**Mini Project – Telecom Customer Churn Prediction Assesment**

**13th - September – 2019**

**Submitted**

By: Shilpa pandey

Contents page No

**1. Project Objectives** .................................................................................................................................

**2. Steps and approach** ..............................................................................................................................

**3. assumption** ................................................................................................................... 4

**4 explanatory Data Analysis**.................................................................................................................................. 4

**5 Data description** ................................................................................................................................ 5

**6. Univariate analysis**............................................................... 5

7. **bivariate analysis**...............................................................

**8. outlier detection** .................................................................................................................................... 5

**9. check missing value and negative values** ................................................................................................................................. 5

**10. check multicollinearity** ........................................................................................................................... 5

**11. summarize the insights** ........................................................................................................................ 6

**12 Logistic Regression**........................................................................................... 6

**13. interpret Logistic Regression** ..................................................................................................................................... 6

**14. applying KNN Model** ............................................................................................................................................... 7

**15. Interpret KNN Model**............................................................................................................................ 7

**16. Confusion matrix interpretation for all models**...................................................................................................... 8

**17. Interpret Naive Bayes Model**......................................................................................................................................... 9

**18. Change of structure of variables from numeric to factors.** .......................................................... 9

**19. Interpretation of other Model Performance Measures for logistic <KS, AUC, GINI>**....................................................... 10

**18. compare models and interpret result. .......................................................... 9**

**19. Model validation exercise ………………………………………….. 10**

**20. Actionable Insights and Recommendations**

**21. Code for reference**

**1. Project Objectives**

Customer Churn is a burning problem for Telecom companies. In this project, we simulate one such case of customer churn where we work on a data of postpaid customers with a contract. The data has information about the customer usage behavior, contract details and the payment details. The data also indicates which were the customers who canceled their service. Based on this past data, we need to build a model which can predict whether a customer will cancel their service in the future or not.

Exploratory data analysis on the dataset.

**2. Steps and approach**

We shall follow step by step approach to arrive to the conclusion as follows:

* Basic data summary, Univariate, Bivariate analysis, graphs
* Check for Outliers and missing values and check the summary of the dataset
* Check for Multicollinearity - Plot the graph based on Multicollinearity & treat it.
* Summarize the insights you get from EDA
* Applying and Interpret Logistic Regression
* Applying and Interpret KNN Model
* Applying and Interpret Naive Bayes Model
* Confusion matrix interpretation for all models
* Interpretation of other Model Performance Measures for logistic <KS, AUC, GINI>
* Remarks on Model validation exercise <Which model performed the best>
* Actionable Insights and Recommendations

**3 Assumptions**

* none

**4 Exploratory Data Analysis – Step by Step approach**

* 1. Environment Set Up and Data Import
     1. Install Necessary Packages and Invoke Libraries

* + 1. Set up working Directory

* + 1. Import and read the dataset
  1. Variable Identification
     1. Variable identification – inferences

|  |  |
| --- | --- |
| **5. Data Description:** |  |

Str function indicates all the var are numerica nad integer but we will change into factor as below:

Machine generated alternative text:
• data. frame': 
churn 
3333 obs. 
. int 
. int 
int 
. int 
num 
. int 
num 
: int 
num 
num 
num 
of 
128 
2.7 
265 
110 
11 
107 
3.7 
162 
123 
variables : 
Accountweeks 
contractRenewa1 : 
Datapl an 
Datausage 
custservcal s 
Dayw i ns 
Daycal s 
S Monthlycharge 
over ageFee 
Roan* i ns 
137 
243 
114 
113 98 88 79 97 84 . 
57 87.3 36 63.9 93.2 
3.1 7.42 . 
10 13.7 12.2 6.6 10.1 6.3 7.5 7.1 
84 75 
118 121 147 117 
101 
o.. 
100 
oo 2.03 0 0.19 3.02 
301 
299 167 
71 
41 
89 82 52 57 
9.87 9.78 6.06 
141 . 
8.7 11.2 . 

|  |  |  |
| --- | --- | --- |
| **Variables** | Variable info | Data type |
| Churn | 1 if customer cancelled service, 0 if not | Categorical with value 1 / 0 |
| AccountWeeks | number of weeks customer has had active account | integer |
| ContractRenewal | 1 if customer recently renewed contract, 0 if not | Categorical with value 1 / 0 |
| DataPlan | 1 if customer has data plan, 0 if not | Categorical with value 1 / 0 |
| DataUsage | gigabytes of monthly data usage | numeric |
| CustServCalls | number of calls into customer service | integer |
| DayMins | average daytime minutes per month | numeric |
| DayCalls | average number of daytime calls | integer |
| MonthlyCharge | average monthly bill | numeric |
| OverageFee | largest overage fee in last 12 months | numeric |
| RoamMins | average number of roaming minutes | numeric |

Machine generated alternative text:
> str (phone) 
'data. frame': 
2.7 3.7 2.03 0 0.19 3.02 
churn 
Accountweeks 
contractRenewa1: Factor w/ 2 levels D 
3333 obs. of 11 variables: 
. Factor w/ 2 levels D 
. int 128 107 137 84 75 
118 121 147 117 141 
" 2221112121 
Datapl an 
Datausage 
custservcal s 
Dayw i ns 
Daycal s 
S Monthlycharge 
over ageFee 
Roan" i ns 
. Factor w/ 2 levels "D 
num 
: int 
num 
: int 
num 
num 
num 
110230301 
265 162 243 299 167 
110 123 114 71 
89 82 52 57 41 
9.87 9.78 6.06 
113 98 88 79 97 84 . 
57 87.3 36 63.9 93.2 
3.1 7.42 . 
10 13.7 12.2 6.6 10.1 6.3 7.5 7.1 
8.7 11.2 . 

Dimension shows it has 3333 rows and 11 columns

Machine generated alternative text:
> di m(phone) 
[1] 3333 11 

Variance: of the overall data

Machine generated alternative text:
> var (phone) 
churn 
Accountweeks 
contr act Renewal 
Datapl an 
Datausage 
custservcal 1 s 
DayM i ns 
Daycal 1 s 
Monthlycharge 
over ageFee 
Roan" i ns 
churn 
o. 12395147 
o. 23190230 
-o. 02706861 
-o. 01608978 
-o. 03906880 
o. 09668080 
3. 93401198 
o. 13042742 
0.41818952 
o. 08285751 
o. 06707297 
Accountweeks 
o. 23190230 
1585. 80012059 
-o. 29143683 
o. 05199533 
o. 72932817 
-o. 19885263 
13. 48258672 
30. 74486824 
8. 22925730 
-o. 68154305 
1. 05772646 
contr act Renewal 
-o. 0270686132 
-o. 2914 368268 
o. 0875444687 
-o. 0007950975 
-o. 0072 385021 
o. 0095445919 
-o. 7960514719 
-o. 0222950566 
-o. 2298418827 
-o. 014 3335318 
-o. 0378913942 
Datapl an 
-o. 0160897842 
o. 0519953256 
-o. 0007950975 
o. 2001648544 
o. 5386315651 
-o. 0104902447 
-o. 0410383866 
-o. 0995388254 
5. 4197905175 
o. 0244201461 
-o. 0016461040 
Dat aus age 
-o. 039068804 
o. 729328174 
-o. 007238502 
o. 538631565 
1. 619683937 
-o. 036367479 
o. 220153480 
-o. 203361701 
16. 340521800 
o. 063372145 
o. 578248860 
custservcal Is 
o. 096680797 
-o. 198852628 
o. 009544592 
-o. 010490245 
-o. 036367479 
1. 730516689 
-o. 961789594 
-o. 500080230 
-o. 605396497 
-o. 043244828 
-o. 035403072 
DayMi ns 
3. 93401198 
13.48258672 
-o. 79605147 
-o. 04103839 
o. 22015348 
-o. 96178959 
2966. 69648652 
7. 37894910 
508.15128101 
o. 97207308 
-1. 54414905 
Daycal Is 
o. 13042742 
30. 74486824 
-o. 02229506 
-o. 09953883 
-o. 20336170 
-o. 50008023 
7. 37894910 
402. 76814092 
-2. 62511797 
-1. 09150684 
1. 20826816 
Monthlycharge 
o. 4181895 
8. 2292573 
-o. 2298419 
5.4197905 
16. 3405218 
-o. 6053965 
508.1512810 
2. 62 51180 
269. 814 5172 
11. 7360304 
5. 3853237 
over ageFee 
o. 08285751 
-o. 68154305 
-o. 014 33353 
o. 02442015 
o. 06337214 
-o. 04324483 
o. 97207308 
-1. 09150684 
11. 73603043 
6.42983488 
-o. 07803751 
Roanw i ns 
o. 067072968 
1. 057726457 
-o. 037891394 
-o. 001646104 
o. 578248860 
-o. 035403072 
-1. 544149046 
1. 208268159 
5. 385323694 
-o. 078037 508 
7. 794368064 

Summary : of the data

Machine generated alternative text:
> summary(phone) 
1.0 
0.0 
74.0 
.:o. 
87.0 
:10. 30 
:10.24 
3rd Qu. :2.000 
:11. 77 
churn 
Accountweeks 
cont r act Renewal 
Datap 1 an 
custservcal Is 
DayMi ns 
Monthlycharge 
overageFee 
Min. 
1st Qu 
Medi an 
Mean 
3rd Qu 
0000 
. 0000 
0000 
: 0.1449 
. 0000 
0000 
Min. 
1st Qu. : 
Medi an 
: 101. 
: 101. 
Mean 
3rd Qu 
. .•127. 
.•243. 
Max. 
:o. 
O 
1 
O 
O 
Min. 
1st Qu. 
Medi an 
Mean 
3rd Qu. 
Max. 
0000 
0000 
0000 
9031 
0000 
0000 
Min. 
1st Qu 
Medi an 
Mean 
3rd Qu. 
Max. 
.•o. 
0000 
0000 
.•o. 
0000 
.•o. 2766 
0000 
0000 
Datausage 
Min. 
:o. 0000 
1st Qu. 
:o. 0000 
Medi an 
:o. 0000 
:o. 8165 
Mean 
3rd Qu 
. 7800 
: 5.4000 
Max. 
Min. 
1st Qu 
Medi an 
Mean 
Max. 
000 
000 
000 
563 
000 
Min. 
1st Qu 
Medi an 
Mean 
3rd Qu 
Max. 
o. 
.:143.7 
:179. 
:179. 
. :216. 
: 350. 
O 
4 
8 
4 
8 
Daycal Is 
Min. 
1st Qu. : 
Medi an 
Mean 
3rd Qu. 
Max. 
:101. 
:100.4 
:114. 
:165. 
O 
O 
O 
Min. 
1st Qu • 
Median • 
Mean 
3rd Qu. • 
Max. 
. 14 
45. 
. 53. 
. 56. 
. 66. 
:111. 30 
. 00 
00 
50 
31 
20 
Min. 
1st Qu. • 
Medi an 
Mean 
3rd Qu. 
Max. 
o 
8. 
:10. 
:10. 
: 18. 
. 00 
33 
07 
05 
19 
Roan"i ns 
Min. 
1st Qu • 
8.50 
Medi an 
Mean 
3rd Qu 
. :12.10 
Max. 

**6. Univariate analysis**

Histogram for numerical variables and box plot for categorical variables

Histogram: for numeric variables

Machine generated alternative text:
OverageFee 
RoamMins 
DayMins 
DayCalls 
DataUsaae 
AccountWeeks 
MonthlyCharge 
CustServCalls 

Boxplot for categorical variables:

Machine generated alternative text:
Chi Irn 
ContractRenewal 
ataPIan 

**7. Bivariate Analysis:**

**Can do box plot for categorical variable vs numeric variable:**

Machine generated alternative text:
Churn 
AccountWeeks 
ContractRenewal 
DataPlan 
DataUsage 
CustSevCalls 
8 
DayMins 
Daycalls 
MonthlyCharge 
OverageFee 
RoamMins 

Analysis using qplot

Inferences:

1. If number of calls into customer service is high then customer will cancel the service.
2. If average daytime minutes per month is high then customer will cancel the service.

Machine generated alternative text:
Churn 
0 75 
025 
DataUsage 
Churn 
150 
075 
0 100 
DataUsage 
Churn 
075 
CustServCalls 
DataUsaqe 
DataUsage 
DavMins 
Churn 
075 
Churn 
075 
Churn 
0 75 
150 
0 100 
DayMins 
DayMins 
DayMins 
Churn 
075 
Churn 
075 
Churn 
0 75 

If average daytime minutes per month is high the customer will cancel the service.

Bivariate analysis for numeric variables can be done using correlation plot

Machine generated alternative text:
> cor (scatter) 
Accountweeks 
Datausage 
custservcal 1 s 
DayM i ns 
Daycal 1 s 
Monthlycharge 
over ageFee 
Roan" i ns 
Accountweeks 
1. 000000000 
o. 014 390757 
-o. 003795939 
o. 006216021 
o. 038469882 
o. 012 580670 
-o. 006749462 
o. 009513902 
Datausage 
o. 014390757 
1. 000000000 
-o. 021722518 
o. 003175951 
-o. 007962079 
o. 781660429 
o. 019637372 
0.162745576 
custservcal 1 s 
-o. 003795939 
-o. 021722518 
1. 000000000 
-o. 013423186 
-o. 018941930 
-o. 028016853 
-o. 012964219 
-o. 009639680 
DayM i ns 
o. 006216021 
o. 003175951 
-o. 013423186 
1. 000000000 
o. 006750414 
o. 567967924 
o. 007038214 
-o. 010154586 
Daycal 1 s 
o. 038469882 
-o. 007962079 
-o. 018941930 
o. 0067 50414 
1. 000000000 
-o. 007963218 
-o. 021448602 
o. 021564794 
Monthl ycharge 
o. 012580670 
o. 781660429 
-o. 028016853 
o. 567967924 
-o. 007963218 
1. 000000000 
o. 281766048 
o. 117432607 
over ageFee 
-o. 006749462 
o. 019637372 
-o. 012964219 
o. 007038214 
-o. 021448602 
o. 281766048 
1. 000000000 
-o. 011023336 
Roan" i ns 
o. 009513902 
o. 162745576 
-o. 009639680 
-o. 010154586 
o. 021564794 
o. 1174 32607 
-o. 011023336 
1. 000000000 

Machine generated alternative text:
AccountWeeks 
DataUsage 
CustServCalls 
DayMins 
Daycalls 
MonthlyCharge 
OverageFee 
RoamMins 
0.8 
0.6 
0.2 
-0.2 
-0.6 
-0.8 

As per graph :

**1. "MonthlyCharge" and "DayMins" seems to be correlated**

**2."MonthlyCharge" and "Datausage" seems to be highly correlated**

**8. outlier detection**

Checking for the outliers in dataset, the below result shows no outlier

Machine generated alternative text:
> outlier 
NULL 

We can check the bar plot as well for outliers , also to find correlations between numerical and categorical variable. As per the graph below

outliers value is seen in “AccountWeeks” column and “DayMins” column, but models doesn’t get affected much with it.

Machine generated alternative text:
Churn 
AccountWeeks 
ContractRenewal 
DataPlan 
DataUsage 
CustSevCalls 
8 
DayMins 
Daycalls 
MonthlyCharge 
OverageFee 
RoamMins 

**9. Missing or negative values**

There are no missing values, negative values and outlier in the dataset

Checking for the missing values in dataset

Machine generated alternative text:
> co] sums (is. na(phone)) 
churn 
Accountweeks 
cont r act Renewal 
Datap Ian 
Dat aus age 
custservcal 1 s 
Day" i ns 
Daycal Is 
Monthlycharge 
over ag eFee 
Roan"i ns 

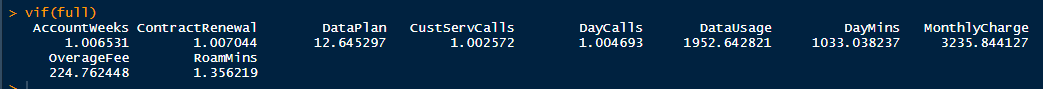
Checking for the Negative values in dataset

We have some negative values in the Experience column which can’t be possible so we have to identify those and clean the dataset.

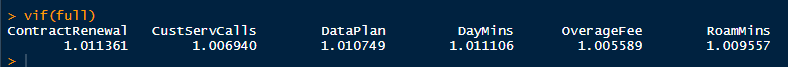
Machine generated alternative text:
phone<O 
> neg 
> sum(neg) 

1. **Check for multicollinearity & treat it**

variable dataplan, datausage, daymins, monthlycharge and overagefee are very collinear as per graph (full model)

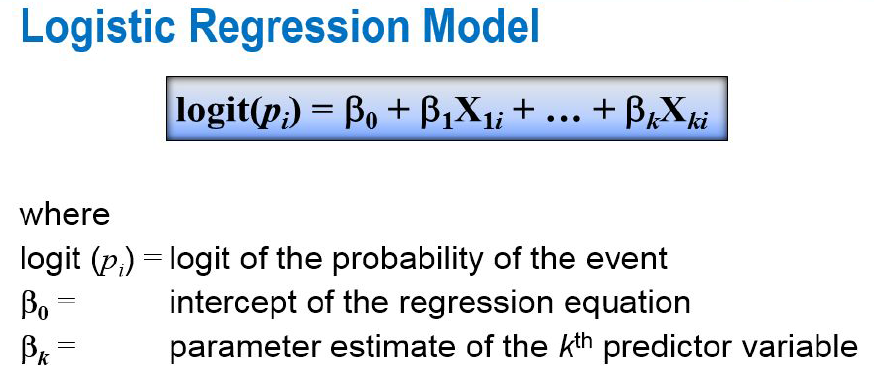


Removing insignificant variables like datausuage, monthlycharge and accountsweek, none of the variables are multicollinear



1. **Summarize the insights you get from EDA**
2. If average daytime minutes per month is high the customer will cancel the service.
3. If number of calls into customer service is high then customer will cancel the service
4. Also, collinearity is less.
5. As per linear regression customerservcalls, daycalls, roammins seems to be significant varables
6. Data need to be scaled for more clarity.
7. Churn is the target variable.

* **12. Logistic Regression**



What logistic regression predicts

The variate or value produced by logistic regression is a probability value

between 0.0 and 1.0.

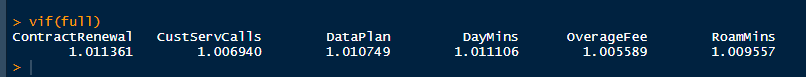
Machine generated alternative text:
CoeTT1 c. ents : 
Estimate std. 
' 0.001 
(Intercept) 
cont r act Renewal 
signif. codes: 
-1. 7083 
Error 
o. 1378 
0.1538 
z value 
-2.499 
-11.104 
Pr (>lzl) 
o. 0124 
<2e-16 
' 0.05 ' 
(Di spersion parameter for 
Null devi ance: 1906. O 
Resi dual deviance: 1795.6 
AIC: 1799.6 
binomial family taken to be 1) 
on 2334 degrees of freedom 
on 2333 degrees of freedom 
Number of Fisher scoring iterations: 4 

**Contract renewal seems to be significant**

**Full model is not significant, find the significant variables in model: by adding variables and checking significant variables**

Machine generated alternative text:
glm(formula = Logitl, family 
-14.252 
' 0.001 
bi nomial , 
data 
train. logit) 
Devi ance Residuals: 
Min 
IQ Medi an 
-1. 6033 
-o. 5686 
-0.4056 
-o. 2611 
Error 
o. 487964 
o. 162459 
o. 023762 
o. 001239 
o. 026070 
o. 044203 
2. 9130 
coeffi ci ents : 
(Intercept) 
Datapl an 
Roanw i ns 
DayM i ns 
over ageFee 
custservcal 1 s 
signif. codes: 
Estimate std. 
-6. 954298 
-o. 812977 
o. 070416 
o. 012709 
o. 140580 
o. 404198 
z value 
5. 004 
2. 963 
10. 257 
5. 392 
9.144 
Pr Izl) 
< 2e-16 
5.61e-07 
o. 00304 
< 2e-16 
6.95e-08 
< 2e-16 
' 0.05 ' 
(Di spersion parameter for 
Null devi ance: 1906.0 
Resi dual deviance: 1654.8 
AIC: 1666.8 
bi nomi al 
on 2334 
on 2329 
family taken to be 1) 
degrees of freedom 
degrees of freedom 
Number of Fisher scoring iterations: 5 

No collinearity between significant data:



* Confusion matrix:
* obs  
  pred 0 1  
   0 841 136  
   1 5 16

* Accuracy:
* Machine generated alternative text:
  > accuracy = 
  > accuracy 
  [1] o. 8587174 

Machine generated alternative text:
10 0.8 0.6 0.4 0.2 oo 
Specificity 

* After chaninging threshold to 0.25 , no changes accuracy doesnot matter much as much as loss

Machine generated alternative text:
> tab. logit 
obs 
pred 0 1 
0 841 136 
1 5 16 
attr(, "class" 
[1] "confusion. matrix" 
> response_imbal ance = 
> response_imbal ance 
[1] 0.8476954 
(841+5) / (841+136+5+16) 

* **12. Interpretation of Logistic Regression**
* **1. Variables dataplan, roammins, daymins, overagefee, custservcalls are significant**
* **2. The accuracy of model is 84.7 per which is not very good**

* **13. KNN model**
* **What is kNN Algorithm?**
* Let’s assume we have several groups of labeled samples. The items present in the groups are homogeneous in nature. Now, suppose we have an unlabeled example which needs to be classified into one of the several labeled groups. How do you do that? Unhesitatingly, using kNN Algorithm.
* k nearest neighbors is a simple algorithm that stores all available cases and classifies new cases by a majority vote of its k neighbors. This algorithms segregates unlabeled data points into well defined groups.
* **Pros:** The algorithm is highly unbiased in nature and makes no prior assumption of the underlying data. Being simple and effective in nature, it is easy to implement and has gained good popularity.
* **Cons:** Indeed it is simple but kNN algorithm has drawn a lot of flake for being extremely simple! If we take a deeper look, this doesn’t create a model since there’s no abstraction process involved. Yes, the training process is really fast as the data is stored verbatim (hence lazy learner) but the prediction time is pretty high with useful insights missing at times. Therefore, building this algorithm requires time to be invested in data preparation (especially treating the missing data and categorical features) to obtain a robust model.

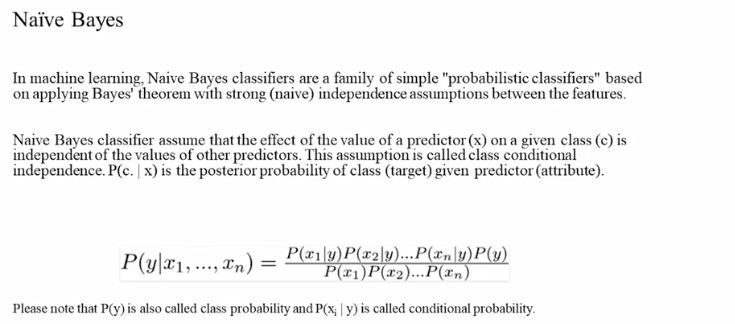
**Output of knn model for K=3**

Machine generated alternative text:
> tab. knn. 3 
y_pred. 3 
0 814 32 
1 70 82 
> accuracy. knn. ag(tab. knn. 3))/sum(tab. knn. 3) 
> accuracy. knn. 3 
[1] o. 8977956 
> loss. knn. 3<-tab. knn. 3C2 knn. knn. 3 CI ,1]) 
> loss. knn.3 
[1] o. 07918552 

* **Output of knn model for** K-5
* Machine generated alternative text:
  > accuracy. Knn. ag(taD. 
  > accuracy. knn. 5 
  [1] o. 9008016 
  > loss. knn. 5<-tab. knn. 
  > loss. knn.5 
  [1] o. 08820287 
  b) ) / sum (t 
  5 C 2 , knn. , 1] *tab. knn. 5 CI , 1]) 
* **Output of knn model for k=7**
* Machine generated alternative text:
  > accuracy. knn. 7< 
  > accuracy. knn. 7 
  [1] o. 8977956 
  > loss. knn.7 
  [1] o. 09388646 
  -sum 
  (di ag (tab. knn. 7))/sum(t 
  > loss. knn. 7<-tab. knn. 7 C 2 knn. knn. 7 CI ,1]) 
* **Output of knn model for k=9**
* Machine generated alternative text:
  > accuracy. knn. 9 
  [1] o. 8977956 
  > loss. knn. 9<-tab. 
  > loss. knn.9 
  [1] o. 0974026 
  1. **Interpretation of KNN model** 
     1. **k=3 is best as ( loss function is also very imp along with accuracy and we see there is not much difference in accuracy when k=5 but loss is increased from 7.9 to 8.8 percent.**

**Also for higher k values the accuracy is dropping and loss is increasing hence we will chose k =3**

* + 1. **Normalized data is must for knn model as distance is the measure for this model**
    2. **The accuracy given is 89.7 per and loss is 7.9 percent**
  1. **Analysis of Naive Bayes**



Naïve based: for categorical x, but continouus variable has been accomodated into naïve based.

Machine generated alternative text:
> tab. NB 
y_pred. NB 
0 780 66 
1 77 75 
> accuracy. ag(tab. NS)) /sum(tab. NS) 
> accuracy. NB 
[1] 0.8567134 
> loss. NB<-tab. NBC2 
> loss. NB 
[1] o. 08984831 
> opp. loss. NB<-tab. NBCI , 
> opp. loss. NB 
[1] 0.4680851 

* Machine generated alternative text:
  [1] 0.4680851 
  > tot. loss. NB<-O. 95* loss. NB+O. 05"opp. loss. NB 
  > tot. loss. NB 
  [1] 0.1087601 

[1] 0.8977956  
> loss.knn.3<-tab.knn.3[2,1]/(tab.knn.3[2,1]+tab.knn.3[1,1])  
> loss.knn.3  
[1] 0.07918552  
> opp.loss.knn.3<-tab.knn.3[1,2]/(tab.knn.3[1,2]+tab.knn.3[2,2])  
> opp.loss.knn.3  
[1] 0.2807018  
> tot.loss.knn.3<-0.95\*loss.knn.3+0.05\*opp.loss.knn.3  
> tot.loss.knn.3  
[1] 0.08926133

* 1. **Interpretation of Naive Bayes**

is it applicable here? comment and if it is not applicable, how can you build an NB model in this case?)

* + 1. **It is initially based on categorical variable, when both x and y are categorical but here we have continuous variable**

**Hence we have a version of Naïve bayes algorithm where it uses some methods where x need not to be discrete.**

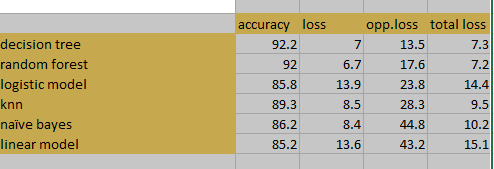
**There are two primary ways to incorporate continuous features into the Naive Bayes model:**

1. **Discretization**: One can transform continuous features into discrete features by categorizing different values into discrete buckets. For instance, a continuous feature can be binarized by treating all values that exceed a threshold as "Large" and all values the don't "Small". Over course, more fine grained discretization that categorizes values into any arbitrary number of buckets is possible as well.
2. **Distribution modeling**: To incorporate continuous features into the Naive Bayes model without discretization, one can replace the conditional probability of that feature given each class label in the Naive Bayes
   * 1. **It also performs the analysis on normalized data.**
     2. **The accuracy measured by this model is 89.7 percent and the loss measured is 10.8 percent.**
   1. **Confusion matrix interpretation for all models**

Checking with other models as well: linear regression, logistic, naïve bayes, knn, decision tree and random forest Model

Interpretations:

* + 1. As per confusion matrix the best model seems to be random forest and decision tree where accuracy and loss are similar.
    2. Logistic and linear model are worst performing model as per confusion matrix



* + 1. Decision tree

1. Machine generated alternative text:
   > tab.DT 
   pred. DT 
   0 832 
   14 
   1 63 
   89 
3. Machine generated alternative text:
   > accuracy. DT) 
   > accuracy. DT 
   [1] 0.9228457 
   > DTC2 
   > loss.OT 
   . 07039106 
   > opp. loss. DT<-tab. DTCI , 
   . loss.DT 
   > opp 
   1359223 
   . loss. OT<-O. 95*10ss. DT+O. 05"opp. loss. 
   > tot 
   > tot. loss.DT 
   . 07366762 

* Random forest:

Machine generated alternative text:
> tab. REM 
FALSE TRUE 
0 826 

Machine generated alternative text:
> accuracy. REM 
[1] o. 9208417 
> loss. 
> loss. REM 
. 06666667 
> opp. loss. RFM<-tab. REM Cl , RFM[2 , 2]) 
. loss. 
> opp 
1769912 
. loss. RFM<-O. 95•10ss. RFM+O. 05"opp. loss. REM 
> tot 
> tot. loss.RFM 
. 07218289 

1. Logistic model:

Machine generated alternative text:
> tab. logit 
obs 
pred 0 1 
0 841 136 
5 16 
attr(, "class" 
[1] "confusion. matrix" 
> accuracy. logic) 
> accuracy. logic 
[1] o. 8587174 
> loss. logit<-tab. logitC1 , logitC1 ,1]) 
> loss. logit 
1392016 
> opp. loss. logit<-tab. logitC2 ,l]/(tab. logitC2 ,2]) 
> opp. loss. logic 
. 2380952 
. loss. logit<-O. 95•10ss. logit+O. 05*opp. loss. logit 
> tot 
. loss. logic 
> tot 
1441463 

1. Knn model

Machine generated alternative text:
> accuracy. NB 
[1] o. 8937876 
> loss. NB<-tab. knn. 3 C 2 knn. 3 [2 knn. ,1]) 
> loss. NB 
. 08520179 
> opp. loss. NB<-tab. knn. 3 CI , knn. knn. 3C2 , 2]) 
. loss. NB 
> opp 
. 2830189 
. loss. NB<-O. 95* loss. NB+O. 05"opp. loss. NB 
> tot 
> tot. loss. NB 
. 09509265 

1. Naïve bayes

Machine generated alternative text:
> accuracy. NB 
[1] o. 8627255 
> loss. NB<-tab. NBC2 
> loss. NB 
. 08440797 
. loss. NB<-tab. , 
> opp 
. loss. NB 
> opp 
.4482759 
. loss. NB<-O. 95* loss. NB+O. 05"opp. loss. NB 
> tot 
> tot. loss. NB 
1026014 

1. Linear model

Taking full model and selecting significant variable:

Machine generated alternative text:
coeffi ci ents : 
(Intercept) 
contr act Renewal 
-12. 924 
custservcal 1 s 
Datapl an 
DayM i ns 
over ageFee 
Roan" i ns 
signif. codes: 
' 0.05 
Es t i mate 
-o. 0867208 
-o. 2948867 
o. 0514124 
-o. 0782733 
o. 0013130 
o. 0141218 
o. 0058763 
std. Error 
o. 0490331 
o. 0228171 
o. 0050377 
o. 0147211 
o. 0001215 
o. 0026122 
o. 0024156 
t value 
-1. 769 
10. 205 
5.317 
10. 806 
5.406 
2.433 
Pr It l) 
0.0771 . 
< 2e-16 
< 2e-16 
1.15e-07 
< 2e-16 
7.10e-08 
o. 0151 
' 0.001 ' 
Resi dual standard error: O. 3195 on 2328 degrees of 
Multiple R-squared: O. 1636, 
Adjusted R-squared: 
—statistic: 75.87 on 6 and 2328 DF-, 
p-value: < 2. 
freedom 
o. 1614 
2e-16 

Machine generated alternative text:
> tab. regSchurn, 
> tab. LPM 
FALSE TRUE 
0 830 
1 131 21 
pred. reg > O. 5) 

Machine generated alternative text:
> accuracy. LPM 
[1] o. 8527054 
> loss. LPM<-tab. 
> loss. LPM 
1363163 
> opp. loss. LPM<-tab. LPM Cl , LPM[2 , 2]) 
. loss. LPM 
> opp 
.4324324 
. loss. LPM<-O. 95•10ss. LPM+O. 05"opp. loss. LPM 
> tot 
> tot. loss. LPM 
.1511221 

**18. Interpretation of other Model Performance Measures for logistic <KS, AUC, GINI>**

GINI, AUC and KS value for decision tree model

Machine generated alternative text:
> auc 
> auc 
> auc 
.4681513 
<- performance(pred, "auc"); 
<- as. numeric(aucay. values) 
. 9021353 
> KS <- max(attr (perf, •y. values CCI]] -attr (perf, 
[1] o. 6999415 
'x. values CCI]]) 

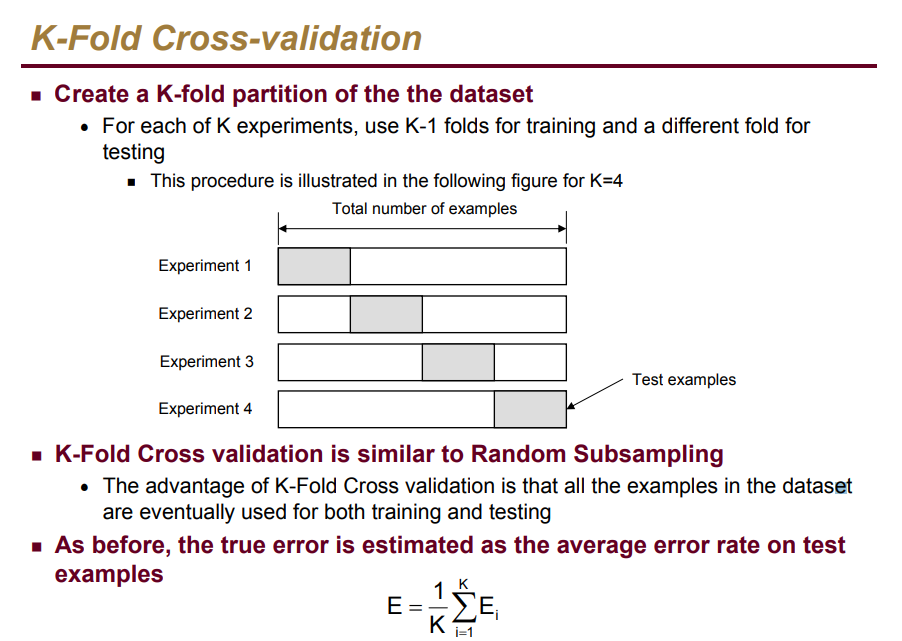
GINI, AUC and KS value for logistic tree model

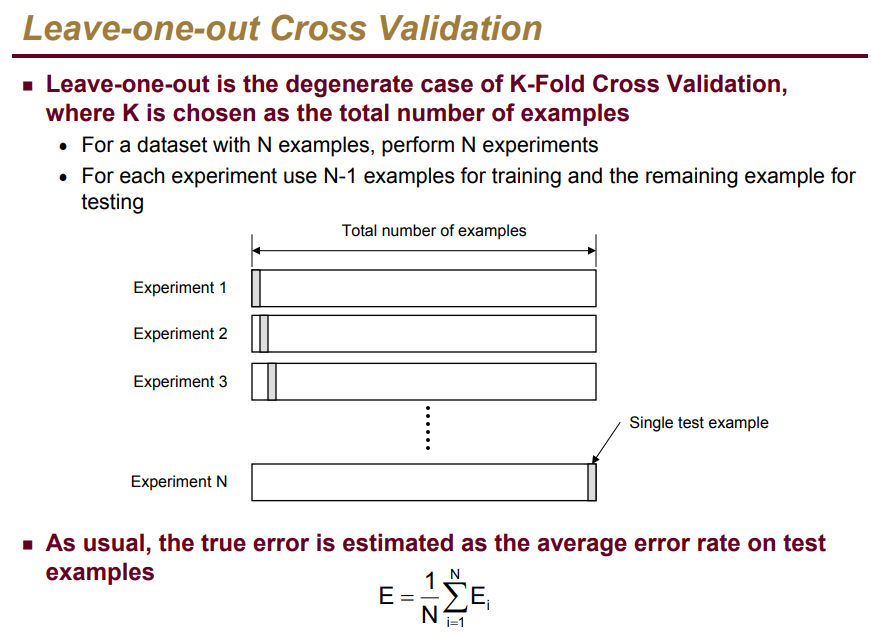
**19. Model validation exercise**

use k fold validation to interpret best model: **As per k fold, Naïve bayes is best model**

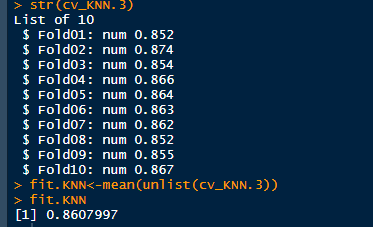
Our model runs on sample data but it may be sensitive to test data, that is why we need to validate our model with test data.

Below are the techniques used for validation:

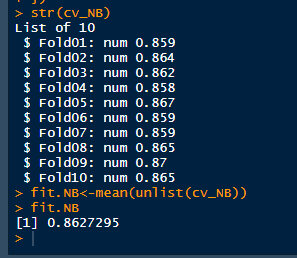




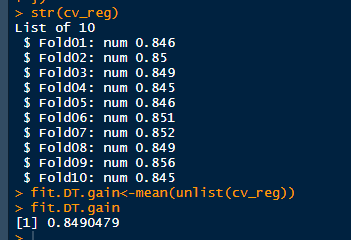
* + 1. Kfold validation for knn model for k=3 is 86 percent



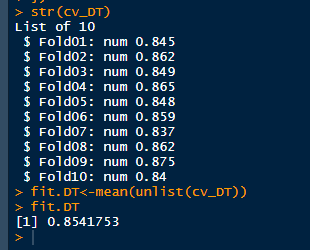
* + 1. Naïve bayes:



* + 1. Linear model



* + 1. Decision tree



**20. Actionable Insights and Recommendations**

**As far as models are concerned, the confusion matrix states that models decision tree and random forest are doing better, however as per**

**K fold validation the models which are doing better are knn and naivebayes.**

**21. Code for reference**

**library(car)**

**library(caret)**

**library(class)**

**library(devtools)**

**library(e1071)**

**install.packages("ggord")**

**library(ggord)**

**library(ggplot2)**

**library(Hmisc)**

**library(klaR)**

**library(klaR)**

**library(MASS)**

**library(nnet)**

**library(plyr)**

**library(pROC)**

**library(psych)**

**library(scatterplot3d)**

**library(SDMTools)**

**install.packages("dplyr")**

**library(dplyr)**

**library(ElemStatLearn)**

**library(rpart)**

**library(rpart.plot)**

**library(randomForest)**

**library(neuralnet)**

**library(caTools)**

**library(rpart)**

**library(rpart.plot)**

**library(rattle)**

**library(RColorBrewer)**

**library(data.table)**

**library(ROCR)**

**library(corrplot)**

**library(tidyverse)**

**library(VIF)**

**library(car)**

**library(caret)**

**install.packages(c("SDMTools","pROC", "Hmisc"))**

**library(SDMTools)**

**library(pROC)**

**library(Hmisc)**

**library(devtools)**

**install.packages("ggplot2")**

**library(ggplot2)**

**install.packages("caret")**

**library(caret)**

**setwd("C:/Users/spandey/Desktop")**

**getwd()**

**phone = read.csv("Cellphone.csv", header = TRUE)**

**attach(phone)**

**head(phone)**

**str(phone)**

**# univariate analysis**

**dim(phone)**

**sd(phone)**

**var(phone)**

**str(phone)**

**summary(phone)**

**attach(phone)**

**#histogram for numerical variables**

**qplot(DataUsage, data= phone)**

**qplot(AccountWeeks, data= phone)**

**qplot(CustServCalls, data= phone)**

**qplot(DayMins, data= phone)**

**qplot(DayCalls, data= phone)**

**qplot(MonthlyCharge, data= phone)**

**qplot(OverageFee, data= phone)**

**qplot(RoamMins, data= phone)**

**#bar plot for categorical variables**

**qplot(ContractRenewal, data= phone, geom = "bar")**

**qplot(Churn, data= phone, geom = "bar")**

**qplot(DataPlan, data= phone, geom = "bar")**

**## bivariate analysis (of only numerical variables)**

**# qplot()**

**library(corrplot)**

**qplot(DataUsage, DayMins, colour = Churn, data=phone)**

**qplot(DataUsage, CustServCalls, colour = Churn, data=phone)**

**qplot(DataUsage, MonthlyCharge, colour = Churn, data=phone)**

**qplot(DataUsage, AccountWeeks, colour = Churn, data=phone)**

**qplot(DataUsage, RoamMins, colour = Churn, data=phone)**

**qplot(CustServCalls, RoamMins, colour = Churn, data=phone)**

**qplot(DayMins, CustServCalls, colour = Churn, data=phone)**

**qplot(DayMins, MonthlyCharge, colour = Churn, data=phone)**

**qplot(DayMins, AccountWeeks, colour = Churn, data=phone)**

**qplot(DayMins, RoamMins, colour = Churn, data=phone)**

**#boxplot**

**boxplot(phone)**

**#correlation plot**

**Scatter<- phone[,-c(1,3,4)]**

**cor(Scatter)**

**correlations<- cor(Scatter)**

**corrplot(correlations, method="circle")**

**#treat missing values and negative values in data set**

**colSums(is.na(phone))**

**outlier = (phone)$out**

**outlier**

**boxplot(phone)**

**neg = phone<0**

**sum(neg)**

**#create dummy variables**

**phone$Contractcancel<-ifelse(phone$Churn=="cancelContract",0,1)**

**head(phone)**

**#str(phone)**

**# Split dataset into train and test**

**set.seed(1)**

**rows <- sample(2, size = nrow(phone),replace=TRUE, prob=c(0.7,0.3))**

**train <- phone[rows==1,]**

**test <- phone[rows==2,]**

**dim(phone)**

**dim(train)**

**dim(test)**

**#Data Frame for Linear Regression**

**head(phone)**

**train.reg<-train**

**head(train)**

**test.reg<-test**

**head(test)**

**#check multicollinearity using vif factor (take full datset)**

**library(car)**

**str(phone)**

**head(phone)**

**linear1= Churn ~ AccountWeeks+ContractRenewal +DataPlan+ContractRenewal+CustServCalls+DayCalls +DataUsage+DayMins+MonthlyCharge+OverageFee+RoamMins**

**full = lm(linear1, data = train.reg)**

**summary(full)**

**vif(full)**

**# keeping only significant variables ( from 1 variable to adding significant variable): here monthly charge and overage fee seems to be collinear hence dropping monthly charge**

**linear= Churn ~ ContractRenewal+CustServCalls+DataPlan+DayMins+OverageFee+RoamMins**

**fullnew = lm(linear, data = test.reg)**

**summary(fullnew)**

**vif(full)**

**mse1 <- mean((test.reg$Churn- pred.reg)^2)**

**print(mse1)**

**#Now some Predictions**

**pred.reg<-predict(full,newdata=test.reg)**

**pred.reg**

**#Confusionmatrix**

**tab.LPM<-table(test.reg$Churn, pred.reg > 0.5)**

**tab.LPM**

**accuracy.LPM<-sum(diag(tab.LPM))/sum(tab.LPM)**

**accuracy.LPM**

**loss.LPM<-tab.LPM[2,1]/(tab.LPM[2,1]+tab.LPM[1,1])**

**loss.LPM**

**opp.loss.LPM<-tab.LPM[1,2]/(tab.LPM[1,2]+tab.LPM[2,2])**

**opp.loss.LPM**

**tot.loss.LPM<-0.95\*loss.LPM+0.05\*opp.loss.LPM**

**tot.loss.LPM**

**#10 Fold Validation with Linear regression**

**library(caret)**

**set.seed(123)**

**folds<-createFolds(train$Churn,k=10)**

**str(folds)**

**installed.packages("plyr")**

**library(plyr)**

**linear= Churn ~ ContractRenewal+CustServCalls+DataPlan+DayMins+OverageFee++RoamMins**

**cv\_reg<-lapply(folds,function(x){**

**train.reg.kval<-train.reg[x,]**

**test.reg.kval<-test.reg[-x,]**

**reg.kval<-lm(linear, train.reg.kval)**

**reg.kval.pred<-predict(reg.kval, test.reg.kval)**

**tab.reg.kval<-table(test.reg.kval[,1], reg.kval.pred>0.5)**

**sum(diag(tab.reg.kval))/sum(tab.reg.kval)**

**})**

**str(cv\_reg)**

**fit.OLS<-mean(unlist(cv\_reg))**

**fit.OLS**

**##################logistic Model###########################**

**#data frame**

**train.logit<-train**

**test.logit<-test**

**head(train.logit)**

**head(test.logit)**

**# Fit the Sigmoid function**

**head(phone)**

**#using one variable**

**Logit1 <-Churn ~CustServCalls**

**Logit <- glm(Logit1 , train.logit, family = binomial)**

**summary(Logit)**

**vif(Logit)**

**#####Retain only significant ones after result from logit model**

**Logitfull <-Contractcancel ~DataPlan+RoamMins+DayMins+OverageFee+CustServCalls+Churn**

**Logit.final <- glm(Logitfull , train.logit, family = binomial)**

**summary(Logit.final)**

**vif(Logit.final)**

**pred.logit.final <- predict.glm(Logit.final, newdata=test.logit, type="response")**

**pred.logit.final**

**#Classification**

**head(test)**

**dim(test)**

**library(caret)**

**tab.logit= confusion.matrix(test.logit$Churn, pred.logit.final,threshold = 0.5)**

**tab.logit**

**accuracy.logit<-sum(diag(tab.logit))/sum(tab.logit)**

**accuracy.logit**

**loss.logit<-tab.logit[1,2]/(tab.logit[1,2]+tab.logit[1,1])**

**loss.logit**

**opp.loss.logit<-tab.logit[2,1]/(tab.logit[2,1]+tab.logit[2,2])**

**opp.loss.logit**

**tot.loss.logit<-0.95\*loss.logit+0.05\*opp.loss.logit**

**tot.loss.logit**

**#############gini, ks, auc###########**

**library(ROCR)**

**library(ineq)**

**pred <- prediction(CD.sample$predict.score[,2], CD.sample$Target)**

**perf <- performance(pred, "tpr", "fpr")**

**plot(perf)**

**KS <- max(attr(perf, 'y.values')[[1]]-attr(perf, 'x.values')[[1]])**

**auc <- performance(pred,"auc");**

**auc <- as.numeric(auc@y.values)**

**gini = ineq(CD.sample$predict.score[,2], type="Gini")**

**with(CD.sample, table(Target, predict.class))**

**auc**

**KS**

**gini**

**################ KNN Model ##########################**

**setwd("C:/Users/spandey/Desktop")**

**getwd()**

**phone = read.csv("Cellphone.csv", header = TRUE)**

**attach(phone)**

**# But before that, we will normalize**

**normalize<-function(x){**

**+return((x-min(x))/(max(x)-min(x)))}**

**head(phone)**

**names(phone)**

**#create dummy variables**

**#phone$Contractcancel<-ifelse(phone$Churn=="cancelContract",1,0)**

**head(phone)**

**#str(phone)**

**phone$norm.AccountWeeks<-normalize(AccountWeeks)**

**phone$norm.ContractRenewal<-normalize(ContractRenewal)**

**phone$norm.DataPlan<-normalize(DataPlan)**

**phone$norm.DataUsage<-normalize(DataUsage)**

**phone$norm.CustServCalls<-normalize(CustServCalls)**

**phone$norm.DayMins<-normalize(DayMins)**

**phone$norm.DayCalls<-normalize(DayCalls)**

**phone$norm.MonthlyCharge<-normalize(MonthlyCharge)**

**phone$norm.RoamMins<-normalize(RoamMins)**

**phone$norm.OverageFee<-normalize(OverageFee)**

**head(phone)**

**#divide data into training and val**

**set.seed(1)**

**rows <- sample(2, size = nrow(phone),replace=TRUE, prob=c(0.7,0.3))**

**train <- phone[rows==1,]**

**test <- phone[rows==2,]**

**dim(phone)**

**dim(train)**

**dim(test)**

**#Data Frame for KNN**

**#Normalization is must**

**train.NB<-train[,c(1,12:20)]**

**test.NB<-test[,c(1,12:20)]**

**head(train.NB)**

**head(test.NB)**

**####KNN**

**#knn3**

**y\_pred.3<-knn(train=train.NB[,-1],test=test.NB[-1], cl=train.NB[,1],k=3)**

**tab.knn.3<-table(test.NB[,1],y\_pred.3)**

**tab.knn.3**

**accuracy.NB<-sum(diag(tab.knn.3))/sum(tab.knn.3)**

**accuracy.NB**

**loss.NB<-tab.knn.3[2,1]/(tab.knn.3[2,1]+tab.knn.3[1,1])**

**loss.NB**

**opp.loss.NB<-tab.knn.3[1,2]/(tab.knn.3[1,2]+tab.knn.3[2,2])**

**opp.loss.NB**

**tot.loss.NB<-0.95\*loss.NB+0.05\*opp.loss.NB**

**tot.loss.NB**

**#knn5**

**y\_pred.5<-knn(train=train.NB[,-1],test=test.NB[-1], cl=train.NB[,1],k=5)**

**tab.knn.5<-table(test.NB[,1],y\_pred.5)**

**tab.knn.5**

**accuracy.knn.5<-sum(diag(tab.knn.5))/sum(tab.knn.5)**

**accuracy.knn.5**

**loss.knn.5<-tab.knn.5[2,1]/(tab.knn.5[2,1]+tab.knn.5[1,1])**

**loss.knn.5**

**#knn7**

**y\_pred.7<-knn(train=train.NB[,-1],test=test.NB[-1], cl=train.NB[,1],k=7)**

**tab.knn.7<-table(test.NB[,1],y\_pred.7)**

**tab.knn.7**

**accuracy.knn.7<-sum(diag(tab.knn.7))/sum(tab.knn.7)**

**accuracy.knn.7**

**loss.knn.7<-tab.knn.7[2,1]/(tab.knn.7[2,1]+tab.knn.7[1,1])**

**loss.knn.7**

**#knn9**

**y\_pred.9<-knn(train=train.NB[,-1],test=test.NB[-1], cl=train.NB[,1],k=9)**

**tab.knn.9<-table(test.NB[,1],y\_pred.9)**

**tab.knn.9**

**accuracy.knn.9<-sum(diag(tab.knn.9))/sum(tab.knn.9)**

**accuracy.knn.9**

**loss.knn.9<-tab.knn.9[2,1]/(tab.knn.9[2,1]+tab.knn.9[1,1])**

**loss.knn.9**

**#####kfold validation #####################,]**

**library(caret)**

**set.seed(123)**

**folds<-createFolds(train$Churn,k=10)**

**str(folds)**

**cv\_KNN.3<-lapply(folds,function(x){**

**train.NB.kval<-train.NB[x,]**

**test.NB.kval<-test.NB[-x,]**

**train.knn.kval<-as.data.frame(lapply(train.NB.kval[,c(2:10)],normalize))**

**test.knn.kval<-as.data.frame(lapply(test.NB.kval[,c(2:10)],normalize))**

**train\_target.kval<-train.NB.kval[,1]**

**test\_target.kval<-test.NB.kval[,1]**

**knn3.kval<-knn(train=train.knn.kval,test=test.knn.kval,cl=train\_target.kval,k=3)**

**tab.knn.3.kval<-table(test\_target.kval,knn3.kval)**

**sum(diag(tab.knn.3.kval))/sum(tab.knn.3.kval)**

**})**

**str(cv\_KNN.3)**

**fit.KNN<-mean(unlist(cv\_KNN.3))**

**fit.KNN**

**##################### Naive Bayes####################**

**train.NB$Churn<-as.factor(train.NB$Churn)**

**test.NB$Churn<-as.factor(test.NB$Churn)**

**library(naivebayes)**

**NB<-naive\_bayes(x=train.NB[-1], y=train.NB$Churn)**

**#pedict**

**y\_pred.NB<-predict(NB,newdata=test.NB[-1])**

**y\_pred.NB**

**#Confusion matrix**

**tab.NB=table(test.NB[,1],y\_pred.NB)**

**tab.NB**

**accuracy.NB<-sum(diag(tab.NB))/sum(tab.NB)**

**accuracy.NB**

**loss.NB<-tab.NB[2,1]/(tab.NB[2,1]+tab.NB[1,1])**

**loss.NB**

**opp.loss.NB<-tab.NB[1,2]/(tab.NB[1,2]+tab.NB[2,2])**

**opp.loss.NB**

**tot.loss.NB<-0.95\*loss.NB+0.05\*opp.loss.NB**

**tot.loss.NB**

**library(caret)**

**set.seed(123)**

**folds<-createFolds(train$Churn,k=10)**

**str(folds)**

**cv\_NB<-lapply(folds,function(x){**

**train.NB.kval<-train.NB[x,]**

**test.NB.kval<-test.NB[-x,]**

**NB.kval<-naive\_bayes(x=train.NB.kval[-1], y=train.NB.kval[,1])**

**y\_pred.NB.kval<-predict(NB,newdata=test.NB.kval[-1])**

**cm.NB.kval=table(test.NB.kval[,1],y\_pred.NB.kval)**

**sum(diag(cm.NB.kval))/sum(cm.NB.kval)**

**})**

**str(cv\_NB)**

**fit.NB<-mean(unlist(cv\_NB))**

**fit.NB**

**#Decision Tree and RFM**

**#Data frame**

**head(phone)**

**train.NB<-train[,c(1,12:20)]**

**test.NB<-test[,c(1,12:20)]**

**head(train.NB)**

**head(test.NB)**

**###Now Decision Trees**

**library(rpart)**

**library(rpart.plot)**

**library(randomForest)**

**DT<-rpart(Churn~., method="class",train.NB)**

**rpart.plot(DT, type=3,extra=101,fallen.leaves = T)**

**pred.DT= predict(DT, type = "class",test.NB)**

**############### ks, gini, auc###################**

**pred.1 = predict(DT, test.NB)**

**library(ROCR)**

**pred <- prediction(pred.1[,2], train.NB$Churn)**

**perf <- performance(pred, "tpr", "fpr")**

**#install.packages("ineq")**

**library(ineq)**

**gini = ineq(pred.1, type="Gini")**

**gini**

**auc <- performance(pred,"auc");**

**auc <- as.numeric(auc@y.values)**

**auc**

**KS <- max(attr(perf, 'y.values')[[1]]-attr(perf, 'x.values')[[1]])**

**KS**

**###############confusion matrix###########**

**tab.DT<-table( test.NB$Churn,pred.DT)**

**tab.DT**

**accuracy.DT<-sum(diag(tab.DT))/sum(tab.DT)**

**accuracy.DT**

**loss.DT<-tab.DT[2,1]/(tab.DT[2,1]+tab.DT[1,1])**

**loss.DT**

**opp.loss.DT<-tab.DT[1,2]/(tab.DT[1,2]+tab.DT[2,2])**

**opp.loss.DT**

**tot.loss.DT<-0.95\*loss.DT+0.05\*opp.loss.DT**

**tot.loss.DT**

**###############kfold###################**

**#########**

**#10 Fold on Decision Trees**

**DT<-Churn ~ norm.AccountWeeks+norm.ContractRenewal+norm.DataPlan+norm.DataUsage+norm.CustServCalls+norm.DayMins+norm.DayCalls+norm.MonthlyCharge+norm.RoamMins**

**set.seed(123)**

**folds<-createFolds(train$Churn,k=10)**

**str(folds)**

**cv\_DT<-lapply(folds,function(x){**

**train.DT.kval<-train.NB[x,]**

**test.DT.kval<-test.NB[-x,]**

**DT.kval<-rpart(Churn~., method="class",train.DT.kval)**

**pred.DT.kval = predict(DT.kval, type="class",newdata=test.DT.kval)**

**tab.DT.kval<-table( pred.DT.kval,test.DT.kval[,1])**

**sum(diag(tab.DT.kval))/sum(tab.DT.kval)**

**})**

**str(cv\_DT)**

**fit.DT<-mean(unlist(cv\_DT))**

**fit.DT**

**############NOW RANDOM FOREST**

**#Run RFM**

**rfm<-randomForest(Churn~.,train.NB)**

**predictrfm<-predict(rfm,test.NB)**

**tab.RFM<-table(test.NB$Churn,predictrfm>0.5)**

**tab.RFM**

**accuracy.RFM<-sum(diag(tab.RFM))/sum(tab.RFM)**

**accuracy.RFM**

**loss.RFM<-tab.RFM[2,1]/(tab.RFM[2,1]+tab.RFM[1,1])**

**loss.RFM**

**opp.loss.RFM<-tab.RFM[1,2]/(tab.RFM[1,2]+tab.RFM[2,2])**

**opp.loss.RFM**

**tot.loss.RFM<-0.95\*loss.RFM+0.05\*opp.loss.RFM**

**tot.loss.RFM**