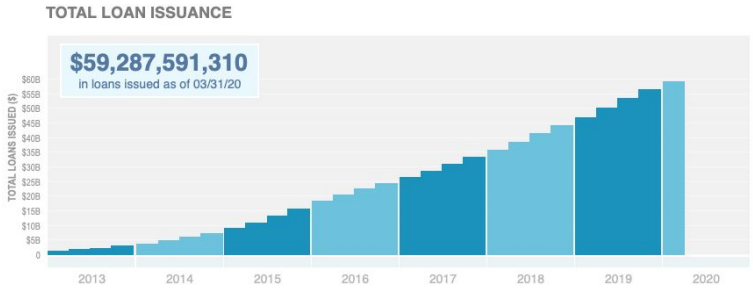
Machine Learning:

Investing on P2P Loans

Ricky Gunawan
Akshay Jindal
Onur Baser

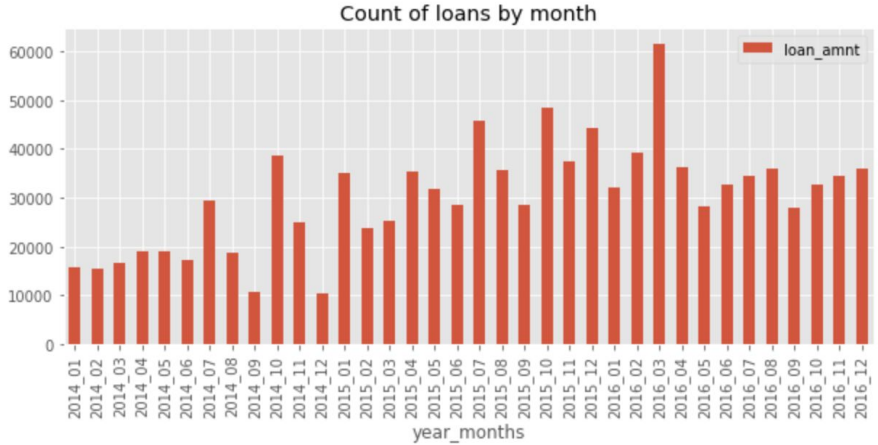
Background

- As the global interest rates benchmark are heading to zero, it is interesting to look at alternative loan investment that provides promising potential return.
- We choose Lending Club loan investing and the platform gives us access to large loan datasets (up to ~5millions loans count per quarter)

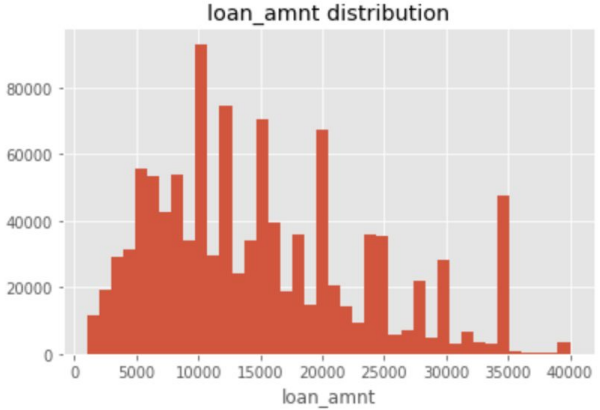


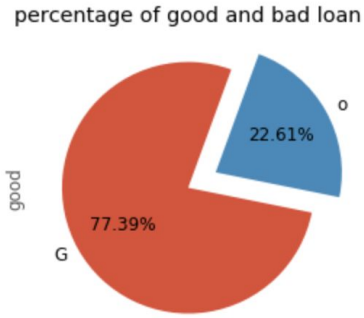
Initial Data Analysis

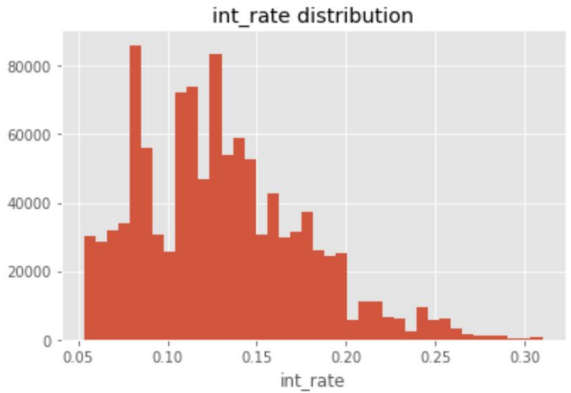
- We choose to focus our analysis on 2014 - 2016 period, which had similar low interest rates regime as today (i.e. Fed Fund Rates range between 0 to 25bps)
- During this period we have about 1.1 **millions loan data** with 151 **features**
- We did TONS of data cleaning and apply data visualizations

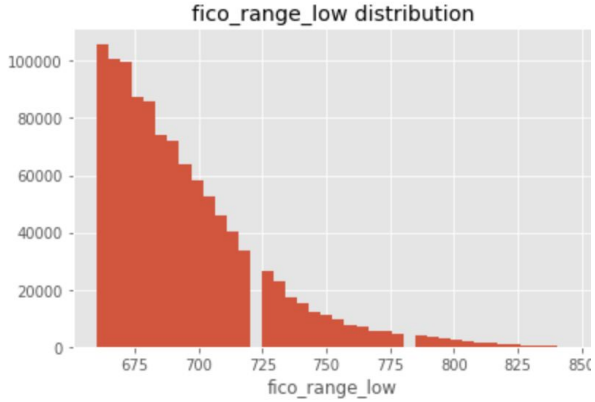


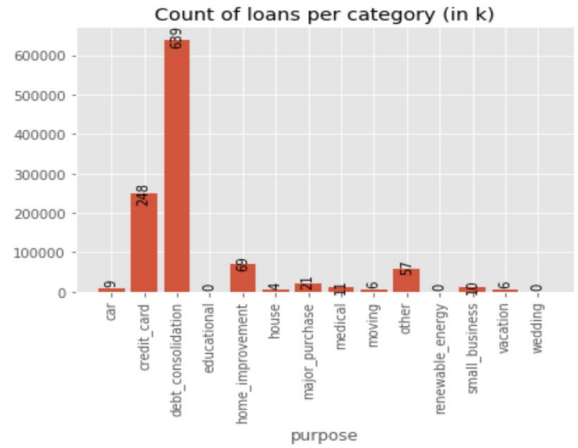
Initial Data Visualization Analysis

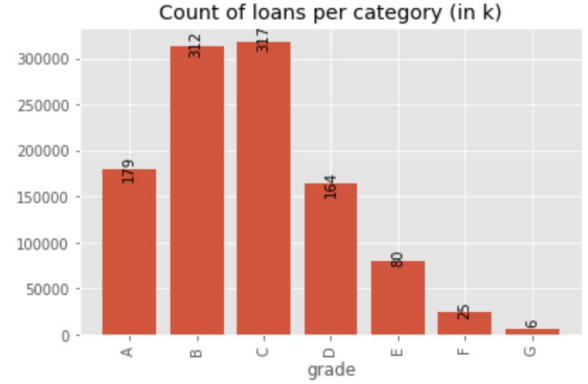




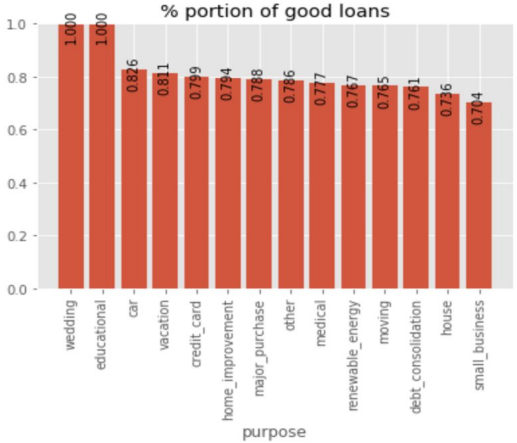


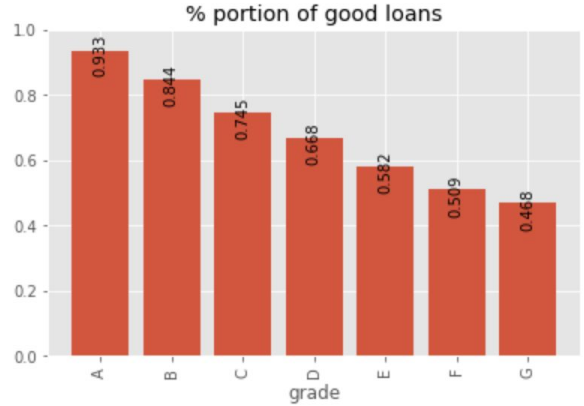


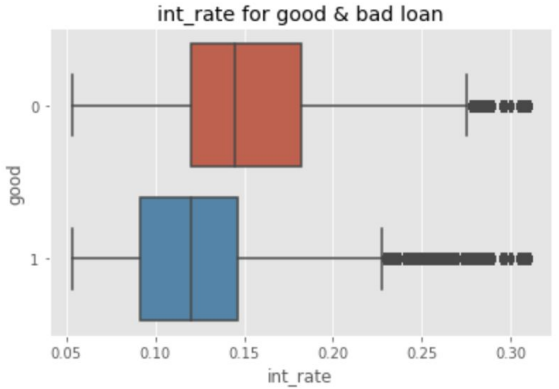


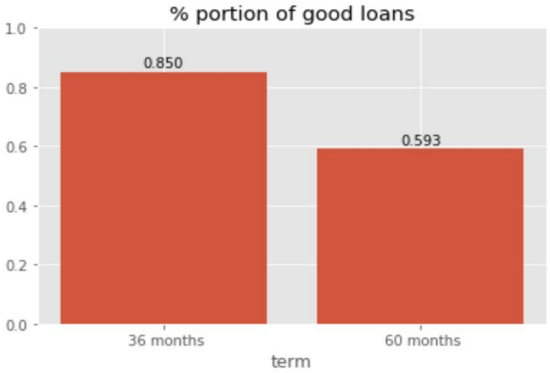


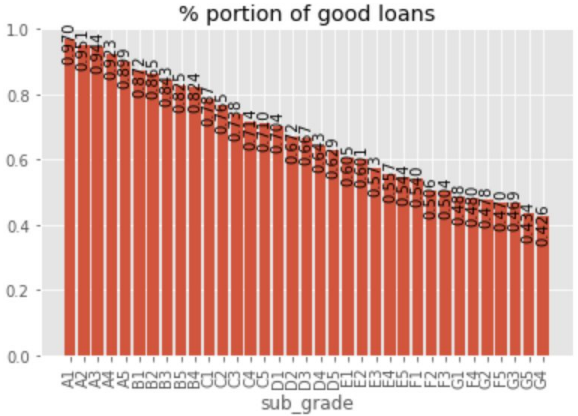
Good Loans Data Visualization Analysis

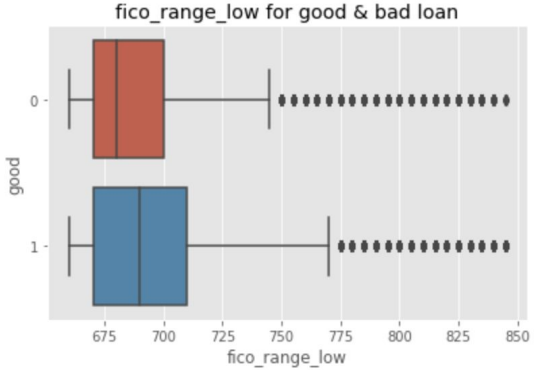












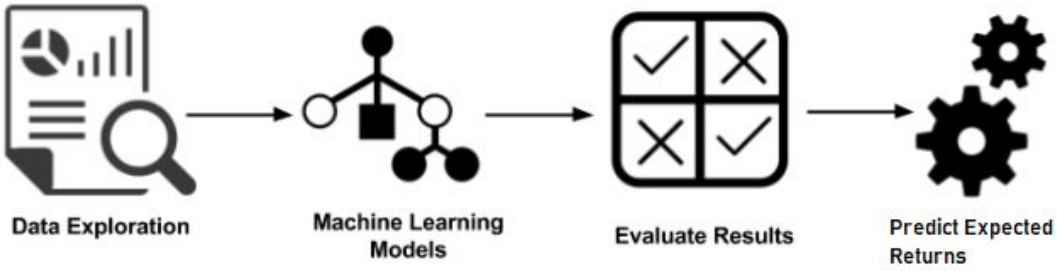
Programming Objectives

1. **Determine** the features in the dataset and calculate returns for each loans in the dataset

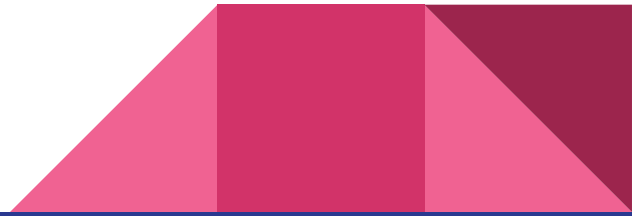
2. **Test** multiple classifiers model to predict good loan and bad loan
    - Random Forest, Logistic Regression, Decision Tree, Gradient Boosted Tree, LSTM RNN

3. **Select** one model and explore on how to increase precision by taking the top percentile of probability of predicting good loan

4. **Calculate** the loans expected return - Hopefully we can see **enhancement** on return here :)



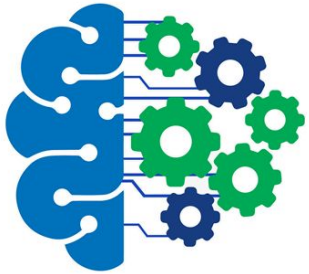
1. Features Selection



Features Selection Methodology

- **Machine Learning Method:**
  - Use classification models to determine important features
    - Random Forest Decision Tree
    - XGBoost

- **Research Method:**
  - Academic articles
  - Case studies
  - Kaggle

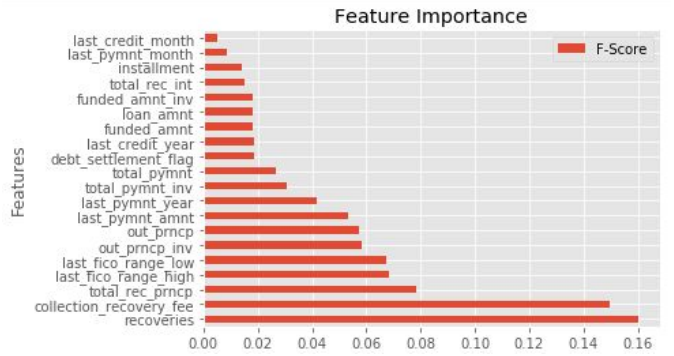


Identifying Important Features

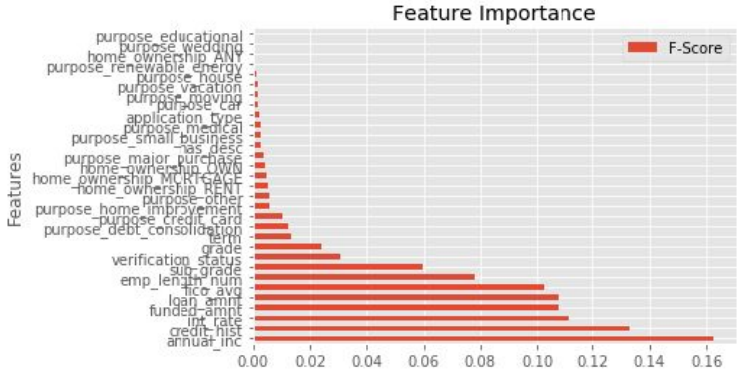
- 9 **additional** features were created on top of the 22 features
- We are selecting features that are known during the initiation of the loan, so features like recoveries, total payment, collection recovery fee are excluded for X training

```
columns = [
 'application_type',
 'loan_amnt',
 'funded_amnt',
 'int_rate',
 'grade',
 'sub_grade',
 'home_ownership',
 'annual_inc',
 'verification_status',
 'purpose',
 'dti',
 'delinq_2yrs',
 'open_acc',
 'pub_rec',
 'fico_range_high',
 'fico_range_low',
 'revol_bal',
 'revol_util',
 'total_pymnt',
 'recoveries',
 'last_pymnt_d',
 'fico_avg',
 'has_desc', # created features
 'credit_hist', # created features
 'loan_length',# created features
 'term',# created features
 'ret_low',# created features
 'ret_high',# created features
 'good',# created features
 'emp_length_num'# created features
]

target = ["good"] # created features
```

Random Forest with all raw features



Random Forest with selected features

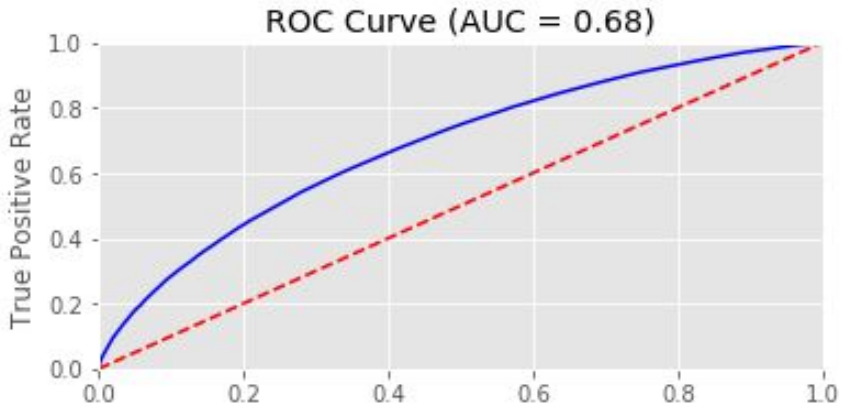


2. Classification Model Selection

Random Forest Tree

	Predicted 0	Predicted 1
Actual 0	3944	45956
Actual 1	4340	205948

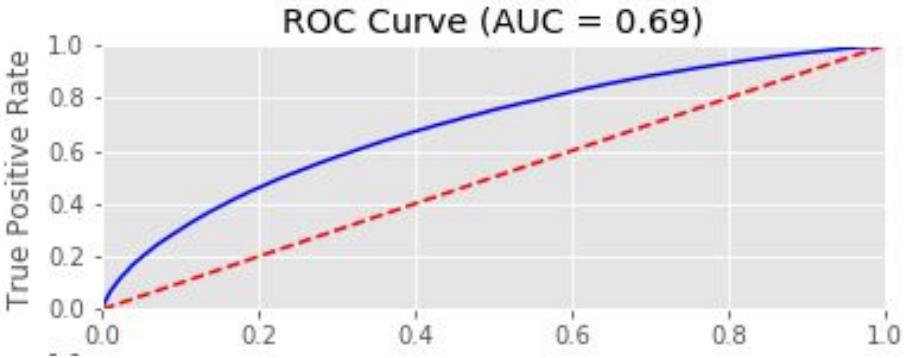
```
Accuracy Score : 0.8066936215351976
Classification Report
              precision    recall  f1-score   support

           0       0.48      0.08      0.14     49900
           1       0.82      0.98      0.89    210288

    accuracy                           0.81    260188
   macro avg       0.65      0.53      0.51    260188
weighted avg       0.75      0.81      0.75    260188
```



Logistic Regression

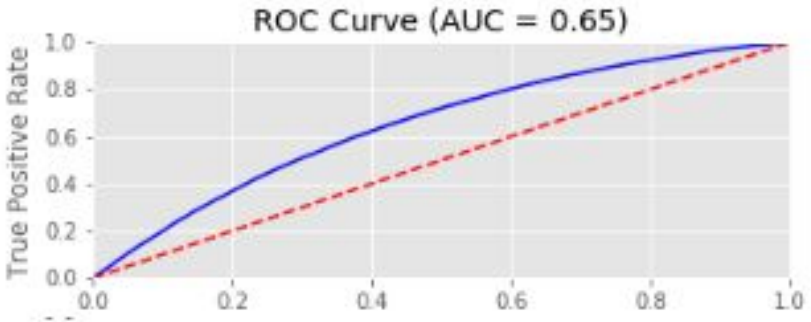
	Predicted 0	Predicted 1
Actual 0	0	49900
Actual 1	0	210288

```
Accuracy Score : 0.80821559795225
Classification Report
              precision    recall  f1-score   support

           0       0.00      0.00      0.00     49900
           1       0.81      1.00      0.89    210288

    accuracy                           0.81    260188
   macro avg       0.40      0.50      0.45    260188
weighted avg       0.65      0.81      0.72    260188
```



LSTM Neural Network

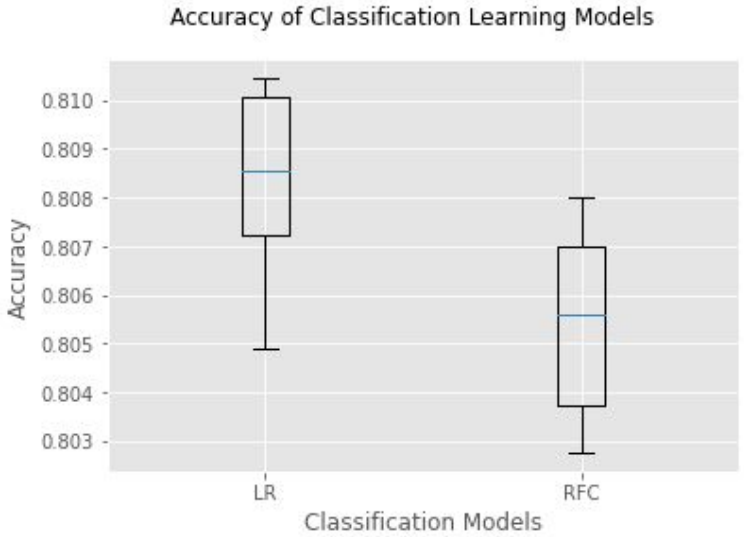
Predicted	Positive (1)	Negative (0)
Actual		
Positive(1)	TP=203597.0	FN=44760.0
Negative(0)	FP=6691.0	TN=5140.0



	precision	recall	f1-score	support
0	0.10	0.43	0.17	11831
1	0.97	0.82	0.89	248357
accuracy			0.80	260188
macro avg	0.54	0.63	0.53	260188
weighted avg	0.93	0.80	0.86	260188

Why we choose Random Forest

```
LR: 0.808412 (0.001757)
RFC: 0.805402 (0.001831)
```



Logistic Regression: High accuracy, **High** standard deviation, more outliers

Random Forest: High accuracy **Low** standard deviation, less outliers

3. Random Forest

Random Forest at sub grade level

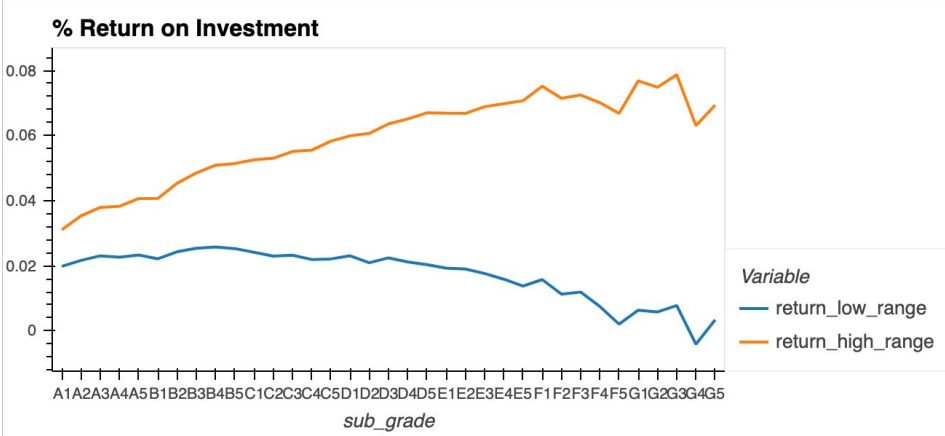
- Training using 31 features for each sub grade from A1 to G5
- Calculate probability of good loans (precision) for random investment and top 10th percentile predicted probability
- Looking at C1, random investment produces 82% of good loans. If we take top 10% of predicted probability, it would produce 86% of good loans (5% improvement)

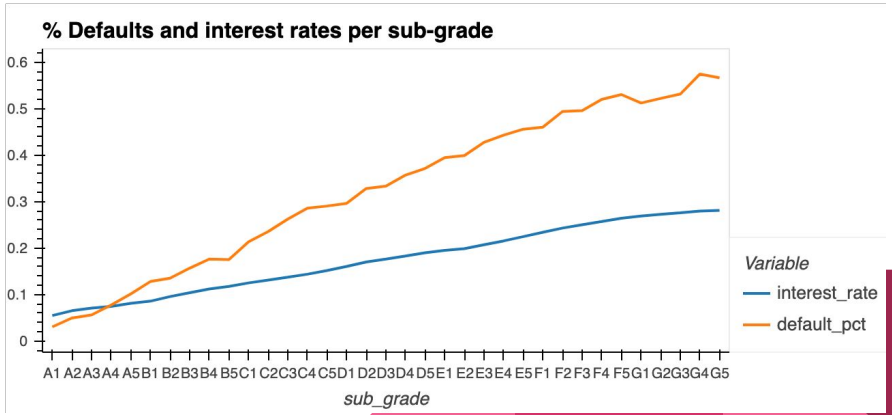
sub_grade	prec_10	random	ratio_10
A1	0.9777	0.9681	1.0099
A2	0.9585	0.9499	1.0091
A3	0.9620	0.9495	1.0132
A4	0.9471	0.9346	1.0133
A5	0.9352	0.9194	1.0171
B1	0.9220	0.9027	1.0214
B2	0.8953	0.8912	1.0046
B3	0.8937	0.8764	1.0197
B4	0.8782	0.8600	1.0212
B5	0.8661	0.8465	1.0231
C1	0.8627	0.8226	1.0487
C2	0.8460	0.8053	1.0506
C3	0.8233	0.7875	1.0454
C4	0.8253	0.7651	1.0787
C5	0.8064	0.7528	1.0711
D1	0.8161	0.7444	1.0962
D2	0.7820	0.7132	1.0966
D3	0.7742	0.7087	1.0924
D4	0.7174	0.6829	1.0504
D5	0.7260	0.6747	1.0760
E1	0.6839	0.6424	1.0646
E2	0.7161	0.6391	1.1205
E3	0.6633	0.6213	1.0675
E4	0.7161	0.5996	1.1943
E5	0.6813	0.5807	1.1733

4. Return on Investment Analysis

Return calculation for each loan

- For every loan in the given dataset, we calculated annualized return on investment
  - High range (optimistic) return: (total loan payment - initial investment) * 12 / age of the loan in month
  - Low range (pessimistic) return: (total loan payment - initial investment) * 12 / term of the loan in month
- High range return ranging from 3% to 8% and Low range return ranging from 2% to -0.4%





Returns Calculation after applying Random Forest

- For every loan in the given dataset, we calculated annualized return on investment
  - From here, we grouped by “Good Loan” and “Defaulted Loan” and able to calculate the expected return of each Good and Defaulted loan at sub_grade level

- Utilizing Random Forest Classifier that we applied at sub_grade level and selecting to 10th percentile probability, we were able to get the precision or probability of predicting good loan P(good loan)

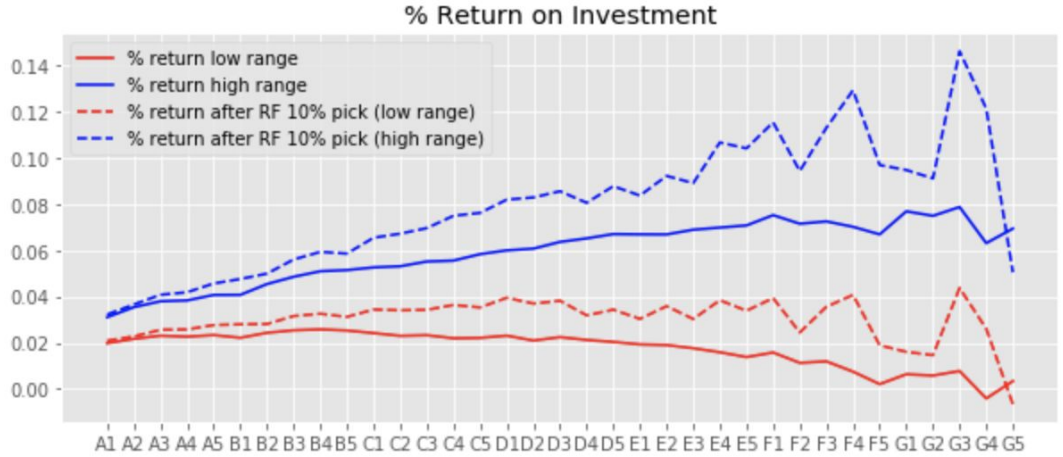
- With the given probability above, we can calculate the new expected return at sub_grade level:
  - Expected Return = P(good loan) * R_{good} + P(defaulted) * $R_{defaulted}$

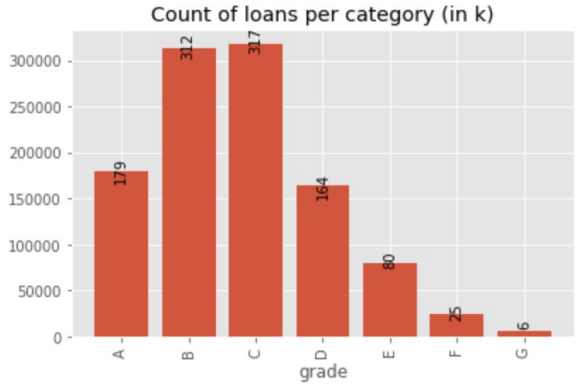
Return Dataframe

sub_grade	return_low_range	return_high_range	default_ret_low	good_ret_low	default_ret_high	good_ret_high	model	prec_10	ret_10_low	ret_10_high
A1	0.019801	0.031039	-0.114755	0.023988	-0.114139	0.035557	random forest	0.9777	0.020894	0.032218
A2	0.021669	0.035367	-0.108059	0.028390	-0.107583	0.042773	random forest	0.9585	0.022728	0.036533
A3	0.023058	0.037923	-0.110451	0.030957	-0.110146	0.046682	random forest	0.9620	0.025583	0.040723
A4	0.022622	0.038289	-0.096301	0.032568	-0.095981	0.049519	random forest	0.9471	0.025750	0.041822
A5	0.023322	0.040659	-0.083352	0.035335	-0.081817	0.054452	random forest	0.9352	0.027644	0.045622
B1	0.022145	0.040700	-0.079595	0.037061	-0.079035	0.058254	random forest	0.9220	0.027961	0.047545
B2	0.024304	0.045376	-0.083665	0.041165	-0.082999	0.065423	random forest	0.8953	0.028095	0.049883
B3	0.025386	0.048514	-0.077903	0.044556	-0.077070	0.071821	random forest	0.8937	0.031538	0.055994
B4	0.025776	0.050904	-0.076724	0.047694	-0.075872	0.078013	random forest	0.8782	0.032540	0.059270
B5	0.025295	0.051411	-0.092627	0.050310	-0.091645	0.081759	random forest	0.8661	0.031171	0.058540
C1	0.024167	0.052560	-0.081297	0.052761	-0.080223	0.088559	random forest	0.8627	0.034354	0.065386
C2	0.022986	0.053053	-0.080765	0.054905	-0.079493	0.093830	random forest	0.8460	0.034012	0.067138
C3	0.023261	0.055107	-0.071300	0.056856	-0.069662	0.099434	random forest	0.8233	0.034211	0.069554
C4	0.021936	0.055508	-0.070681	0.058997	-0.069047	0.105349	random forest	0.8253	0.036342	0.074882
C5	0.022109	0.058300	-0.074446	0.061602	-0.072595	0.111837	random forest	0.8064	0.035263	0.076131
D1	0.023071	0.059928	-0.079635	0.066265	-0.077691	0.117806	random forest	0.8161	0.039434	0.081854
D2	0.020933	0.060674	-0.076785	0.068634	-0.074459	0.126639	random forest	0.7820	0.036933	0.082800
D3	0.022436	0.063585	-0.075285	0.071223	-0.072666	0.131608	random forest	0.7742	0.038142	0.085483
D4	0.021206	0.065114	-0.070674	0.072196	-0.067894	0.138930	random forest	0.7174	0.031821	0.080481
D5	0.020338	0.066962	-0.069976	0.073649	-0.067029	0.146056	random forest	0.7260	0.034296	0.087670
E1	0.019273	0.066855	-0.065843	0.074735	-0.062541	0.151170	random forest	0.6839	0.030298	0.083616
E2	0.018997	0.066814	-0.068900	0.077322	-0.065441	0.154573	random forest	0.7161	0.035810	0.092111
E3	0.017631	0.068855	-0.062113	0.077161	-0.058301	0.163780	random forest	0.6633	0.030267	0.089005
E4	0.015883	0.069782	-0.062948	0.078524	-0.059260	0.172322	random forest	0.7161	0.038360	0.106576
E5	0.013803	0.070703	-0.065882	0.080492	-0.061709	0.181517	random forest	0.6813	0.033842	0.104001

Returns Calculation after applying RF

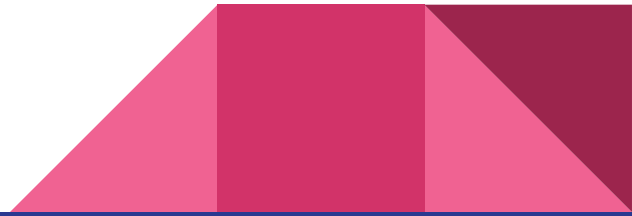
- For both low and high range return, the new expected returns are higher
- Looking at C1 sub grade, the low range return increased to 3.4% from 2.4% and high range return to 6.5% from 5.3% after taking top 10% probability of good loan
- At lower grades (F1 to G5) we started to see model instability, likely due to sparse data availability for training and testing





What would we do if we have more time?

- Extend the study utilizing neural network model
- Review the model using the more recent data and potentially add the ability to incrementally train the model as we get new data
- Utilizing more of Google Colab or AWS as we are are dealing with larger datasets

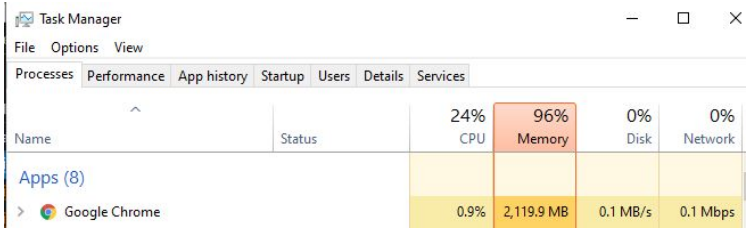


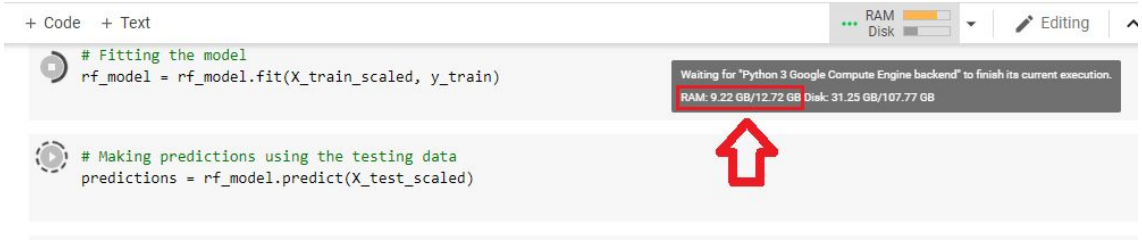
BIG DATA

Processing Big Data

Cloud Computing application (Google Colab)

- Utilized more memory for processing
  - Local Runtime: 20 - 45 min
  - Cloud Runtime: 5 - 10 min





Citations

- https://www.lendingclub.com/info/statistics.action
- https://www.liebertpub.com/doi/pdf/10.1089/big.2018.0092
- https://towardsdatascience.com/understanding-auc-roc-curve-68b2303cc9c5
- https://towardsdatascience.com/accuracy-recall-precision-f-score-specificity-which-to-optimize-on-867d3f11124
- http://cs229.stanford.edu/proj2015/199_report.pdf
- http://abhay.harpale.net/blog/machine-learning/threshold-tuning-using-roc/
- https://machinelearningmastery.com/feature-selection-with-real-and-categorical-data/
- https://www.researchgate.net/publication/322098019_Combination_of_Random_Forests_and_Neural_Networks_in_Social_Lending
- https://www.cs.huji.ac.il/labs/learning/Papers/giladbachrachnavottishby04b.pdf
- https://www.kaggle.com/janiobachmann/lending-club-risk-analysis-and-metrics
- https://www.kaggle.com/pavlofesenko/minimizing-risks-for-loan-investments