one question. Is this one general classifier enough to reliably predict the outcome of the loan across several key features of interest or was a new one needed to correctly predict a certain group of borrowers?

A key feature was defined as one identified by sklearn preprocessing's feature importance, sklearn random forest's own feature importance, and features an investor maybe interested in such as individual/joint, grades, income, etc.

No new classifier was needed, our general classifier outperformed all the locally trained ones.

Scoring

There were several metrics used to evaluate our models as it mirrors the many ways one can invest in LC loans. While you can see that put out the standard mix of model evaluation, I mainly focused on score (accuracy), balanced score and precision. I compared these metrics to the loan's failure rate and failure rate adjusted for loan amount. If a loan was charged off, I assumed that the investors would lose all the money.

Within LC one can choose to invest a fixed amount across all loans, this is similar to the canonical diversification method. On the opposite end one can choose individual loans, hopefully through meticulous process. Or somewhere in between.

The diversification method can be best reflected in the accuracy score or the balance accuracy score. The individual selection by method/model can be best reflected in the precision score. Any user of a method/model should be very sensitive to false positives.

While these were my personal metrics, the standard metrics for model evaluation were not ignored. AUC was the main metric used and recall, precision, and F1 was used as back up to confirm the AUC. AUC was weighted with loan amounts.

Please note that all scores reflect the "1" label (Positive, Good Loan, will not be Charged Off)

The Journey

Many classifiers were used, tuned, and finally tossed out in favor of Random Forest. The classifiers that were thrown out were Logistic (predicted that nearly every loan would paid in full), Gradient Boost (did slightly better than Logistic), Naive Bayes (used 3 versions and all of them managed to do worse than the invest in everything method), Decision Tree (why use 1 tree when we can have a forest of trees), and Keras(preformed similarly to the gradient boosting models). All models were trained against untouched data and normalized inputs after nans were filled with 0s and averages. Normalization was the z-score function.

In Keras I added and expanded layers until my accuracy score no longer had any gains then, I used a variety of activation functions and got similar results. Although ReLu seem to (very slightly) outperform every other activation function.

Gradient boosted models

```
from sklearn.ensemble import GradientBoostingClassifier

gbm = GradientBoostingClassifier()
gbm.fit(X_train, y_train)
print('GRADIENT BOOSTING')
scoring(gbm, X_test, y_test)
```

```
GRADIENT BOOSTING
score:  0.805088149016917
[[ 11623 138054]
 [  8524 593821]]
F1 score:  0.8901395571944657
precision_score:  0.8113694278394534
recall_score:  0.9858486415592393
roc:  0.7235511327115599
```

```
gbm = GradientBoostingClassifier(loss = 'exponential')
gbm.fit(X_train, y_train)
print('GRADIENT BOOSTING')
scoring(gbm, X_test, y_test)
```

```
GRADIENT BOOSTING
score:  0.8047956043839143
[[ 10015 139662]
 [  7136 595209]]
F1 score:  0.8902211759356754
precision_score:  0.8099503178108811
recall_score:  0.9881529688135537
roc:  0.7239684702409301
```

Keras

```
model.fit(Xn_train, yn_train, validation_split=0.25, epochs=5)
print(np.mean(yn_train))
```

```
Train on 1316037 samples, validate on 438679 samples
Epoch 1/5
1316037/1316037 [==============================] - 268s 204us/sample - loss: 0.4545 - acc: 0.8026 - categorical_accuracy: 0.013
3 - val_loss: 0.4525 - val_acc: 0.8040 - val_categorical_accuracy: 0.0346
Epoch 2/5
1316037/1316037 [==============================] - 265s 202us/sample - loss: 0.4514 - acc: 0.8036 - categorical_accuracy: 0.022
2 - val_loss: 0.4503 - val_acc: 0.8047 - val_categorical_accuracy: 0.0253
Epoch 3/5
1316037/1316037 [==============================] - 268s 203us/sample - loss: 0.4504 - acc: 0.8042 - categorical_accuracy: 0.025
1 - val_loss: 0.4500 - val_acc: 0.8050 - val_categorical_accuracy: 0.0288
Epoch 4/5
1316037/1316037 [==============================] - 267s 203us/sample - loss: 0.4496 - acc: 0.8045 - categorical_accuracy: 0.026
2 - val_loss: 0.4491 - val_acc: 0.8049 - val_categorical_accuracy: 0.0248
Epoch 5/5
1316037/1316037 [==============================] - 266s 202us/sample - loss: 0.4488 - acc: 0.8047 - categorical_accuracy: 0.027
3 - val_loss: 0.4489 - val_acc: 0.8052 - val_categorical_accuracy: 0.0207
0.8015205879469954
```

```
score(Xn_test,yn_test,model)
```

```
0.8009672589365736
[[  8971 140706]
 [  6820 595525]]
              precision    recall  f1-score   support

           0       0.57      0.06      0.11    149677
           1       0.81      0.99      0.89    602345

    accuracy                           0.80    752022
   macro avg       0.69      0.52      0.50    752022
weighted avg       0.76      0.80      0.73    752022
```

The best classifier identified was Random Forest.

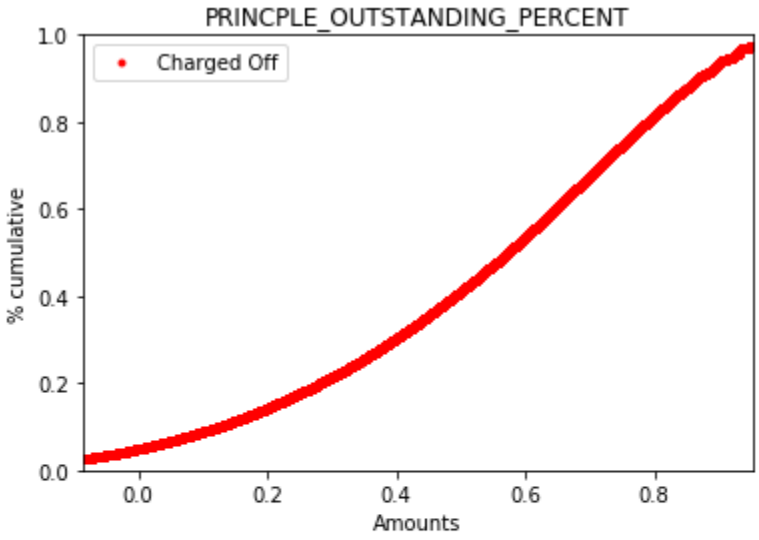
```
Loan passing rate: 0.8009672589365736
Balanced loan passing rate: 0.7855872316459166


score:  0.9414485214528299
balanced_accuracy_score:  0.8602194324067285
[[108035  41642]
 [  2390 599955]]
F1 score:  0.9646028512583384
precision_score:  0.9350963299391986
average_precision_score:  0.929240714442075
recall_score:  0.9960321742522973
roc:  0.9644819356626336
roc_weighted:  0.9653726069638308

{'class_weight': None, 'criterion': 'entropy', 'n_estimators': 101}
```

Reservations- an opinion

I find the rise of this alternative investment troubling. While LC's products do offer higher returns than conventional fixed income products, there is little recourse if the borrower defaults. In the case of a default we can expect total payments to be about 47% of the principle. This stresses the need to identify false positives.



	Charged Off
count	498120.000000
mean	0.530862
std	0.282095
min	-0.721846
25%	0.344073
50%	0.572631
75%	0.753026
max	1.000000

My biggest issue with LC is their marketing to investors. The one thing constantly promoted is diversification. One should add LC's products into their portfolio for diversification and then scoop up a bunch of loans for diversification. This is just an over used term that was the same logic behind MBS.

Another issue I have with LC is fees. 1% of all payments are taken by LC, effectively a 1% fee on all assets invested. While other fee-based investments charge a similar percentage fee annually, there are many that offer better returns for less risk. While we did get a great precision and roc score across different segments, remember that all models are bad but, some are useful.

There is almost no recourse LC or the investor can take when a loan defaults. There is no upfront collateral and any legal recourse to reimburse investor's principal is weak. If a loan was to go into a default, LC would take all legal measures to get payment for the loan then charge a 40% of

all payments after legal expenses are deducted. If no legal measure was taken, then only 30% of all payments are taken by LC.

Not all is doom and gloom, I can see A, B, and C (if investor is risk tolerant enough) class loans as a substitute for some people's longer-term fixed income investments such as CDs. The tax treatment as far as I can see is similar. However, liquidity may be a different story as we will see later.

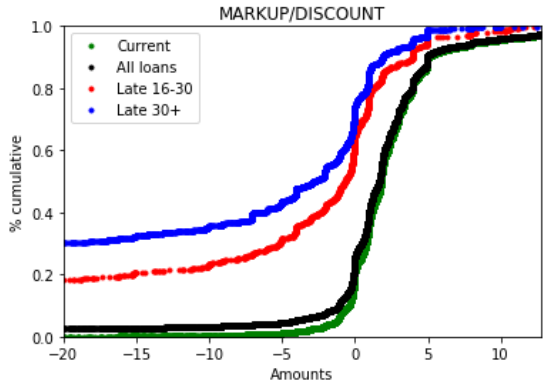
The rest of the grades have yields similar to credit card debt and high enough fail rates to start dragging the yield curve down.

LOAN PERFORMANCE DETAILS

ISSUE DATE START 2007 Q1 ISSUE DATE END 2018 Q4 UNITS Number of loans

	TOTAL ISSUED	FULLY PAID	CURRENT	LATE	CHARGED OFF	AVG. INTEREST RATE	ADJ. NET ANNUALIZED RETURN[1]
A	433,027	220,979	196,171	1,668	14,209	7.07%	4.66%
B	663,557	340,441	264,674	5,788	52,654	10.67%	5.80%
C	650,053	296,518	258,700	9,038	85,797	14.18%	6.11%
D	324,424	140,393	117,037	5,743	61,251	18.18%	5.78%
E	135,639	57,993	38,879	2,574	36,193	21.79%	5.09%
FG	53,968	22,415	11,292	1,045	19,216	26.10%	3.00%
All	2,260,668	1,078,739	886,753	25,856	269,320	13.38%	5.55%

Liquidity may be an issue depending on the loan. Lending Club uses a third party, FolioInvesting to handle the secondary market. From the markup percentages, the market seems liquid if the loan is current. Otherwise the loans are subjected to a fire sale.



	All loans	Current	Late 16-30	Late 30+
count	126461.000000	115616.000000	1866.000000	8979.000000
mean	0.796134	2.300448	-7.154411	-16.921551
std	9.491068	3.708379	15.064838	26.236657
min	-92.000000	-52.910000	-69.600000	-92.000000
25%	0.000000	0.410000	-8.027500	-25.300000
50%	1.680000	1.810000	-0.660000	-2.720000
75%	3.310000	3.380000	0.940000	0.100000
max	39.610000	39.610000	20.190000	27.480000

Despite my reservations, LC does serve a rising and much needed function. Anecdotally most LC loan applicants were rejected by a bank. Although only about 36 billion dollars have been originated. Since the Great Recession, either from scars and/or regulations, banks have heavily scaled back on lending to anyone who wasn't pristine. In addition from what I can see in the loan process, LC is less onerous than a bank.

Disclaimers

As of the time I have written this report, I have no financial interest/stake in Lending Club nor in any of its financial products. It is because I failed to meet their criteria of a suitable investor.

If I was to invest in LC's loans, I would use my random forest model and use it on the more speculative loans (Grade C and below). I would give all my loans equal dollar amounts and periodically retrain the model with new data. After that I would check my portfolio's loans and anything newly labeled to be Charged Off would be sold on the secondary market at par.