Final Report

Data Science for Business Analytics

Professor Foster Provost

Group 8

Jahnavi Kalyani

Sakshi Mishra

Yash Rajan Raval

Date

May 9, 2017

Appendix:

Business Understanding

Data Understanding

Data Preparation

Data Modeling

Evaluation

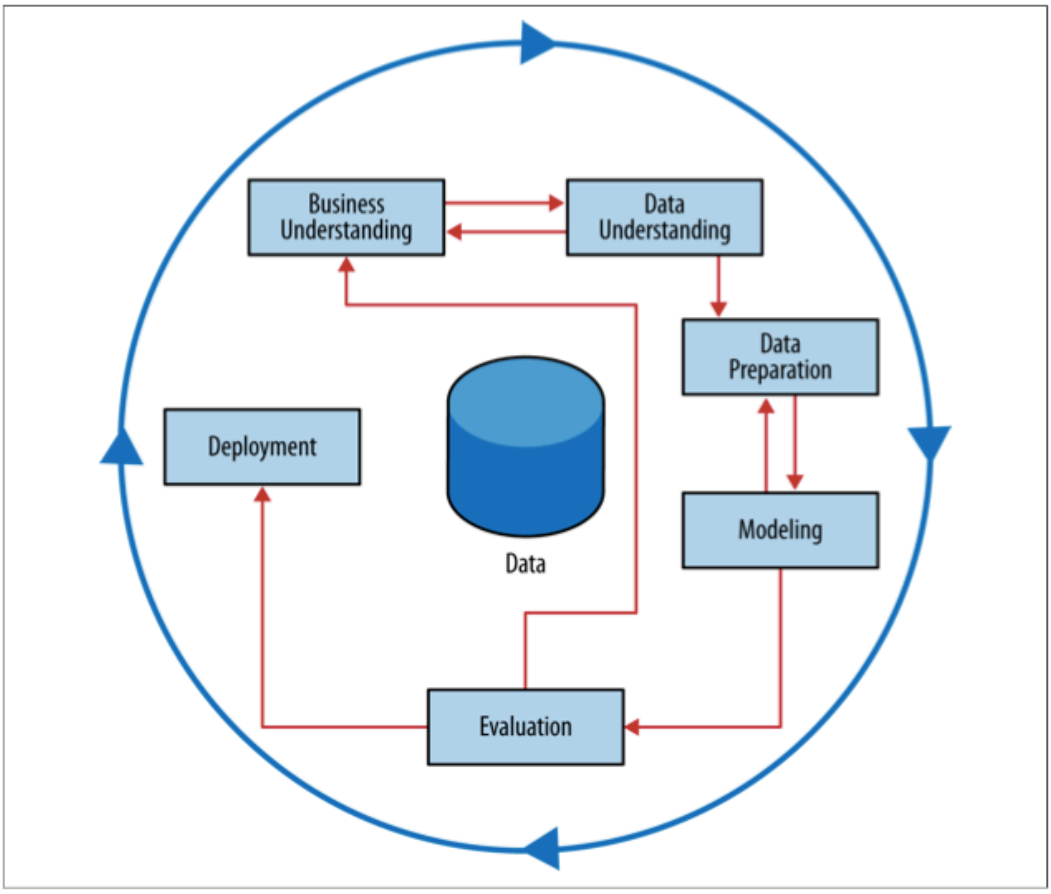
Deployment

Team Contributions:

Jahnavi Kalyani: Data Cleaning of partial attributes, Models and their evaluation - Decision Tree Model, Bernoulli Naive Bayes models, Report preparation.

Yash Rajan Raval: Data Cleaning of partial attributes, Models and their evaluation - Logistic Regression, SVM and their evaluations, Report preparation.

Sakshi Mishra: Data Cleaning of partial attributes, Models and their evaluation - Random Forests, Gaussian Naive Bayes, Report preparation.



Business Understanding

Peer-to-Peer (P2P) lending services encompasses the practice of lending or borrowing money involving two individuals or businesses. With the advent of online marketplaces whereby two or more individuals can participate in an exchange/trade it only behooves the arrival of the P2P platform on the digital stage.

PROSPER LOAN DEFAULT

P2P, which is a growing trend in America, stands to attract a plethora of individuals who seek to diversify or expand their investments with maximal returns beyond the traditional investment avenues such as mutual funds, banks, stocks and so on.

Subsequently, on the basis of the aforementioned scenarios, there have been several companies such as Prosper Loans, Lending Club et al, that provide a platform on which individuals can take part as either lenders or borrowers. These companies tend to operate with relatively less overhead due to the digital nature of the venture thus enabling them to provide competitive rates as opposed to other unregulated sources which may command notoriously high interest rates.

Moreover, it also paves way for those who seek a stable, fruitful investment in a regulated environment and are too risk averse to invest in relatively riskier avenues like the stock market, which has a great degree of uncertainty.

While the lenders have an opportunity to earn higher returns as compared to savings and investment schemes offered by banks, they are at a risk of borrower default on these loans.

In lieu of the nature of the transaction that takes place between two individuals, it is quite a challenge for P2P lending companies to identify possible defaulters. Due to this possibility, lenders who may get good returns also bear a significant amount of risk, pertaining to a default from the borrower's end. This eventually boils down to the company having to bear a certain amount of responsibility for the same. Furthermore, P2P organisations also find it difficult to attract prospective lenders on their platform and hence effectively curbing potential revenue streams.

Upon recognising this issue, our team has decided to leverage data to facilitate lenders entering a transaction to vet prospective borrowers, notably those who possess a high risk of defaulting on a loan which would in turn effectively lower the risks they carry when they commit to a transaction. This will also enable the P2P lending platform to safeguard their borrowers.

Specifically, we examined the P2P lending site called Prosper. Prosper was founded in 2006 and since then has transacted over $6 billion through its platform.

Our business problem strives to help predict such instances of defaults made by the borrowers, so as to ensure that the investors can safeguard themselves and prevent a

scenario where they would lose out on their investments, thereby minimising risk in lending money through P2P platforms.

Moreover the business plan can also be used as a method to identify borrowers who have a less chance of defaulting and target them with loan offers at competitive APRs. Prosper can use our model to help lenders avoid potential flight risks and take preventive measures beforehand to shield their position against a borrower who is likely to default. Prosper can benefit from this model by instilling faith in lenders to use their platform as opposed to other traditional methods of investment or other competitive companies.

Data Understanding

The dataset we used is from 'Prosper Marketplace Inc.', a California based peer-to-peer money lending and borrowing platform. Borrowers post a request with the loan amount and the purpose. Following this, lenders can choose to invest in the loan listing of their choice. The dataset contains **113,937** records with **81** predictor variables

Considering we have a particular target variable to focus on, we will follow a **supervised learning** approach. We have chosen the target variable such that it can improve the stated business problem. We now define the target variable - precisely.

Target Variable

"Will the borrower default on the loan?"

The target variable is recorded as a categorical variable where 1 indicates a default on the loan and 0 indicates a non-default by a particular borrower. The target variable 'Loan Status' has been derived by combining 'Defaulted' and 'ChargedOff" since both indicate borrower default.

Features

There are 80 attributes in our dataset. Some of the attributes that we could use as features to gives us more information on the target variable are as follows:
Prosper Rating (Alpha and numeric), Employment status, Employment duration, Occupation, IsBorrowerHomeOwner, CreditScoreRangeLower, CreditScoreRangeUpper, IncomeRange, BorrowerRate, TotalCreditLinesPast7Years, CurrentCreditLines, DelinquenciesLast7Years, RevolvingCreditBalance, AvailableBankcardCredit, StatedMonthlyIncome, PublicRecordsLast12Months, OnTimeProsperPayments.

A critical part of the data understanding phase is estimating the costs and benefits of each data source and deciding whether further investment is merited. We are not considering costs in our analysis, as we are not aware of the industry costs incurred in the process and factors like customer satisfaction and future business benefits. Hence, we are not including the same in this project. Though, as we are following a Supervised learning approach in this process we are rest assured that our costs won't be very high, considering our data is labeled. This is another advantage of this data-set which helps

us in reaching our business goal without having to follow other approaches in order to predict loan default.

Some key initial observations from the data are:

- Majority of loan defaults occurred before 2009
- Majority of borrowers have income between $25,000 and $75,000; 92.4% of these incomes are verifiable (we can trust the numbers)
- Majority of borrowers kept their debt-to-income ratio below 50%
- Majority of the loans have a term of 36 months

Later, during the Data Preparation stage, we have further estimated which features are more informative based on their information gains and correlations with each other.

Data Preparation

In this phase, we tried to see the practicality of feature retrieval and modification. Since a significant part of the data wasn't formatted to fit the models we would eventually use, we had to do a lot of data cleaning before we could proceed further.

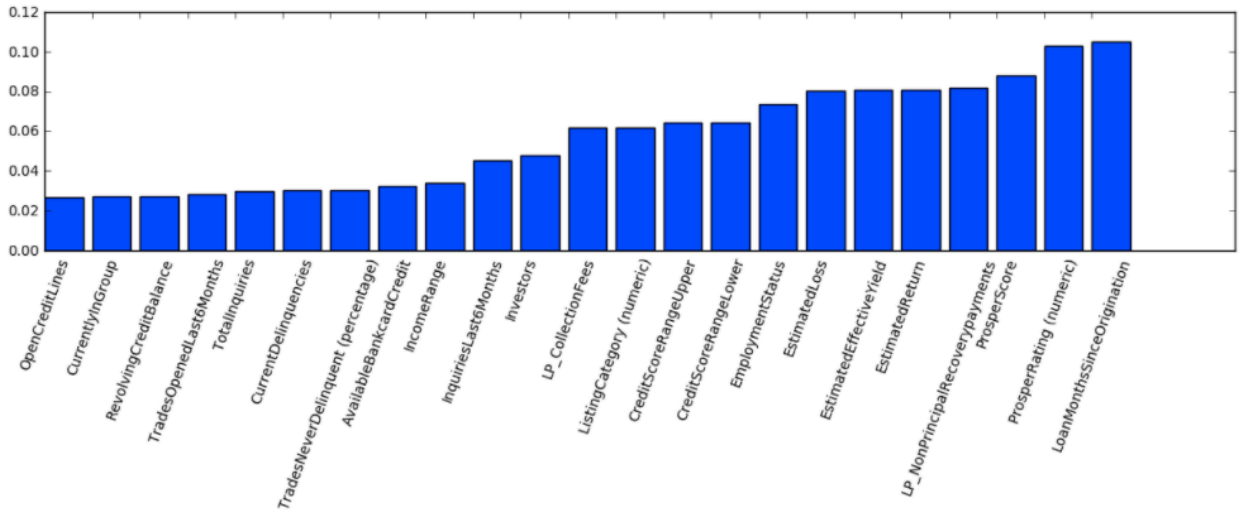
The first step was to import the data into a pandas data frame. We then proceeded to modify the columns to suit our analysis.

Data cleaning:

1. Dealing with missing data: We wanted to analyse if ‘missingness’ can be predictive and hence instead of removing attributes or rows with a lot of missing data, we chose to include it in our analysis. Also, instead of using standard methods of replacing the missing values with mean or mode or median, we have introduced an outlier value ‘-9999’ or ‘NA’ to treat such missing values. We then analyse each column to understand how to manipulate empty values in columns so that they do not affect impact analysis.
2. We removed duplicate and redundant features features. For example, ‘Prosper Rating (Alpha)’ is a duplicate of ‘Prosper Rating (Numeric)’, ‘CreditGrade’ was a duplicate (with many empty rows) assigned based on ‘CreditScoreRangeLower’ and ‘CreditScoreRangeLower’ etc. Also some variables for dates had equivalent duration attributes in the date - For example, ‘ClosedDate’, ‘DataCreditPulled’, etc.
3. We also created best possible additional features from existing ones that can help us gain more information about the target variable through feature engineering. For example, we created seasonality for the dates based on quarters, etc.
4. We converted categorical variables and text fields to numeric fields and boolean (0/1) for our analysis. For example, ‘IsBorrowerHomeowner’, ‘IncomeRange', etc.
5. We identified potential target leaks and removed them - ‘ListingNumber’, ‘LoanFirstDefaultedCycleNumber’, ‘LP_NetPrincipalLoss’, etc. We removed these target leaks as they have high influence on target and are not very good indicators. Target leaks can introduce selection bias and other discrepancies.

6. For the Target Variable - ‘LoanStatus’, we converted the categorical field to numeric. It takes values 0 and 1. We considered ‘Charged Off’ and ‘Defaulted ‘as 1 (A Default) and ‘Current’, ‘Completed’, ‘Final Payment in Progress’ as 1 (Not a Default).

We then identified most informative set of variables using entropy and information gain (for both continuous and categorical variables). Also, we used a function to determine the best threshold for the continuous variables to calculate the maximum IG.
Some of the highest predictors are - MonthsSinceOrigination, EstimatedReturn, IncomeRange, CurrentDelinquencies, OpenCreditLines, CreditScoreRangeUpper, etc.

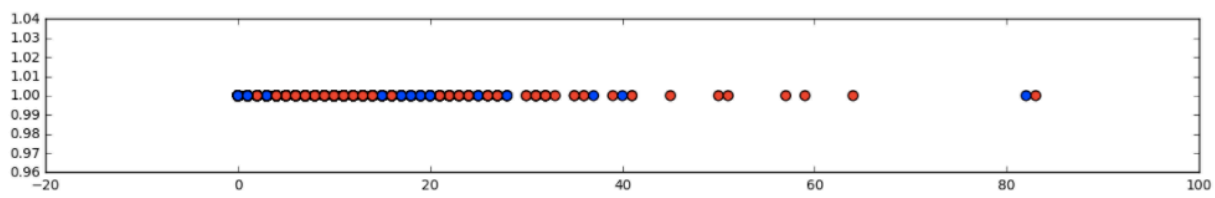


For our understanding we plotted a few variables to understand the data as can be seen from the graph below:

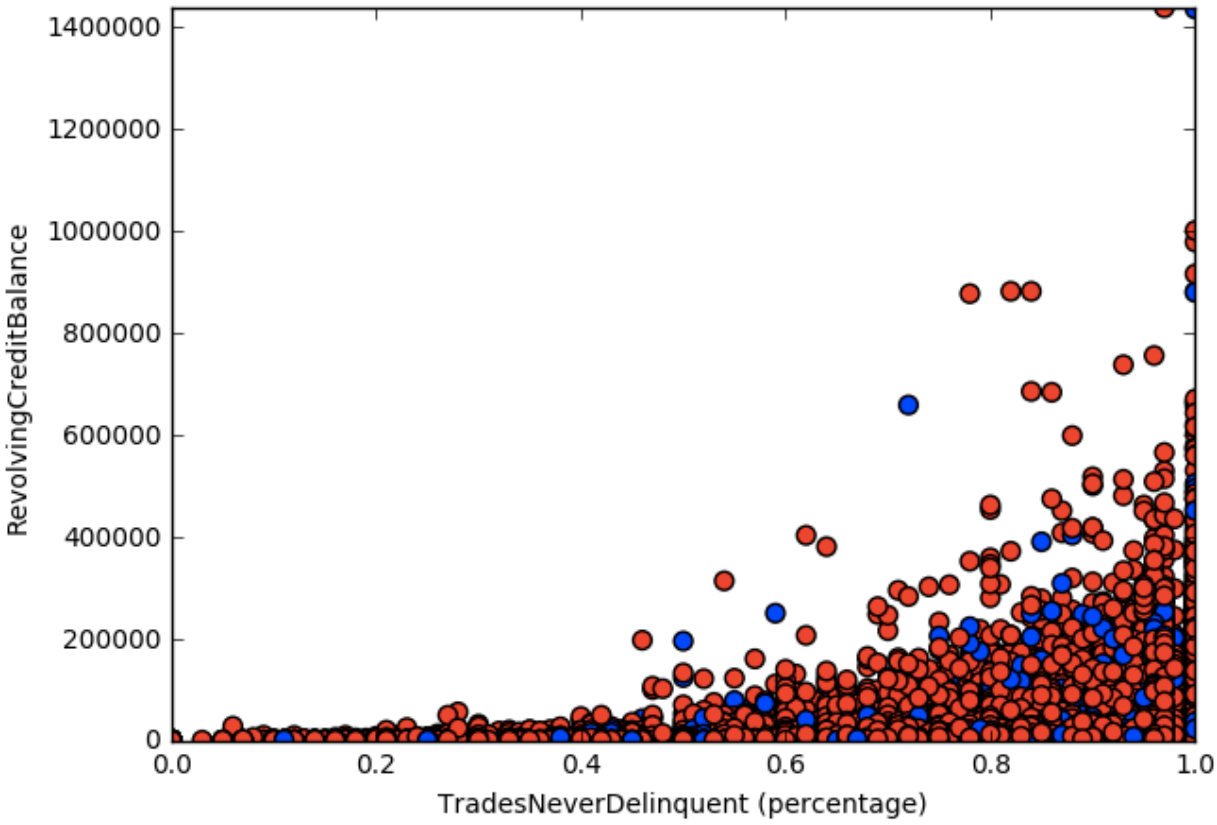
```
# For CurrentDelinquencies
plt.rcParams['figure.figsize'] = [15.0, 2.0]

color = ["red" if x == 0 else "blue" for x in data['LoanStatus']]
plt.scatter(orig_data['CurrentDelinquencies'], [1] * len(orig_data), c=color, s=50)
```

```
<matplotlib.collections.PathCollection at 0x7f1d94357550>
```



Following this, we have split the data into training and holdout data (80:20 ratio) to perform further analysis with various data models. Later, as explained in the evaluation phase, the optimal split for each model based on cross validation is calculated and our final split is done based on the same.



Modeling

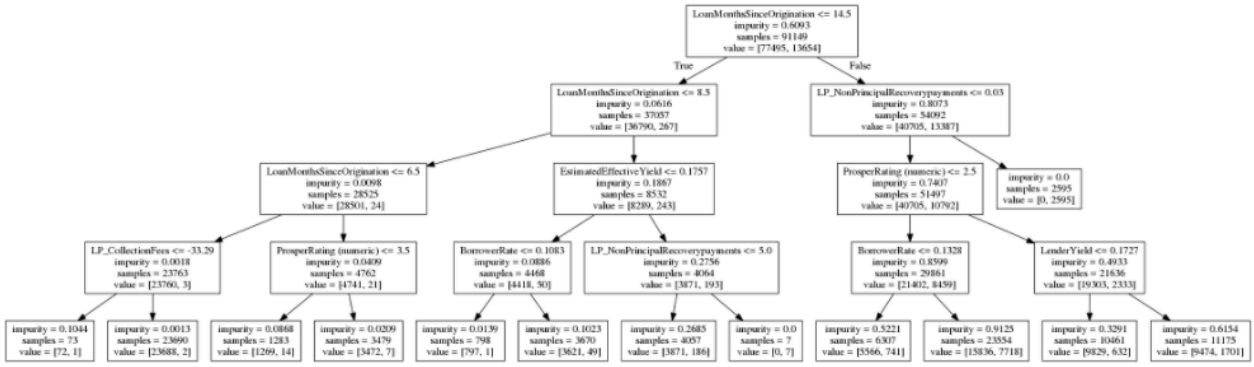
In this phase, we have modelled the data we have prepared in the previous phase to see which model best captures the regularities. Again, while modelling, we ran into various errors like Type Conversion Errors (For example, Logistic Regression doesn't allow strings in attribute values), and had to keep modifying the data to be able to fit the models to the data.

We have tried the following models from the scikit-learn library:

- **DecisionTreeClassifier** : Models via an iterative process where the errors obtained in the previous tree are eliminated in the next tree to the extent possible. Hence, more accurate and refined prediction can be expected.
- **RandomForestClassifier** : Models without considering error factor from the previous tree when creating the new tree. Due to this, model may not have the best predictive value.
- **BernoulliNB**: Implements the naive Bayes training and classification algorithms for data that is distributed according to multivariate Bernoulli distributions; i.e., there may be multiple features but each one is assumed to be a binary-valued (Bernoulli, boolean) variable.
- **GaussianNB**: Implements the Gaussian Naive Bayes algorithm for classification. The likelihood of the features is assumed to be following a Gaussian distribution.
- **LogisticRegression**: Models based on the effect of the dependent variables on the target categorical variable

- **LinearSVC**: Models developed via supervised learning algorithm and clustering

A decision tree with 5 nodes looks like this -



As our business problem requires a class probability estimation, we have explored the above different models commonly used for the same. Based on the results of our evaluation phase, wherein we have analysed the performances of our models on our data-set, we can make an informed decision of choosing the “right” model for our purpose.

As we know that Logistic Regression and Decision Tree classifiers are the most commonly used models in the class probability estimation scenarios, we went ahead and initiated our analysis with the same. To broaden our analysis, we employed another widely used modelling technique, Linear SVM, as it is a simple yet elegant model and serves great purpose for real world scenarios due to its objective function, which performs margin maximising. This is generally considered to pose small errors and thus, is widely used for classification problems. Later, we also wanted to evaluate our data-set

on other alternatives, so we moved to Naive Bayes Classifier as it is another simple classifier which takes all the feature evidence into account. We ran both the Gaussian and Bernoulli NB models in order to attain the best results. Another advantage of Naive Bayes is that it is naturally an 'incremental learner' and it can update its model one training example at a time. However, it should be noted that since our business scenario is not dynamic in nature, we do not need to exploit the advantage that this particular model offers. We need to run our model, in the "Deployment" phase, just once, to reveal potential defaults.

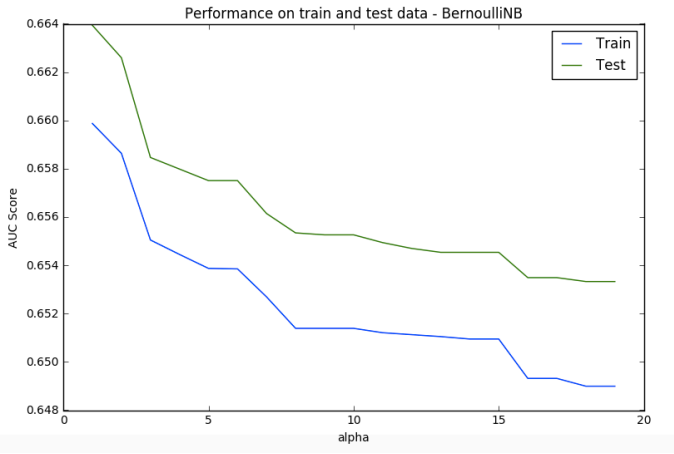
After evaluating the models, we checked the accuracy (Using AUC (Area under ROC Curve)) of each of the models (discussed further in the next section). After incorporating cross validation and complexity control, we further tuned these models so as to get better accuracy.

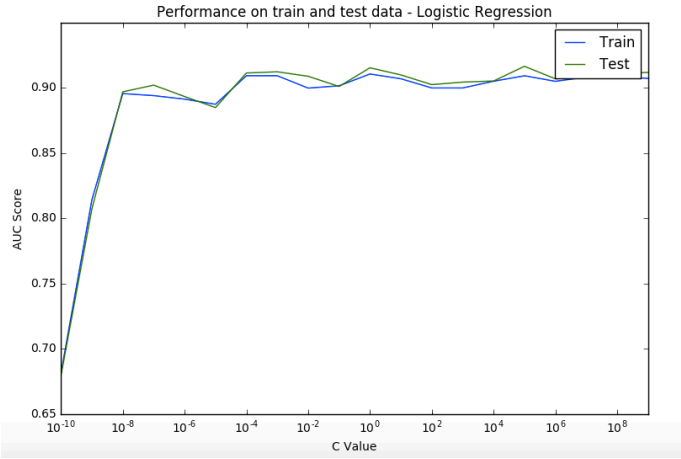
Evaluation

In this phase, we plan to evaluate our model performance and accuracy. We also plan to evaluate the model for accuracy, overfitting, complexity, etc so as to assess whether the metrics are appropriate for the stated business problem. We have assessed the results from the modelling phase rigorously so as to gain confidence that they are not only valid, but also reliable i.e. they are actual regularities and not idiosyncrasies or anomalies.

We started off by calculating the AUC (Area under ROC Curve) and Cross Validation accuracy for each of the models. We noticed that DecisionTreeClassifier, RandomForestClassifier, LogisticRegression and LinearSVC all had AUC (around 0.88) while Naive Bayes (BernoulliNB, GaussianNB) has a much lower AUC (around 0.76).

Following this, we plotted the fitting graphs for each of these based on the complexities. The trends can be clearly observed. This is to introduce complexity control and avoid overfitting as our data has a high likelihood of overfitting (it had 81 attributes to start with!).



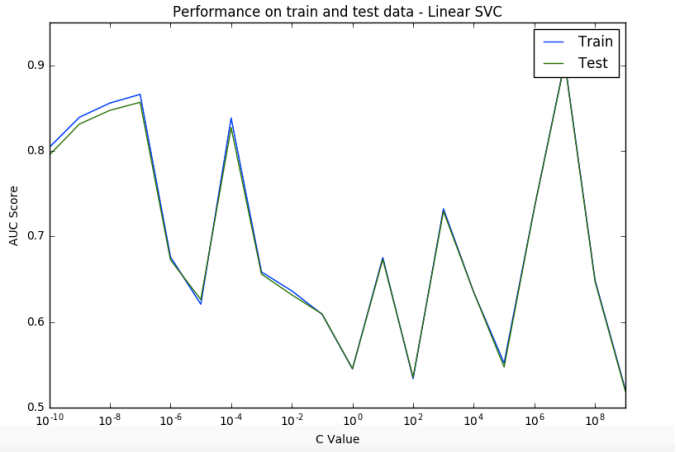


Complexities -

LogisticRegression, LinearSVC - C Value

BernoulliNB - Alpha

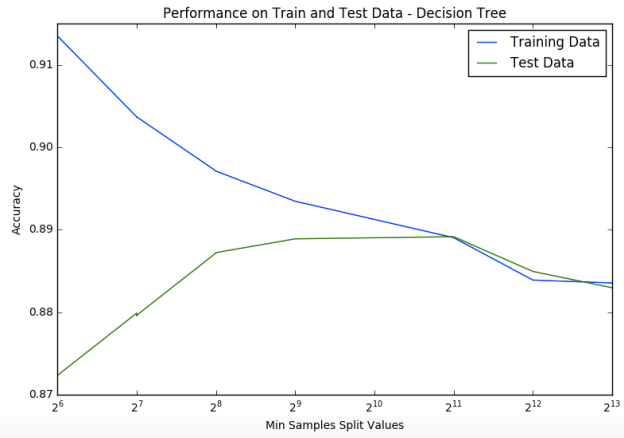
DecisionTreeClassifier, RandomForestClassifier - max_depth, min_samples_leaf, min_samples_split

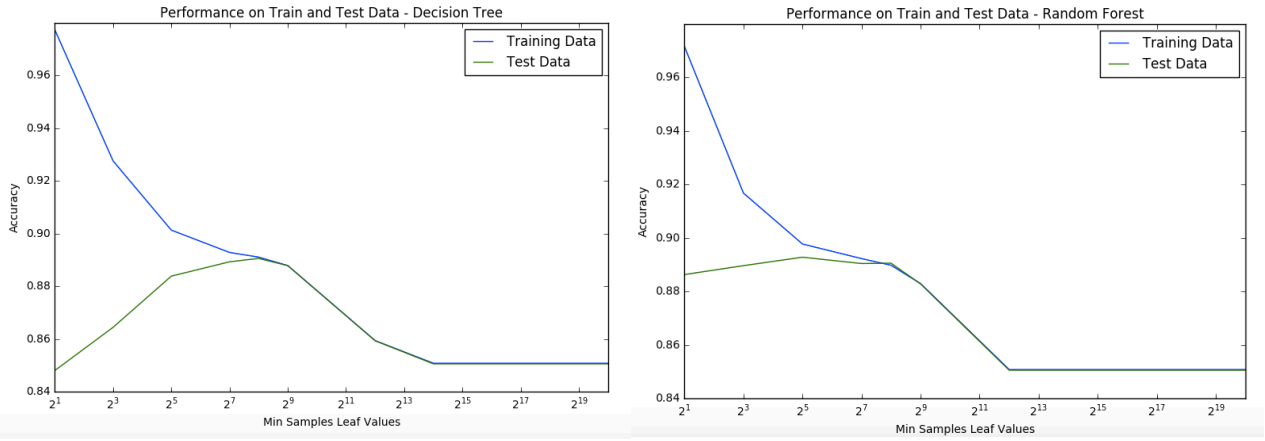


Using the information from this, we figured out the optimal complexities for each of the models.

For BernoulliNB, best complexity was default complexity

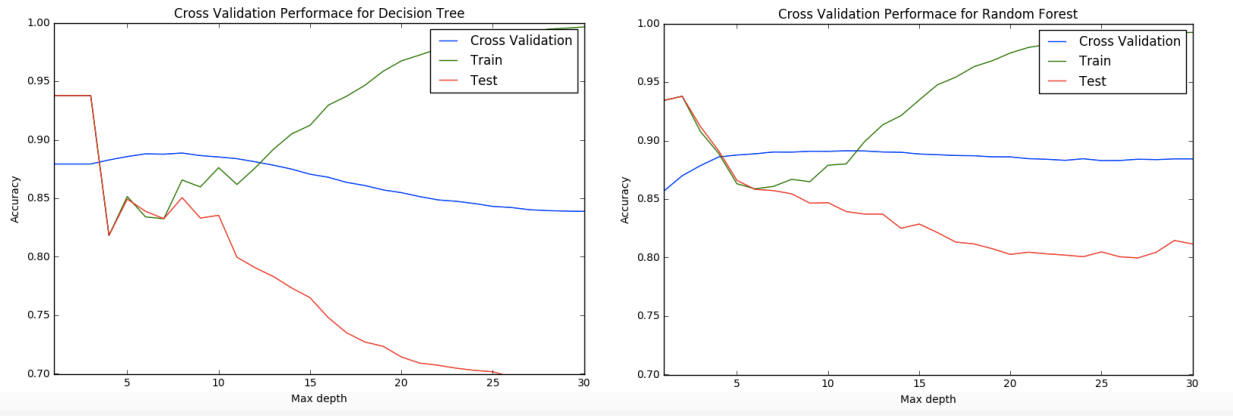
For Decision Tree, best complexities: *Min Samples split - 2^11 and Min Samples Leaf - 2^8 and Max Depth - 8*



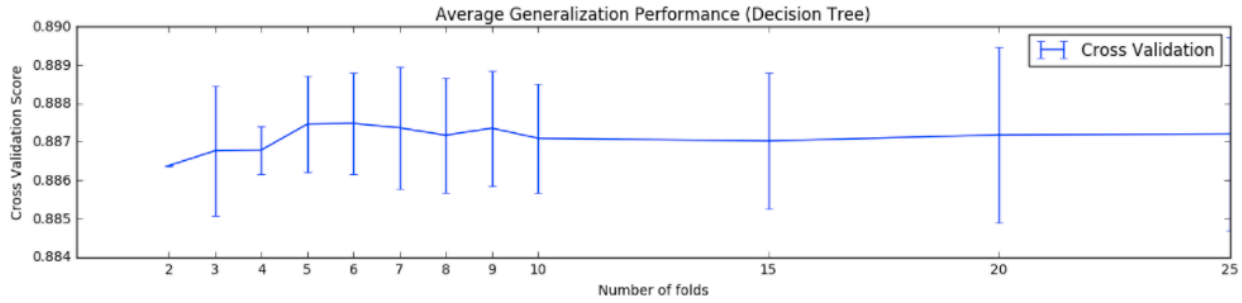


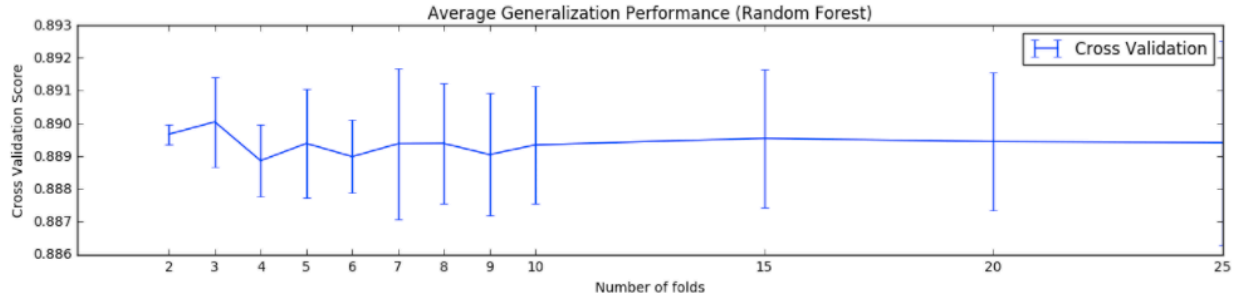
For random forest, best complexities: *Min Samples split - 2^6 and Min Samples Leaf - 2^5 and Max Depth - 7*

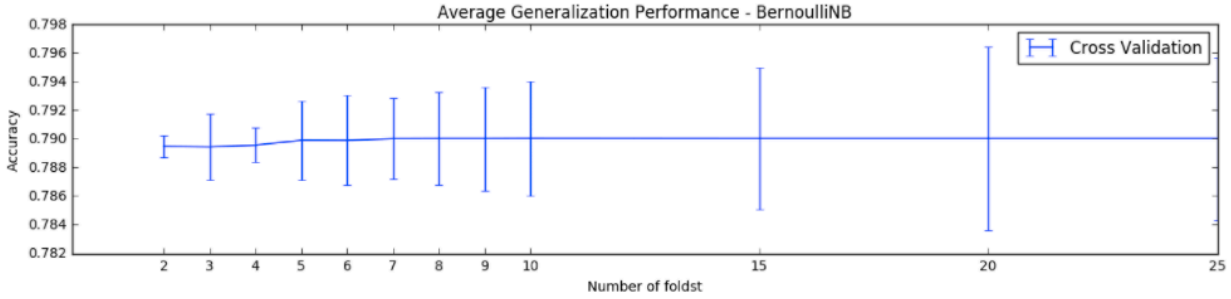
We then tried to look at the cross validation accuracy as well for each of these as well. We also compared the training vs test vs cross validation accuracy to be sure if the models are indeed reliable.

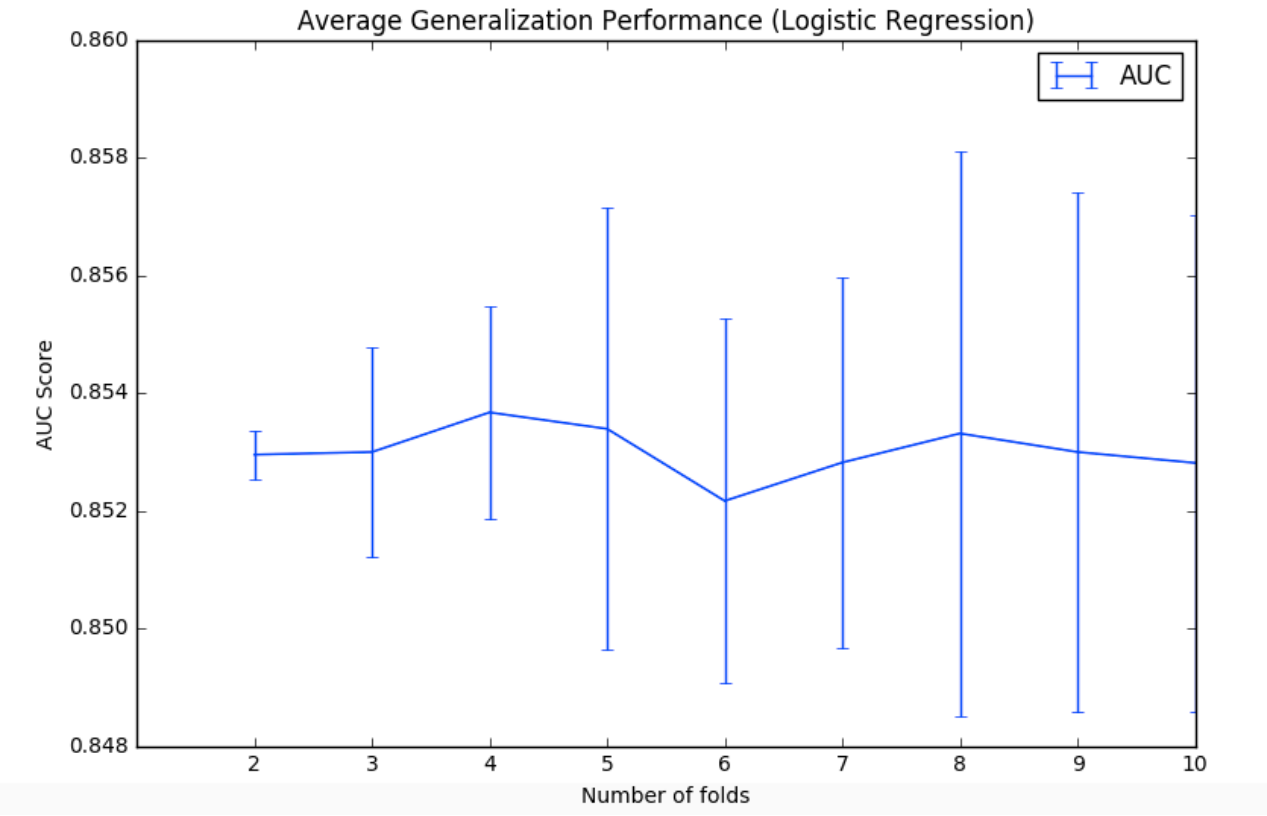


We then decided to find out the best k-fold Cross Validation on AUC Score as well as Cross Validation Score to split the data into training and holdout data for each of the models.









We figured out that each model had it's own optimal fold for the data. For example,

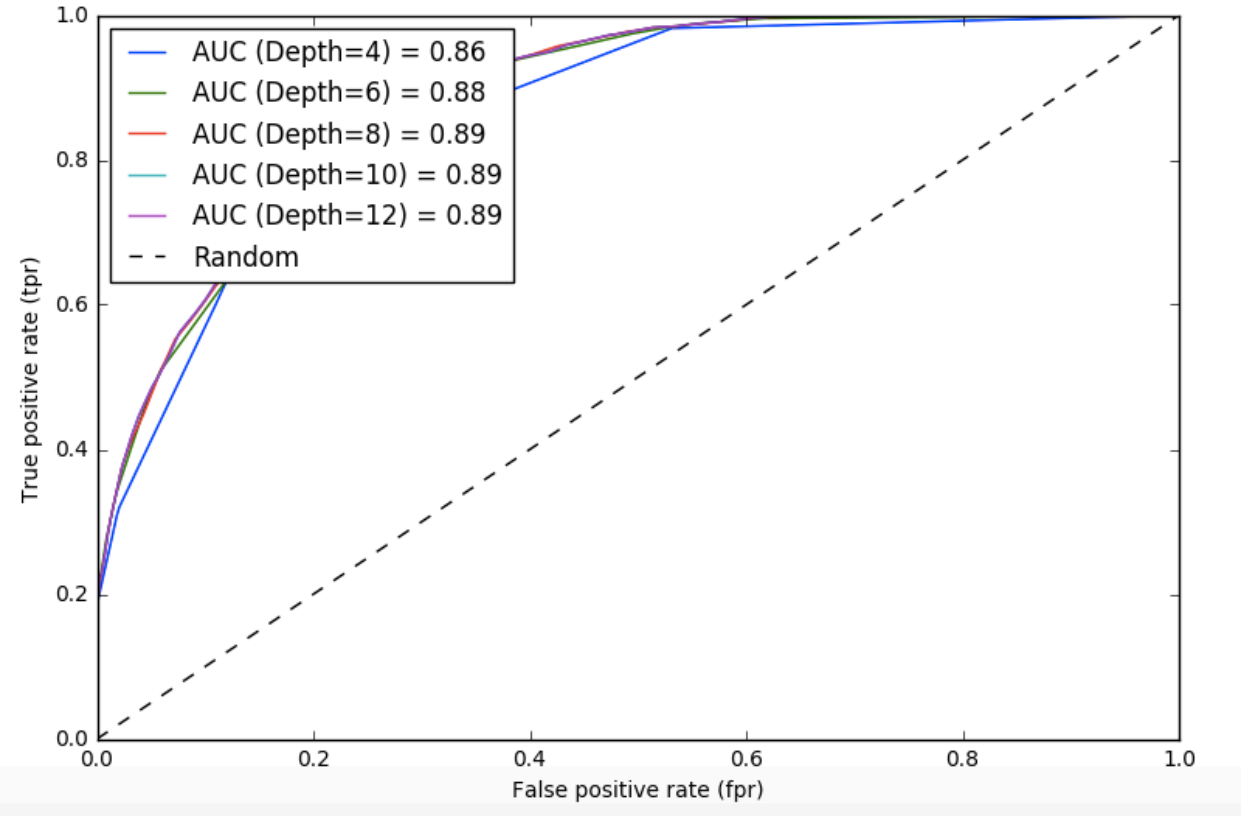
The tradeoff between mean and std is maximum when there are 5-folds for Decision Tree.

The tradeoff between mean and std is maximum when there are 3-folds for Random Forest.

The tradeoff between mean and std is maximum when there are 4-folds for Naive Bayes.

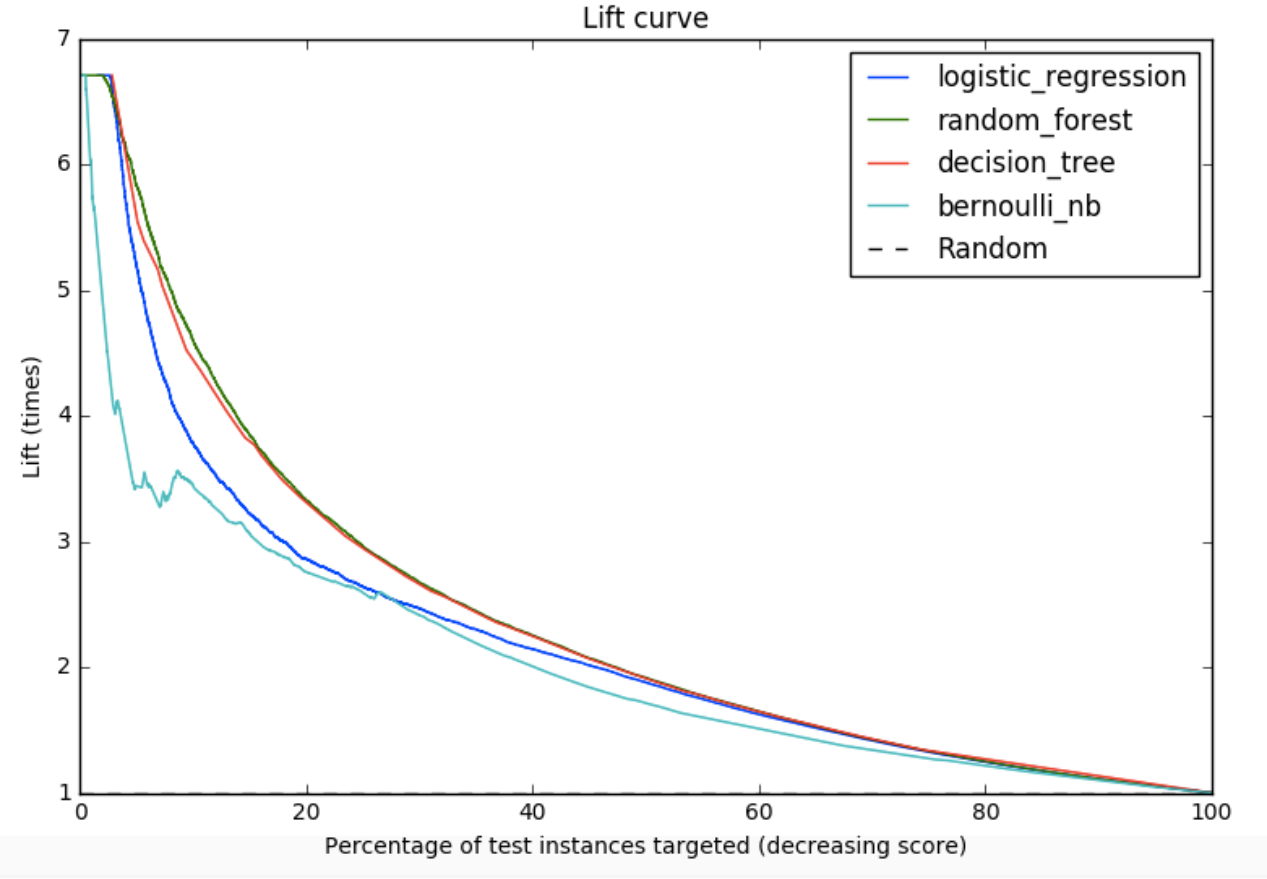
We then built the confusion matrices using the above models using 50% probability. Since our models are scoring models, the confusion matrices are defined with respect a scoring model plus a threshold on the score. For this, we look at the ROC Curves -

For more intuitiveness, we have used the best complexities from the above graphs and plotted the Cumulative Response Curve (CRC) for the models that had comparable accuracies and Naive Bayes as a baseline model.



Conclusion

Based on our analysis and accuracy calculations, we have decided that **Decision Tree Classifier** (with complexities as most optimal) is the best model for our business problem. Also, it offers an added advantage of being easily understood for person with not much technical background or understanding in statistics. This plays an important role as the final decision in choosing the model is mostly at the hands of the Management.



Deployment

The data mining phase saw us utilise data from an existing data set of Prosper to create a model, that would be applied to prospective borrowers on the platform.

At this point to build the model, we have used Python 3.0, to build the data mining system that consists of fetching, pre-processing the data and treating it with the model. Rather than implementing only the model, we can implement the data mining system altogether by integrating it with the existing system so as to ensure a new entry can be flagged if likely to be a defaulter and the lender who may be looking out to lend money

may be alerted of the prospective defaulter if his request to borrow an amount may appear on the Lender's dashboard, post which it would be at the lender's discretion of he/she would want to enter the trade.

Moreover, by integrating the system with an interactive interface we can ensure that different end user's will be able to obtain results from a data mining system with a certain layer of abstraction.

It is essential to conduct sanity data check while obtaining a new data feed.
And executing periodic regressions, tests may have to be conducted to ensure that the data mining system is indeed performing as it was intended to perform. Furthermore, over time, the performance of a model may decrease, due to which it should be monitored for a performance dip to facilitate fixes or a maybe an overhaul.
We, believe that our model is a good indicator of the likelihood of a borrower defaulting. However, it should not be used as concrete evidence to explicitly state that a borrower will be a defaulter if the model predicts him to be one.

Our plan for deployment, involves a certain degree of systems integration, with an existing system. The transition from data science/modelling to production can pose some risks if the transition to production is not gradual, i.e. if the development team become involved only towards the commencement of the development phase, there is a strong chance of an incorrect model being developed. In order to mitigate the risks associated with the transfer from data science/modelling to production, it is imperative

to involve the member of technical staff who are proficient in the architecture of the current system and the software developers slated to build and integrate the model right from the data model creation phase. This will ensure that the data model that is being built will be done with system constraints in mind, moreover the software developers will also acquire knowledge on the capabilities of the model, due to which they would be in a better position to accept ownership of the development phase. This will effectively lead to an adroit integration with minimal roadblocks.

INSIGHTS:

Our top 4 predictors are as below:

1. **‘LoanMonthsSinceOrigination’** - This feature gives the months or the duration since the loan has initiated. As the loan duration plays an important role in the interest values and the overall repayment amount, this seems like a good predictor of the default, which is also evident from out information gain run.
2. **'ProsperRating (numeric)' -** The prosper rating classifies individuals on the basis of ratings. This is a common approach to understand where a customer stands in the process, to get an initial idea before which a lender offers the loan. This looks like the most logical feature which is expected to come up as one of the top features.
3. **'ProsperScore' -** This is like above, another indicator of how reliable the lender is or how safe is it to lend to a particular borrower. Another good predictor for our analysis.

4. **'LP_NonPrincipalRecoverypayments' -** This signifies the fine which is not a part of the principal payment but is to be paid for a late payment. We can clearly note this to be one of our top predictors, which is also shown through our information gain run.

Next Steps:

- Track predictions against new defaults to tune the model
    - This shall be essential in evaluating the model against newer unseen data
    - Probably subject the model to some tweaks
- Evaluate loan defaults across other lenders for comparison analysis
    - Would stand to provide good insight and probably set a benchmark for model performance analysis.
- Adjust interest rates on new loans to reflect default predictions, tuning model for optimal ROI

REFERENCES

[1] https://en.wikipedia.org/wiki/Peer-to-peer_lending

[2] https://www.kaggle.com/jschnessl/prosperloans

[3] https://www.prosper.com/