Lending Club Loan Data Analysis

Springboard Capstone Project

Aug - 2017

Prepared by-Prerna Saxena

Index

Introduction

Lending Club is no new name in the lending industry. It has shifted the lending paradigm with the use of technology devising efficient, convenient and the smartest way about investing and borrowing.

It has revolutionized peer to peer loan lending platform by operating a credit marketplace which is overpowering the loan programs offered by traditional banking institutions. With their lending model, borrowers can take advantage of lower interest rate where as investors can also be benefitted with better returns on their investments.

With Lending Club, an investor can invest in a portfolio of loans. But how will those loan perform, and which loans to pick from? In order to say anything meaningful about what loans to choose, we must first estimate how loans will do over time. What percentage of loans will default? What percentage of loans will get paid off in full?

Through this project we are interested in making a guess at the probability of default, which is intended to support company's decision in approving/not approving the loans given the features that are collected from the lenders through loan application.

Obtaining data

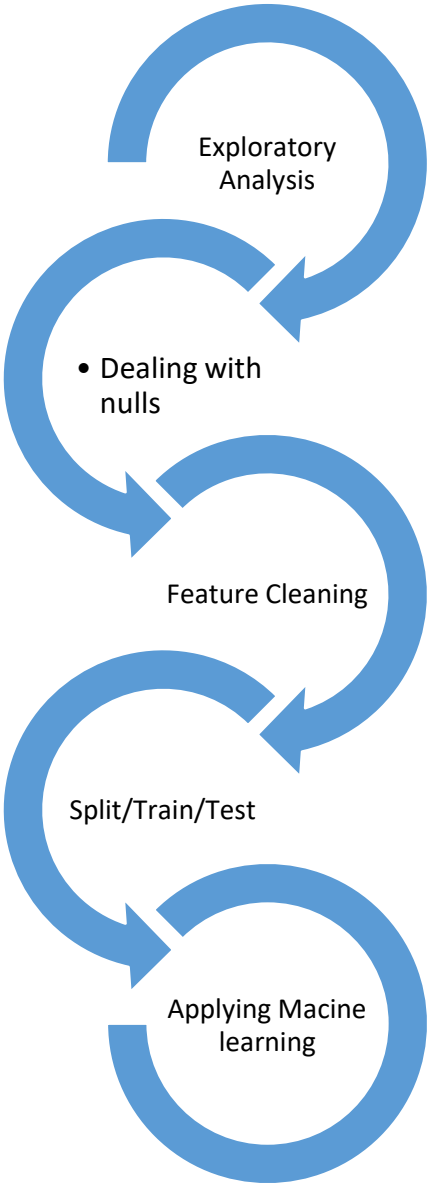
Dataset is publicly available from the Lending Club website. We also referred to the data dictionary provided by the company on the site while exploring features.

Link to data: - (https://www.lendingclub.com/info/download-data.action).

We are going to explore, visualize and analyze 2014-2015 data set and build models for loan predictions. We downloaded two separate csv files each for 2014 and 2015 dataset from the above link and combined them together into one dataset and called it 'custdata'

Process Flow

- Exploratory Analysis
- Dealing with nulls / missing values
- Feature Cleaning
- Preparing Training/Test data set
- Applying Modelling techniques



Tech Stack

Our code is written in Python framework making an effective use of Panda library throughout the project for basic data frame wrangling. To handle numerical calculations, we heavily used libraries like NumPy, SciPy. We used ggplot, matplotlib for exploratory analysis and visualization purposes. To benchmark our results against robust machine learning code, we took advantage of the scikit-learn library, which offers out of the box functionality for the Logistic regression, kNN, Random forest and Naive Bayes algorithms implementation. Scikit-Learn implementations were adopted to verify the accuracy of the results. We've written our code to be modular making use of ipython notebook so that we can easily plug in different implementations to test our code.

About Lending Club dataset

Original data came with 656732 rows and 122 columns. We noticed that , however mentioned in the data dictionary, there are a number of columns which could have been important to our analysis were missing from the data set like fico scores. Also, there were fields which were not making intuitive sense for learning algorithms like ID, Member_ID, URL, month the last payment was received etc. Thus we removed such fields from the dataset. We also dropped rows which were populated with "nan" values. And lastly we removed columns having missing values for greater than 50% of the rows.

To label the dataset, we classified any loan that "defaulted" or were "charged off" as negative examples (0), while we classified any loan that was "fully paid" as positive examples (1). We eliminated loans which were "late" and "in-grace" period or current from the dataset as due to their reversible state.

We would explore a subset of features and test their predictive strength against lenders paying off their loan or defaulting on their loan.

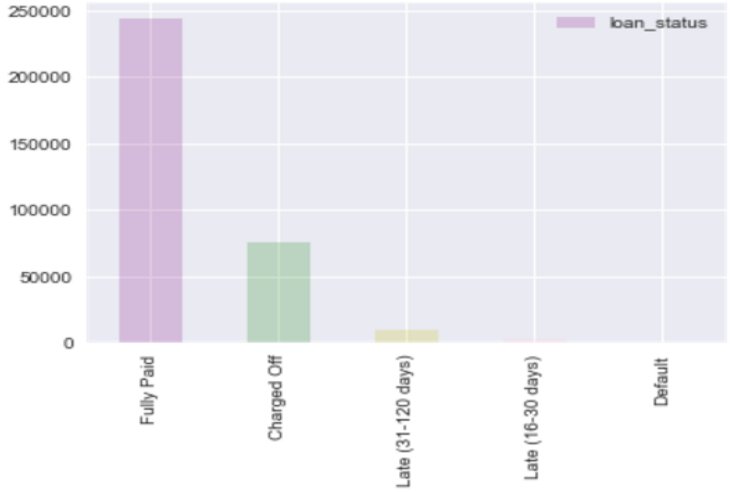
Problem formulation

We decided to tackle the loan quality problem by approaching the simple problem of whether a loan will be fully paid or charged off at completion, simplifying our initial approach to **a** binary classification problem.

Exploratory Analysis

Exploring Loan Status (Target variable)

Loans can transition through the following states over time.



- Current,
- In Grace Period,
- Late (16-30 days),
- Late (31-120 days),
- Default,
- Charged Off,
- Fully Paid

From one month to the next, a loan could move, for example, from Current -> In Grace Period, or Current -> Fully Paid, or Late -> Current.

	Status_Category	No. of applicants
0	Current	319220
1	Fully Paid	244085
2	Charged Off	75589
3	Late (31-120 days)	10123
4	In Grace Period	5187
5	Late (16-30 days)	2506
6	Default	14

Here, we are interested in the final and irreversible stage of a loan activity. We consider a loan being a “good loan” if it is fully paid off and a loan being defaulted or under the Charged-Off categories making it “bad loan”. Any loan falling under the Late or in-grace period could turn up or down so we would filter them from the dataset along with “Current”. So, here is our new groups:

‘1 ’= ‘Fully Paid' and

'0’= ‘Charged Off' + ‘Default’

‘2’ = ‘Current’, ‘Late (31-120 days)’ + ‘In Grace Period’ + ‘Late (16-30 days)’

Cleaning “loan_status”

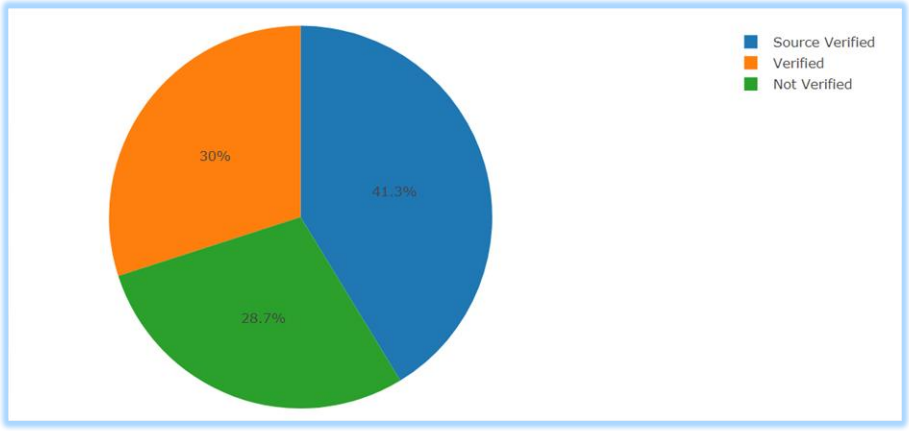
Action	loan_status_category
1	‘Fully Paid'
0	'Charged Off' and Default
Remove	'Current’,‘In-GracePeriod’, ‘Late(16-30days)','Late (31-120 days)' or blank

That reduces our dataset to 332317 rows now out of which there are 256,714 “Fully paid” rows. & “75,603” bad loans. Hence, 29 % of the loans turns out to be bad loans in our dataset.

	loan_status_clean	loan_status	count_of_loan_issued
0	0	Charged Off	75589
1	0	Default	14
2	1	Fully Paid	244085

Exploring “verification status” (Input variable)

“verification_status” indicates if income was verified by LC. There are three distinct values in the column. There is no clear explanation in the data dictionary about what is the difference between ‘Source Verified’ and just ‘verified’. We are going to make an assumption that Source Verified could be a third party confirming the verification of income or it could have been obtained from the credit report. Based on the assumption we could merge them into one and classify the values into two groups.



	Status_Category	No. of applicants
0	Source Verified	137483
1	Verified	100229
2	Not Verified	94605

Cleaning “verification_status”

We cleaned up the column by assigning “Source Verified” and “Verified” as 1 and “Not Verified” as 0. 21% (16202/75603) of the bad loans were not verified in our dataset.

Action	Verification_status
1	'Verified', 'Source Verified'
0	'Not Verified'
Remove	All other keywords or blank

	Bad Loan	Good Loan	Rtotal
Not-Verified	16202	78403	94605
Verified	59401	178311	237712
Ctotal	75603	256714	332317

Exploring “emp_length” (Number of year lender is employed)

It indicates employment length in years. Possible values are between <1, 1...9 and 10+ where <1 means less than one year and 10 + means ten or more years. We see good number of loan application counts are coming from individuals less the 3 years of employment length. We also identified rows with “n/a” which is interesting. To my understanding, this could be because the individual are unemployed or they just didn’t fill that column. When we analyzed n/a records alone, we identified half of the applications have a history of defaulting the payments. We further pulled corresponding verification status just to find out whether these records had their employment verified. And we could see only 1219 records with “n/a” that are not verified which is quite low. That makes us assume that these individuals might not be bound by any employment at the time loan was funded. They might be doing business or their source of income must be something which does not qualify as employment.

```
10+ years    104925
2 years       28829
< 1 year      25992
3 years       25486
1 year        20807
5 years       18608
4 years       18600
n/a           16370
8 years       16365
7 years       16163
6 years       14710
9 years       12833
Name: emp_length, dtype: int64
```

	emp_length	loan_status_clean	count_of_loan_issued
0	n/a	0	5319
1	n/a	1	11051

	emp_length	verification_status	count_of_loan_issued
0	n/a	Not Verified	1219
1	n/a	Source Verified	3217
2	n/a	Verified	11934

For now, we just wanted to retain these records and study them further to determine the impact on loan status, however at this stage it would be nice to give it a suitable label other than n/a may be something like “not_specified”

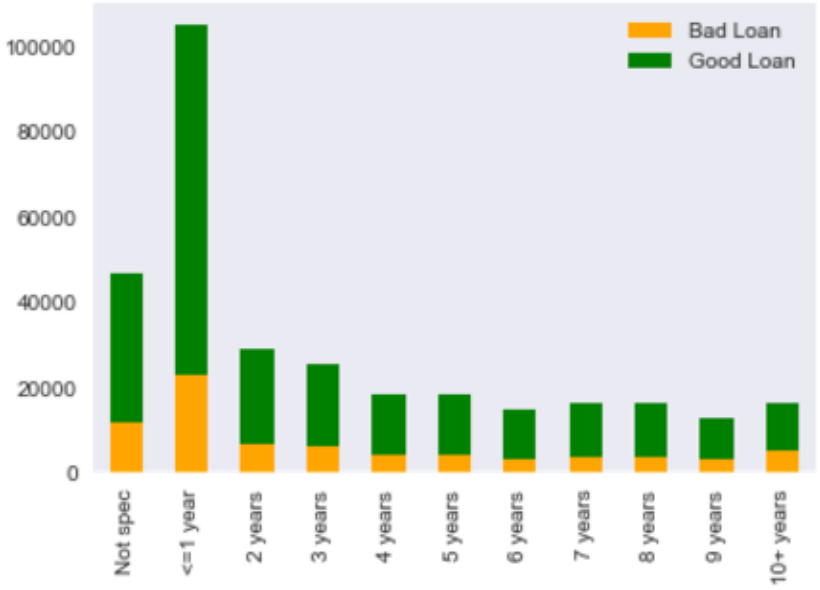
Cleaning Emp_length

We cleaned up the column by replacing “<” embedded in the values with space. We also see many applications having “n/a” against emp_length.

Action	Verification_status
< , +, Years	Replace with ‘ ‘
n/a	'Not Verified'
Remove	Nothing

Below table displays the % contribution of the emp_length towards loan_status.

	Bad Loan%	Good Loan%	Rtotal%
Not spec	3.651373	10.987588	14.638960
<1 year	7.183879	25.637184	32.821063
2 years	2.112059	6.905796	9.017855
3 years	1.879020	6.093128	7.972148
4 years	1.366645	4.451528	5.818173
5 years	1.380721	4.439954	5.820675
6 years	1.059783	3.541578	4.601361
7 years	1.155814	3.900053	5.055867
8 years	1.222755	3.896299	5.119054
9 years	0.973136	3.041090	4.014226
10+ years	1.663810	3.456808	5.120618
Ctotal%	23.648995	76.351005	100.000000



It's intuitive enough to assume that the loan payments would continue as long as individual continues to work but we would like to explore if number of years have anything to do with the loan defaults specially looking at the lower range (1-3 years) of emp_length given the number of loans approved for this group in the 2014-2015 dataset.

Exploring “home_ownership”

This indicates home ownership status provided by the borrower while furnishing the loan application. Values are: RENT, OWN, MORTGAGE, OTHER.

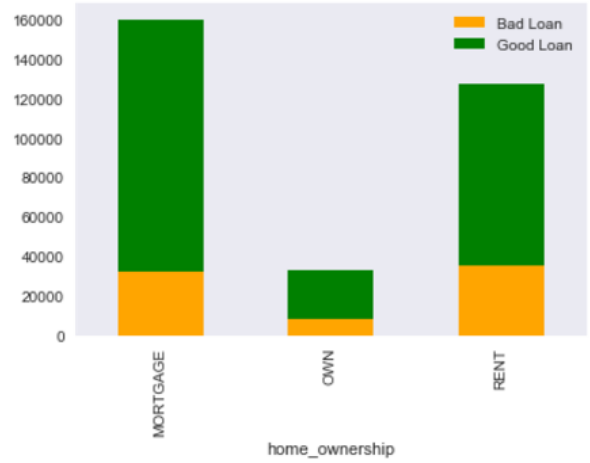
Here is the loan number distribution:-

	Home Ownership	Number of applicants
0	MORTGAGE	159786
1	RENT	127259
2	OWN	32642
3	OTHER	1

It’s interesting to see only one record against “OTHER” category. We verified the loan status for this record and found out its fully paid (loan_status_clean = ‘1’), We eliminated this row from our dataset. We also assigned numeric values to each of the categories.

Cleaning “home_ownership”

Action	loan_status_category
1	‘OWN
2	'MORTGAGE’
3	‘RENT’
REMOVE	‘OTHER’

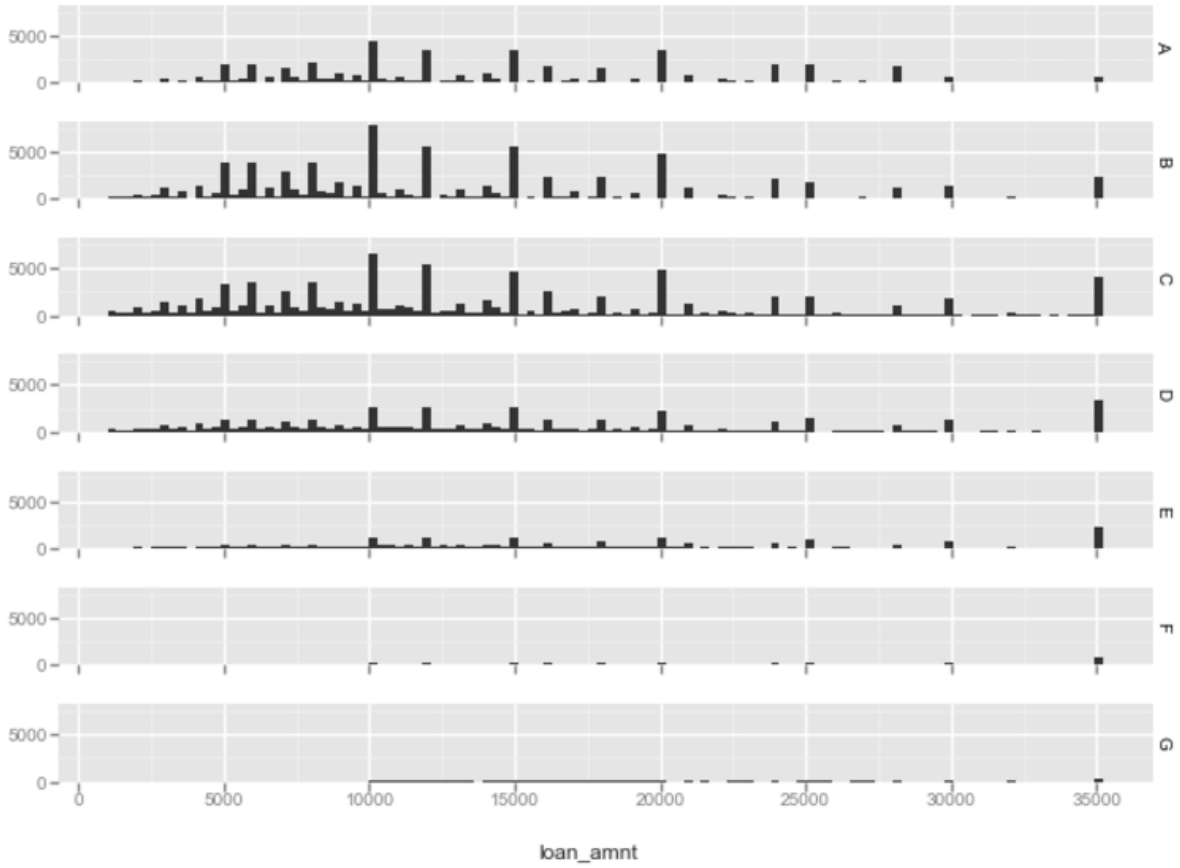


	Bad Loan	Good Loan	Rtotal
Mortgage	10.055460	39.926553	49.982014
Own	2.532790	7.677822	10.210612
Rent	11.060819	28.746555	39.807374
Ctotal	23.649069	76.350931	100.000000

"Rent" status seems to have the biggest contribution to the bad loans as compared to "Mortgage” and “Own" categories.

Exploring “grade”

This is LC assigned loan grade: There are 7 loan grades ranging from A: F, A being the finest and F being the lowest grade. Let’s look at distribution of loan_amnt against grades.



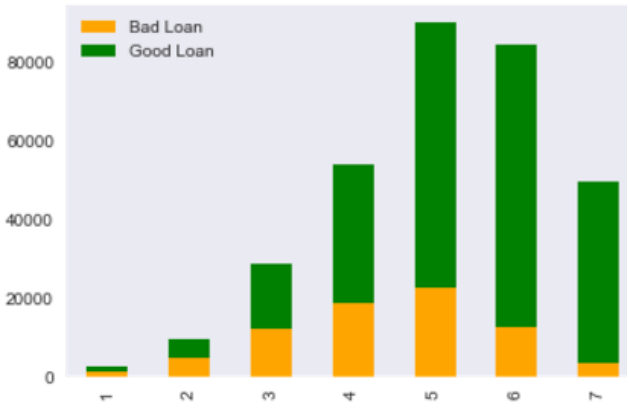
More loans have been allotted to the loan grade A, B, C, D compared to the lower grades.

Cleaning loan_grade

We will assign numeric values to the grades: Since G being the finest we assign 1 for it. And 7 to the lowest grade A.

Action /Assignment	grade_value
1	“G”
2	“F”
3	“E”
4	“D”
5	“C”
6	“B”
7	“A”

	Bad Loan	Good Loan	Rtotal
1	1345	1171	2516
2	4684	4874	9558
3	12386	16618	29004
4	18618	35484	54102
5	22493	67627	90120
6	12519	71965	84484
7	3558	46345	49903
Ctotal	75603	244084	319687



Loan grade 1, 2 & 3 appear to have equally distributed good and bad loans.

Exploring “term”

Term indicates number of payments on the loan. Values are in months and can be either 36 or 60. We will see if term has any impact in a loan getting paid or defaulted. Below is the distribution of applications by “term”. Most of the loans are issued for 36 months timeframe.

	Term	No. of applicants
0	36 months	234926
1	60 months	84762

Later, we found out the grade for “36 month” and looks like many low grade loans have been allotted against 36 months.

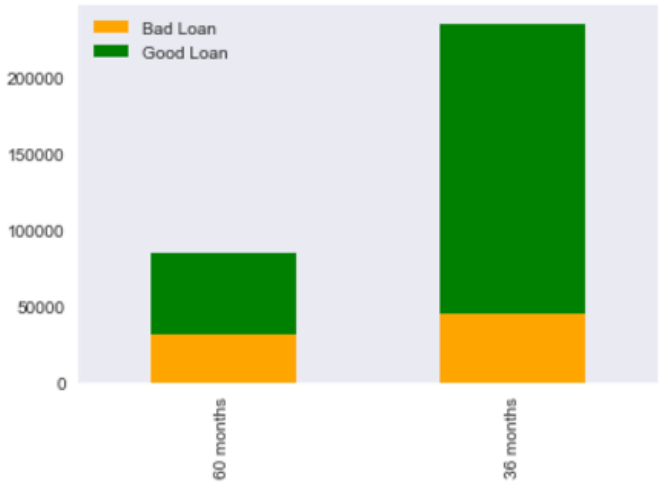
	defaulted_loan	term	grade	count_of_loan_issued
0	0	36 months	A	3471
1	0	36 months	B	10656
2	0	36 months	C	15682
3	0	36 months	D	9773
4	0	36 months	E	3938
5	0	36 months	F	965
6	0	36 months	G	147
7	0	60 months	A	87
8	0	60 months	B	1863
9	0	60 months	C	6811
10	0	60 months	D	8845
11	0	60 months	E	8448
12	0	60 months	F	3719
13	0	60 months	G	1198

When we look at the grade and number of application defaulted for 60 months, we see there are good number riskier loans allotted against 60 months term which got defaulted.

Cleaning “term”

We will assign numeric values to the term values:

Action	term_clean
1	“36 months”
0	“60 months”



18 % of the 36 months term loans have turned bad vs 36 % of loan with 60 months term. We will check its predictive power in the modeling section.

Logistic Regression

Logistic Regression becomes an obvious choice as it encodes a binary outcome.

Logistic regression with “home_ownership”

Before running unique list of values for home ownership using logistic regression against loan status, we have to create individual columns for each value, referred to as dummy variables. Each column will have a True(1) or a False(0) value associated with the individual loan that has either "Rent", "Mortgage", "Own". We eliminated "Other" as there was only one record and that was fully paid. And here is our co-efficient values.

Inference: Home ownership status marked as "Rent" has the lowest chance of 0.14 paying back the loan. An additional study of other related features could have shed more light to determine its predictive strength. It will be interesting to see the corresponding “dti” (debt to income ratio) values for those individuals.

	Status	Coef
0	MORTGAGE	[0.464353997439]
1	OWN	[0.259040650821]
2	RENT	[0.141502939778]

Logistic regression with “emp_length”

Using our column of years employed, we create dummies so that we could easily run the logistic regression. We're trying to see which length of employment is best predictive of someone paying back their loan.

Inference: There does not seem to be striking variation in the coefficient values for the number of employment years between 2 to 9 years. They are hovering over the range of .01 to .05. However, applications with emp_length <= 1 year and also those which dint have the employment length specified” Not-specified” have negative coefficients. They are more likely to be defaulting. To my understanding, anybody not specifying the employment could be because they might not be bound with employment at the time loan was funded or could be in some other business. Their annual_inc and verification status could be analyzed further to draw any co-relation yet this feature does have predictive power to a certain extent as much intuitive it appears to begin with.(a person would be continuing to pay off loans as long as he/she is employed.)

	Emp_length	Coef
0	< 1 year	[-0.0108478832174]
1	2 years	[0.0512277076293]
2	3 years	[0.0653339840592]
3	4 years	[0.0504631305282]
4	5 years	[0.0335182117012]
5	6 years	[0.0456009552521]
6	7 years	[0.0648643222256]
7	8 years	[0.0386935922295]
8	9 years	[0.0158223785232]
9	10+ years	[0.12079719402]
10	Not specified	[-0.289418274895]

Logistic regression with “verification status” and “term”

Both the features have been converted into numeric values: 0 and 1. We will apply LR to both and see the results:-

	Status	Coef
0	verification_status_clean	[-0.352843271276]
1	term_clean	[0.842305079757]

Inference: -> Any application which does not have the source of income verified is more likely to fall in the default category. This could be a good predictor. And about the term for the loan is funded there is .84 chance of individuals paying off the loan if they are granted loan for 36 months.

Algorithms

Here we split the dataset into train: test in the ratio 7:3. We are going to apply below techniques and record model performances.

- Logistic Regression
- Random forest
- KNN
- Decision Tree

Target Variable = [loan_status_clean]

Input Variables= [emp_length_clean, verification_status_clean, term_clean, home_ownership_clean, grade_clean

Logistic Regression Results

	precision	recall	f1-score	support
0	0.51	0.13	0.20	24949
1	0.78	0.96	0.86	80548
avg / total	0.72	0.76	0.71	105497

Accuracy:
76.4315572955

Random Forest

```
from sklearn.ensemble import RandomForestClassifier
from sklearn.neighbors import KNeighborsClassifier
rf = RandomForestClassifier(n_estimators=10, min_samples_split=2)

rf_result=rf.fit(X_train,y_train)

rf_pred = rf_result.predict(X_test)
accuracy = accuracy_score(y_test, rf_pred)
accuracy
```

```
0.76482743585125645
```

kNN

```
knn = KNeighborsClassifier(n_neighbors=21)
knn_result=knn.fit(X_train,y_train)
knn_pred = knn.predict(X_test)
accuracy = accuracy_score(y_test, knn_pred)
accuracy
```

```
0.756049935069243 67
```

Logistic Regression gave us an overall accuracy of about 0.764

Random forest gave us an overall accuracy of about .764

Logistic Regression gave us an overall accuracy of about .756

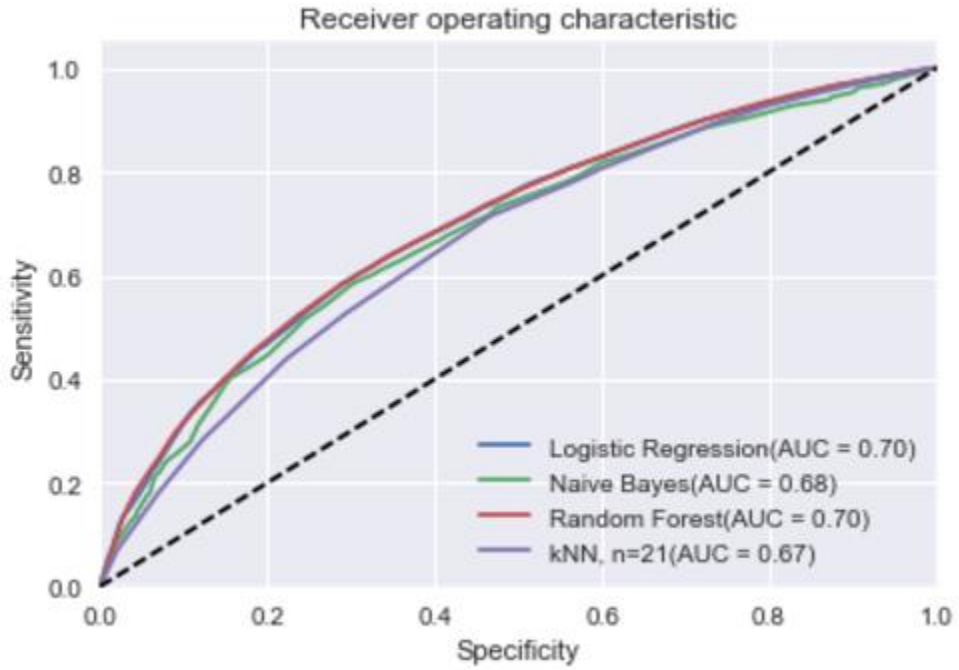
Decision Tree

```
Accuracy:0.739

Classification report
             precision    recall  f1-score   support

          0       0.42      0.28      0.34     24949
          1       0.80      0.88      0.84     80548

avg / total       0.71      0.74      0.72    105497


Confusion matrix
[[ 7061 17888]
 [ 9646 70902]]
```

Confusion matrix is often used to describe the performance of a classification model on a set of test data for which the true values are known. As much as this model provides us the decent accuracy value of .73, yet it gives us terrible specificity value. We see out of 24949 class 0 loans in our test dataset, our model could predict only 7061 defaulted loans correctly. Predictions for Class '1' is better with high precision and high recall values. So, overall our model made correct prediction for about 77,963 (7061+70902) applications and incorrect prediction for about 27534 (9646+17888) loans. So, now the question is why is the model bad in predicting Charged Off loans when it gives out a decent accuracy?

Possible causes could be the imbalance in the data. We have 76 % of the 319688 records classified as "Fully Paid" which hints us towards imbalance distribution of data in our dataset due to which minority classes (0) are not able to make a big impact making accuracy almost meaningless and a poor metric for inferring model quality. AUC curve could be the way to go!

ROC and AUC



We tested our dataset through four modelling techniques. Looking at the ROC graphs above, we find not a single model standing out to be significantly different than the others all and of them having comparable AUCs. Although the overall prediction accuracy is good, yet the prediction accuracy for defaulter instances (class 0) are not as good as expected across the board. Out of all the classification algorithms used on the Lending Club dataset for the year 2014 -2015, Logistic regression and Random forest stand at .70 as the best overall accuracy. KNN and Naive Bayes scores are weak as compared to other algorithms.

Our input variables Term, Emp_length, home_ownership_status, verification_status seem to have a decent predictive strength against loan_status (default/paid off).

There is a need to eliminate data imbalance by adopting oversampling/under sampling techniques which could serve as the future direction of this project. For now, if we go by the AUC, and by the simplicity of model implementation Logistic regression should be our model of choice in predicting Loan Defaults in the Lending Club dataset (2014-15)

Conclusion & Future Direction

Determining the loan outcome is clearly not an easy task. Going through various techniques, we found out that some of the intrinsic characteristics of a loan by itself have a major role to play in determining the loan results. Like grade, term in our case. On the borrower side, verification status (source of income verified) was an important factor. Length of lender's employment was certainly a good predictor. Homeownership status gave us an insights about how the defaults predictability improved with the individuals who are renting than the individuals who own a house or have a mortgage. Borrower's Information collected from the loan application is equally important as we saw "n/a" for employment length yielded greater chances of loan defaults.

No model can guarantee 100 percent accuracy. But there will always be value in a model that can help discern future outcomes. Likewise, this model by no means guarantee best results to drive any investment/financial decisions, but it certainly will be a great source to gain insight into potential lending outcomes at the time of loan's origination making us all smarter investors or diligent borrowers!

Future Direction:

Some possible reasons and ways to improve the results:

It will be interesting to see how our model performs against relatively larger dataset. We can combine all the 2007-15 data tables and perform the prediction again.

Not all default cases are the same, and some late payments could still be recovered later on, we could include late and in-grace status in our model so instead of a binary classification, multinomial predictions could be employed to take into account the different types of statuses so as to make much more granular prediction.

We played with limited features to make the predictions. A further analysis may incorporate some of the discarded features, such as emp_title, debt to income ratio, annual inc interest rate or external information like relevant economic status into the model to produce more meaningful results. Also, some of the variables, which probably are known to offer the strongest predictive power, are removed from the publicly available data-set, such as fico/credit scores. That information would have been helpful as well.

Future work would also include looking into each of the various methods for fixing up imbalanced data by oversampling, undersampling, or synthetic data generation and then generating AUC for better results.

Learnings and Credits

In this project I was able to make use of a range of skills learned through Springboard's Data Science career track. The capstone project gave me a platform to apply the fundamental concepts in the real world scenario. Lending Club publicly available data set was very streamlined and clean to work with. I gained experience in exploratory analysis, data-munging, and data-analysis and was able to touch upon the basics of several modelling techniques.

Code and the synopsis can be located in the following github link.

GitHub repo: https://github.com/DataGalore/Capstone

Special credit goes to my mentor **Kenneth Kihara** for the valuable suggestions, feedback and discussions.

References

http://www.dataschool.io/simple-guide-to-confusion-matrix-terminology/

http://kevinyuan.ca/2016/08/01/Interpreting-Logistic-Regression/

https://www.youtube.com/watch?v=yLsKZTWyEDg

http://rpubs.com/jfdarre/119147

https://www.analyticsvidhya.com/blog/2015/09/naive-bayes-explained/

https://www.kaggle.com/amanemisa/want-to-get-low-interest-rates

http://blog.yhat.com/posts/machine-learning-for-predicting-bad-loans.html

http://cs229.stanford.edu/proj2015/199_report.pdf

https://kevinzakka.github.io/2016/07/13/k-nearest-neighbor/

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