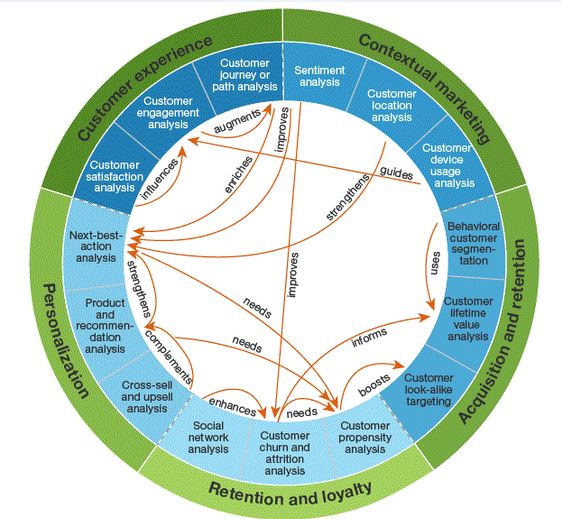
**Churn Modeling and Analysis**

Customer churn is a serious problem in the competitive world. Almost every organization face this issue of customer churn. Best way to reduce the churn is by identifying the customers who may churn and then take some action to retain the customer and keep them loyal and happy. Customer churn analysis is important part of customer analysis and we have various frame work available for the same. One of the most popular frame-work for customer analysis is given by Forrester.

**What is Forrester customer analytics framework?**

The scale and diversity of customer data provide rich new sources of insight, letting firms engage with customers in new ways and enable the digital disruption of entire industries. This will help us to understand customer and provide better understanding and insights. Forrester Research Inc. has provided with reference model called Forrester customer analytics model and Framework, which can be used as a reference for developing any solutions for customer analytics.

Shown below:



**Fig: Forrester customer analytics model**

The above model shows major domains in customer analytics, along with the dependencies between each domain. Each domain is further broken down to 15 sub domains and each sub domains places important role in customer analytics.

**1) Contextual Marketing:**

Customers are everything to any organization but there is very rare chance that customer will approach you first. In the completive world the most important part is to target best clients who will get engage with us in near future.

Contextual marketing refers to online and mobile marketing that provides targeted advertising based upon user information, such as the search terms they’re using or recent web-browsing activity (See also Computational Marketing).The goal is to present ads to customers representing products and services they are already interested in. For example, a customer performs an Internet search for cars and fuel efficiency. Afterwards, they check their daily news website and the ads which show up alongside of news includes for hybrid cars. The customer, already thinking about saving fuel on their commute, clicks on the ad to check out the latest hybrids.

**2) Acquisition and retention:**

Once we get the details of potential customer then the next task is to acquire the customer and get him engage with our products and services.

Best example will be POC done to our clients. Here we know what they require and now it is our task to acquire those clients by providing them confidence with our product and/or service.

**3) Retention and loyalty:**

Customer retention refers to the activities and actions companies and organizations take to reduce the number of customer defections. The goal of customer retention programs is to help companies retain as many customers as possible, often through customer loyalty and brand loyalty initiatives.

**4) Personalization:**

This involves generation of recommendation in which customer will be interested in, identifying the best possible action to make business from the customer along with keeping customer happy.

**5) Customer Experience:**

Customer experience is most important for the business. When customer has good experience they remain loyal and are retained with the business. We can understand customer experience by understanding customer engagement with the organization, satisfaction with the requirement fulfilled also by understanding the journey from the start of service.

**Whenever we need to work on data modeling we follow certain approach given below:**

1. **Understand the use-case (Problem)**

Understanding the use-case is very important as this helps us in taking right step in getting accurate and desired result.

Eg: Identify and predict the customer who may churn in near future.

So we are clear that we need to do some sort of prediction also we need to classify the data based on some condition to churned or non-churned class. We can conclude that the solution will be based on Machine Learning (Classifier Problem).

1. **Understand the data**

This is one and most important step while working with discovery phase. This involves understanding of meanings and value of data provided. It also includes clarifying the doubts in data if present.

1. **Discover the relations among data**

We gather, analyze relationship among the data. We can use different approach for the same.

1. Using python libraries such as seaborn, matplotlib, etc.
2. Visualization tools.

The best approach is to use visualization tools as it reduces the coding and also give better picture of data. We can track the changes in data using UI controllers and filters. This helps in understanding the story.

The data which has the highest relationship (dependency) with the problem are used as Features during the modeling phase.

**4) Featuring engineering**

Once we know the relationship among the data we work for getting the Features for modeling.Feature engineering is the process of using domain knowledge of the data to create features that make machine learning algorithms work. Feature engineering is fundamental to the application of machine learning, and is both difficult and expensive. The need for manual feature engineering can be obviated by automated feature learning.

A feature is an attribute or property shared by all of the independent units on which analysis or prediction is to be done. Any attribute could be a feature, as long as it is useful to the model. The features in your data are important to the predictive models you use and will influence the results you are going to achieve. The quality and quantity of the features will have great influence on whether the model is good or not.

General approach for Featuring engineering:

1. Brainstorming or Testing features
2. Deciding what features to create
3. Creating features
4. Checking how the features work with your model
5. Improving your features if needed
6. Go back to brainstorming/creating more features until the work is done.

**5) Modeling**

There are many models available for prediction and classification so we don’t need to make all of our own. In python we have sklearn library which provide almost all the models which we require for machine learning.

For churn modeling following models are mostly used:

(To know modeling using sklearn in depth click on names)

1. [Random Forest Classifier](http://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html)
2. [Gradient Boosting Classifier](http://scikit-learn.org/stable/modules/generated/sklearn.ensemble.GradientBoostingClassifier.html)
3. [SVC – Linear SVC](http://scikit-learn.org/stable/modules/generated/sklearn.svm.LinearSVC.html)
4. [Logistic Regression](http://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html)
5. [Naive Bayes Classifier](http://scikit-learn.org/stable/modules/naive_bayes.html)

**5.1) Random Forest Classifier**

**Random forests** or **random decision forests** are an ensemble learning method for classification, regression and other tasks, that operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees. Random decision forests correct for decision trees' habit of overfitting to their training set.

A random forest is a meta estimator that fits a number of decision tree classifiers on various sub-samples of the dataset and use averaging to improve the predictive accuracy and control over-fitting. The sub-sample size is always the same as the original input sample size but the samples are drawn with replacement if *bootstrap=True* (default).

**5.2) Gradient Boosting Classifier**

**Gradient boosting** is a machine learning technique for regression and classification problems, which produces a prediction model in the form of an ensemble of weak prediction models, typically decision trees. It builds the model in a stage-wise fashion like other boosting methods do, and it generalizes them by allowing optimization. This functional gradient view of boosting has led to the development of boosting algorithms in many areas of machine learning and statistics beyond regression and classification.

**5.3) Linear SVC**

“Support Vector Machine” (SVM) is a supervised machine learning algorithm which can be used for both classification or regression challenges. However, it is mostly used in classification problems. In this algorithm, we plot each data item as a point in n-dimensional space (where n is number of features you have) with the value of each feature being the value of a particular coordinate. Then, we perform classification by finding the hyper-plane that differentiate the two classes very well (look at the below snapshot).



**Fig: Hyper Plane differentiating the classes**

Support Vectors are simply the co-ordinates of individual observation. Support Vector Machine is a frontier which best segregates the two classes (hyper-plane/ line).

**5.4) Logistic Regression**

Logistic regression measures the relationship between the categorical dependent variable and one or more independent variables by estimating probabilities using a logistic function**.**

## Logistic Function: Logistic regression is named for the function used at the core of the method, the logistic function. The logistic function, also called the sigmoid function was developed by statisticians to describe properties of population growth in ecology, rising quickly and maxing out at the carrying capacity of the environment. It’s an S-shaped curve that can take any real-valued number and map it into a value between 0 and 1, but never exactly at those limits.

**sigmoid function = 1 / (1 + e^-value)**

**e**: It is the base of the natural logarithm exponent

**value**: It is the actual numerical value that you want to transform.

Below is a plot of the numbers between -5 and 5 transformed into the range 0 and 1 using the logistic function.



**Fig: Sigmoid function**

**5.5)Naive Bayes Classifier**

### In machine learning, *naive Bayes classifiers* are a family of simple probabilistic classifiers based on applying Bayes' theorem with strong (naive) independence assumptions between the features. Naive Bayes classifiers are highly scalable, requiring a number of parameters linear in the number of variables (features/predictors) in a learning problem. Maximum-likelihood training can be done by evaluating a closed-form expression, which takes linear time, rather than by expensive iterative approximation as used for many other types of classifiers. Gaussian naive Bayes

When dealing with continuous data, a typical assumption is that the continuous values associated with each class are distributed according to a Gaussian distribution. For example, suppose the training data contains a continuous attribute. Another common technique for handling continuous values is to use binning to discretize the feature values, to obtain a new set of Bernoulli-distributed features; some literature in fact suggests that this is necessary to apply naïve. GaussianNB implements the Gaussian Naive Bayes algorithm for classification.

**Train & test Split**

Another important part of modeling is splitting the data set into test and train dataset. We can perform this using sklearn method or by using manual splitting. In general we use sklearn’s train\_test\_split method. Now the splitting ration depends on the type of data in general we split into 70% train and 30% test or 75% train and 25% test.

**Model Evaluation Basics**

We perform model evaluation to identify best model for our purpose. For model evaluation we check value of following parameters.

**1) Accuracy:** It refers to the closeness of a measured value to a standard or known value.

The formula for **accuracy** is:   
 A(M) = (TN+TP)/(TN+TP+FP+FN)

**TP** is the number of true positives

In simple language, Predicted as the customer has churned and they really churn.

**TN** is the number of true negatives:

In simple language, Predicted as the customer will remaining with us and they do remain.

**FP** is the number of false positives:

In simple language, Predicted as they will churn, but they stay with us.

**FN** is the number of false negatives: In simple language, Predicted as the customer won’t churn but they actually churn.

We need to really take into consideration of FN in model evaluation along with others.

1. **Sensitivity/Recall** - how well the model recalls/identifies those that will leave. Also known as the true positive rate.

The formula for **Sensitivity** is: TP / (TP + FN)

**In simple, recall shows correctly predicting the churner.**

1. **Specificity -** How well the model identifies those that will stay.

The formula for **Specificity** is: TN / (TN + FP)

1. **Precision-** This help to identify how believable is the model.

The formula for **Precision** is: TP / (TP + FP)

A low precision model will tell you; those who are predicted leaving but that are actually staying.

1. **F1 score-** This is the harmonic mean between precision and recall or the balance.

The formula for **F1 score** is: 2 \* (precision \* recall)/(precision + recall)

F1 **score** (also **F**-**score** or **F**-measure) is a measure of a test's accuracy. The F1 **score** can be interpreted as a weighted average of the precision and recall, where an F1 **score** reaches its best value at 1 and worst at 0.

For better understanding you may refer this [site](https://tryolabs.com/blog/2013/03/25/why-accuracy-alone-bad-measure-classification-tasks-and-what-we-can-do-about-it/).

**6) Predict the result.**

Once we have trained model we can use the same model to predict the churn and we have required solution for the given problem.

**Let’s understand the churn modeling concept by using real life example.**

**Current Problem Statement:**

There is a US based telecom company who are face a problem with customer churn and that company has approached us (Acrotrend) to help them.

Then are seeking help to identify and predict the customer who will churn in near future and provide the list of the same.

**Data Set Provided Overview:**

For our modeling they have provided with dataset of existing users for the month of May 2017. And also they have provided us with the description of different columns so that we can understand data better.

Company keeps track of following customer activities listed below.

* State: They are state code (2 character) representing the state from US.
* Account length: This represents customer timeline.
* Area code: It is to specify the exact location of customer, just like pin-code /postcode.
* Phone number: can be used as customer identifier.
* International plan: It states if the customer as opt for international service plan.
* Voice mail plan: It states if the customer as opt for voice mail service plan.
* Number vmail messages: Total voice-mail send.
* Total day minutes: Shows total minute spent in call during the day.
* Total day calls: Total number of calls during the day.
* Total day charge: Total charge the customer has to pay for all the calls during the day.
* Total eve minutes: Shows total minute spent in call during the evening.
* Total eve calls: Total number of calls during the evening.
* Total eve charge: Total charge the customer has to pay for all the calls during the evening.
* Total night minutes: Shows total minute spent in call during the night.
* Total night calls: Total number of calls during the night.
* Total night charge: Total charge the customer has to pay for all the calls during the night.
* Total intl minutes: Shows total minute spent in international calls.
* Total intl calls: Total number of international calls.
* Total intl charge: Total charge the customer has to pay for all the international calls.
* Customer service calls: Calls made to customer service center.
* Churn: Shows whether the customer has churned or not.

Link of dataset: <https://www.kaggle.com/c/customer-churn-prediction/data>

The dataset has 5000 records and 21 columns.

**Discovering the relationship**

For the given dataset let’s find some relationship using visualization tool. In our case we are using Tableau to determine the insights form the data.

By using visualization we should be able to conclude following:

* Whether selected filed is feature or not?
* Do we need to perform some calculation of fields?
* What type of dependencies we have among different columns?
* Do we have some common pattern or policy used in our data?
* Is dataset balance?

**Few screenshot :**

Let’s check if state can be taken as feature or not. We saw that most of the churners are from few states only. As churn rate is not equally divided over states we can consider state in our feature.

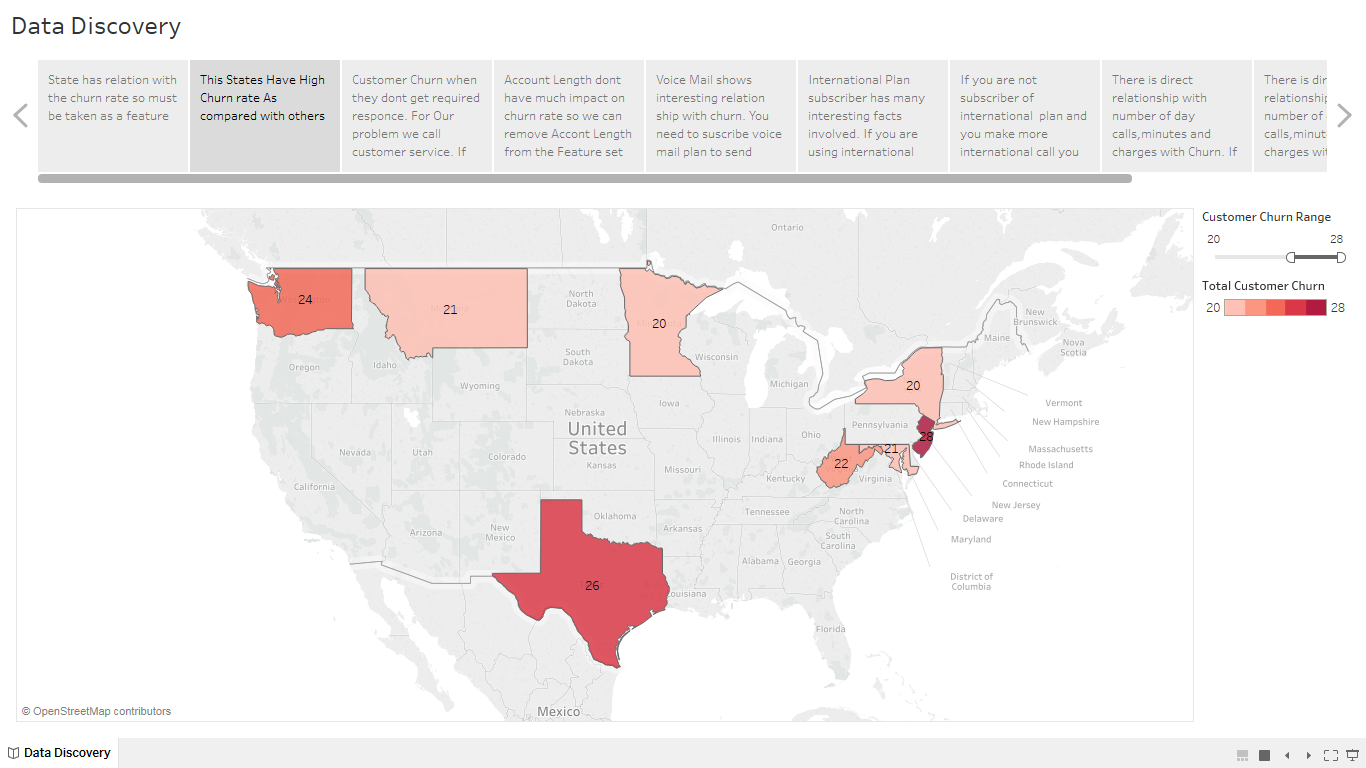


Fig: Above shown states have max number of churns

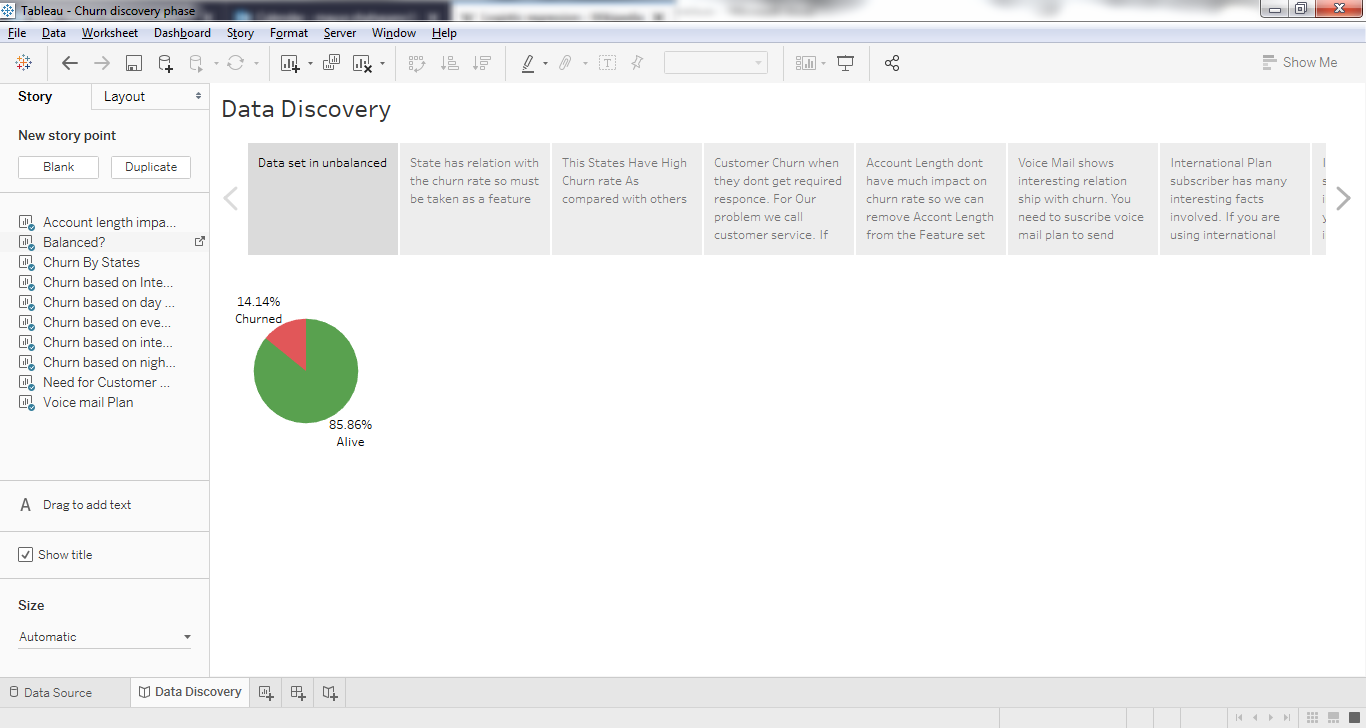


Fig: Show balancing of data set.

We can see from the pi chart, dataset is unbalance.

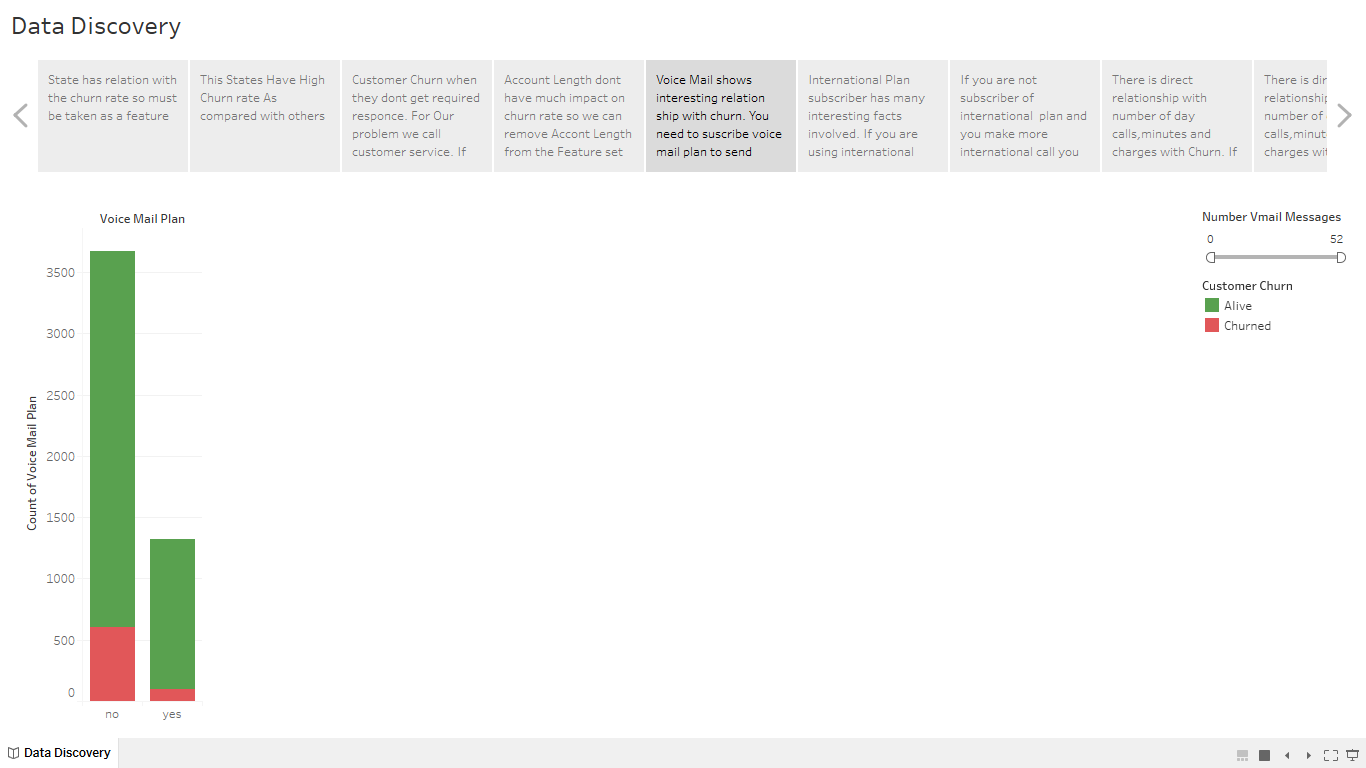
As we only have 14.14% Churn data in dataset so it become difficult to model evaluation. Eg: If model perform worst then too we will have accuracy around 86%. So we need different approach for model evaluation. Which we have seen in modeling section. 

Fig: Voice mail plan relationship with churn

We found that voice mail plan has policy applied: To send voice mail you shoud subscribe to voice mail plan. Futher we saw that churn rate is low for customer using voice mail plan. So we can consider this as feature.

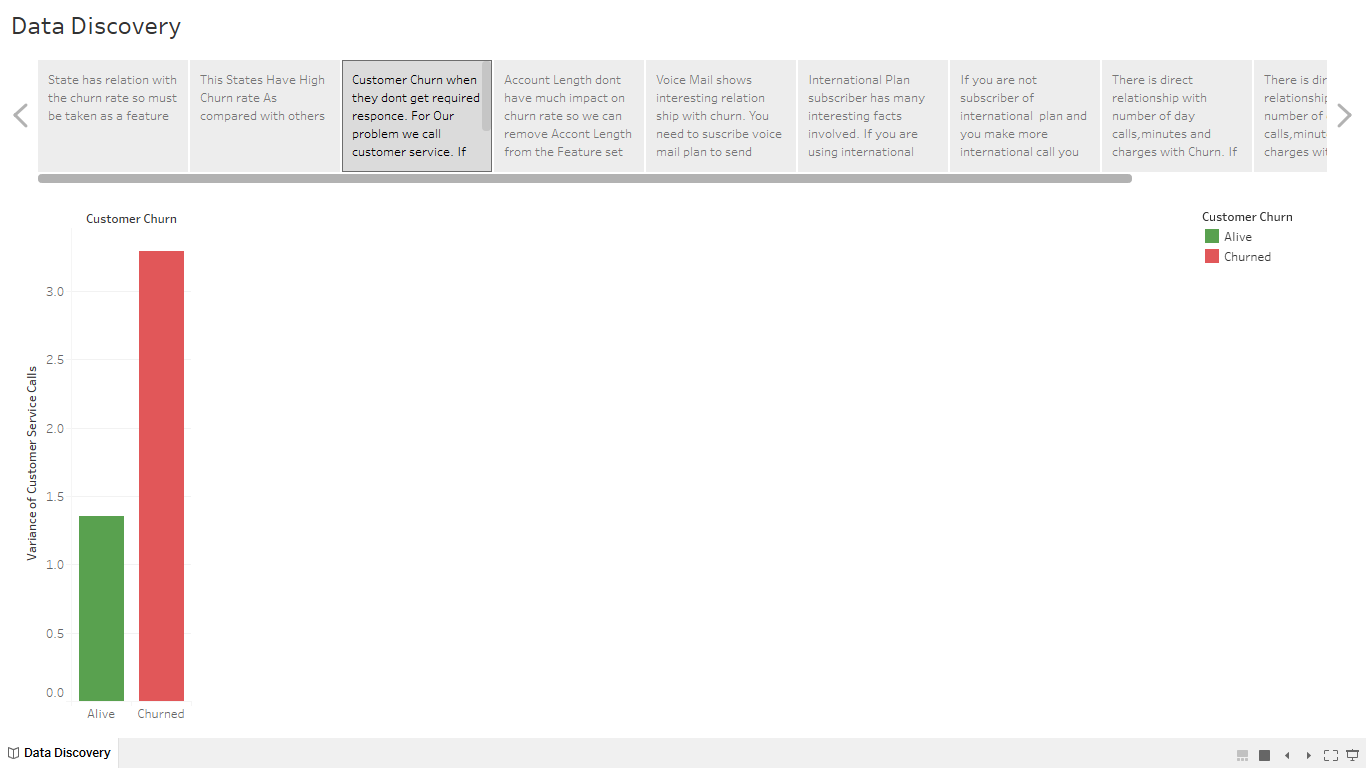


Fig: Customer Service calls role in churn

We can say that if customer is calling more then 3 times then they are more likely to churn

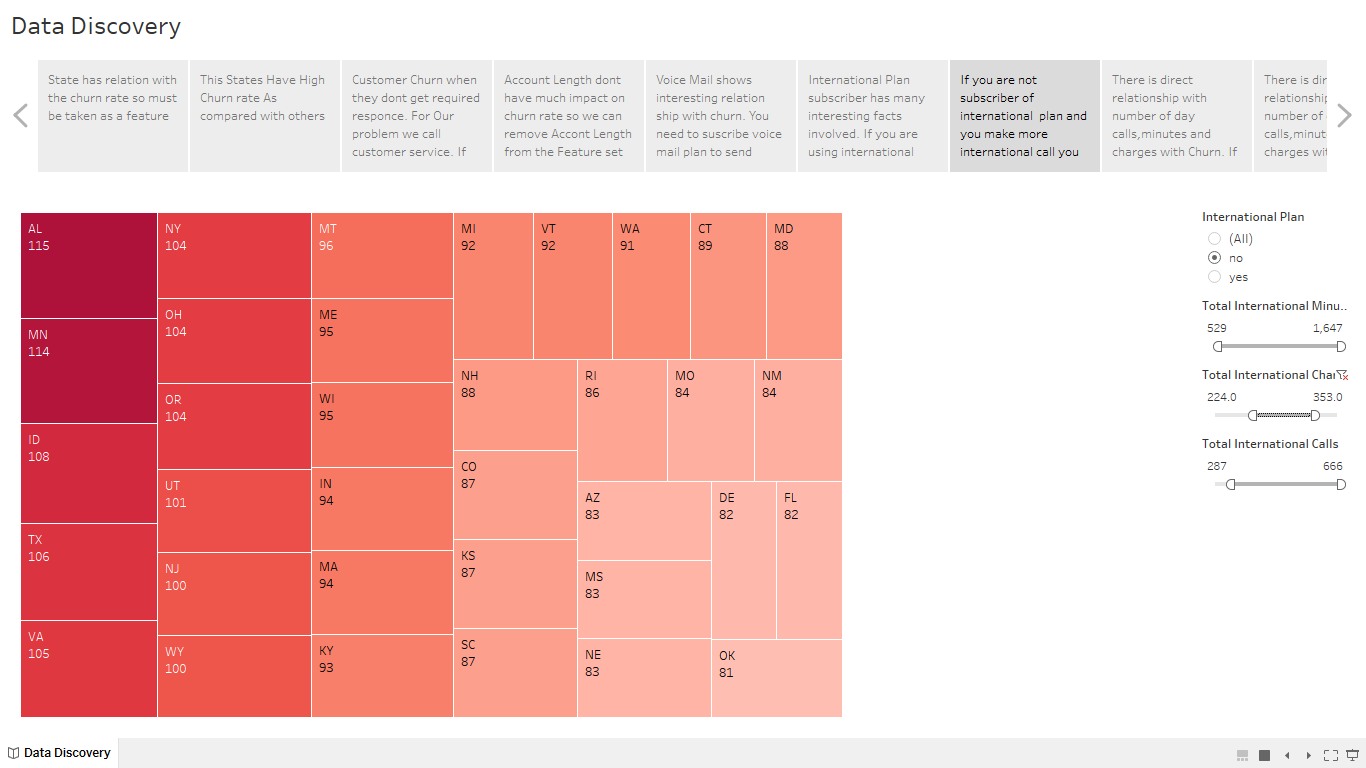


Fig: Relationship of state, international plan, international call (numbers of call,cost,minutes) with churn

Rest relationships and story board can be checked in Tableau file in repository.

**Selection of Features**

Based on above data discovery we found following relevant features which can be used for modeling:

* State
* International plan
* Voice mail plan
* Number vmail messages
* Total day minutes
* Total day calls
* Total day charge
* Total eve minutes
* Total eve calls
* Total eve charge
* Total night minutes
* Total night calls
* Total night charge
* Total intl minutes
* Total intl calls
* Total intl charge
* Customer service calls

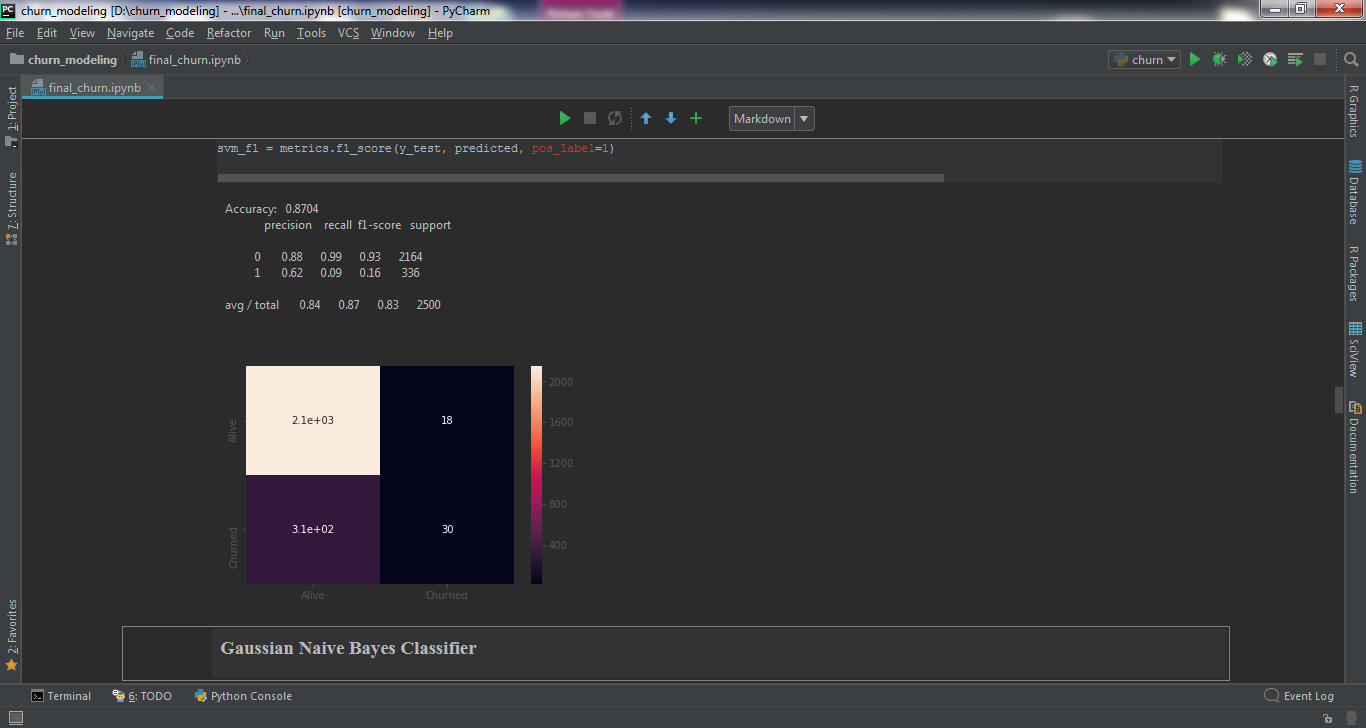
Now we need to decide on **train & test split:** Now in my case I had small dataset with very less churns so in this case I decided to go with 50% train and 50% test

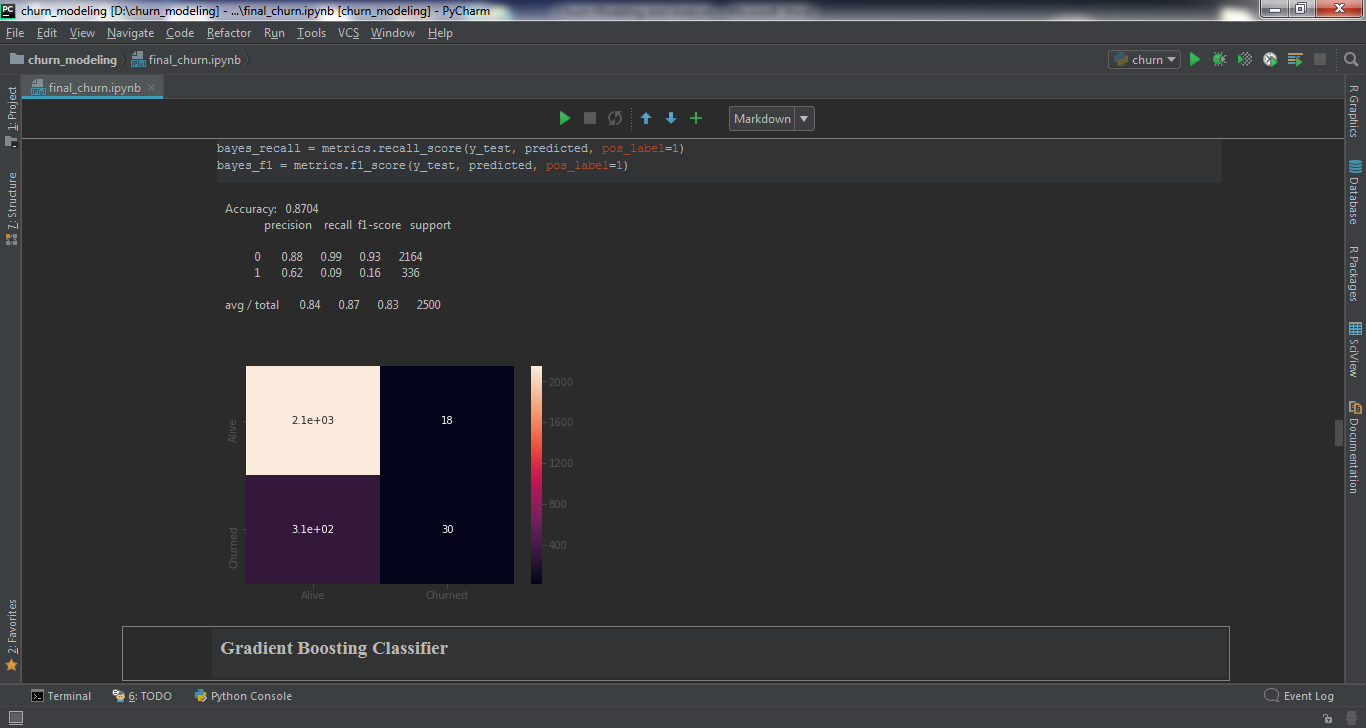
**Modeling**

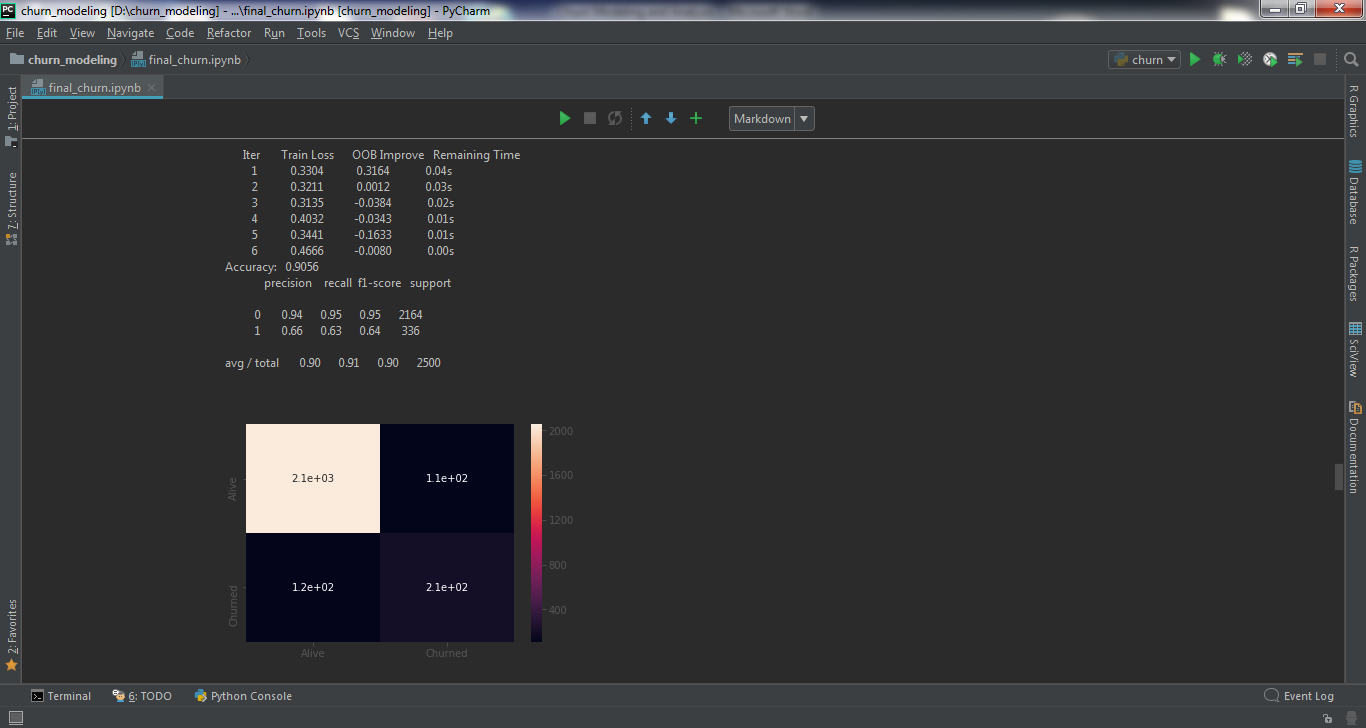
**Modeling code can be referred in churn-modeling.ipynb (Jupyter Notebook)**

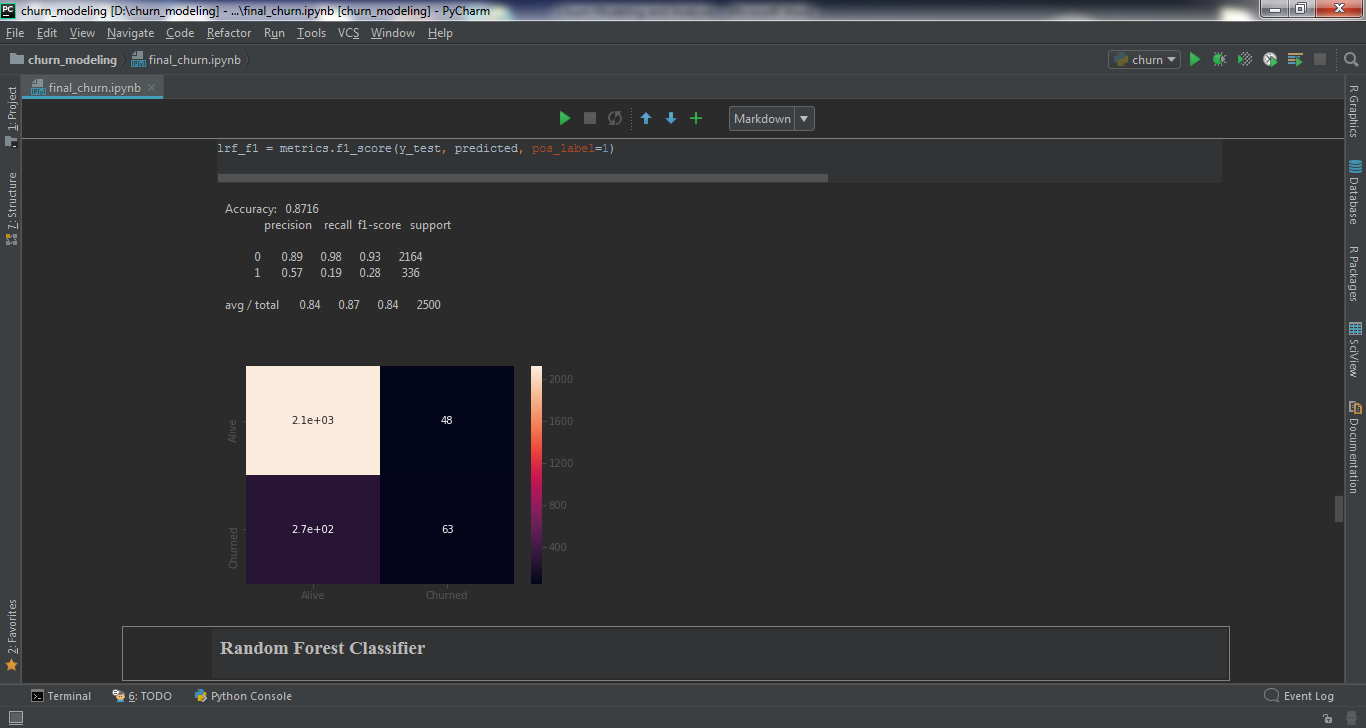
**Results of Models**

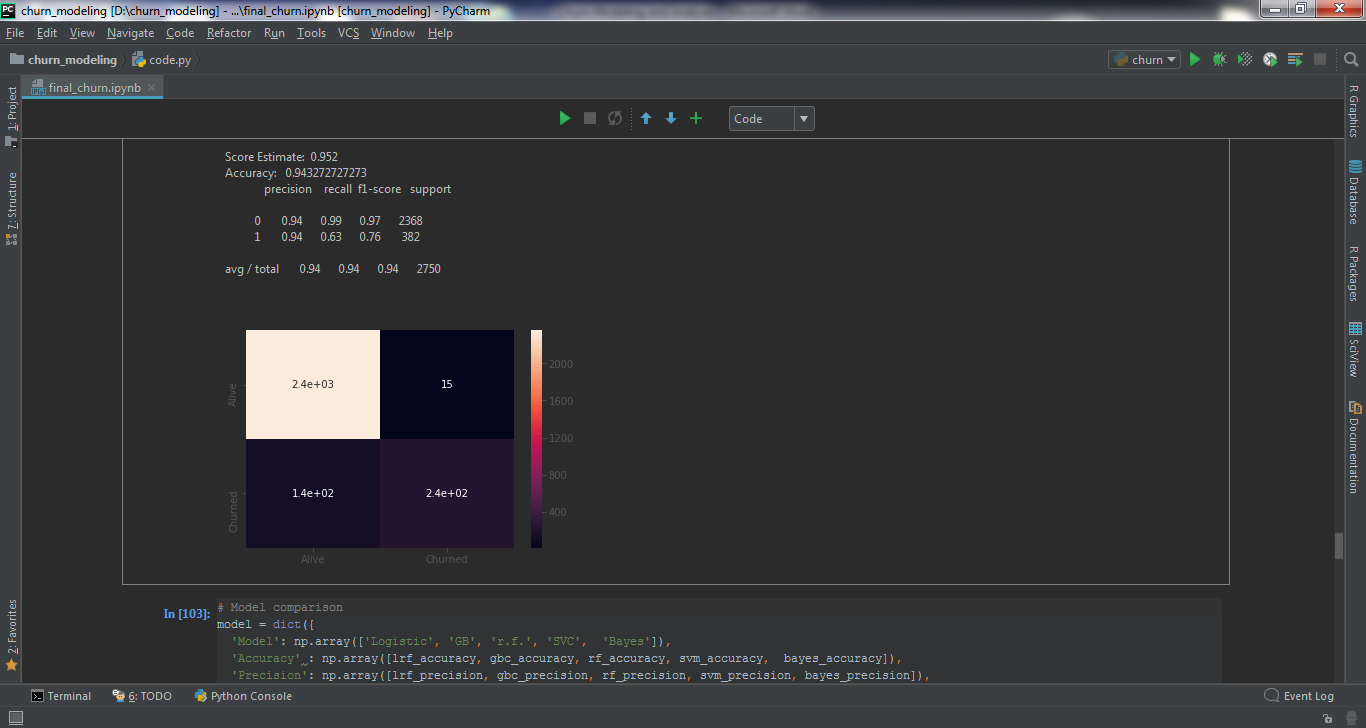
**As our data set id unbalanced so we can’t only relay on accuracy every time. So let’s check the values of precision, recall, F1-score for model evaluation.**

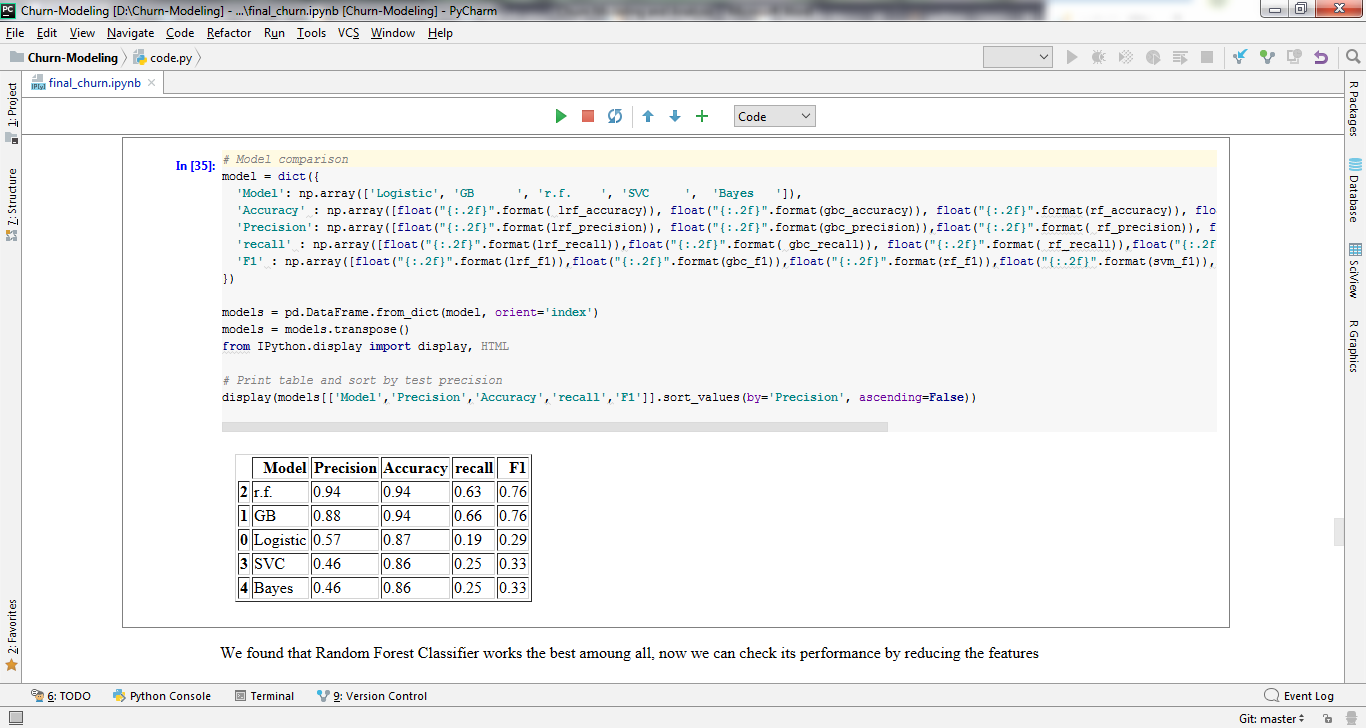
 **Linear SVC**

 **Naive Bayes Classifier**

 **Gradient Boosting Classifier**

 **Logistic Regression**

 **Random Forest Classifier**



**Models performance matrix**

**By looking at the above results we can conclude that Random Forest is best model for our dataset as Random forest has highest precision, recall, F1 score along with Accuracy.**

**Predict the result**

**B) Appendix**

**When to use which type of models: Tips and Advice.**

There is no fixed approach which can be used for churn modeling. As type of model and its parameters all depend on the Features and relationship of the data.

Mostly used models are

**Random Forest Classifier:**

This model is best in most of the cases as it takes care of model overfitting which make the model bias to trained condition. Also it can easily balance the features by assigning weight to the branches. Random forest runtimes are quite fast, and they are able to deal with unbalanced and missing data.

Generally use when:

* + 1. Have missing data
    2. Training data set size is too large.
    3. Have unbalance data set
    4. Overfitting is major concerned.

**Logistic Regression**

Logistic regression is used to predict the odds of being a case based on the values of the independent variables (predictors). The odds are defined as the probability that a particular outcome is a case divided by the probability that it is a non-case. Thismodel is best when we can see log graph like pattern in relationship among data.

**Gradient Boosting Classifier**

This is also quite efficient as random forest and takes care of Unbalanced data structure. When the complexity of relationship increased Gradient Boosting Classifier works much better than others.

**Naive Bayes Classifier**

This is a simple classifier and works when the complexity is low with simple dataset. This is suitable for small sized dataset. This model do have issue with overfitting so not much used for real time solutions but all depends on type and structure of the data.