**METHODOLOGY**

**Data Exploration**

Call Centre Data:-

**a**. First of all we splitted the full statement of the enquiry using split method in Python.

**b**. After doing data pre-processing, we made new Data Frame which consists of 19 columns those are:-

i. Account Number (1)

ii. And there are 6 columns associated with each month (6\*3=18)

"Account", "Initial\_Level\_Payment\_Settlement", "Initial\_Level\_Tech", "Operational\_Capabilities", "Payment\_Settlement", "Utilities"

**c.** Missing values – Imputed with 0.

**d**. Feature Engineering – we made 6 new columns

1. "Account” = 6\* “Account\_Month\_3” + 3\* “Account\_Month\_2” + 2\*“Account\_Month\_1”.
2. Similarly the above calculation was done for others (“Initial\_Level\_Payment\_Settlement", “Initial\_Level\_Tech", etc.).

**e.** Final DataFrame for Call – Centre consists of 7 columns:

‘Account Number’, 'Initial\_Level\_Tech', 'Account', 'Payment\_Settlement', 'Utilities', 'Operational\_Capabilities', 'Initial\_Level\_Payment\_Settlement'

Payment Data:-

a. After doing pre-processing, we made DataFrame which consists 5 columns:-

1. Account Number (1)
2. Mode\_of\_Payment(1) = Mode of (Mode\_of\_Payment\_month1, Mode\_of\_Payment\_month2, Mode\_of\_Payment\_month3).
3. For each month (3 X 1): Sum\_paid\_Month\_x

Default Data:-

a. After doing pre-processing, we made DataFrame which consists 4 columns:-

1. Account Number (1)
2. For each month (3 X 1): Default\_sum\_paid\_Month\_x

Combining Default and Payment Data (Processed Version):-

a. Missing values –

1. If for an account number Default\_Sum\_paid\_month\_x is Null, and Sum\_paid\_month\_x is not null then it is imputed with 0 or vice versa.
2. If for an account number both Default\_Sum\_paid\_month\_x and Sum\_paid\_month\_x are Null then Default\_Sum\_paid\_month\_x is imputed with average value of Default\_Sum\_paid of other months and same calculation is done for Sum\_paid\_month\_x.

b. Feature Engineering –

1. “Number of Default” = 6\*(Default\_Sum\_paid\_month\_3 != 0) +3\*(Default\_Sum\_paid\_month\_2 != 0) +2\*(Default\_Sum\_paid\_month\_1 != 0).
2. “Last\_Month\_Payment” = Sum\_paid\_month\_3
3. “Average\_of\_payment\_of\_last\_two\_month” = (Sum\_paid\_month\_3 +Sum\_paid\_month\_2)/2
4. “Average\_of\_payment\_of\_last\_three\_month” = (Sum\_paid\_month\_3 +Sum\_paid\_month\_2+ Sum\_paid\_month\_1)/3
5. “Last\_Month\_default” = Default\_sum\_paid\_month\_3
6. “Average\_of\_default\_of\_last\_two\_month” = (Default\_sum \_paid\_month\_3 + Default\_sum \_paid\_month\_2)/2
7. “Average\_of\_default\_of\_last\_three\_month” = (Default\_sum \_paid\_month\_3 + Default\_sum \_paid\_month\_2+ Sum\_paid\_month\_1)/3

Pattern Use:-

a. After doing pre-processing, we made DataFrame which consists 9 columns:-

1. Account Number (1),
2. For each month (3 X 3):

1. Used pattern\_Month\_x.

2. Usage\_Post\_Limit\_Month\_x

3. data used\_Month\_x

b. Missing Values: Null values in Used\_pattern\_month\_x, Usage Post Limit\_Month\_x, and data\_used\_month\_x was imputed with the mean of other month’s data.

c. Feature Engineering:

1. “Number of Limit\_Default” = 6\*(Usage Post Limit\_Month\_3 != 0) +3\*(Usage Post Limit\_Month\_2 != 0) +2\*(Usage Post Limit\_Month\_1!= 0).
2. “Last\_Month\_used\_pattern” = Used pattern\_Month\_3
3. “Average\_of\_used\_pattern\_of\_last\_two\_month” = (Used pattern\_Month\_3+ Used pattern\_Month\_2)/2
4. “Average\_of\_used\_pattern\_of\_last\_three\_month” = (Used pattern\_Month\_3+ Used pattern\_Month\_2+ Used pattern\_Month\_1)/3
5. Similarly for Usage\_Post\_Limit and data\_used, three features (Last\_month, Average of last two month and average of last three month) were created.

Demographic data:-

a. Feature Engineering:

1. “Time Period”: Equal to number of days between 31st March 2015 and the customer commence date.
2. “Contract\_Time\_Remaining”: Equal to number of days between Equipment warranty expiry date and 31st March 2015.

Combining data:-

 Missing Values Treatment (After combining all modified DataFrames, new missing values were generated).

1. Deleting rows in Training data in which more than 70% of the features are not known and the customer did not churn.
2. Call Centre data features: Imputed with 0.
3. Gender: One new label is created.
4. Professional Info: Replaced with its label 0(Missing).
5. Equipment Warranty, Address, Mode of payment imputed with mode.
6. For remaining features, Missing values were imputed with median after grouping down the data according to Scheme.

 For Scheme Feature: Training data had two unique labels which are not there in test data, and test data had one unique label which are not there in training data. Therefore all three of these were replaced with mode after grouping down the data according to Region.

 Feature Engineering: As the data contains too many categorical variables therefore dummy encoding of all these may create too many variables. Therefore each categorical variable were replaced with two new features:-

1. Mean of Churn (Target Variable) after grouping down the data according to categorical variable.
2. Standard deviation of Churn (Target Variable) after grouping down the data according to categorical variable.

(Technique picked up from Kaggle forums.)

 Normalization of the continuous variables.

**Business Model:**

We aimed to minimize the cost/maximize the profit for the telecom. To do this, we write an equivalently, cost matrix associated with the four scenarios.

o True Negative = people we predicted not to churn and who won’t churn. We associate no cost with this as they continue being our customers.

o False Positive = people we predict to churn and who will not churn. We associated a `admin\_cost+offer\_cost` cost per customer with this scenario as we will spend some money on getting them not to churn, but we will lose this money.

o False Negative = people we predict will not churn and we send them nothing. But they will. This will be a big loss, i.e. customer lifetime value (clv).

o True Positive = people, who we predict will churn. And they will. These are the people, we can do something with. So we make them an offer. Say a fraction of .5 accept it. So, our cost is `f \* offer\_cost + (1-f)\*(clv+admin\_cost) `.Assuming a conversion fraction of 0.5.

Assuming, Administrative cost = Rs. 1,

Offer cost = Rs. 100,

Customer lifetime value (clv) = Rs. 300 (assuming 1 years and Rs. 30 month margin per user)

Therefore, COST MATRIX = [[0,101], [300,201]]

**Model Implementation:**

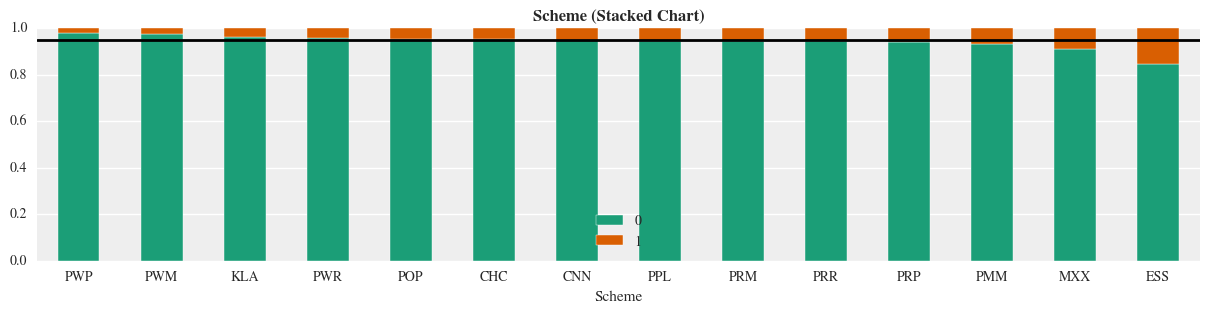
First of all we divided the training data into train set (75%) and validation set (25%).

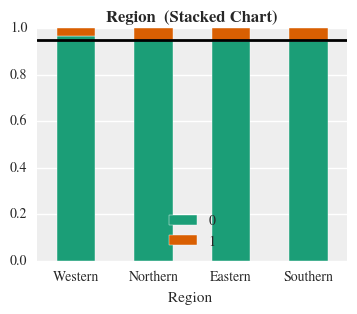
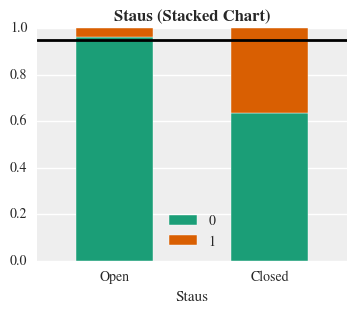
Our dataset is a very unbalanced data set with 95% of samples being negative. We know that in such a case, accuracy is not a very good metric to judge the classifier. Therefore we made a metric which ought to maximize the profit of the telecom company (We calculated the average cost per person. The Average cost can be calculated by multiplying the cost matrix by the confusion matrix elementwise and then dividing by the test set size).

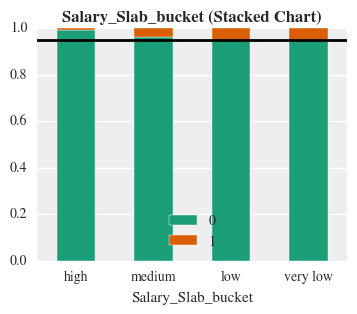
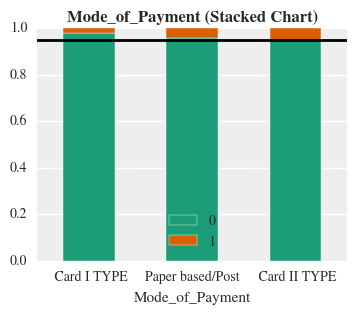
Then we tried few classifiers and calculated the average cost per person for each Algorithm (Logistic Regression, Random Forest, Adaboost, Gradient Boosting, Xgboost).

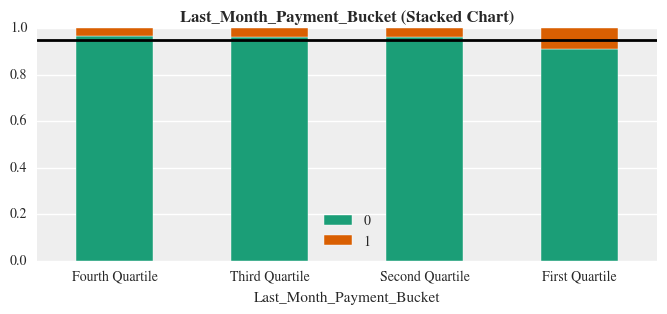
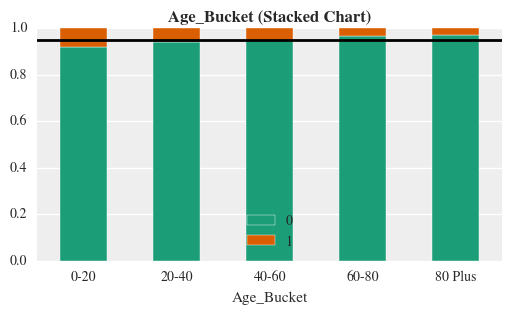
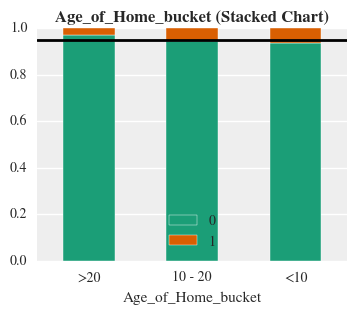
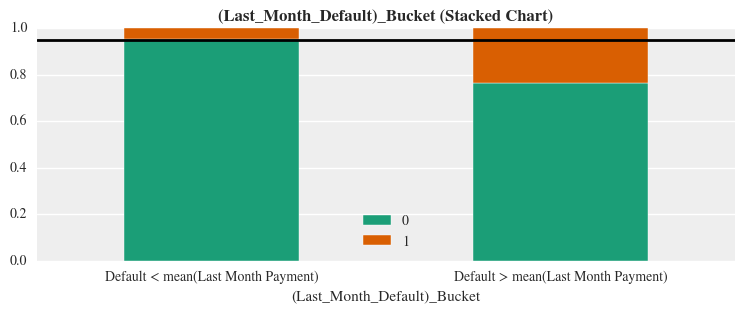
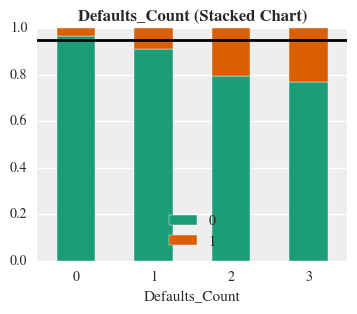
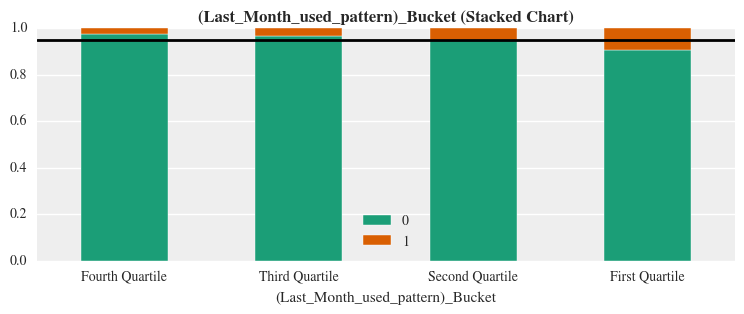
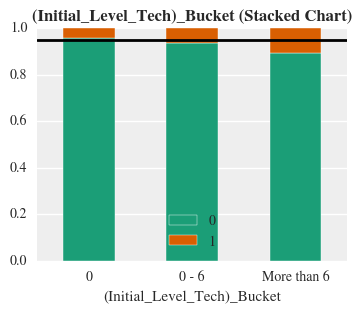
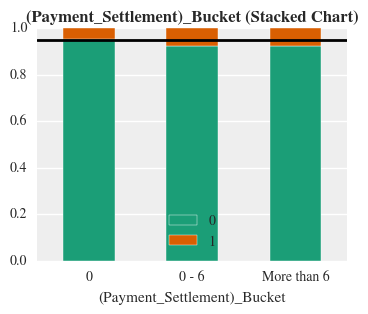
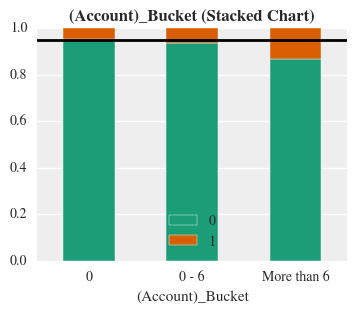
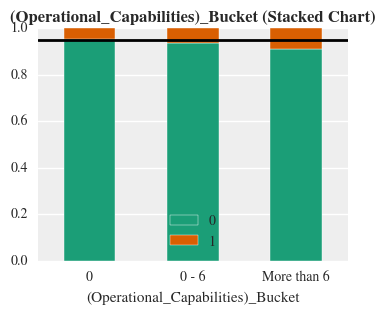
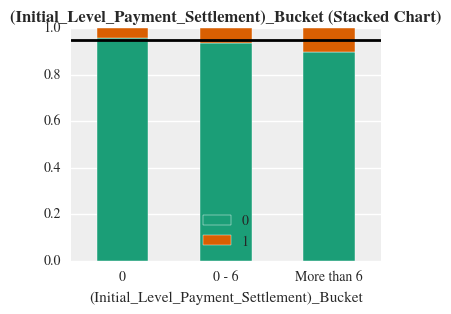
After experimenting with various algorithms, finally Ensembled model of Xgboost and Logistic was used for final modelling as it was giving the best validation score.

**Important Plots:**







\*\*\*\*\*