Capstone Project

Final Report

Done by: Daoud Tebbakh

27.09.2020

Problem statement

Falcon airlines determine the importance role Loyal Customer has the impact of losing market share, so a good starting point would be swaying a passenger feedback towards 'satisfied' by understand which parameters important to 'satisfied' passenger and predict whether will be satisfied or not to give the right treatment, using the best model we can portion of the population should be targeted to get the highest response rate with less amount of money better than portion randomly selected, also classify new customer either satisfied or dissatisfied.

Tools:

1. Laptop(core i7 8th Gen , 8GB RAM, Windows x64)
2. RStudio 3.6.2

Data Exploratory:

Collecting Aviation data by using: random sample selected by flight booking sites data by CSV file named 'Flight data' and Survey online forms feedback by CSV file named 'Survey data'. , flight data collected while booking the flight, survey data collected after the flight.

Qualitative Method: Likert Scale (extremely poor....excellent) provides depth, effective, efficient and detail for 90917 individuals.

Flight_data: 90917 Customer x 9 Variables:

- **CustomerID**: [numeric] ID customers
- **Gender**: [character] Female and Male
- **CustomerType**: [character] 'disloyal Customer' and 'Loyal Customer'
- **Age**: [numeric] age years number.
- **TypeTravel**: [character] travel type 'Business travel' and 'Personal Travel'
- **Class**: [character] 'Business' and 'Eco' and 'Eco Plus'
- **Flight_Distance**: [numeric] distance in Kilometer.
- **DepartureDelayin_Mins**: [numeric] departure delaying in minutes.
- **ArrivalDelayin_Mins**: [numeric] arrival delaying in minutes.

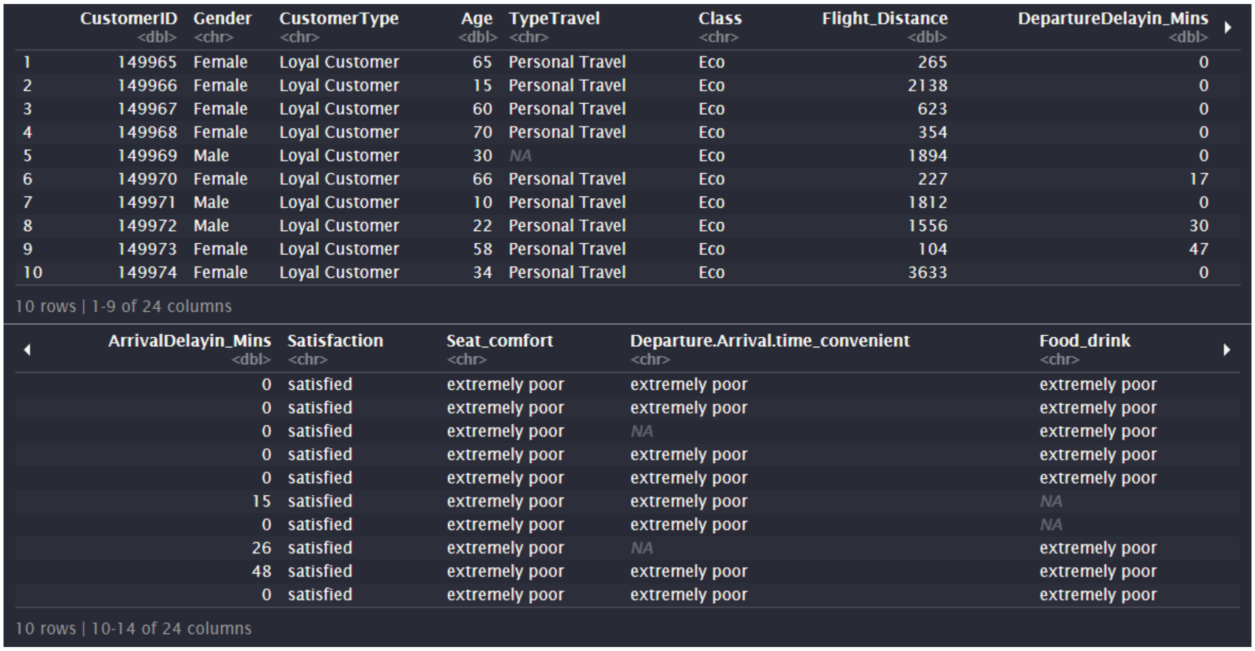
Survey_data: 90917 Customer x 16 Variables:

- **CustomerId**: [numeric] ID customers
- **Satisfaction**: [character] 'neutral or dissatisfied' and 'satisfied'
- **Gate_location**: [character] [very convenient, convenient, manageable, need improvement, Inconvinient, very inconvenient]
- **Seat_comfort, Departure.Arrival.time_convenient, Food_drink, Inflightwifi_service, Inflight_entertainment, Online_support, Ease_of_Onlinebooking, Onboard_service, Leg_room_service, Baggage_handling, Checkin_service, Cleanliness, Online_boarding** : [character] [excellent, good, acceptable, need improvement, poor, extremely poor]

Dependent variable: **Satisfaction.**

Independent variables: **all the other variables.**

Show first 10 Rows unclean data: 'head()'

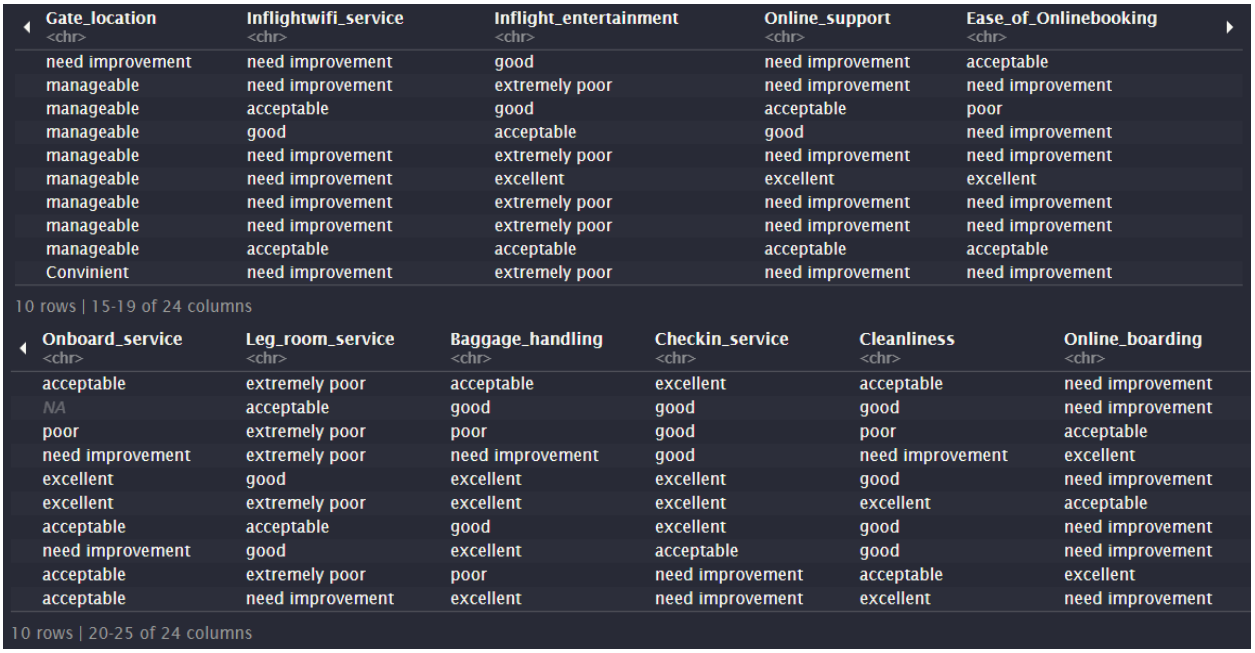


	CustomerID <dbl>	Gender <chr>	CustomerType <chr>	Age <dbl>	TypeTravel <chr>	Class <chr>	Flight_Distance <dbl>	DepartureDelayin_Mins <dbl>
1	149965	Female	Loyal Customer	65	Personal Travel	Eco	265	0
2	149966	Female	Loyal Customer	15	Personal Travel	Eco	2138	0
3	149967	Female	Loyal Customer	60	Personal Travel	Eco	623	0
4	149968	Female	Loyal Customer	70	Personal Travel	Eco	354	0
5	149969	Male	Loyal Customer	30	NA	Eco	1894	0
6	149970	Female	Loyal Customer	66	Personal Travel	Eco	227	17
7	149971	Male	Loyal Customer	10	Personal Travel	Eco	1812	0
8	149972	Male	Loyal Customer	22	Personal Travel	Eco	1556	30
9	149973	Female	Loyal Customer	58	Personal Travel	Eco	104	47
10	149974	Female	Loyal Customer	34	Personal Travel	Eco	3633	0

10 rows | 1-9 of 24 columns

ArrivalDelayin_Mins <dbl>	Satisfaction <chr>	Seat_comfort <chr>	Departure.Arrival.time_convenient <chr>	Food_drink <chr>
0	satisfied	extremely poor	extremely poor	extremely poor
0	satisfied	extremely poor	extremely poor	extremely poor
0	satisfied	extremely poor	NA	extremely poor
0	satisfied	extremely poor	extremely poor	extremely poor
0	satisfied	extremely poor	extremely poor	extremely poor
15	satisfied	extremely poor	extremely poor	NA
0	satisfied	extremely poor	extremely poor	NA
26	satisfied	extremely poor	NA	extremely poor
48	satisfied	extremely poor	extremely poor	extremely poor
0	satisfied	extremely poor	extremely poor	extremely poor

10 rows | 10-14 of 24 columns



```
Gate_location       Inflightwifi_service  Inflight_entertainment  Online_support     Ease_of_Onlinebooking
<chr>               <chr>                 <chr>                   <chr>              <chr>
need improvement    need improvement      good                    need improvement   acceptable
manageable          need improvement      extremely poor          need improvement   need improvement
manageable          acceptable            good                    acceptable         poor
manageable          good                  acceptable              good               need improvement
manageable          need improvement      extremely poor          need improvement   need improvement
manageable          need improvement      excellent               excellent          excellent
manageable          need improvement      extremely poor          need improvement   need improvement
manageable          need improvement      extremely poor          need improvement   need improvement
manageable          acceptable            acceptable              acceptable         acceptable
Convinient          need improvement      extremely poor          need improvement   need improvement
10 rows | 15-19 of 24 columns
```

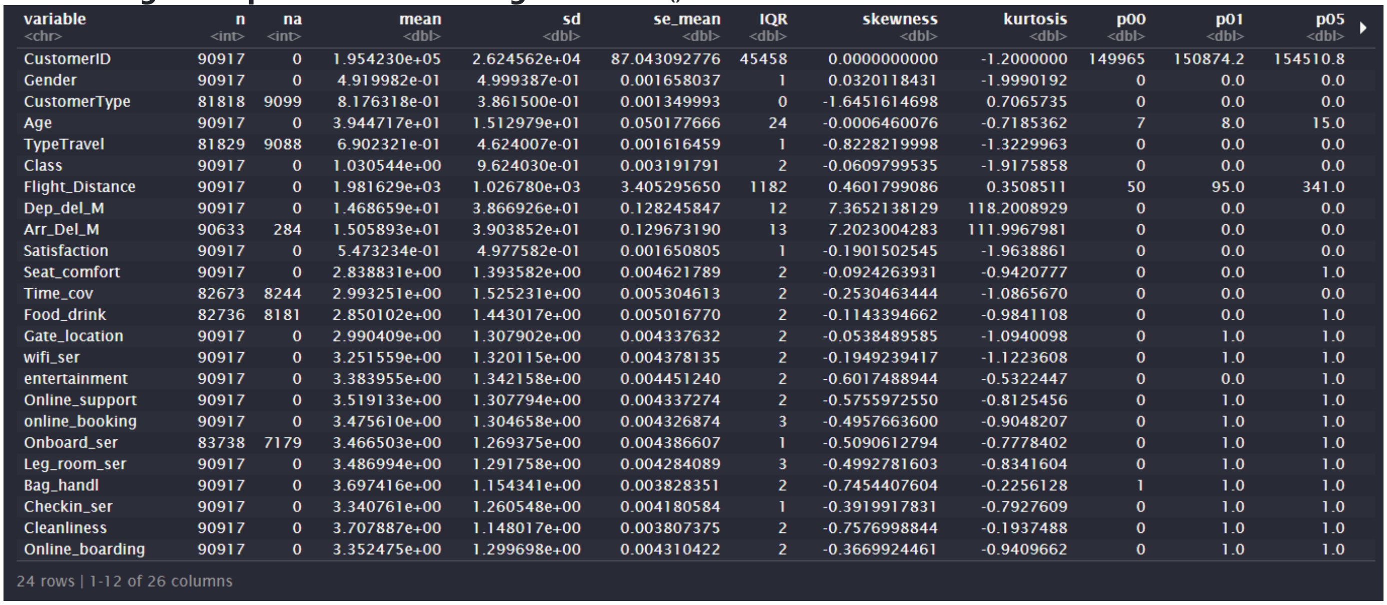
```
Onboard_service     Leg_room_service   Baggage_handling   Checkin_service    Cleanliness        Online_boarding
<chr>               <chr>              <chr>              <chr>              <chr>              <chr>
acceptable          extremely poor     acceptable         excellent          acceptable         need improvement
NA                  acceptable         good               good               good               need improvement
poor                extremely poor     poor               good               poor               acceptable
need improvement    extremely poor     need improvement   good               need improvement   excellent
excellent           good               excellent          excellent          good               need improvement
excellent           extremely poor     excellent          excellent          excellent          acceptable
acceptable          acceptable         good               excellent          good               need improvement
need improvement    good               excellent          acceptable         good               need improvement
acceptable          extremely poor     poor               need improvement   acceptable         excellent
acceptable          need improvement   excellent          need improvement   excellent          need improvement
10 rows | 20-25 of 24 columns
```

Initial Exploratory Data Analysis:

Renaming some variables: 'rename()'

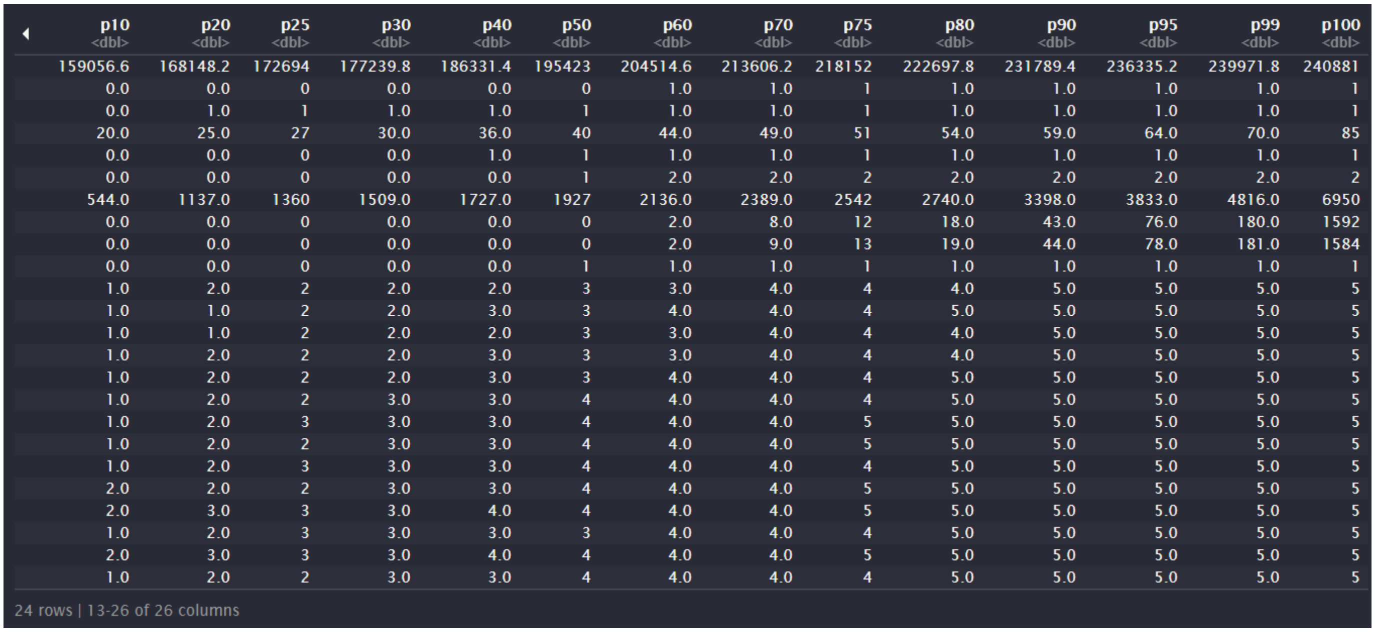
- Dep_del_M=DepartureDelayin_Mins,
- Arr_Del_M=ArrivalDelayin_Mins,
- Time_cov=Departure.Arrival.time_convenient,
- wifi_ser=Inflightwifi_service,
- entertainment=Inflight_entertainment,
- online_booking=Ease_of_Onlinebooking,
- Onboard_ser=Onboard_service,
- Leg_room_ser=Leg_room_service,
- Bag_handl=Baggage_handling,
- Checkin_ser=Checkin_service

Calculating descriptive statistics using: ‘describe()’



variable <chr>	n <int>	na <int>	mean <dbl>	sd <dbl>	se_mean <dbl>	IQR <dbl>	skewness <dbl>	kurtosis <dbl>	p00 <dbl>	p01 <dbl>	p05 <dbl>
CustomerID	90917	0	1.954230e+05	2.624562e+04	87.043092776	45458	0.0000000000	-1.2000000	149965	150874.2	154510.8
Gender	90917	0	4.919982e-01	4.999387e-01	0.001658037	1	0.0320118431	-1.9990192	0	0.0	0.0
CustomerType	81818	9099	8.176318e-01	3.861500e-01	0.001349993	0	-1.6451614698	0.7065735	0	0.0	0.0
Age	90917	0	3.944717e+01	1.512979e+01	0.050177666	24	-0.0006460076	-0.7185362	7	8.0	15.0
TypeTravel	81829	9088	6.902321e-01	4.624007e-01	0.001616459	1	-0.8228219998	-1.3229963	0	0.0	0.0
Class	90917	0	1.030544e+00	9.624030e-01	0.003191791	2	-0.0609799535	-1.9175858	0	0.0	0.0
Flight_Distance	90917	0	1.981629e+03	1.026780e+03	3.405295650	1182	0.4601799086	0.3508511	50	95.0	341.0
Dep_del_M	90917	0	1.468659e+01	3.866926e+01	0.128245847	12	7.3652138129	118.2008929	0	0.0	0.0
Arr_Del_M	90633	284	1.505893e+01	3.903852e+01	0.129673190	13	7.2023004283	111.9967981	0	0.0	0.0
Satisfaction	90917	0	5.473234e-01	4.977582e-01	0.001650805	1	-0.1901502545	-1.9638861	0	0.0	0.0
Seat_comfort	90917	0	2.838831e+00	1.393582e+00	0.004621789	2	-0.0924263931	-0.9420777	0	0.0	1.0
Time_cov	82673	8244	2.993251e+00	1.525231e+00	0.005304613	2	-0.2530463444	-1.0865670	0	0.0	0.0
Food_drink	82736	8181	2.850102e+00	1.443017e+00	0.005016770	2	-0.1143394662	-0.9841108	0	0.0	1.0
Gate_location	90917	0	2.990409e+00	1.307902e+00	0.004337632	2	-0.0538489585	-1.0940098	0	1.0	1.0
wifi_ser	90917	0	3.251559e+00	1.320115e+00	0.004378135	2	-0.1949239417	-1.1223608	0	1.0	1.0
entertainment	90917	0	3.383955e+00	1.342158e+00	0.004451240	2	-0.6017488944	-0.5322447	0	0.0	1.0
Online_support	90917	0	3.519133e+00	1.307794e+00	0.004337274	2	-0.5755972550	-0.8125456	0	1.0	1.0
online_booking	90917	0	3.475610e+00	1.304658e+00	0.004326874	3	-0.4957663600	-0.9048207	0	1.0	1.0
Onboard_ser	83738	7179	3.466503e+00	1.269375e+00	0.004386607	1	-0.5090612794	-0.7778402	0	1.0	1.0
Leg_room_ser	90917	0	3.486994e+00	1.291758e+00	0.004284089	3	-0.4992781603	-0.8341604	0	1.0	1.0
Bag_handl	90917	0	3.697416e+00	1.154341e+00	0.003828351	2	-0.7454407604	-0.2256128	1	1.0	1.0
Checkin_ser	90917	0	3.340761e+00	1.260548e+00	0.004180584	1	-0.3919917831	-0.7927609	0	1.0	1.0
Cleanliness	90917	0	3.707887e+00	1.148017e+00	0.003807375	2	-0.7576998844	-0.1937488	0	1.0	1.0
Online_boarding	90917	0	3.352475e+00	1.299698e+00	0.004310422	2	-0.3669924461	-0.9409662	0	1.0	1.0

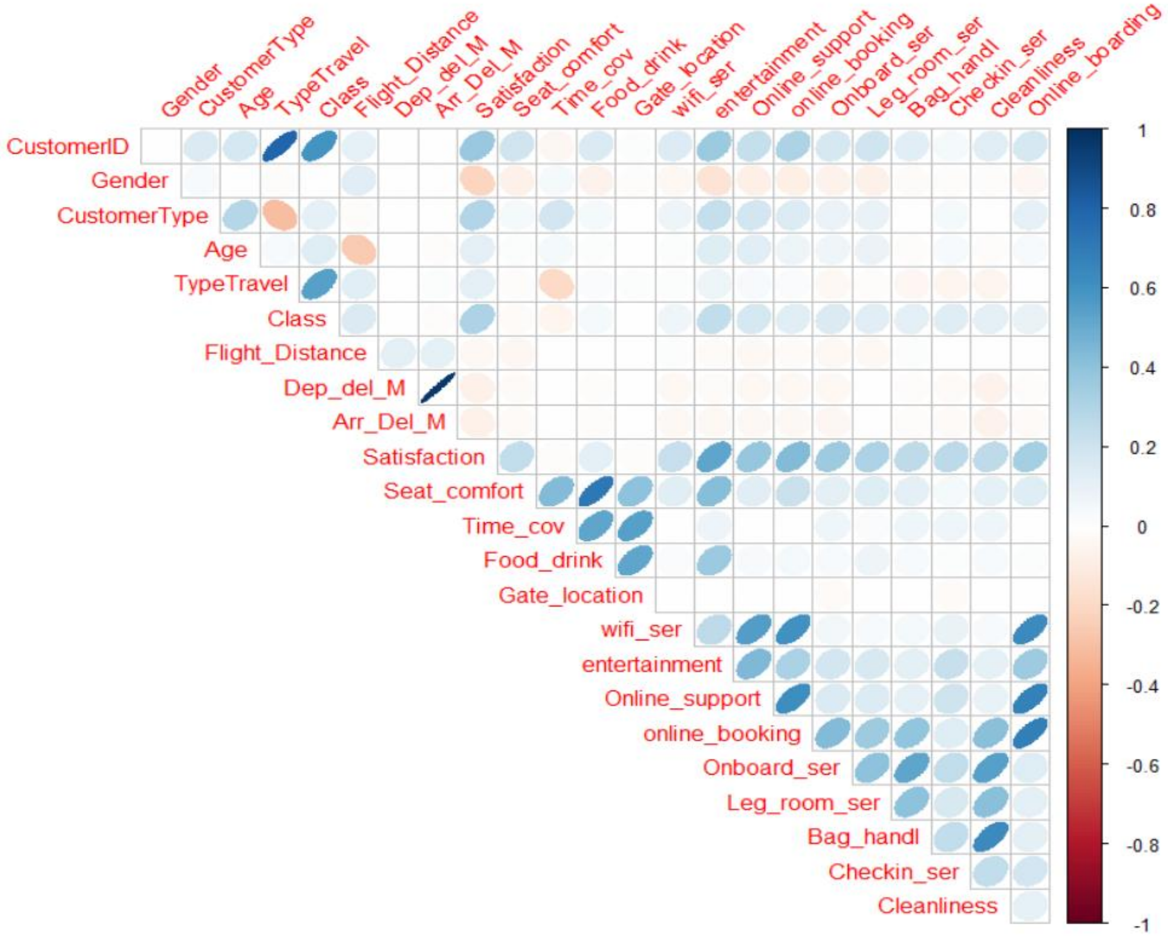
24 rows | 1-12 of 26 columns



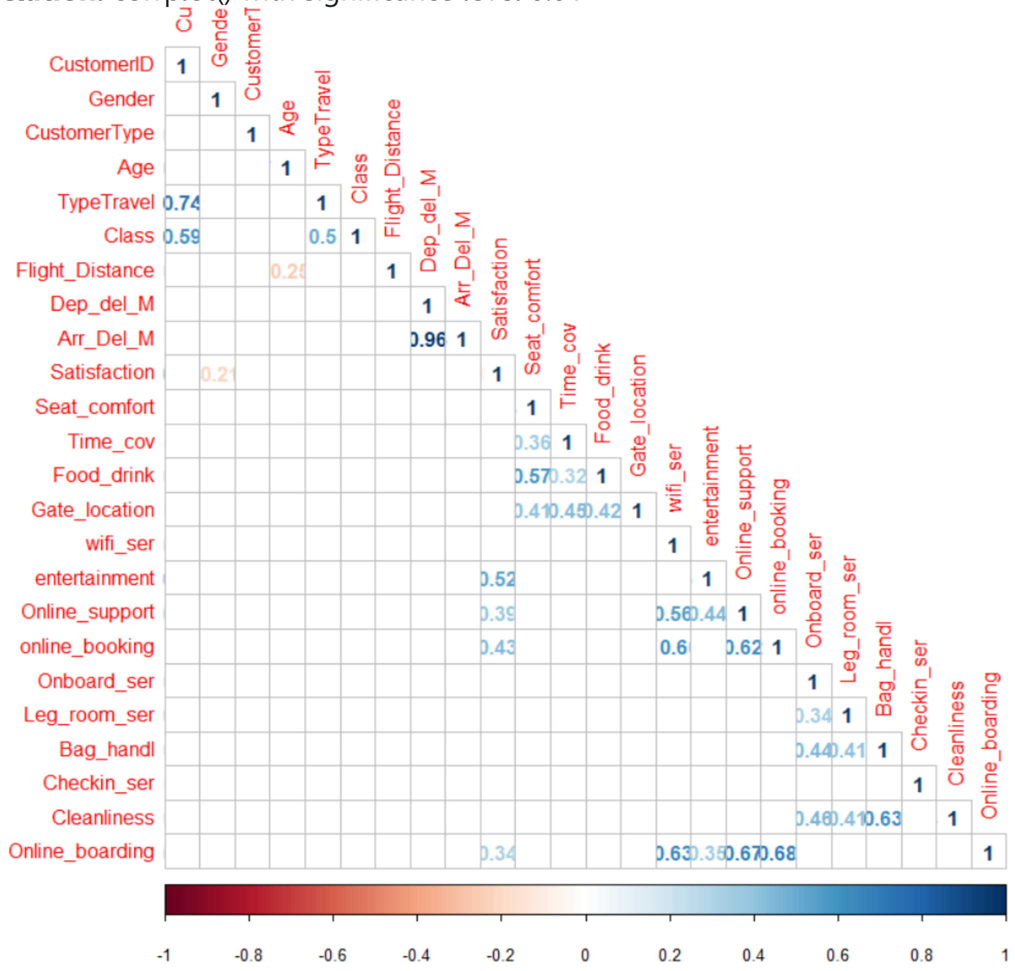
p10 <dbl>	p20 <dbl>	p25 <dbl>	p30 <dbl>	p40 <dbl>	p50 <dbl>	p60 <dbl>	p70 <dbl>	p75 <dbl>	p80 <dbl>	p90 <dbl>	p95 <dbl>	p99 <dbl>	p100 <dbl>
159056.6	168148.2	172694	177239.8	186331.4	195423	204514.6	213606.2	218152	222697.8	231789.4	236335.2	239971.8	240881
0.0	0.0	0	0.0	0.0	0	1.0	1.0	1	1.0	1.0	1.0	1.0	1
0.0	1.0	1	1.0	1.0	1	1.0	1.0	1	1.0	1.0	1.0	1.0	1
20.0	25.0	27	30.0	36.0	40	44.0	49.0	51	54.0	59.0	64.0	70.0	85
0.0	0.0	0	0.0	1.0	1	1.0	1.0	1	1.0	1.0	1.0	1.0	1
0.0	0.0	0	0.0	0.0	1	2.0	2.0	2	2.0	2.0	2.0	2.0	2
544.0	1137.0	1360	1509.0	1727.0	1927	2136.0	2389.0	2542	2740.0	3398.0	3833.0	4816.0	6950
0.0	0.0	0	0.0	0.0	0	2.0	8.0	12	18.0	43.0	76.0	180.0	1592
0.0	0.0	0	0.0	0.0	0	2.0	9.0	13	19.0	44.0	78.0	181.0	1584
0.0	0.0	0	0.0	0.0	1	1.0	1.0	1	1.0	1.0	1.0	1.0	1
1.0	2.0	2	2.0	2.0	3	3.0	4.0	4	4.0	5.0	5.0	5.0	5
1.0	1.0	2	2.0	3.0	3	4.0	4.0	4	5.0	5.0	5.0	5.0	5
1.0	1.0	2	2.0	2.0	3	3.0	4.0	4	4.0	5.0	5.0	5.0	5
1.0	2.0	2	2.0	3.0	3	3.0	4.0	4	4.0	5.0	5.0	5.0	5
1.0	2.0	2	2.0	3.0	3	4.0	4.0	4	5.0	5.0	5.0	5.0	5
1.0	2.0	2	3.0	3.0	4	4.0	4.0	4	5.0	5.0	5.0	5.0	5
1.0	2.0	3	3.0	3.0	4	4.0	4.0	5	5.0	5.0	5.0	5.0	5
1.0	2.0	2	3.0	3.0	4	4.0	4.0	5	5.0	5.0	5.0	5.0	5
1.0	2.0	3	3.0	3.0	4	4.0	4.0	4	5.0	5.0	5.0	5.0	5
2.0	2.0	2	3.0	3.0	4	4.0	4.0	5	5.0	5.0	5.0	5.0	5
2.0	3.0	3	3.0	4.0	4	4.0	4.0	5	5.0	5.0	5.0	5.0	5
1.0	2.0	3	3.0	3.0	3	4.0	4.0	4	5.0	5.0	5.0	5.0	5
2.0	3.0	3	3.0	4.0	4	4.0	4.0	5	5.0	5.0	5.0	5.0	5
1.0	2.0	2	3.0	3.0	4	4.0	4.0	4	5.0	5.0	5.0	5.0	5

24 rows | 13-26 of 26 columns

Visualization of the correlation matrix using: ‘plot_correlate()’



correlation: corrplot() with significance level 0.01

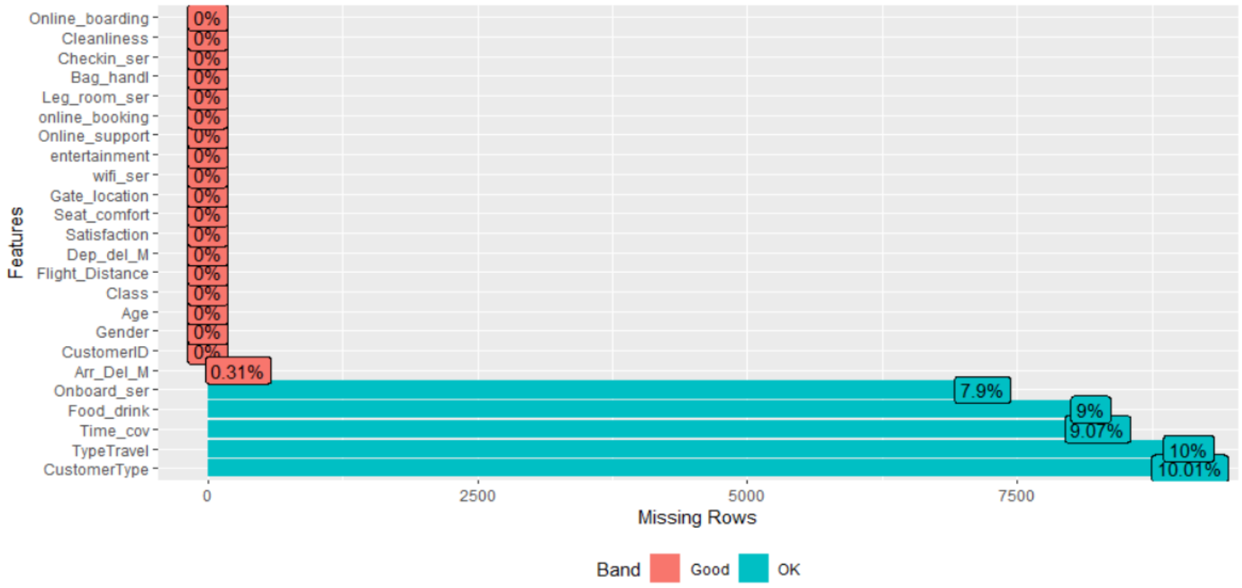


Observation: The variables are sorted by correlation top 6:

- 96% Dep_del_M --------- Arr_Del_M:
- " DepartureDelayin_Mins " have very strong connection of 96% with "ArrivalDelayin_Mins ", Very logical delay at the depart with create delay at the arrive.
- 74% TypeTravel --------- Class:
  " TypeTravel " have good connection of 74% with "Class", logically customer with 'Business travel' will take 'Business' class.
- 68% online_booking ---- Online_boarding:
  "online_booking" have connection of 68% with "Online_boarding",
- 67% online_booking ---- Online_support:
  "online_booking" have connection of 67% with "Online_support"
- 63% Bag_handl ---------- Cleanliness :
  "Bag_handl" have connection of 63% with "Cleanliness"
- 63% online_booking ---- wifi_ser:

“online_booking” have connection of 63% with “wifi_ser”

Plot Missing Values using: ‘plot_missing()’



Observation: missing values will be treated in Project Notes 2

Missing values for Total_data:

- Arr_Del_M: 0.31% which are 284 values.
- Onboard_ser: 7.9% which are 7179 values.
- Food_drink: 9% which are 8181 values.
- Time_cov: 9.07% which are 8244 values.
- TypeTravel: 10% which are 9088 values.
- CustomerType : 10.01% which are 9099 values.

Variable Transformation: ‘ifelse()’

variable	transform
Gender	
Male	1
Female	0
CustomerType	
Disloyal Customer	0
Loyal Customer	1
TypeTravel	
Personal Travel	0

Business travel	1
Class	
Eco	0
Eco Plus	1
Business	2
Satisfaction	
neutral or dissatisfied	0
satisfied	1

The rest of Survey variables: 'levels()'

variable	variable	transform
very inconvinient	extremely poor	0
Inconvinient	poor	1
need improvement	need improvement	2
Manageable	acceptable	3
Convinient	good	4
very convinient	excellent	5

1) Data pre-processing:

Removal of unwanted variables:

All the variables are need except for 'CustomerId' from 'Survey_data', we drop it on merging 'Survey_data' and 'Flight_data' dataframes together by CustomerID.

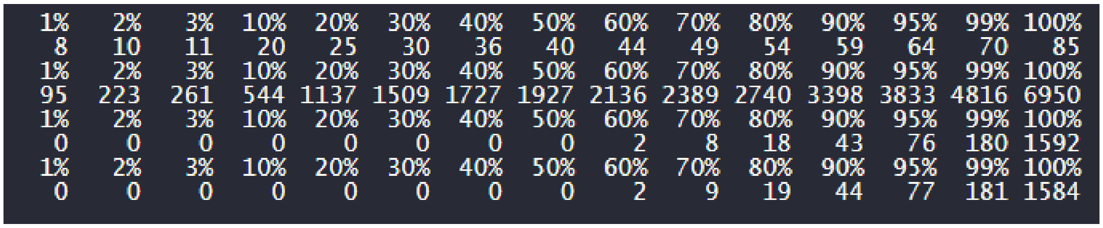
Missing Value Treatment:

- We replace NA values for 'CustomerType' and 'TypeTravel' with 'median' because the variables are binary, must be data_collection Error from Flight company
- We replace NA values for 'Arr_Del_M' with 'mean' because the variable is continuous, must be data_collection Error from Flight company
- Since the missing values are survey collected from the customer and the missing values almost 10%, they chose Not to fill the survey, replace NA for 'Time_cov' and 'Food_drink' and 'Onboard_ser' with 6.

Outlier treatment:

- Age: Density plot looks normal-skewed distribution, No Treatment required
- Flight_Distance: Density plot looks right-skewed, No Treatment required
- Dep_del_M: Density plot looks extremely right-skewed.
- Arr_Del_M: Density plot looks extremely right-skewed.

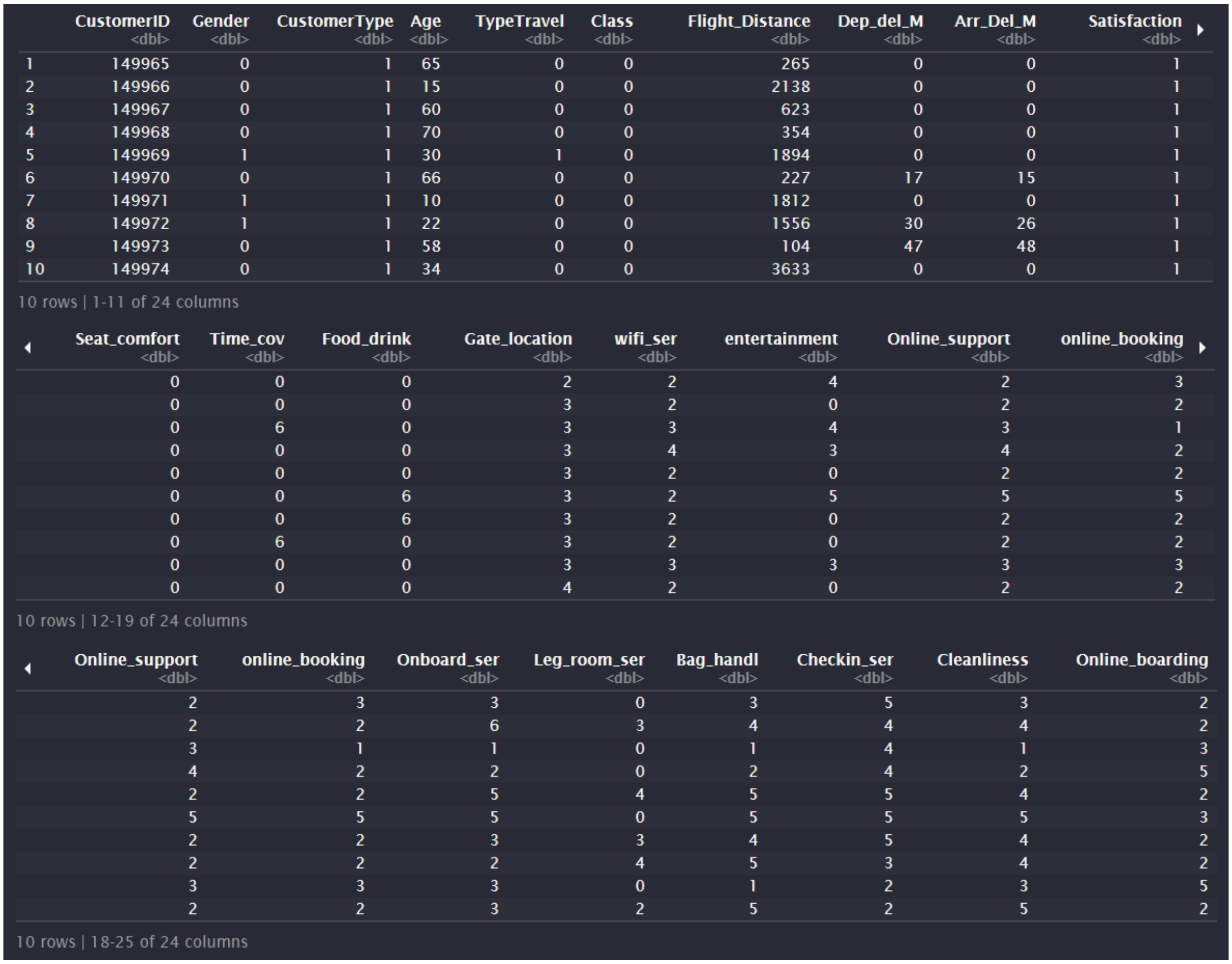
- 'Dep_del_M' and 'Arr_Del_M': are almost identical and normal Natural, No Treatment required



```
  1%    2%    3%   10%   20%   30%   40%   50%   60%   70%   80%   90%   95%   99%  100% 
   8    10    11    20    25    30    36    40    44    49    54    59    64    70    85 
  1%    2%    3%   10%   20%   30%   40%   50%   60%   70%   80%   90%   95%   99%  100% 
  95   223   261   544  1137  1509  1727  1927  2136  2389  2740  3398  3833  4816  6950 
  1%    2%    3%   10%   20%   30%   40%   50%   60%   70%   80%   90%   95%   99%  100% 
   0     0     0     0     0     0     0     0     2     8    18    43    76   180  1592 
  1%    2%    3%   10%   20%   30%   40%   50%   60%   70%   80%   90%   95%   99%  100% 
   0     0     0     0     0     0     0     0     2     9    19    44    77   181  1584 
```

Addition of New variables: No New addition required

Clean Data:



	CustomerID	Gender	CustomerType	Age	TypeTravel	Class	Flight_Distance	Dep_del_M	Arr_Del_M	Satisfaction
	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
1	149965	0	1	65	0	0	265	0	0	1
2	149966	0	1	15	0	0	2138	0	0	1
3	149967	0	1	60	0	0	623	0	0	1
4	149968	0	1	70	0	0	354	0	0	1
5	149969	1	1	30	1	0	1894	0	0	1
6	149970	0	1	66	0	0	227	17	15	1
7	149971	1	1	10	0	0	1812	0	0	1
8	149972	1	1	22	0	0	1556	30	26	1
9	149973	0	1	58	0	0	104	47	48	1
10	149974	0	1	34	0	0	3633	0	0	1

10 rows | 1-11 of 24 columns

Seat_comfort	Time_cov	Food_drink	Gate_location	wifi_ser	entertainment	Online_support	online_booking
<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
0	0	0	2	2	4	2	3
0	0	0	3	2	0	2	2
0	6	0	3	3	4	3	1
0	0	0	3	4	3	4	2
0	0	0	3	2	0	2	2
0	0	6	3	2	5	5	5
0	0	6	3	2	0	2	2
0	6	0	3	2	0	2	2
0	0	0	3	3	3	3	3
0	0	0	4	2	0	2	2

10 rows | 12-19 of 24 columns

Online_support	online_booking	Onboard_ser	Leg_room_ser	Bag_handl	Checkin_ser	Cleanliness	Online_boarding
<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
2	3	3	0	3	5	3	2
2	2	6	3	4	4	4	2
3	1	1	0	1	4	1	3
4	2	2	0	2	4	2	5
2	2	5	4	5	5	4	2
5	5	5	0	5	5	5	3
2	2	3	3	4	5	4	2
2	2	2	4	5	3	4	2
3	3	3	0	1	2	3	5
2	2	3	2	5	2	5	2

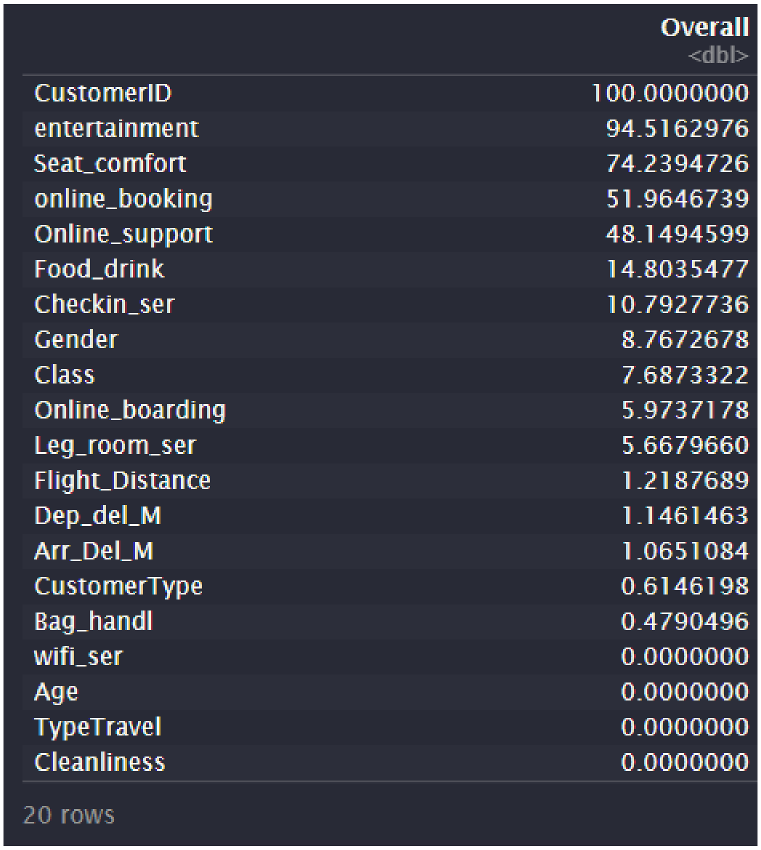
10 rows | 18-25 of 24 columns

Important variables: we apply “Single decision tree” and “Random Forest” and have the same result: top 5

1. CustomerID
2. Entertainment
3. Seat_comfort
4. Online_booking
5. Online_support

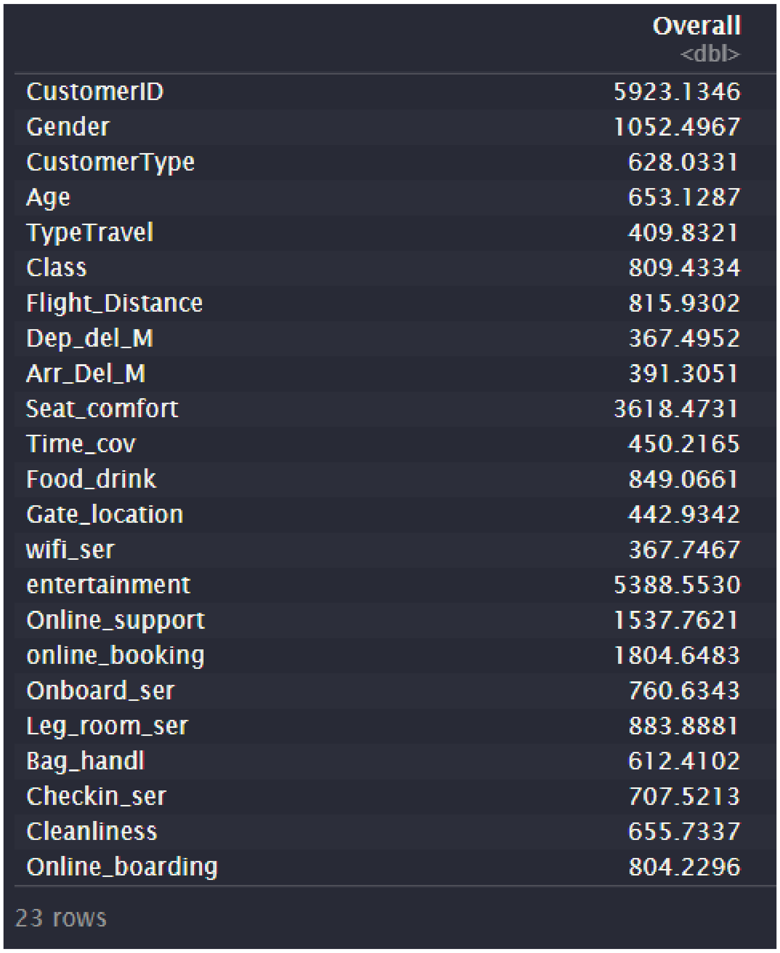
Sorted variables by important effect for customer to be satisfied, this will give as chance to keep satisfied customer by keeping up great of those variables.

Rpart: Single decision tree:



```
                           Overall
                             <dbl>
CustomerID             100.0000000
entertainment           94.5162976
Seat_comfort            74.2394726
online_booking          51.9646739
Online_support          48.1494599
Food_drink              14.8035477
Checkin_ser             10.7927736
Gender                   8.7672678
Class                    7.6873322
Online_boarding          5.9737178
Leg_room_ser             5.6679660
Flight_Distance          1.2187689
Dep_del_M                1.1461463
Arr_Del_M                1.0651084
CustomerType             0.6146198
Bag_handl                0.4790496
wifi_ser                 0.0000000
Age                      0.0000000
TypeTravel               0.0000000
Cleanliness              0.0000000
20 rows
```

Random Forest:

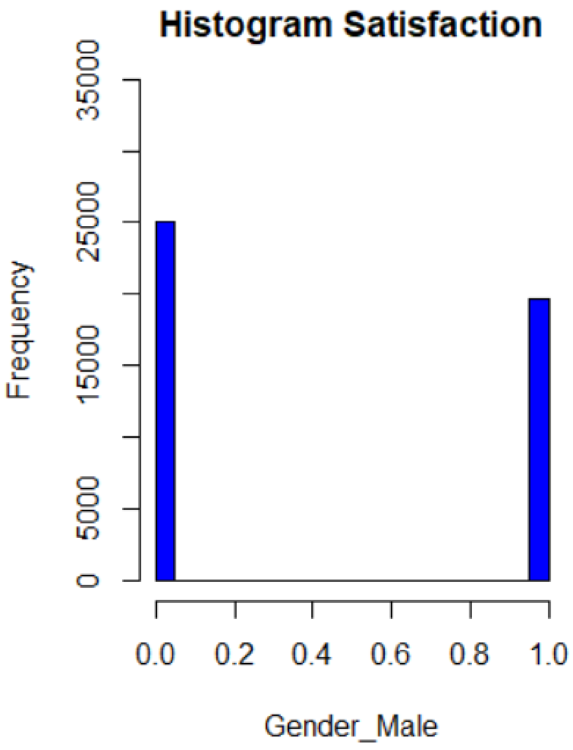


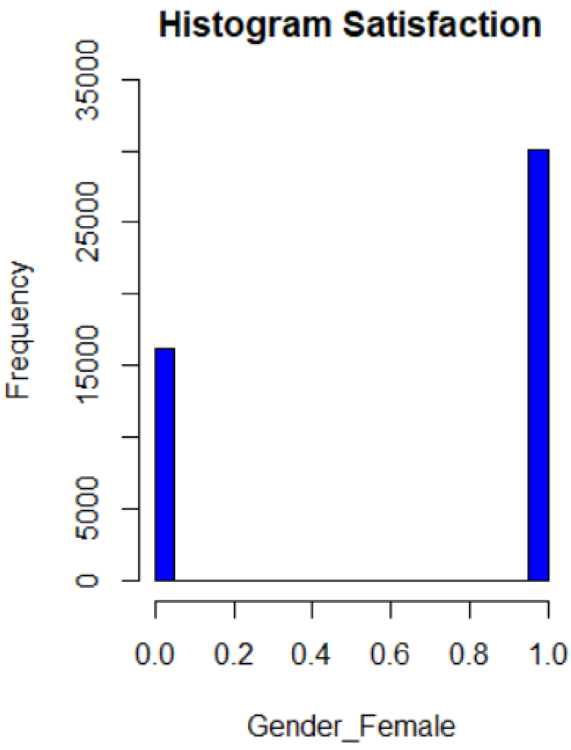
	Overall <dbl>
CustomerID	5923.1346
Gender	1052.4967
CustomerType	628.0331
Age	653.1287
TypeTravel	409.8321
Class	809.4334
Flight_Distance	815.9302
Dep_del_M	367.4952
Arr_Del_M	391.3051
Seat_comfort	3618.4731
Time_cov	450.2165
Food_drink	849.0661
Gate_location	442.9342
wifi_ser	367.7467
entertainment	5388.5530
Online_support	1537.7621
online_booking	1804.6483
Onboard_ser	760.6343
Leg_room_ser	883.8881
Bag_handl	612.4102
Checkin_ser	707.5213
Cleanliness	655.7337
Online_boarding	804.2296

23 rows

Visualizations:

Satisfaction VS Grender:

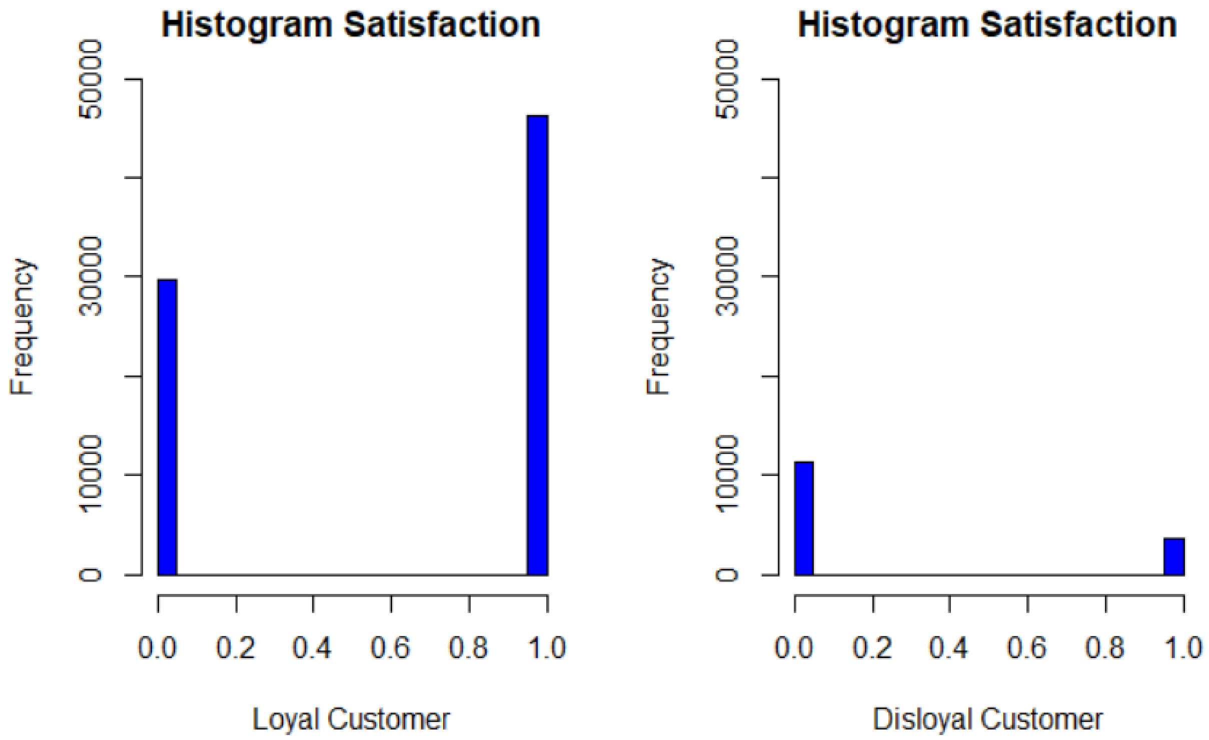




Observation:

- Female more Satisfied then male.
- Male are more dissatisfied.

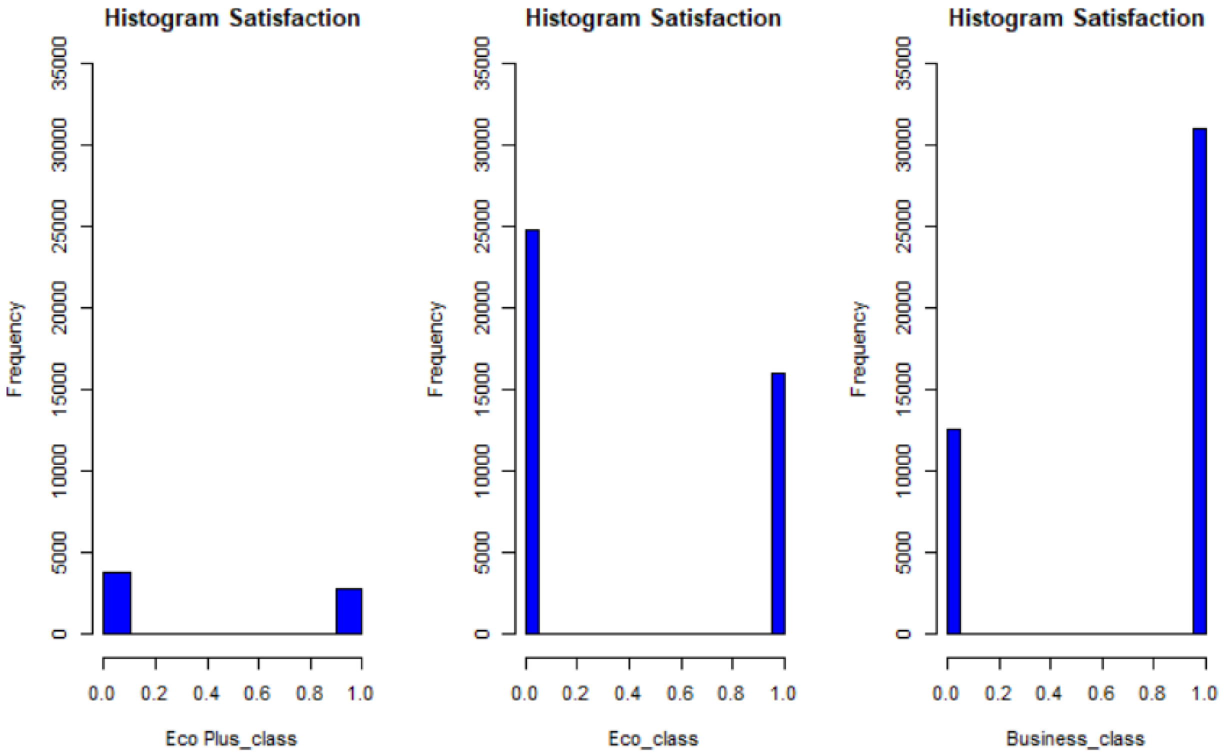
Satisfaction VS CustomerType:



Observation:

- Loyal customer are more Satisfied, almost half of loyal customer are dissatisfied.
- Disloyal Customer are more dissatisfied

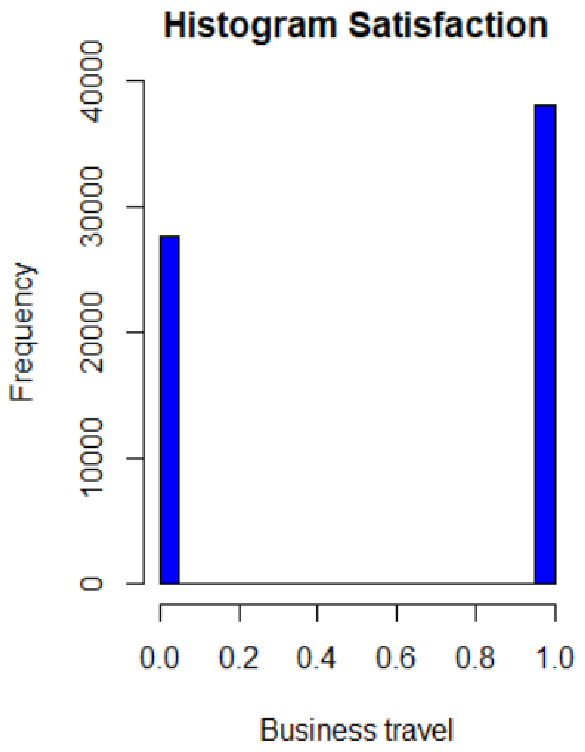
Satisfaction VS class:

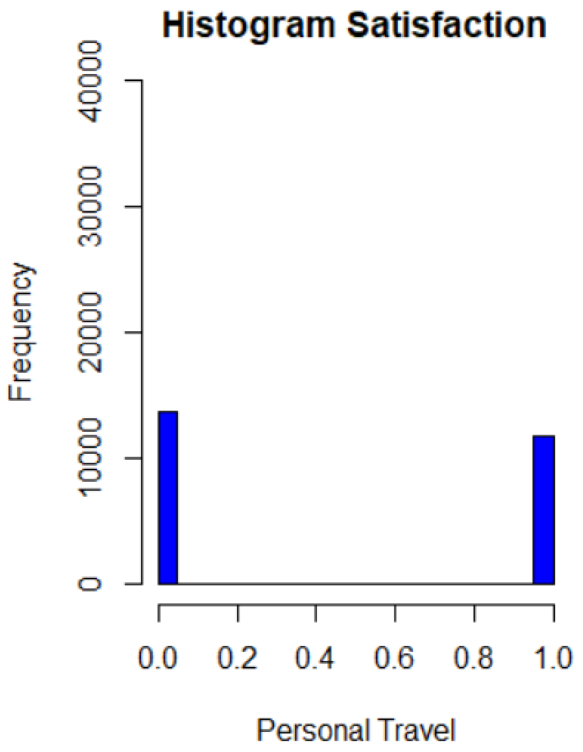


Observation:

- The majorty of Business_class are Satisfied but 1/3 are dissatisfied.
- The dissatisfied are more for Eco_class then Satisfied.
- Eco_Plus_class almost Satisfied same as dissatisfied.

Satisfaction VS TypeTravel:

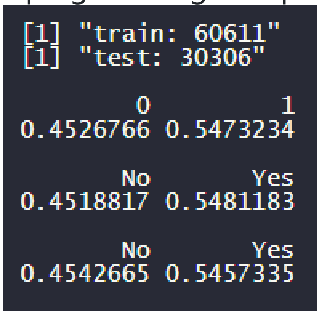




Observation:

- For Business_travel larg are Satisfied but a lot of are dissatisfied.
- Personal_Travel most likely same for dissatisfied and Satisfied.

Data split in to test and train: sample.split() 70% train data , 30% test data randomly with keeping the original split of 54% satisfied and 45% dissatisfied.



```
[1] "train: 60611"
[1] "test: 30306"

        0         1
0.4526766 0.5473234

       No       Yes
0.4518817 0.5481183

       No       Yes
0.4542665 0.5457335
```

Our target: the optimal Model: avoid over-fitting and under-fitting: we will apply multiple "7" models:

1. Single decision tree
2. Random forest
3. k-Nearest Neighbors
4. Naïve Bayes
5. Logistic Regression
6. Bagging
7. Xtreme Gradient boosting

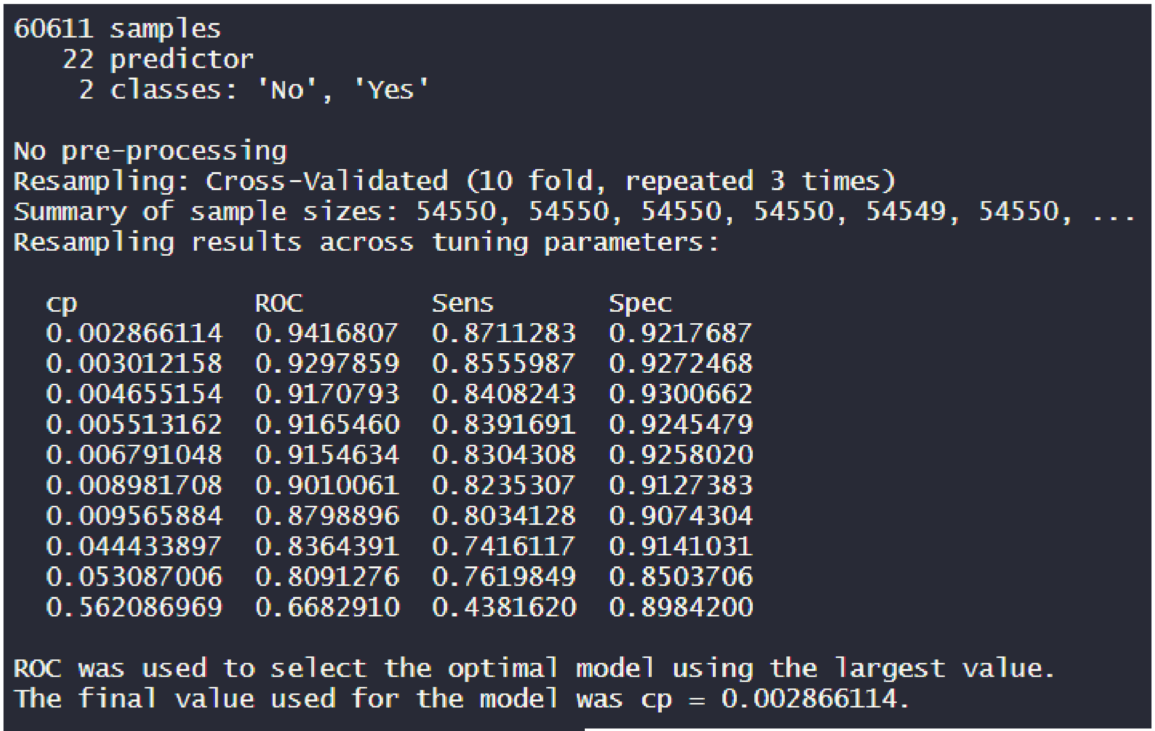
Setting general parameter for the best model:

1. **Random:** repeated random sub-sampling validation
2. **Cross-validation:** high-quality training for model to use all data of training
3. **Accuracy:** the highest accuracy
4. **Sensitivity :** percentage of all 1's were correctly predicted. The highest
5. **Specificity :** percentage of all 0's were correctly predicted. the highest
6. **Sensitivity and Specificity :** must be very close and very high
7. **Concordance:** the higher Concordance the better the model on cutoff value and easy for observation to be classified with very high prediction.
8. **ROC curve:** can be used to know what cutoff gives the best sensitivity, specificity or both. the highest
9. **Gini:** Coefficient is an indicator of how well the model outperforms random predictions. the highest
10. **KS:** used to make decisions like: How many customers to target for a marketing campaign? or How many customers should we pay for to show ads, also helps to understand?, **KS statistic** is the perfect portion of the population should be targeted to get the highest response rate. The highest

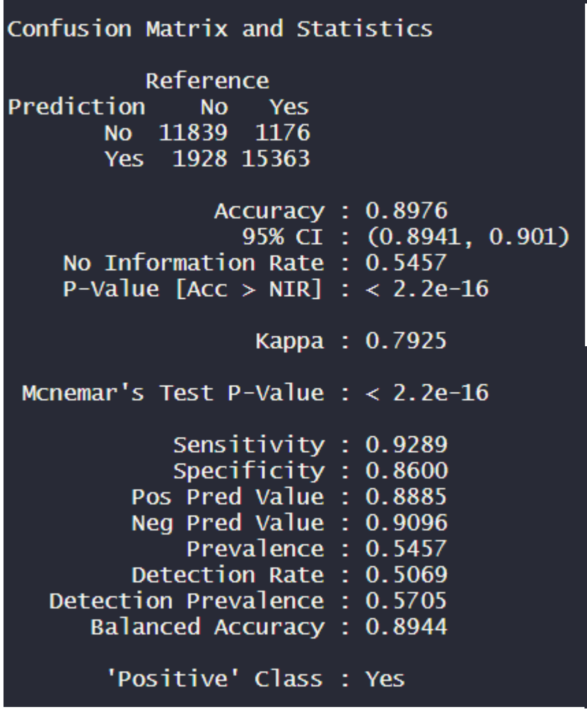
11. **Interpretability:** easy or possible

Modelling Process:

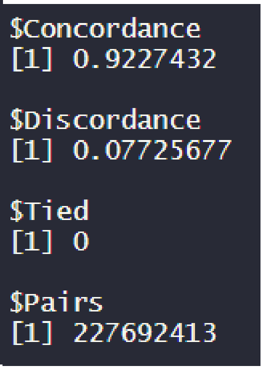
1- single decision tree:



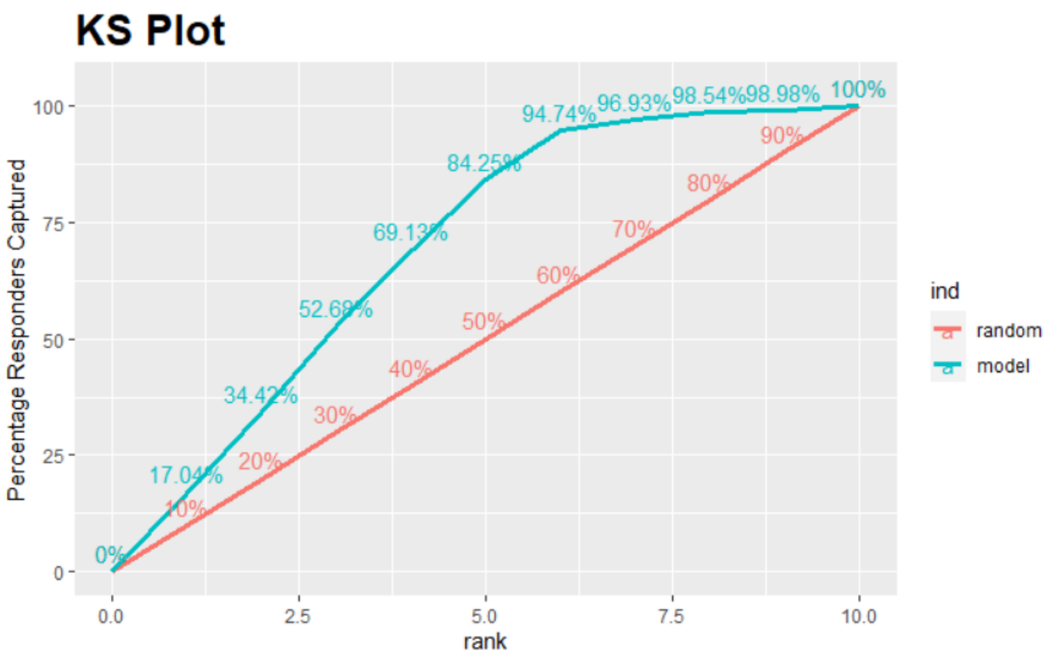
```
60611 samples
   22 predictor
    2 classes: 'No', 'Yes'

No pre-processing
Resampling: Cross-Validated (10 fold, repeated 3 times)
Summary of sample sizes: 54550, 54550, 54550, 54550, 54549, 54550, ...
Resampling results across tuning parameters:

  cp           ROC        Sens       Spec
  0.002866114  0.9416807  0.8711283  0.9217687
  0.003012158  0.9297859  0.8555987  0.9272468
  0.004655154  0.9170793  0.8408243  0.9300662
  0.005513162  0.9165460  0.8391691  0.9245479
  0.006791048  0.9154634  0.8304308  0.9258020
  0.008981708  0.9010061  0.8235307  0.9127383
  0.009565884  0.8798896  0.8034128  0.9074304
  0.044433897  0.8364391  0.7416117  0.9141031
  0.053087006  0.8091276  0.7619849  0.8503706
  0.562086969  0.6682910  0.4381620  0.8984200

ROC was used to select the optimal model using the largest value.
The final value used for the model was cp = 0.002866114.
```



```
Confusion Matrix and Statistics

          Reference
Prediction    No   Yes
       No  11839  1176
       Yes  1928 15363

               Accuracy : 0.8976
                 95% CI : (0.8941, 0.901)
    No Information Rate : 0.5457
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.7925

 Mcnemar's Test P-Value : < 2.2e-16

            Sensitivity : 0.9289
            Specificity : 0.8600
         Pos Pred Value : 0.8885
         Neg Pred Value : 0.9096
             Prevalence : 0.5457
         Detection Rate : 0.5069
   Detection Prevalence : 0.5705
      Balanced Accuracy : 0.8944

       'Positive' Class : Yes
```

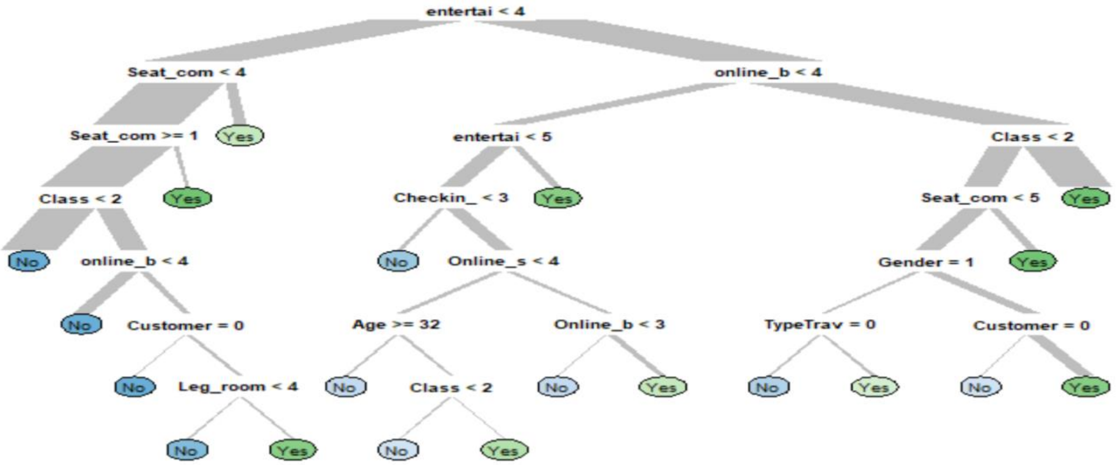


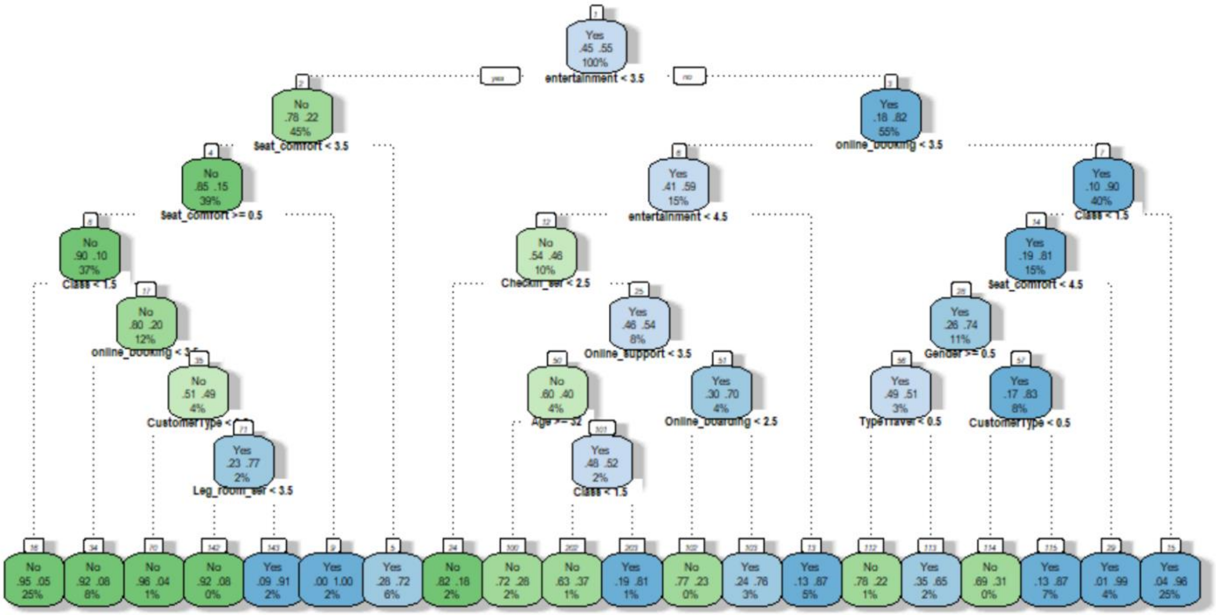
```
$Concordance
[1] 0.9227432

$Discordance
[1] 0.07725677

$Tied
[1] 0

$Pairs
[1] 227692413
```



Observe：

- accuracy for single decision tree: 89.7%.
- some difference between Sensitivity and Specificity.
- concordance is 92.27%, Probability of (Right) is 92.27% which is Good.
- discordance is 7.7%.
- the area under the curve: ROC : 93.98%.
- Gini : 39.88%.
- Kolomogorov-Smirnov :KS statistic : 76.46%.
- 60% of data give us 95% respond with this model.
- ROC reaches 1 when complexity parameter reaches 0.2 .
- The final value used for the optimal model was cp = 0.002866114

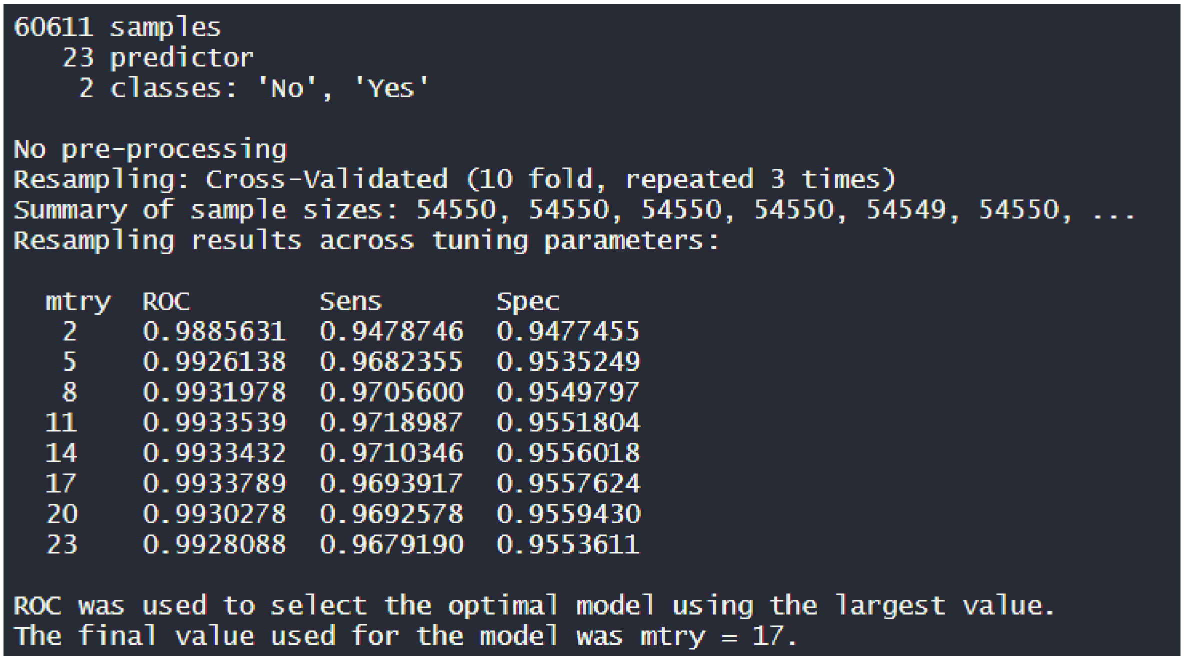




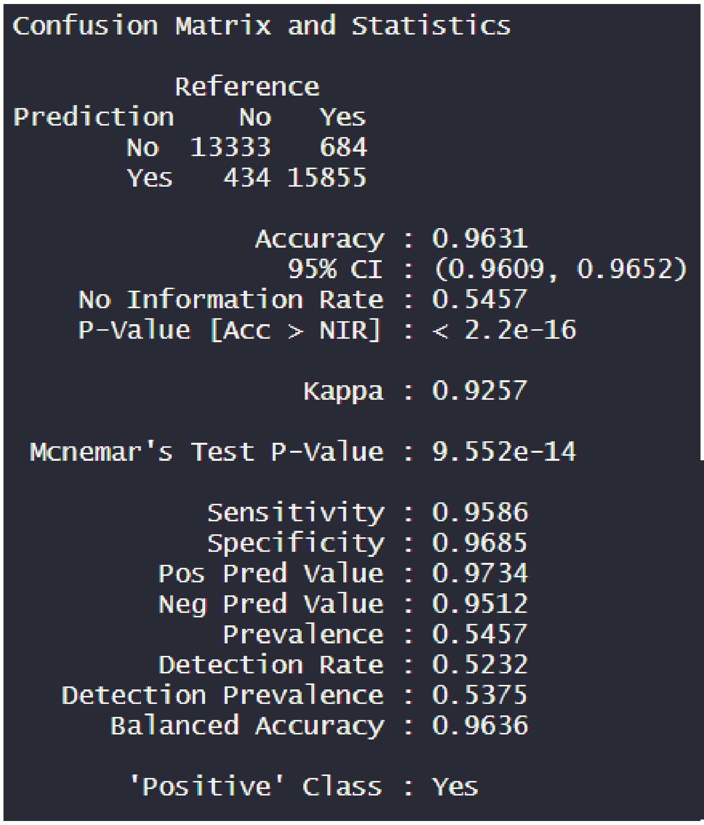
Observe:

- 25% of customer satisfied have (entertainment< 3.5 ,online_booking>=3.5, Class< 1.5)
- 25% of customer dissatisfied have (entertainment< 3.5 , Seat_comfort< 3.5, Seat_comfort>=0.5, Class< 1.5)

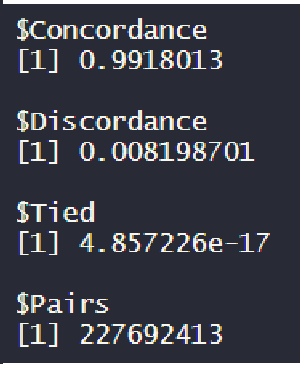
2- Random Forest:



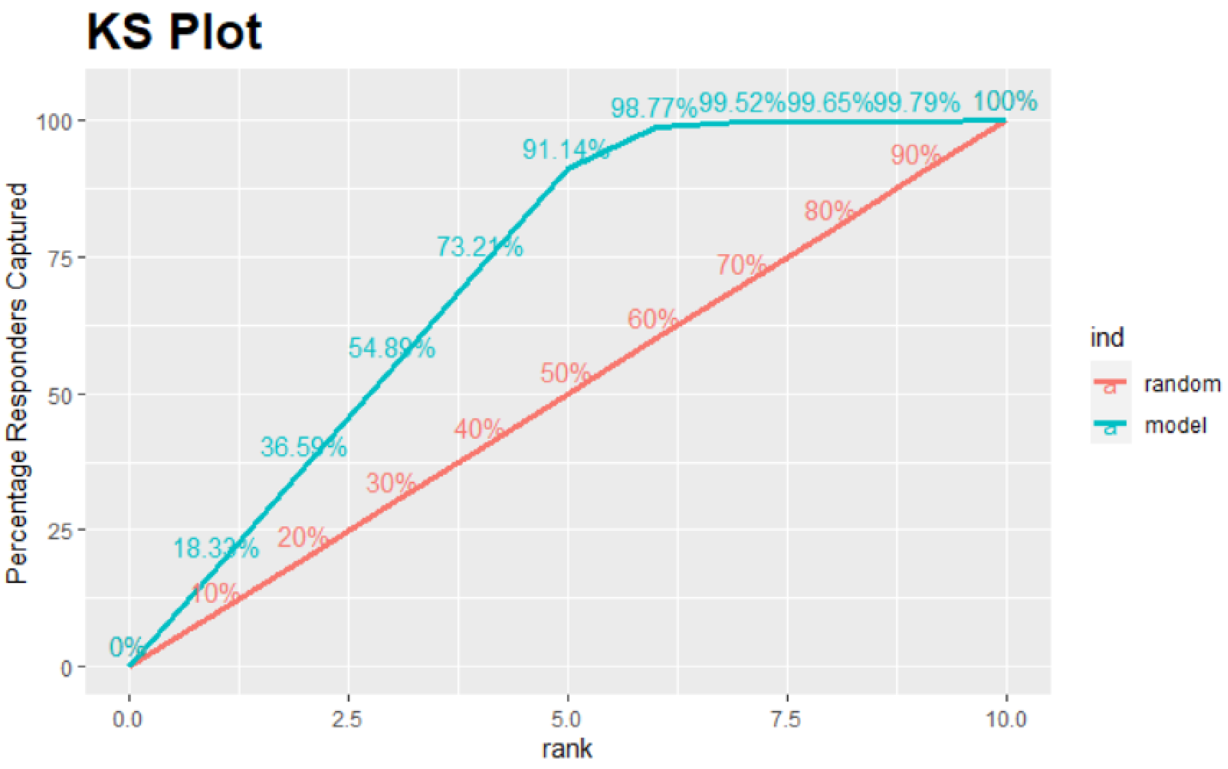
```
60611 samples
   23 predictor
    2 classes: 'No', 'Yes'

No pre-processing
Resampling: Cross-Validated (10 fold, repeated 3 times)
Summary of sample sizes: 54550, 54550, 54550, 54550, 54549, 54550, ...
Resampling results across tuning parameters:

  mtry  ROC        Sens       Spec
   2    0.9885631  0.9478746  0.9477455
   5    0.9926138  0.9682355  0.9535249
   8    0.9931978  0.9705600  0.9549797
  11    0.9933539  0.9718987  0.9551804
  14    0.9933432  0.9710346  0.9556018
  17    0.9933789  0.9693917  0.9557624
  20    0.9930278  0.9692578  0.9559430
  23    0.9928088  0.9679190  0.9553611

ROC was used to select the optimal model using the largest value.
The final value used for the model was mtry = 17.
```

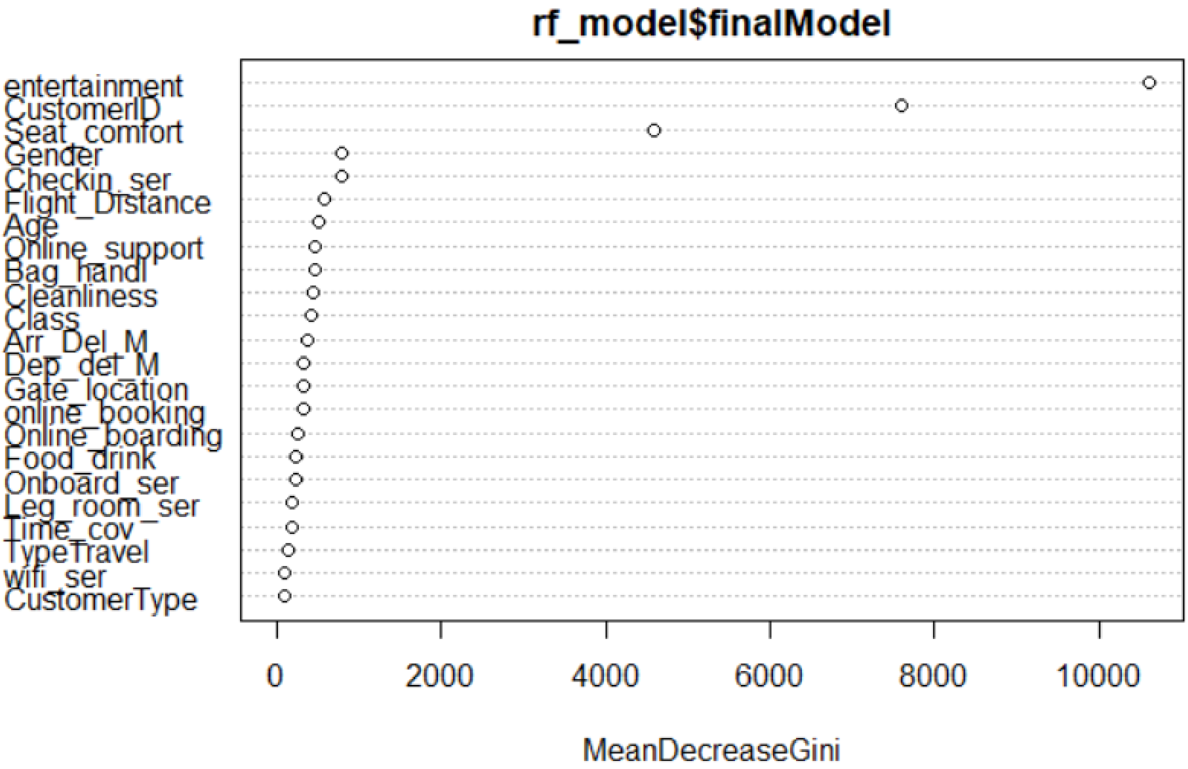


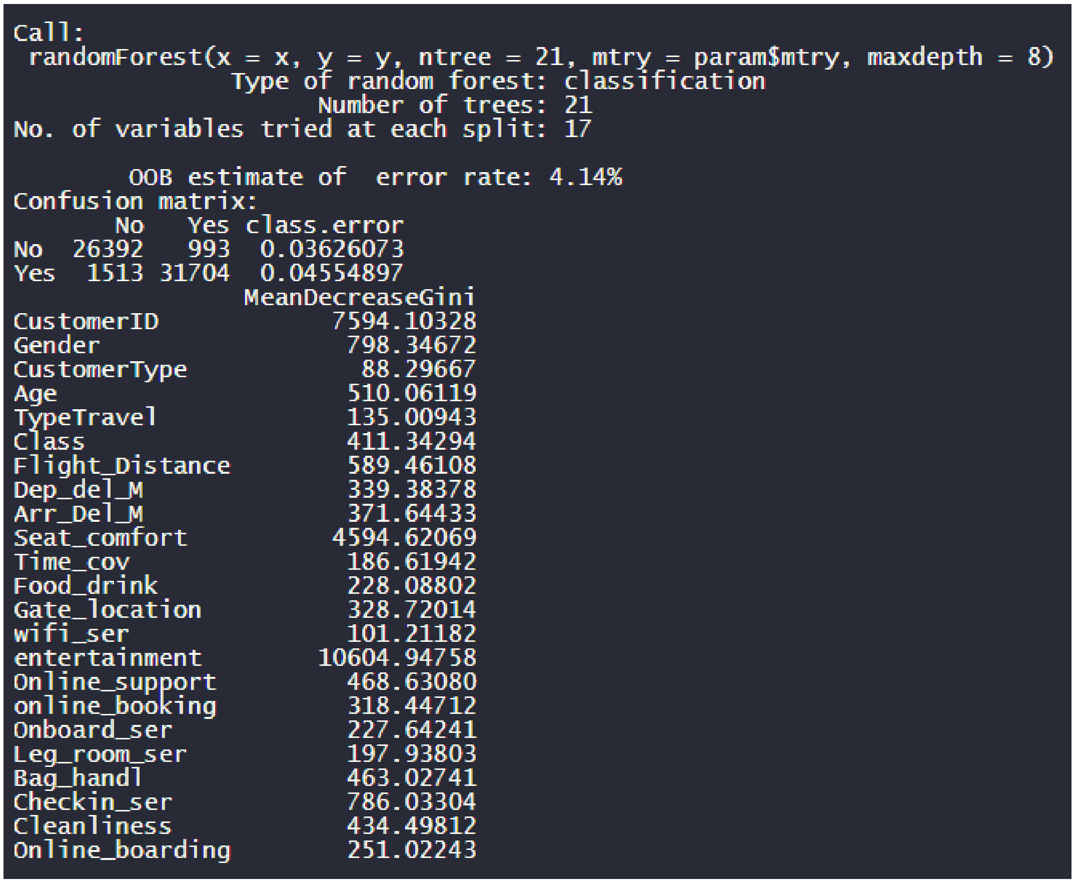
```
Confusion Matrix and Statistics

          Reference
Prediction    No   Yes
       No  13333   684
       Yes   434 15855

               Accuracy : 0.9631
                 95% CI : (0.9609, 0.9652)
    No Information Rate : 0.5457
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.9257

 Mcnemar's Test P-Value : 9.552e-14

            Sensitivity : 0.9586
            Specificity : 0.9685
         Pos Pred Value : 0.9734
         Neg Pred Value : 0.9512
             Prevalence : 0.5457
         Detection Rate : 0.5232
   Detection Prevalence : 0.5375
      Balanced Accuracy : 0.9636

       'Positive' Class : Yes
```



```
$Concordance
[1] 0.9918013

$Discordance
[1] 0.008198701

$Tied
[1] 4.857226e-17

$Pairs
[1] 227692413
```


KS Plot
Percentage Responders Captured
100
75
50
25
0
0.0
2.5
5.0
7.5
10.0
rank
ind
random
model
0%
18.33%
36.59%
54.89%
73.21%
91.14%
98.77%
99.52%
99.65%
99.79%
100%
10%
20%
30%
40%
50%
60%
70%
80%
90%


rf_model$finalModel
entertainment
CustomerID
Seat_comfort
Gender
Checkin_ser
Flight_Distance
Age
Online_support
Bag_handl
Cleanliness
Class
Arr_Del_M
Dep_del_M
Gate_location
online_booking
Online_boarding
Food_drink
Onboard_ser
Leg_room_ser
Time_cov
TypeTravel
wifi_ser
CustomerType
0
2000
4000
6000
8000
10000
MeanDecreaseGini



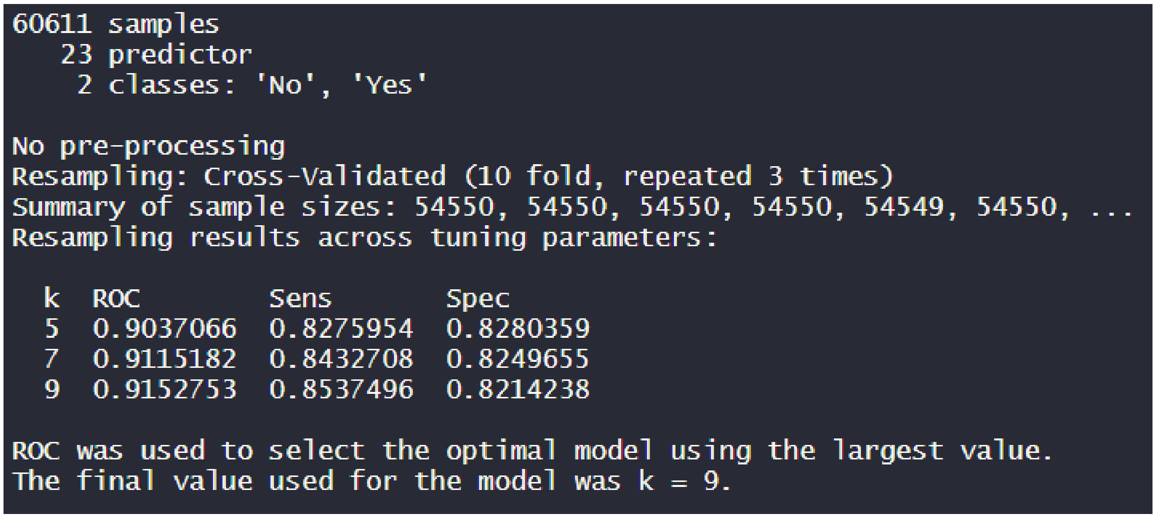
```
Call:
 randomForest(x = x, y = y, ntree = 21, mtry = param$mtry, maxdepth = 8)
               Type of random forest: classification
                     Number of trees: 21
No. of variables tried at each split: 17

        OOB estimate of  error rate: 4.14%
Confusion matrix:
       No   Yes class.error
No  26392   993  0.03626073
Yes  1513 31704  0.04554897
                MeanDecreaseGini
CustomerID            7594.10328
Gender                 798.34672
CustomerType            88.29667
Age                    510.06119
TypeTravel             135.00943
Class                  411.34294
Flight_Distance        589.46108
Dep_del_M              339.38378
Arr_Del_M              371.64433
Seat_comfort          4594.62069
Time_cov               186.61942
Food_drink             228.08802
Gate_location          328.72014
wifi_ser               101.21182
entertainment        10604.94758
Online_support         468.63080
online_booking         318.44712
Onboard_ser            227.64241
Leg_room_ser           197.93803
Bag_handl              463.02741
Checkin_ser            786.03304
Cleanliness            434.49812
Online_boarding        251.02243
```

Observe:

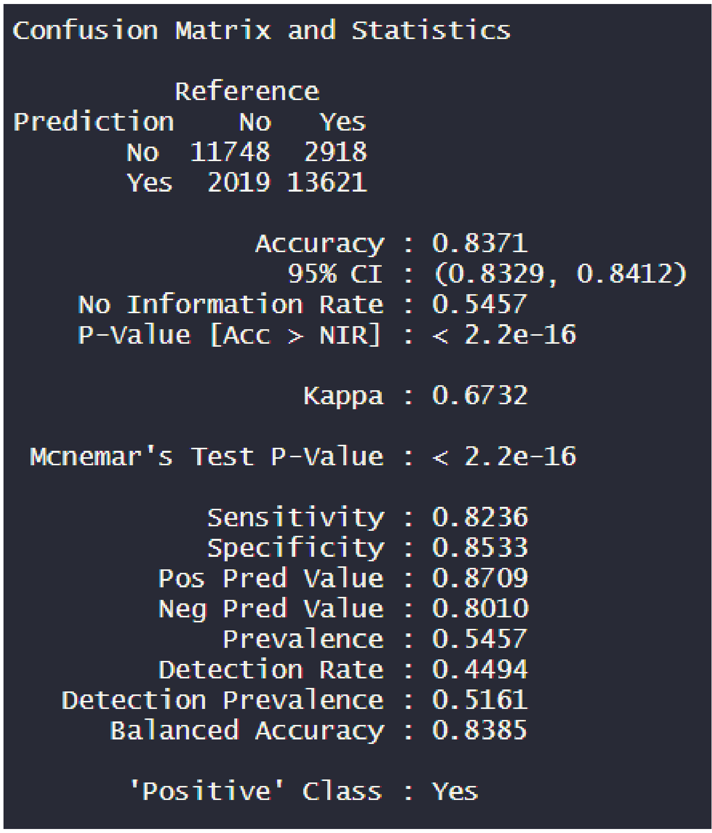
- accuracy for random forest : 96.31%.
- small difference between Sensitivity and Specificity.
- concordance is 99.18%, Probability of (Right) is 99.18% which is Good.
- discordance is 0.8%.
- the area under the curve: ROC : 99.34%.
- Gini : 44.5%.
- Kolomogorov-Smirnov :KS statistic : 90.78%.
- 60% of data give us 98.86% respond with this model.
- OOB estimate of error rate: 4.14%.
- The final value used for the optimal model was mtry = 17
- The most important variables: [entertainment, CustomerID, Seat_comfort]

Remark:
It took very long time to train 21 number of tree and give those results, figuring out the right number of tree will increase the accuracy of the model, unfortunately I don't have enough compute power, knowing my laptop is : core i7 8th Gen , 8GB RAM.

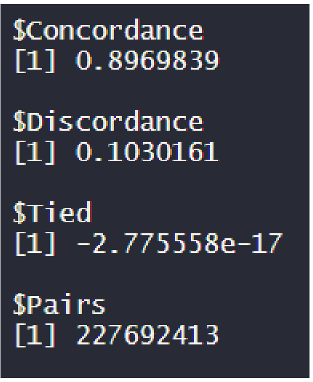
3- KNN: k-Nearest Neighbors



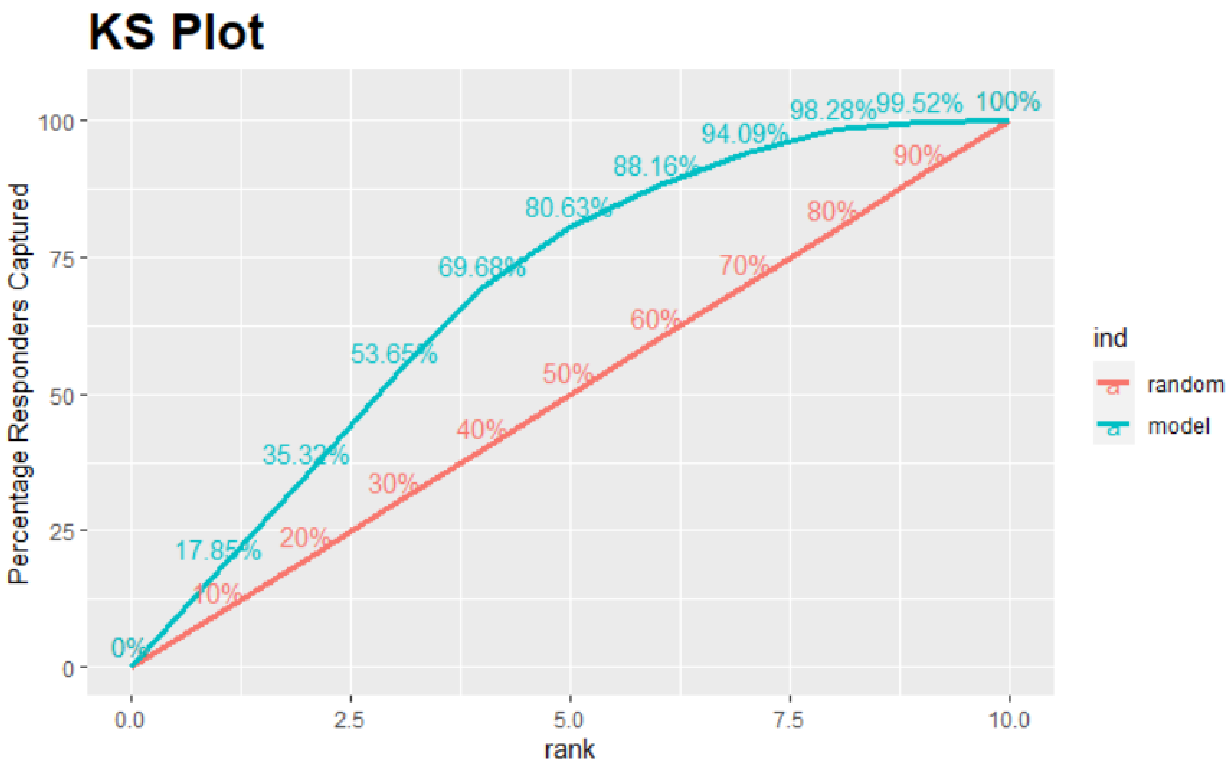
```
60611 samples
   23 predictor
    2 classes: 'No', 'Yes'

No pre-processing
Resampling: Cross-Validated (10 fold, repeated 3 times)
Summary of sample sizes: 54550, 54550, 54550, 54550, 54549, 54550, ...
Resampling results across tuning parameters:

  k  ROC        Sens       Spec
  5  0.9037066  0.8275954  0.8280359
  7  0.9115182  0.8432708  0.8249655
  9  0.9152753  0.8537496  0.8214238

ROC was used to select the optimal model using the largest value.
The final value used for the model was k = 9.
```



```
Confusion Matrix and Statistics

          Reference
Prediction    No   Yes
       No  11748  2918
       Yes  2019 13621

               Accuracy : 0.8371
                 95% CI : (0.8329, 0.8412)
    No Information Rate : 0.5457
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.6732

 Mcnemar's Test P-Value : < 2.2e-16

            Sensitivity : 0.8236
            Specificity : 0.8533
         Pos Pred Value : 0.8709
         Neg Pred Value : 0.8010
             Prevalence : 0.5457
         Detection Rate : 0.4494
   Detection Prevalence : 0.5161
      Balanced Accuracy : 0.8385

       'Positive' Class : Yes
```



```
$Concordance
[1] 0.8969839

$Discordance
[1] 0.1030161

$Tied
[1] -2.775558e-17

$Pairs
[1] 227692413
```

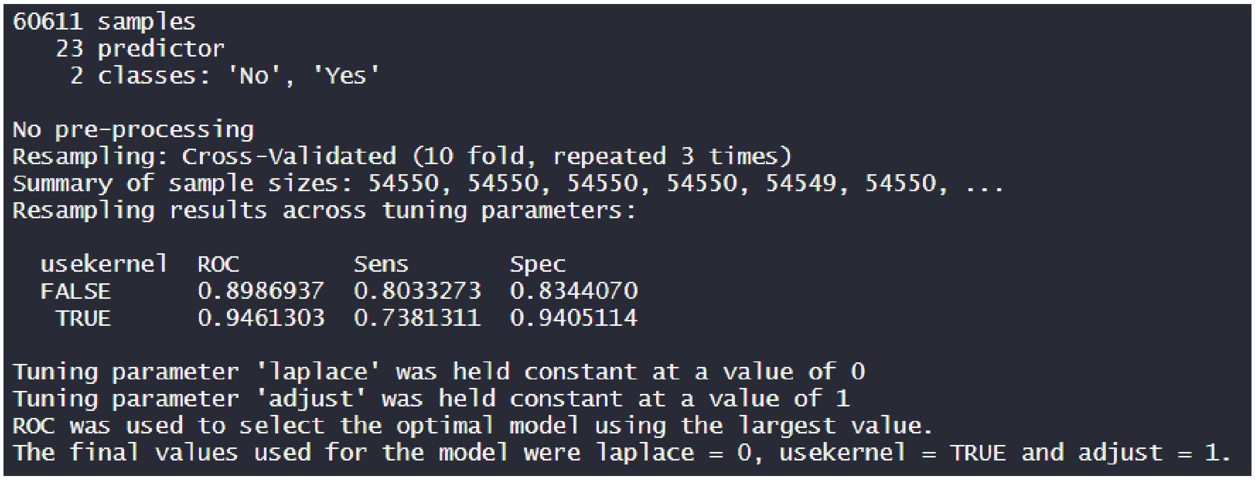


Observe:

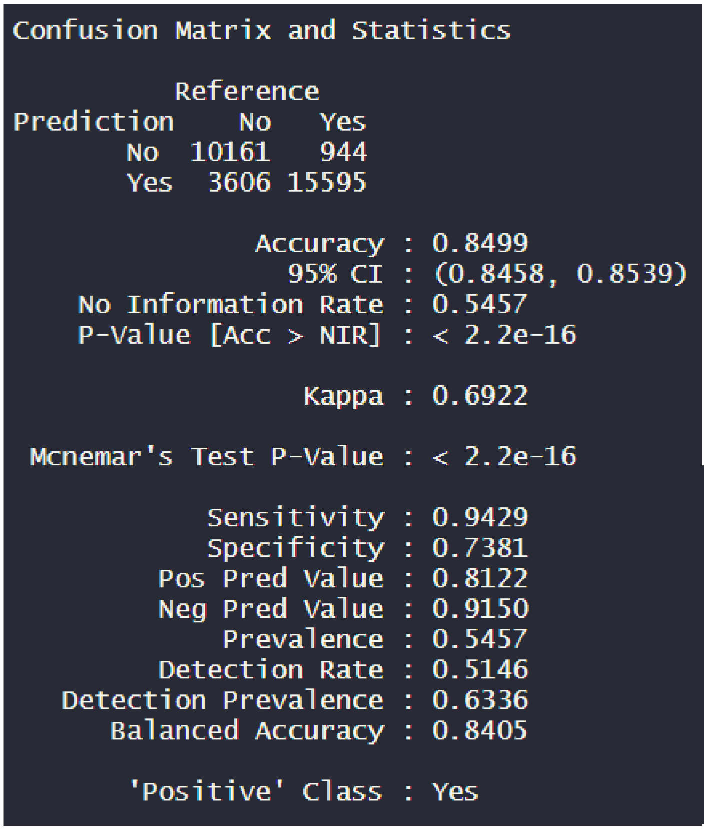
- accuracy for KNN : 83.7%.
- very small difference between Sensitivity and Specificity.
- concordance is 89.6%, Probability of (Right) is 86.9% which is acceptable.
- discordance is 10.3%.
- the area under the curve: ROC : 91.67%.
- Gini : 39.95%.
- Kolomogorov-Smirnov :KS statistic : 67.42%.
- 90% of data give us 99.5% respond with this model.
- ROC reaches 1 when complexity parameter reaches 0.6 .
- The final value used for the optimal model was k = 9

Remark:
It took very long time to train

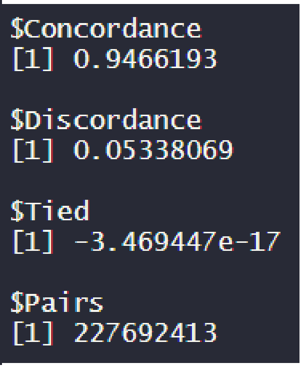
4 - Naive Bayes:



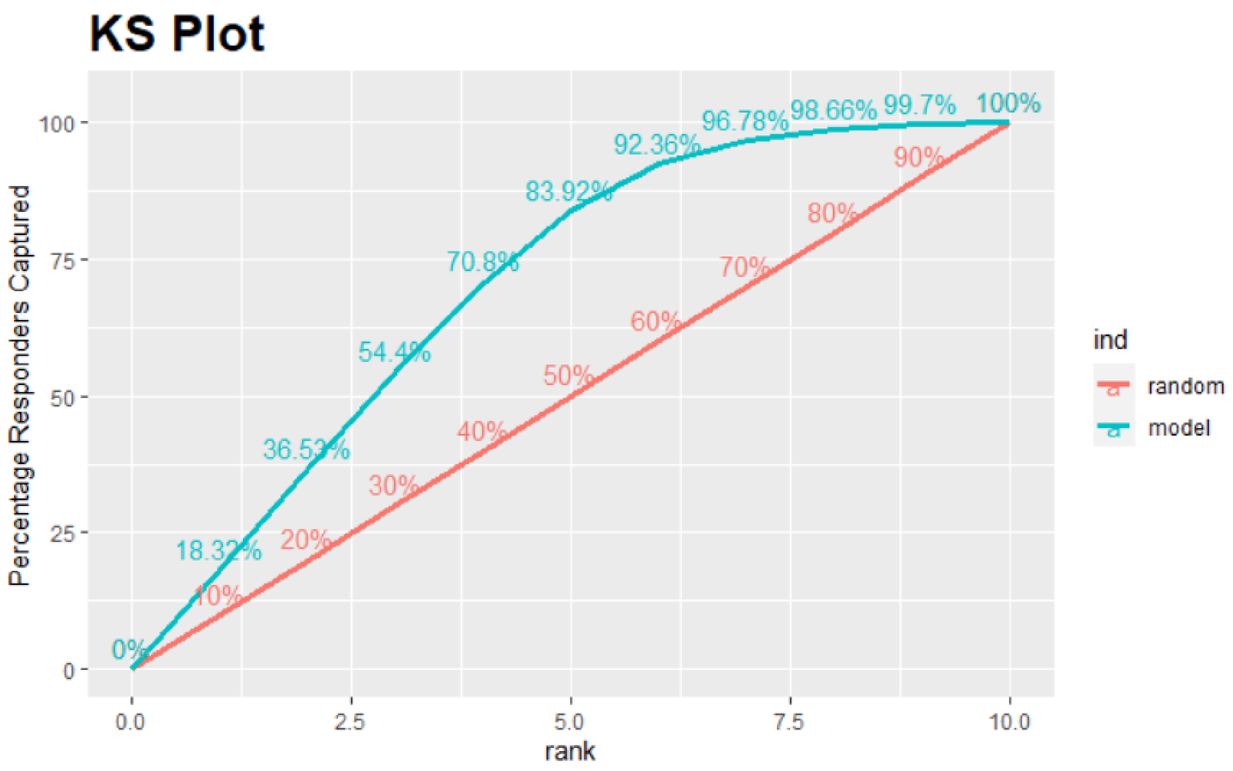
```
60611 samples
   23 predictor
    2 classes: 'No', 'Yes'

No pre-processing
Resampling: Cross-Validated (10 fold, repeated 3 times)
Summary of sample sizes: 54550, 54550, 54550, 54550, 54549, 54550, ...
Resampling results across tuning parameters:

  usekernel  ROC        Sens       Spec
  FALSE      0.8986937  0.8033273  0.8344070
   TRUE      0.9461303  0.7381311  0.9405114

Tuning parameter 'laplace' was held constant at a value of 0
Tuning parameter 'adjust' was held constant at a value of 1
ROC was used to select the optimal model using the largest value.
The final values used for the model were laplace = 0, usekernel = TRUE and adjust = 1.
```



```
Confusion Matrix and Statistics

          Reference
Prediction    No   Yes
       No  10161   944
       Yes  3606 15595

               Accuracy : 0.8499
                 95% CI : (0.8458, 0.8539)
    No Information Rate : 0.5457
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.6922

 Mcnemar's Test P-Value : < 2.2e-16

            Sensitivity : 0.9429
            Specificity : 0.7381
         Pos Pred Value : 0.8122
         Neg Pred Value : 0.9150
             Prevalence : 0.5457
         Detection Rate : 0.5146
   Detection Prevalence : 0.6336
      Balanced Accuracy : 0.8405

       'Positive' Class : Yes
```



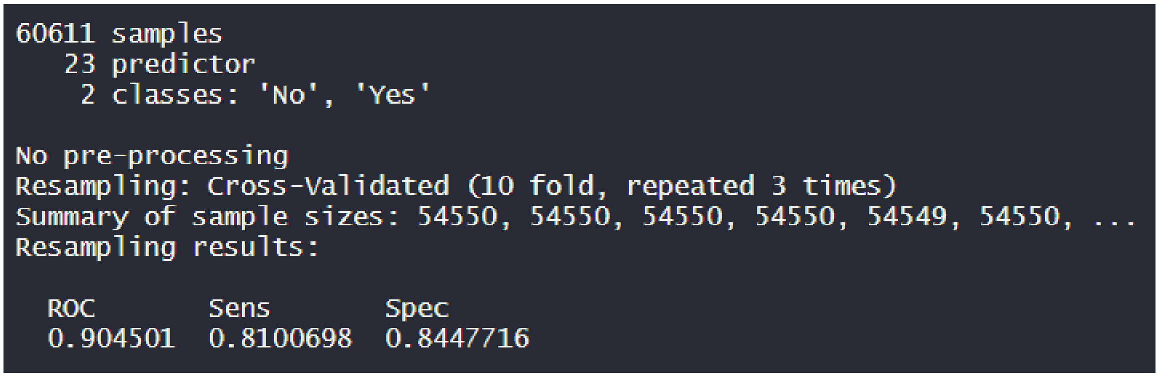
```
$Concordance
[1] 0.9466193

$Discordance
[1] 0.05338069

$Tied
[1] -3.469447e-17

$Pairs
[1] 227692413
```



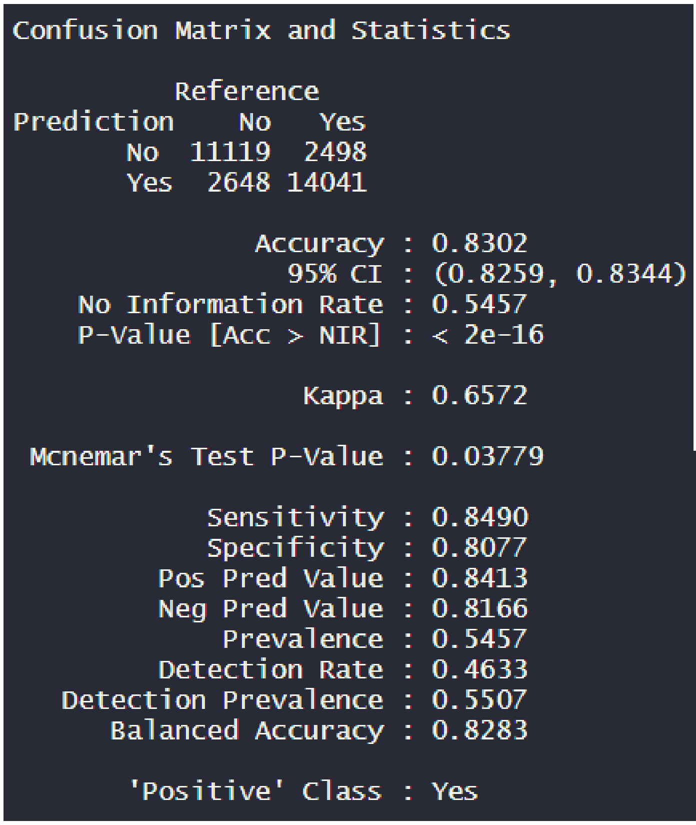
Observe:

- accuracy for naïve bayes : 84.99%.
- large difference between Sensitivity and Specificity.
- concordance is 94.6%, Probability of (Right) is 96.6% which is Good.
- discordance is 5.3%.
- the area under the curve: ROC : 94.66%.
- Gini : 35.69%.
- Kolomogorov-Smirnov : KS statistic : 74.66%.
- 90% of data give us 99.7% respond with this model.
- ROC reaches 1 when complexity parameter reaches 0.5 .

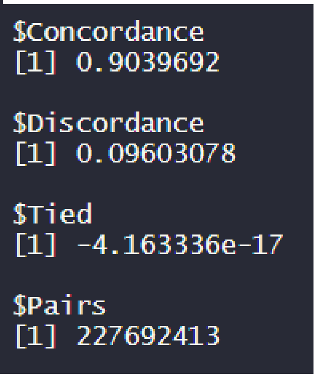
5 - Logistic Regression:



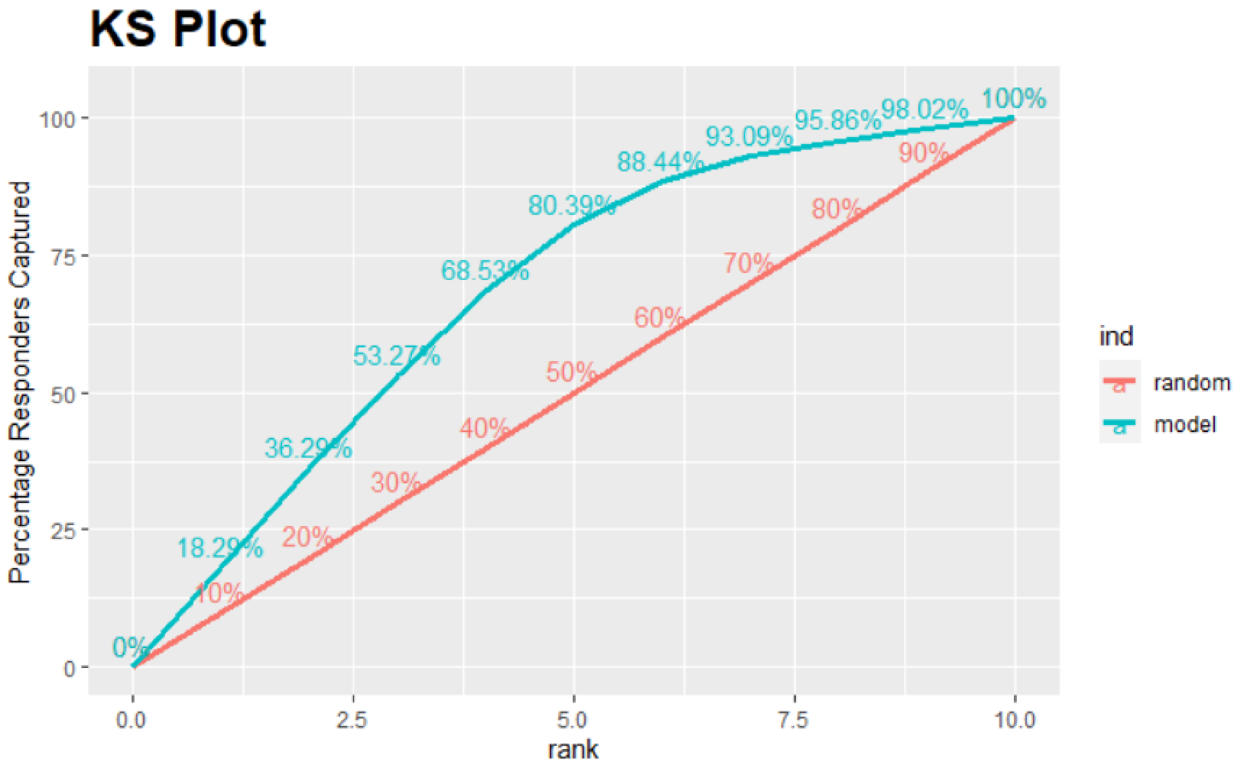
```
60611 samples
   23 predictor
    2 classes: 'No', 'Yes'

No pre-processing
Resampling: Cross-Validated (10 fold, repeated 3 times)
Summary of sample sizes: 54550, 54550, 54550, 54550, 54549, 54550, ...
Resampling results:

  ROC       Sens       Spec     
  0.904501  0.8100698  0.8447716
```



```
Confusion Matrix and Statistics

          Reference
Prediction    No   Yes
       No  11119  2498
       Yes  2648 14041
                                          
               Accuracy : 0.8302          
                 95% CI : (0.8259, 0.8344)
    No Information Rate : 0.5457          
    P-Value [Acc > NIR] : < 2e-16         
                                          
                  Kappa : 0.6572          
                                          
 Mcnemar's Test P-Value : 0.03779         
                                          
            Sensitivity : 0.8490          
            Specificity : 0.8077          
         Pos Pred Value : 0.8413          
         Neg Pred Value : 0.8166          
             Prevalence : 0.5457          
         Detection Rate : 0.4633          
   Detection Prevalence : 0.5507          
      Balanced Accuracy : 0.8283          
                                          
       'Positive' Class : Yes             
```



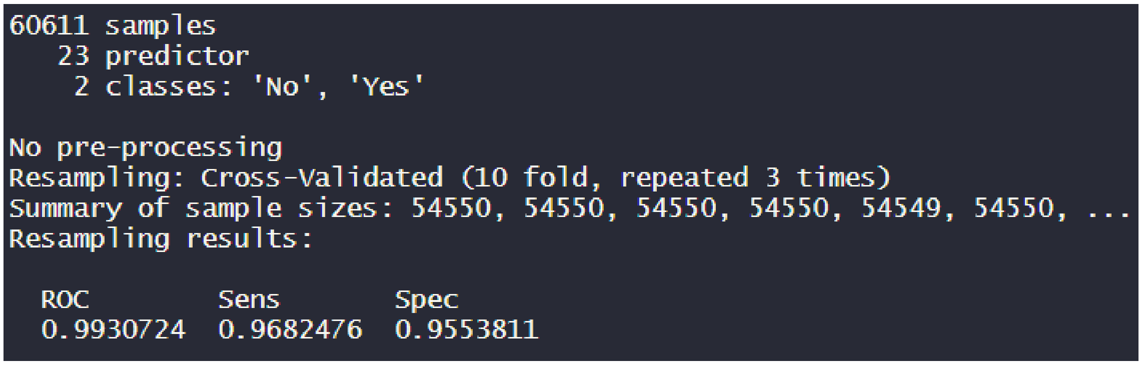
```
$Concordance
[1] 0.9039692

$Discordance
[1] 0.09603078

$Tied
[1] -4.163336e-17

$Pairs
[1] 227692413
```



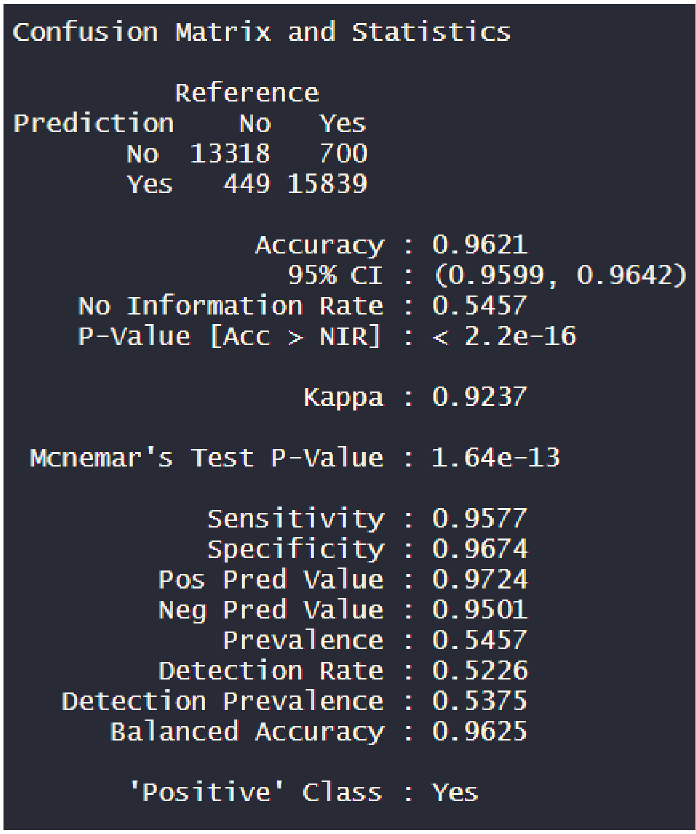
Observe:

- accuracy for Logistic Regression: 83.02%.
- small difference between Sensitivity and Specificity.
- concordance is 90.39%, Probability of (Right) is 90.39% which is Good.
- discordance is 9.6%.
- the area under the curve: ROC : 90.39 %.
- Gini : 36.42 %.
- Kolomogorov-Smirnov : KS statistic : 66.88%.
- 90% of data give us 98.02% respond with this model.
- ROC reaches 0.9 when complexity parameter reaches 0.5 .

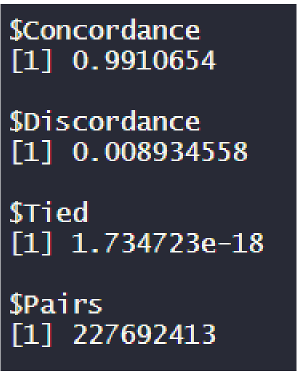
6 – Bagging:



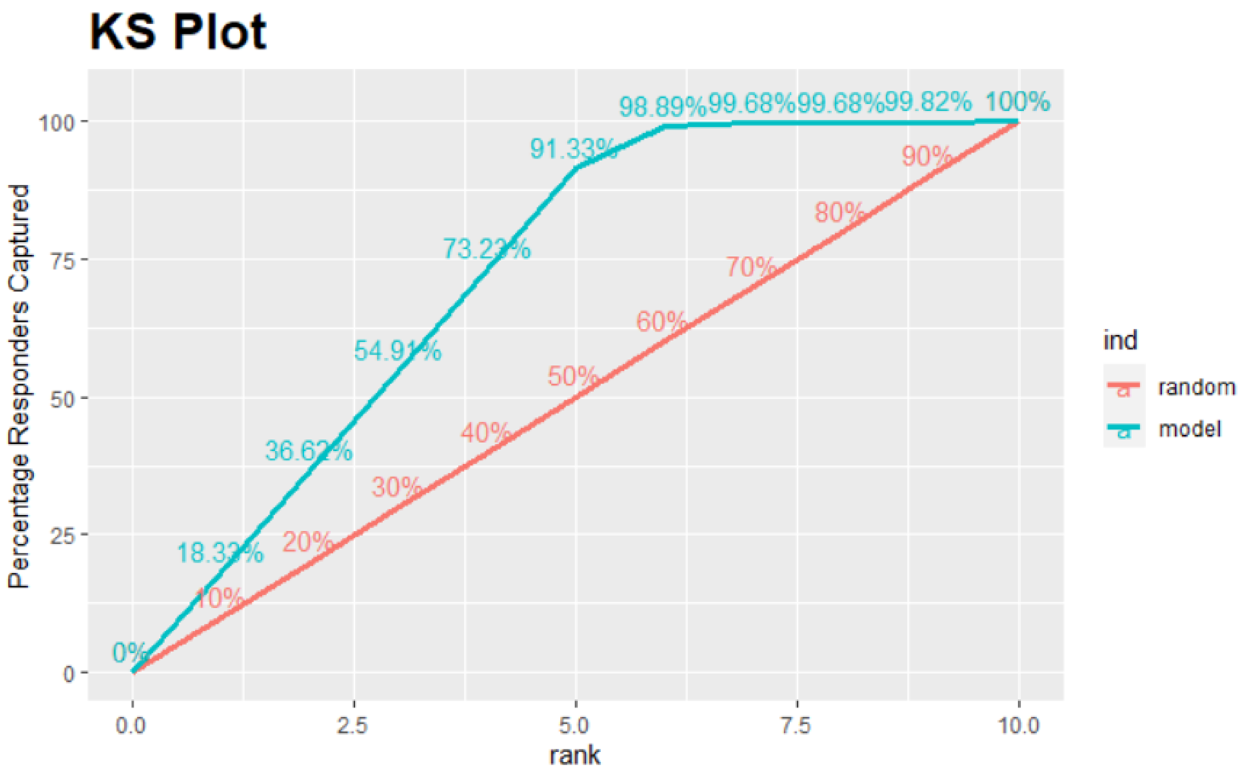
```
60611 samples
   23 predictor
    2 classes: 'No', 'Yes'

No pre-processing
Resampling: Cross-Validated (10 fold, repeated 3 times)
Summary of sample sizes: 54550, 54550, 54550, 54550, 54549, 54550, ...
Resampling results:

  ROC        Sens       Spec     
  0.9930724  0.9682476  0.9553811
```



```
Confusion Matrix and Statistics

          Reference
Prediction    No   Yes
       No  13318   700
       Yes   449 15839
                                          
               Accuracy : 0.9621          
                 95% CI : (0.9599, 0.9642)
    No Information Rate : 0.5457          
    P-Value [Acc > NIR] : < 2.2e-16       
                                          
                  Kappa : 0.9237          
                                          
 Mcnemar's Test P-Value : 1.64e-13        
                                          
            Sensitivity : 0.9577          
            Specificity : 0.9674          
         Pos Pred Value : 0.9724          
         Neg Pred Value : 0.9501          
             Prevalence : 0.5457          
         Detection Rate : 0.5226          
   Detection Prevalence : 0.5375          
      Balanced Accuracy : 0.9625          
                                          
       'Positive' Class : Yes             
```



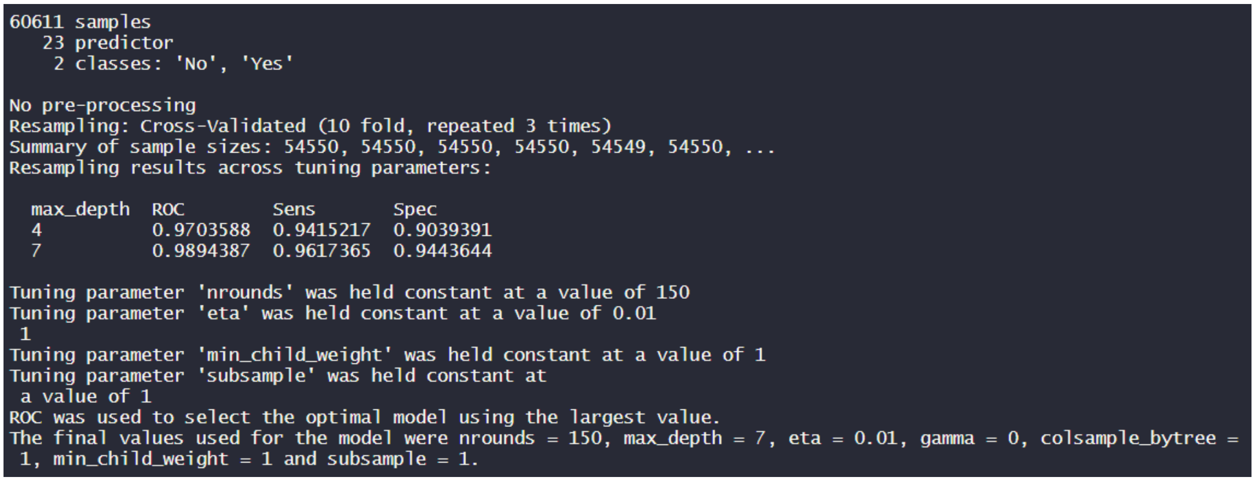
```
$Concordance
[1] 0.9910654

$Discordance
[1] 0.008934558

$Tied
[1] 1.734723e-18

$Pairs
[1] 227692413
```



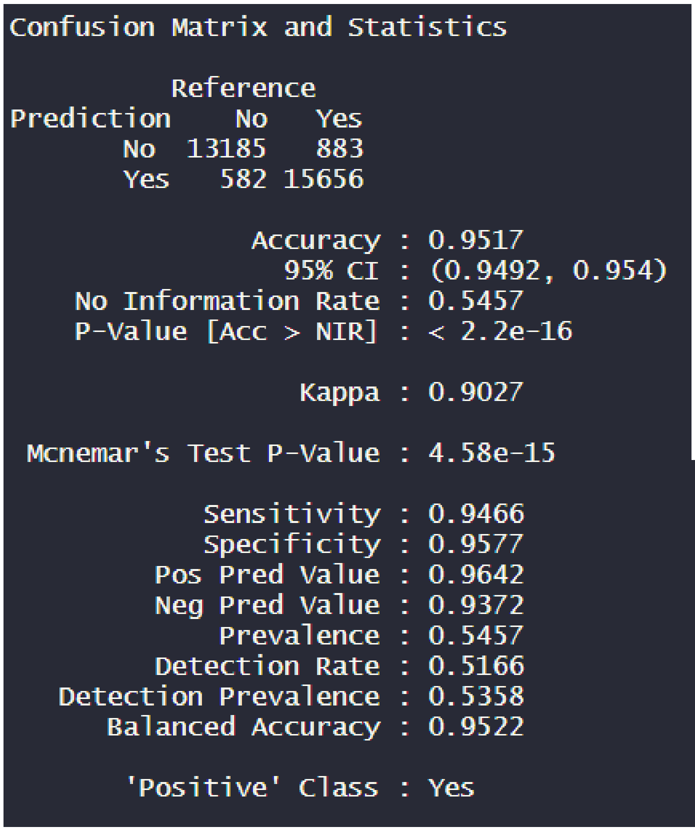
Observe:

- accuracy for bagging: 96.21%.
- very small difference between Sensitivity and Specificity.
- concordance is 99.10%, Probability of (Right) is 99.10% which is very Good.
- discordance is 0.8%.
- the area under the curve: ROC : 99.30 %.
- Gini : 44.68 %.
- Kolomogorov-Smirnov : KS statistic : 90.97%.
- 60% of data give us 98.89% respond with this model.
- ROC reaches 1 when complexity parameter reaches 0.2.

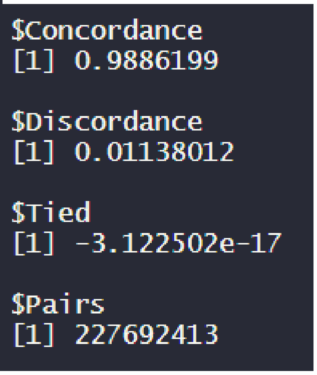
7 - Xtreme Gradient boosting:



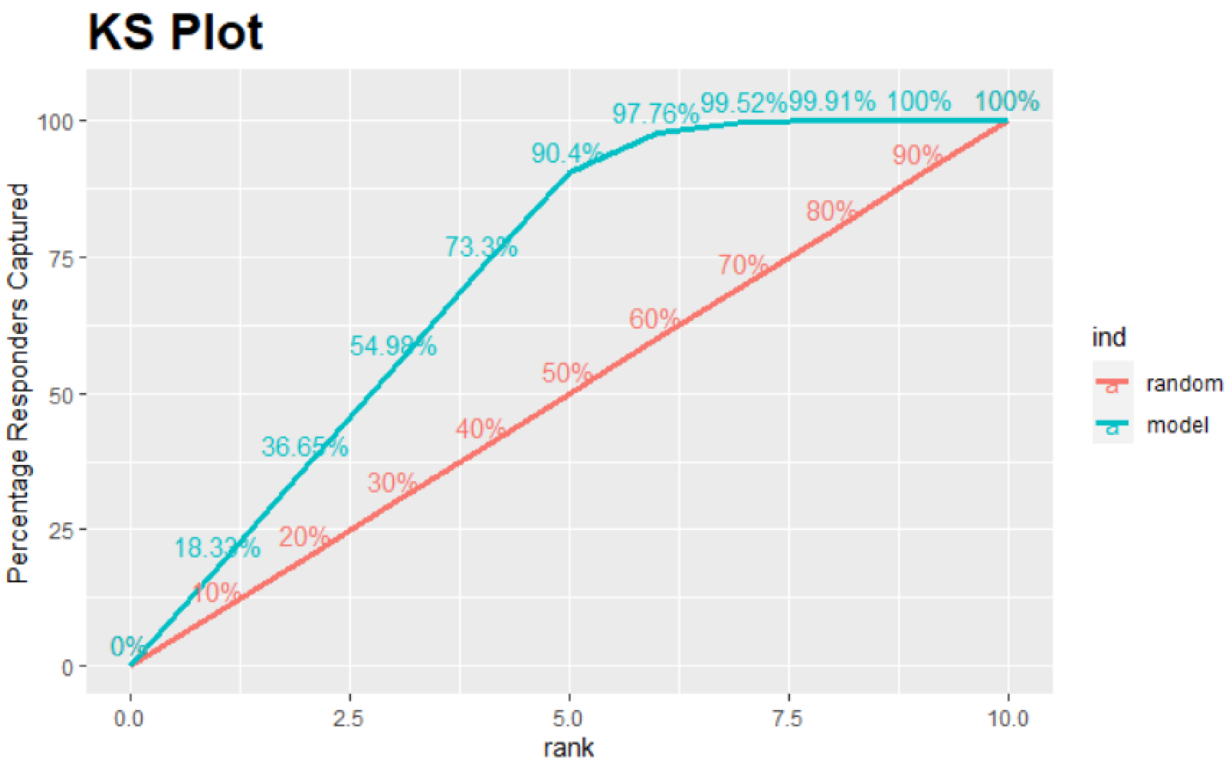
```
60611 samples
   23 predictor
    2 classes: 'No', 'Yes'

No pre-processing
Resampling: Cross-Validated (10 fold, repeated 3 times)
Summary of sample sizes: 54550, 54550, 54550, 54550, 54549, 54550, ...
Resampling results across tuning parameters:

  max_depth  ROC        Sens       Spec
  4          0.9703588  0.9415217  0.9039391
  7          0.9894387  0.9617365  0.9443644

Tuning parameter 'nrounds' was held constant at a value of 150
Tuning parameter 'eta' was held constant at a value of 0.01
 1
Tuning parameter 'min_child_weight' was held constant at a value of 1
Tuning parameter 'subsample' was held constant at
 a value of 1
ROC was used to select the optimal model using the largest value.
The final values used for the model were nrounds = 150, max_depth = 7, eta = 0.01, gamma = 0, colsample_bytree =
 1, min_child_weight = 1 and subsample = 1.
```



```
Confusion Matrix and Statistics

          Reference
Prediction    No   Yes
       No  13185   883
       Yes   582 15656

               Accuracy : 0.9517
                 95% CI : (0.9492, 0.954)
    No Information Rate : 0.5457
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.9027

 Mcnemar's Test P-Value : 4.58e-15

            Sensitivity : 0.9466
            Specificity : 0.9577
         Pos Pred Value : 0.9642
         Neg Pred Value : 0.9372
             Prevalence : 0.5457
         Detection Rate : 0.5166
   Detection Prevalence : 0.5358
      Balanced Accuracy : 0.9522

       'Positive' Class : Yes
```



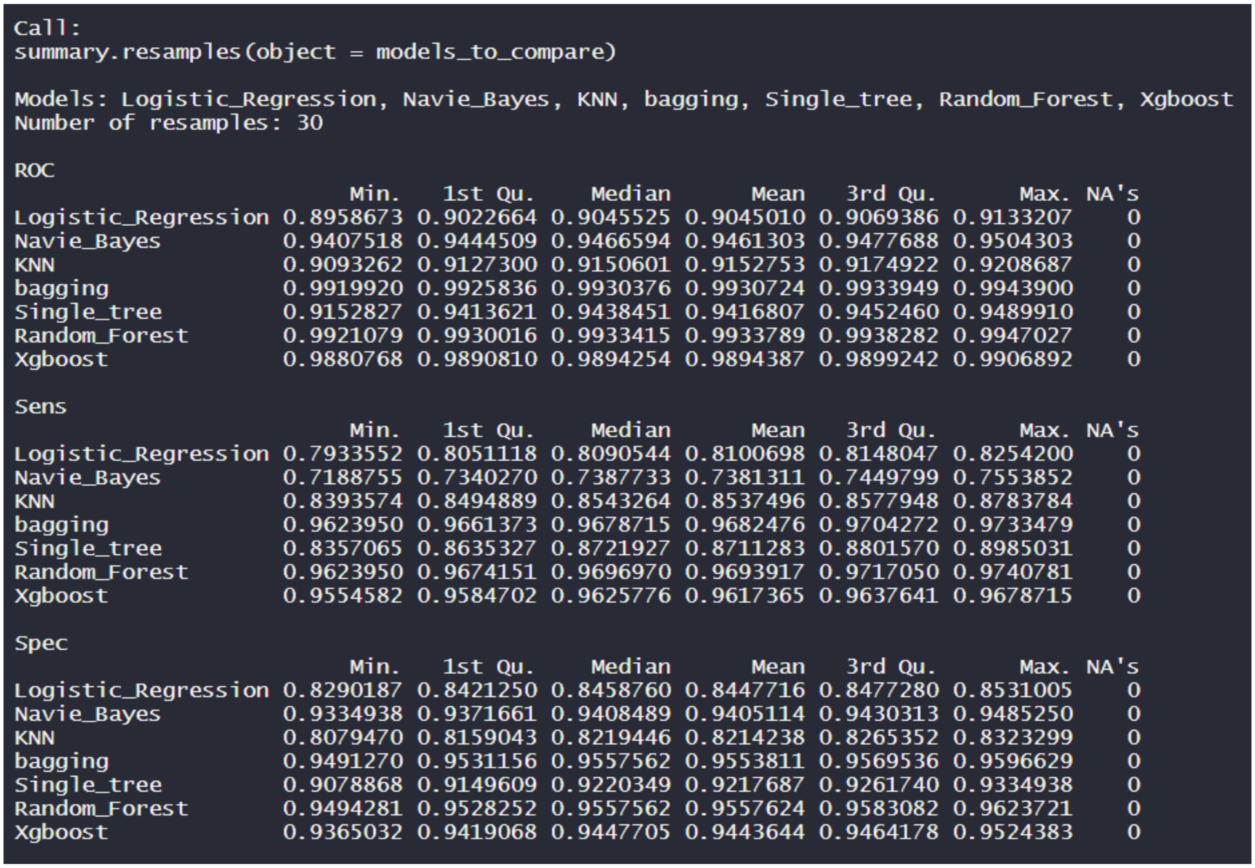
```
$Concordance
[1] 0.9886199

$Discordance
[1] 0.01138012

$Tied
[1] -3.122502e-17

$Pairs
[1] 227692413
```



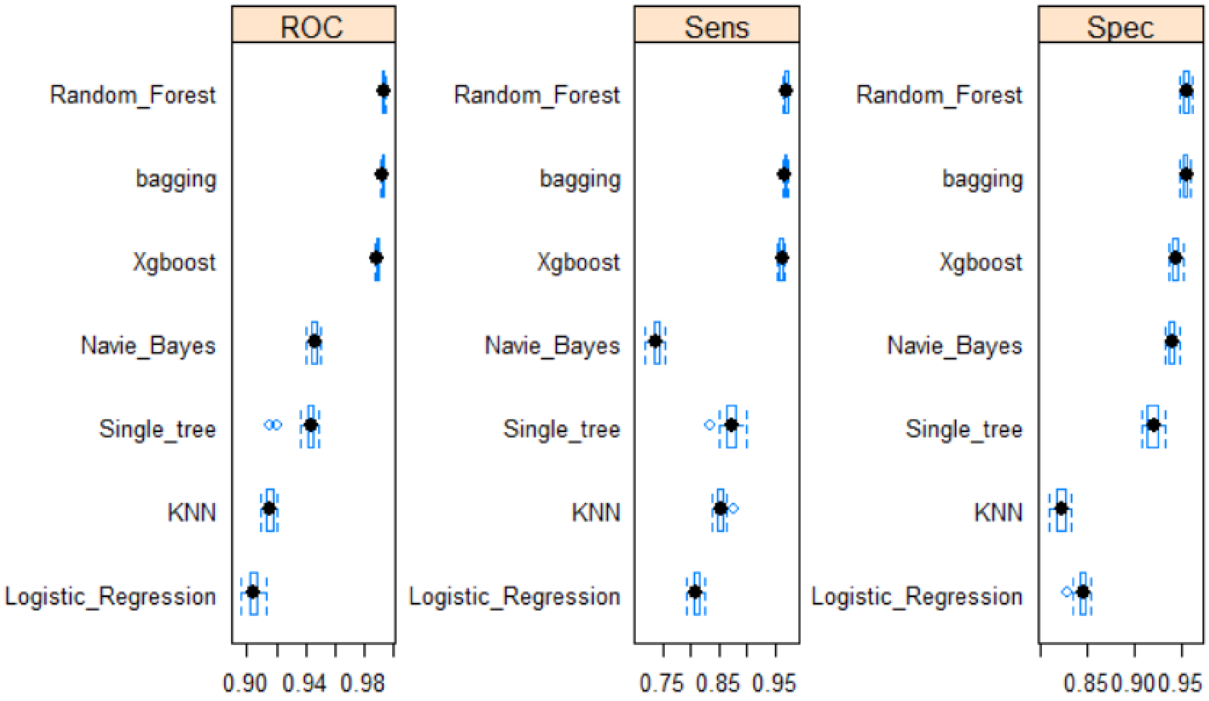
Observe:

- accuracy for Xtreme Gradient boosting: 95.17%.
- very small difference between Sensitivity and Specificity.
- concordance is 98.86%, Probability of (Right) is 98.86% which is Good.
- discordance is 1.13%.
- the area under the curve: ROC : 98.95%.
- Gini : 34.57%.
- Kolomogorov-Smirnov : KS statistic : 88.93%.
- 60% of data give us 97.7% respond with this model.
- ROC reaches 1 when complexity parameter reaches 0.2.

Interpretation from the best model:



```
Call:
summary.resamples(object = models_to_compare)

Models: Logistic_Regression, Navie_Bayes, KNN, bagging, Single_tree, Random_Forest, Xgboost
Number of resamples: 30

ROC
                         Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
Logistic_Regression 0.8958673 0.9022664 0.9045525 0.9045010 0.9069386 0.9133207    0
Navie_Bayes         0.9407518 0.9444509 0.9466594 0.9461303 0.9477688 0.9504303    0
KNN                 0.9093262 0.9127300 0.9150601 0.9152753 0.9174922 0.9208687    0
bagging             0.9919920 0.9925836 0.9930376 0.9930724 0.9933949 0.9943900    0
Single_tree         0.9152827 0.9413621 0.9438451 0.9416807 0.9452460 0.9489910    0
Random_Forest       0.9921079 0.9930016 0.9933415 0.9933789 0.9938282 0.9947027    0
Xgboost             0.9880768 0.9890810 0.9894254 0.9894387 0.9899242 0.9906892    0

Sens
                         Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
Logistic_Regression 0.7933552 0.8051118 0.8090544 0.8100698 0.8148047 0.8254200    0
Navie_Bayes         0.7188755 0.7340270 0.7387733 0.7381311 0.7449799 0.7553852    0
KNN                 0.8393574 0.8494889 0.8543264 0.8537496 0.8577948 0.8783784    0
bagging             0.9623950 0.9661373 0.9678715 0.9682476 0.9704272 0.9733479    0
Single_tree         0.8357065 0.8635327 0.8721927 0.8711283 0.8801570 0.8985031    0
Random_Forest       0.9623950 0.9674151 0.9696970 0.9693917 0.9717050 0.9740781    0
Xgboost             0.9554582 0.9584702 0.9625776 0.9617365 0.9637641 0.9678715    0

Spec
                         Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
Logistic_Regression 0.8290187 0.8421250 0.8458760 0.8447716 0.8477280 0.8531005    0
Navie_Bayes         0.9334938 0.9371661 0.9408489 0.9405114 0.9430313 0.9485250    0
KNN                 0.8079470 0.8159043 0.8219446 0.8214238 0.8265352 0.8323299    0
bagging             0.9491270 0.9531156 0.9557562 0.9553811 0.9569536 0.9596629    0
Single_tree         0.9078868 0.9149609 0.9220349 0.9217687 0.9261740 0.9334938    0
Random_Forest       0.9494281 0.9528252 0.9557562 0.9557624 0.9583082 0.9623721    0
Xgboost             0.9365032 0.9419068 0.9447705 0.9443644 0.9464178 0.9524383    0
```



Remarks:

1. we tried to understand what is the role of Satisfaction Falcon airlines Customer, so we uploaded and Data Preparation and split data in to two part Train,Test and we applied multiple "7" models with general parameters
2. result discuss :
3. biased on the best accuracy : “random forest” had accuracy value : 96.31% and ” bagging” is 96.21%
4. after that” XGBoost” with 95.17% and then “decision tree” with accuracy of 89.7%
5. Before last “naïve bayes” with 84.99% at last “Logistic Regression” and “KNN” with accuracy of : 83.7%.
6. but when we sort biased on ROC and Sensitivity and Specificity: witch are very important for choosing the model, "random forest" is the best of all, and then “bagging”,”XGBoost”.

Actual Accomplishment: final model selection

- ✓ **Random:** randomly selected
- ✓ **Cross-validation:** k=10
- ✓ **Accuracy:** ‘random forest’ 96.31% the highest
- ✓ **Sensitivity:** ‘random forest’ 95.86% the highest
- ✓ **Specificity:** ‘random forest’ 96.85% the highest
- ✓ **Sensitivity and specificity small different :** 96.85%- 95.86%= 0.99% very small
- ✓ **Concordance:** ‘random forest’ 99.18% the highest
- ✓ **ROC :** ‘random forest’ 99.34% the highest
- ✓ **Gini** : ‘random forest’ 44.5% very small different with the highest bagging 44.68%
- ✓ **KS statistic:** ‘random forest’ 90.78% very small different with the highest bagging 90.97%
- ✓ **Interpretability:** possible

The final discussion: since “random forest” is the best match of our optimal model target, which is very good.
Bagging was very good and very close to random forest in most, but Bagging improves prediction accuracy at the cost of interpretability.

We will go with random forest model.
R code file called : “final-report.Rmd”

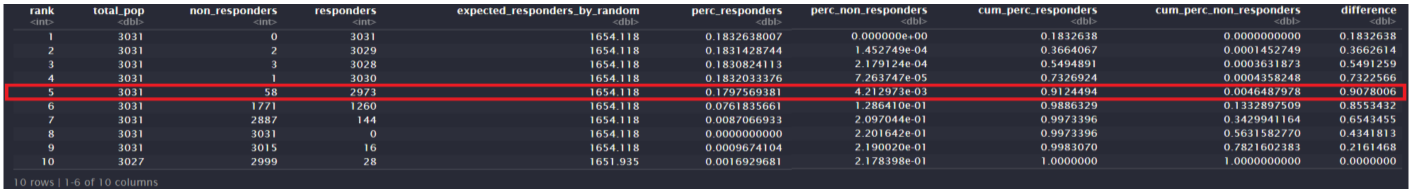
Business insights

As we finally decided that “random forest” is the best model biased on Accuracy and ROC and Gini and Sensitivity and Specificity.

We can use this model for predict and classify the new customers either satisfied or dissatisfied with accuracy of 96.31%.

Biased on Random Forest model the most important variables that make the customer satisfied are: “entertainment” “Seat_comfort” upgraded those will help us to keep satisfied customer away from churning.

KS =90.78% is portion of the population should be targeted to get the highest response rate of 2973 responders out 3031 using random forest with 90.78% customer satisfaction.

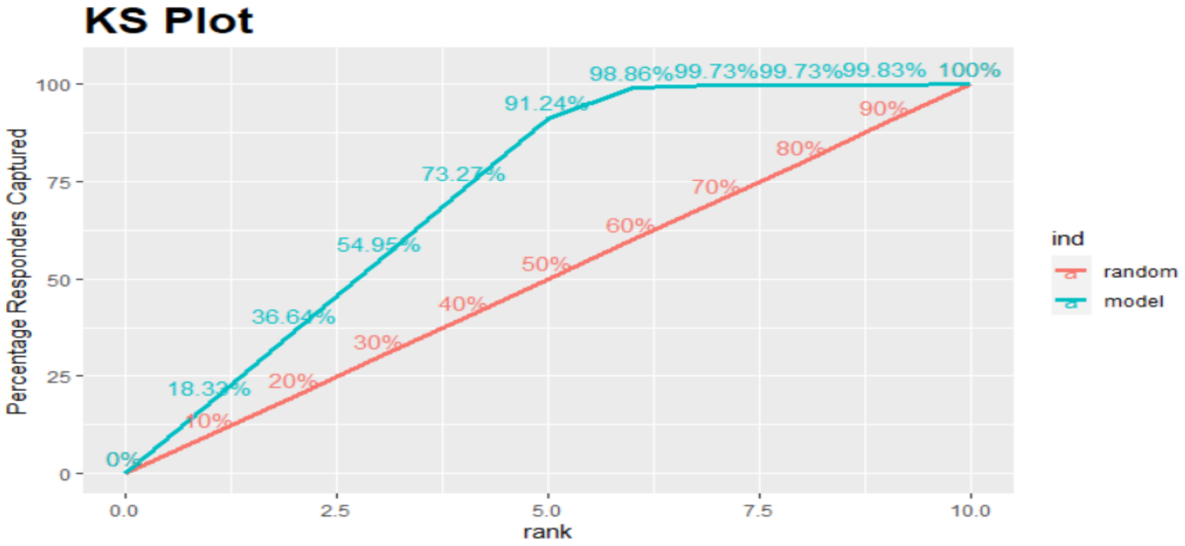


rank	total_pop	non_responders	responders	expected_responders_by_random	perc_responders	perc_non_responders	cum_perc_responders	cum_perc_non_responders	difference
<int>	<dbl>	<int>	<int>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
1	3031	0	3031	1654.118	0.1832638007	0.000000e+00	0.1832638	0.0000000000	0.1832638
2	3031	2	3029	1654.118	0.1831428744	1.452749e-04	0.3664067	0.0001452749	0.3662614
3	3031	3	3028	1654.118	0.1830824113	2.179124e-04	0.5494891	0.0003631873	0.5491259
4	3031	1	3030	1654.118	0.1832033376	7.263747e-05	0.7326924	0.0004358248	0.7322566
5	3031	58	2973	1654.118	0.1797569381	4.212973e-03	0.9124494	0.0046487978	0.9078006
6	3031	1771	1260	1654.118	0.0761835661	1.286410e-01	0.9886329	0.1332897509	0.8553432
7	3031	2887	144	1654.118	0.0087066933	2.097044e-01	0.9973396	0.3429941164	0.6543455
8	3031	3031	0	1654.118	0.0000000000	2.201642e-01	0.9973396	0.5631582770	0.4341813
9	3031	3015	16	1654.118	0.0009674104	2.190020e-01	0.9983070	0.7821602383	0.2161468
10	3027	2999	28	1651.935	0.0016929681	2.178398e-01	1.0000000	1.0000000000	0.0000000

10 rows | 1-6 of 10 columns

The KS Chart is particularly useful in marketing campaigns and ads click predictions where you want to know the right population size to target to get the maximum response rate.

By targeting the top 60% of the population (point it touches the X-axis), the Random forest model is able to cover 98.86% of responders as satisfied customers.


KS Plot
Percentage Responders Captured
rank
ind
random
model