AGH UNIVERITY OF SCIENCE AND TECHNOLOGY
FACULTY OF APPLIED MATHEMATICS

The Selection of a Model for Airlines Customer Satisfaction

Joanna Krężel, Anna Matysek, Piotr Mikler, Adam Szczerba

19 stycznia, 2022

Abstract

The project aims to analyze the data about satisfaction of Investico Airlines passengers. Having customer-granular observations about the cruise and a reported satisfaction level for particular aspects of the flight we try to fit models which predict whether they are satisfied with the service or not. The binary classification models used during the project are {} {} {}, out of which our recommendation is {} based on {}. This document describes the process we undertook and presents the results of data preprocessing, model selection and model validation.

Contents

1 Introduction 3

2 Data 3

2.1 Exploratory Data Analysis . . . 4

2.2 Data Preprocessing . . . 7

2.2.1 Missing data treatment . . . 8

2.2.2 Feature Engineering . . . 9

2.2.3 Feature Encoding . . . 16

3 A train-test split of data 19

4 Baseline Model 19

4.1 Fitting and performance . . . 19

5 The Challenger Models 21

5.1 Random Forest . . . 21

5.1.1 Fitting and performance . . . 21

5.1.2 Validation . . . 26

5.2 Logistic Regression . . . 27

5.2.1 Fitting and performance . . . 27

5.2.2 Validation . . . 36

5.3 State-of-the-art: Neural Network model . . . 37

5.3.1 Fitting and performance . . . 37

5.3.2 Validation . . . 41

6 Conclusion 41

7 Bibliography 43

1 Introduction

Tbd...

2 Data

The data is downloaded from www.kaggle.com and delivered by an airline organization. The dataset consists of the details of customers who have already flown with them. The feedback of the customers on various context and their flight data has been consolidated. The main purpose of this dataset is to predict whether a future customer would be satisfied with their service given the details of the other parameters values. Also the airlines need to know on which aspect of the services offered by them have to be emphasized more to generate more satisfied customers. The data consists of 129880 rows and 23 columns.

Below we list all column names with explanations of the variables' meaning. For categorical variables describing satisfaction level, 0 means *Not Available* and reflects situation in which the passenger did not provide a rating.

Feature	Description	Values
Satisfaction	Airline satisfaction level	satisfied/dissatisfied
Gender	Gender of the passenger	male/female
Customer type	The customer type	loyal / disloyal customer
Age	The age of a passenger	[7; 85] years
Type of travel	Purpose of the flight	personal / business travel
Class	Travel class in the plane	business / eco / eco plus
Flight distance	The flight distance of the journey	[50; 6951] miles
Seat comfort	Satisfaction level of seat comfort	{0-5}
Departure/arrival	Satisfaction level of departure/arrival time	{0-5}
Food and drink	Satisfaction level of food and drink	{0-5}
Gate location	Satisfaction level of gate location	{0-5}

Feature	Description	Values
Inflight WiFi service	Satisfaction level of the inflight wifi service	{0-5}
Inflight entertainment	Satisfaction level of inflight entertainment	{0-5}
Online support	Satisfaction level of online support	{0-5}
Ease of online booking	Satisfaction level of online booking	{0-5}
On-board services	Satisfaction level of on-board service	{0-5}
Leg room service	Satisfaction level of leg room service	{0-5}
Baggage handling	Satisfaction level of baggage handling	{0-5}
Checkin service	Satisfaction level of check-in service	{0-5}
Cleanliness	Satisfaction level of cleanliness	{0-5}
Online boarding	Satisfaction level of online boarding	{0-5}
Departure delay in minutes	Delay upon departure	[0; 1592] minutes
Arrival delay in minutes	Delay upon arrival	[0; 1584] minutes

2.1 Exploratory Data Analysis

Before data wrangling and trying to fit some models, it is great to get an overview of our data set and verify whether it makes sense. We start from summarizing our to see basic statistical analysis. For each quantitive variable we get the minimum, maximum, mean, median and the inter quartile range information. However, for each categorical variable we get the number of observations in each category.

```
summary(AirlinesRaw)
```

```
##        satisfaction      Gender                      Customer Type          Age
##  satisfied   :71087   Female:65899   Loyal Customer   :106100   Min.   : 7.00
##  dissatisfied:58793   Male  :63981   disloyal Customer: 23780   1st Qu.:27.00
##                                                                 Median :40.00
##                                                                 Mean   :39.43
##                                                                 3rd Qu.:51.00
##                                                                 Max.   :85.00
##
##          Type of Travel       Class       Flight Distance Seat comfort
##  Personal Travel:40187   Eco     :58309   Min.   :  50    0: 4797
##  Business travel:89693   Business:62160   1st Qu.:1359    1:20949
##                          Eco Plus: 9411   Median :1925    4:28398
##                                           Mean   :1981    5:17827
##                                           3rd Qu.:2544    2:28726
##                                           Max.   :6951    3:29183
##
##  Departure/Arrival time convenient Food and drink Gate location
##  0: 6664                           0: 5945        2:24518
##  1:20828                           1:21076        3:33546
##  2:22794                           2:27146        4:30088
##  3:23184                           3:28150        1:22565
##  4:29593                           4:27216        5:19161
##  5:26817                           5:20347        0:    2
##
##  Inflight wifi service Inflight entertainment Online support
##  2:27045               4:41879                2:17260
##  0:  132               2:19183                3:21609
##  3:27602               0: 2978                4:41510
##  4:31560               3:24200                5:35563
##  5:28830               5:29831                1:13937
##  1:14711               1:11809                0:    1
##
##  Ease of Online booking On-board service Leg room service Baggage handling
##  3:22418                3:27037          0:  444          3:24485
##  2:19951                4:40675          4:39698          4:48240
##  1:13436                1:13265          3:22467          1: 7975
##  5:34137                2:17174          2:21745          2:13432
##  4:39920                5:31724          5:34385          5:35748
##  0:   18                0:    5          1:11141
```

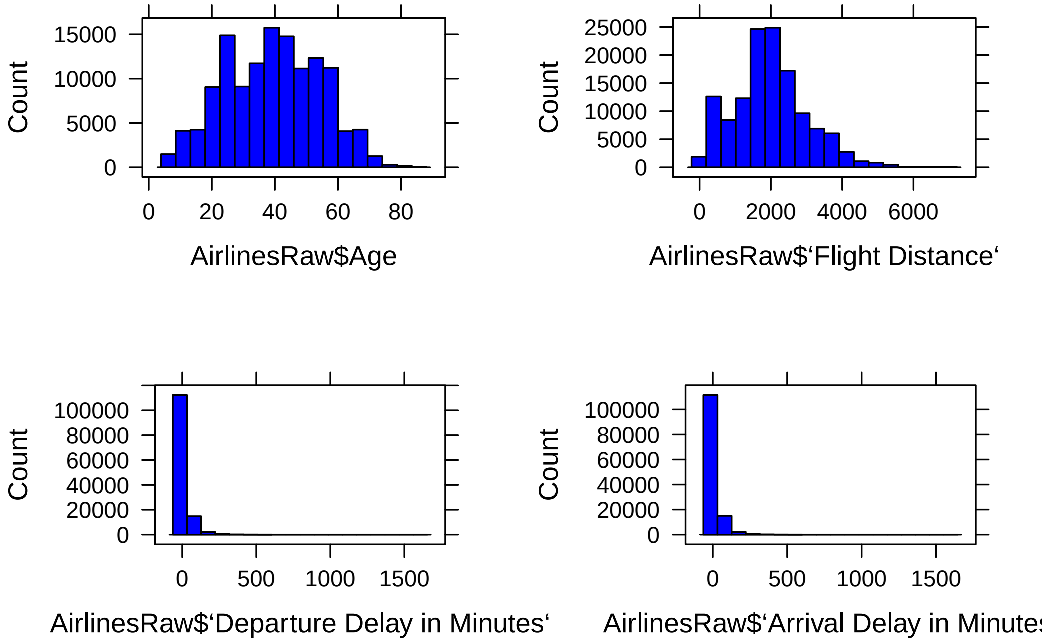
```
##
##  Checkin service Cleanliness Online boarding Departure Delay in Minutes
##  5:27005         3:23984     2:18573         Min.   :   0.00
##  2:15486         4:48795     3:30780         1st Qu.:   0.00
##  4:36481         1: 7768     5:29973         Median :   0.00
##  3:35538         2:13412     4:35181         Mean   :  14.71
##  1:15369         5:35916     1:15359         3rd Qu.:  12.00
##  0:    1         0:    5     0:   14         Max.   :1592.00
##
##  Arrival Delay in Minutes
##  Min.   :   0.00
##  1st Qu.:   0.00
##  Median :   0.00
##  Mean   :  15.09
##  3rd Qu.:  13.00
##  Max.   :1584.00
##  NA's   :393
```

\ In this project we will predict the satisfaction level of a customer in function of the other variables based on past data. All results seems to be reasonable. It is noteworthy that the numbers of satisfied and dissatisfied customers are similar, which means that the data is valuable for future predictions. Moreover, we have almost the same number of males and females, and of business and eco class passengers. However, we have significantly more opinions from loyal customers than from disloyal ones. This may cause some unreliability in data since loyal customers might give greater notes because of some discounts for flights etc. We can also look at each variable connected with customers' notes and check which aspects are considered better and which worse. What we also notice is the fact that there are missing values in some columns. Nevertheless, we will take care about it later.\

Additionally, we plot some histograms to view the distribution of quatitive variables.\

```
attach(AirlinesRaw)
h4<-histogram(AirlinesRaw$Age,AirlinesRaw,col='blue',type='count')
h7<-histogram(AirlinesRaw$`Flight Distance`,AirlinesRaw,col='blue',type='count')
h22<-histogram(AirlinesRaw$`Departure Delay in Minutes`,AirlinesRaw,col='blue',typ
h23<-histogram(AirlinesRaw$`Arrival Delay in Minutes`,AirlinesRaw,col='blue',type=
```

```r
grid.arrange(h4,h7, h22, h23,ncol=2)
```



We see that:\ • departure and arrivals delays are mostly small,\ • the most common flight distance is around 2000 kilometers,\ • the age differs from few years to more than 80, but the biggest number of customers is between 20 and 60.\

Now, when we have an initial overview of our data set, we can move to preprocessing.

2.2 Data Preprocessing

As both academia and business point out, the data-related operations typically constitute about 80% of the whole effort of a modeling pipeline. The performance of any model is heavily driven by the quality of it’s inputs. It can be easily proven in a simple trial by combat that even a suboptimal model running on high quality data can oftentimes bring a sophisticated one with poor inputs to it’s knees. For that reason it is of utmost importance to pay extra care and attention to the data which is fed to the decision making models.

During data preprocessing step we will gain insight about data statistics and information it conveys. First, we'll deal with NAs and outliers. Then we will refactor the features with variable binning and select a subset of features which are likely to display predictive power for our problem. In the end we will encode the variables and prepare a train-test split for model development.

2.2.1 Missing data treatment

Table 2: NA breakdown per feature. NAs span a small portion of data

Feature	NA_count	pct_of_data
ScheduleNote	6664	5.13%
FoodNote	5945	4.58%
SeatNote	4797	3.69%
EntertainmentNote	2978	2.29%
LegRoomNote	444	0.34%
ArrivalDelay	393	0.3%
WifiNote	132	0.1%
eBookingNote	18	0.01%
eBoardingNote	14	0.01%
CleanNote	5	0%
ServiceNote	5	0%
GateNote	2	0%
CheckInNote	1	0%
eSupportNote	1	0%

After examining the data it seems we don't have any critical issue related to missing values. *NAs* are present in 12 variables, but they constitute a minuscule portion of a very large dataset (see fig. 2). We considered employing an imputation strategy based on median, but given that NAs constitute roughly 0.08 of all observations, even if we drop them we would still have 119255 observations left to work with. Based on that we decided not to introduce imputed values to the dataset, but rather work with pure data.

2.2.2 Feature Engineering

The more is not always the better. Feature engineering is a pre-modeling stage which serves identifying features which are significant and filtering out the ones that are not. *Filtering feature selection methods* allow one to discard redundant features in a model independent way. By reducing the number of variables or discretizing them we simplify the model and increase it's interpretability. It is also a step which tackles multicollinearity (high linear codependency of explanatory variables) which kills stability and predictive power of some models.

Across the following sections we are going to introduce a few additional modifications of variables to the dataset and see if it makes sense to keep them. Since we have a big share of $0s$ in column `DepartureDelay`, we wanted to check if it makes sense to include a binary variable $IsDelayed$ as a predictor. We are also going to try discretizing `Age`, `DepartureDelay` and `FlightDistance` continuous variables into bins based on Weight of Evidence metric and evaluate their predictive power.

2.2.2.1 Continuous Features We have four continuous variables in our dataset: `Age`, `DepartureDelay`, `ArrivalDelay` and `FlightDistance`. We start the analysis by analyzing their codependence structure. We note a high linear relationship between `ArrivalDelay` and `DepartureDelay`, visible both in the correlation matrix (fig. **??**) and on figure 1. We can safely drop `ArrivalDelay`, since it doesn't introduce new information and additionally contaminates the dataset with NAs.

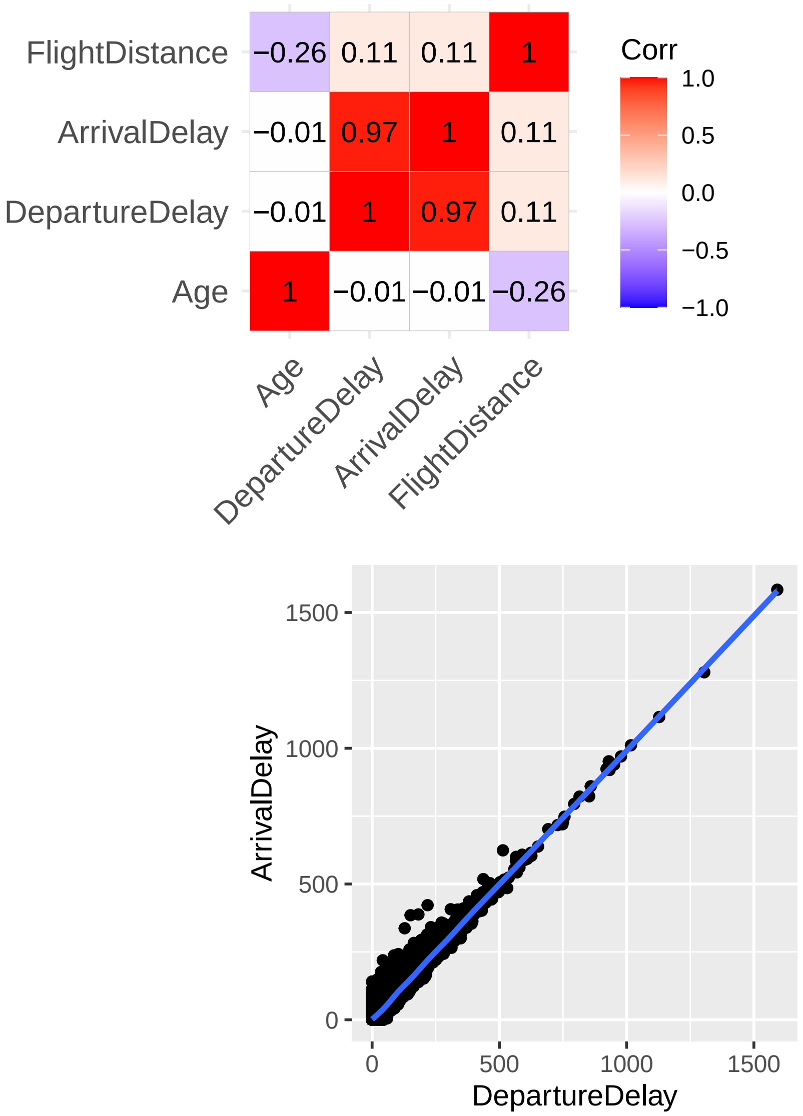
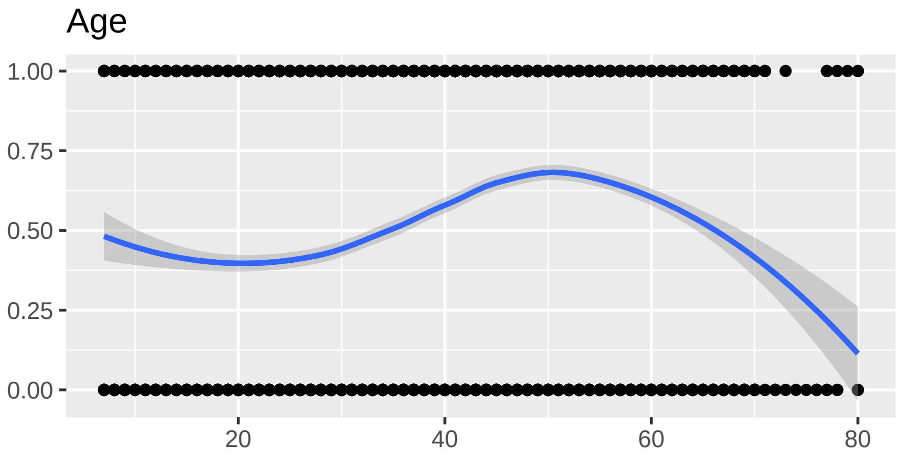
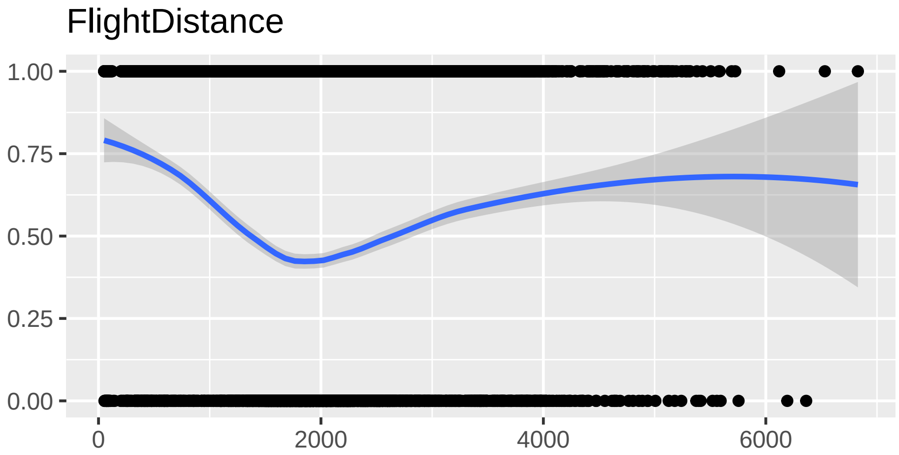
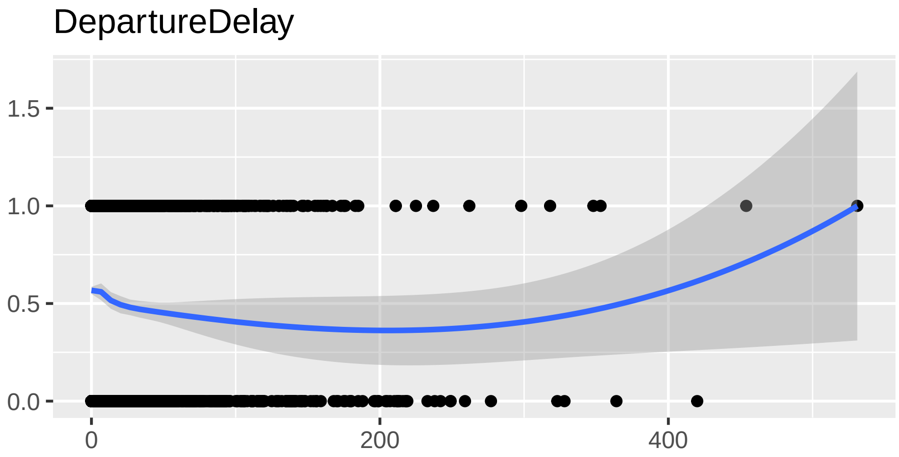


Figure 1: Strong linear relationship between departure delay and arrival delay allows to drop one of them from the dataset.

Next, we're going to examine the loess estimator of satisfaction as a function of the remaining continuous variables to see if any of them looks flat enough to raise suspicion regarding it's utility. Flatness of loess implies that average satisfaction does not change in the explanatory variable, hence the explanatory variable doesn't convey much information. In our case, this is not visible on the figure ??, so we cannot discard any feature based on that. Note, the plot has been generated on a randomized data sub-sample for computational complexity reduction. We made sure however to take a sample large enough, so that the standard errors are tamed, and to ensure the relationship shape is stable regardless of the random seed chosen.


Age
1.00
0.75
0.50
0.25
0.00
20
40
60
80


FlightDistance
1.00
0.75
0.50
0.25
0.00
0
2000
4000
6000


DepartureDelay
1.5
1.0
0.5
0.0
0
200
400

Now we are going to look at possible binnings of our features. We'll use `woeBinning::woe.binning` function which chooses the binning to maximize the information value of the feature. If the optimized binning will yield $IV < 0.1$, we will discard the variable. Otherwise we'll analyze the bins to ensure they are not over-optimized to an unreasonable degree. Based on loess plot shapes/regimes the expectation is to have not more than four bins for `Age` and `FlightDistance`, and a maximum of two bins for `DepartureDelay`.

WOE Table for Age

	Final.Bin	Total.Count	Total.Distr.	0.Rate	WOE	IV
1	<= 28	30896	25.9%	58.6%	-51.4	0.068
2	<= 40	29312	24.6%	49.0%	-12.6	0.004
3	<= 59	47672	40.0%	33.9%	50.1	0.096
4	<= Inf	11375	9.5%	53.2%	-29.3	0.008
6	Total	119255	100.0%	45.9%	NA	0.177

WOE Table for FlightDistance

	Final.Bin	Total.Count	Total.Distr.	0.Rate	WOE	IV
1	<= 1359	29839	25.0%	33.3%	53.0	0.067
2	<= 3053	71534	60.0%	53.4%	-30.2	0.055
3	<= Inf	17882	15.0%	36.7%	38.0	0.021
5	Total	119255	100.0%	45.9%	NA	0.143

WOE Table for DepartureDelay

	Final.Bin	Total.Count	Total.Distr.	0.Rate	WOE	IV
1	<= 19	95920	80.4%	44.0%	7.6	0.005
2	<= Inf	23335	19.6%	53.6%	-31.1	0.019
4	Total	119255	100.0%	45.9%	NA	0.024

From the WOE tables above we see that `DepartureDelay` is a variable of low predictive power, hence we won't use it in modelling. For the other variables, as the data binning chosen by the algorithm seems reasonable given the ex-ante expectations, we’re going to keep them.

2.2.2.2 Ordinal & Categorical Features The main challenge of the data preparation in this dataset is the proper treatment of passenger notes. Take for example the `SeatNote` feature which is a customer note describing their satisfaction level with the seating arrangement. One could ask himself the following questions:

- What did the passenger have in mind? Satisfaction with their seat location? Seat comfort? Possibility of choosing the seat?
- Does ‘$SeatNote$‘ $= 3$ imply a negative attitude towards a service? Or it’s a moderate ‘OK’?
- Is the satisfaction *“difference”* between notes 3 and 2 the same as between notes 5 and 4?
- Is a note ‘$SeatNote$‘ $= 5$ given ‘$Class$‘ $=$ ‘Eco‘ the same as ‘$SeatNote$‘ $= 5$ given ‘$Class$‘ $=$ ‘$Business$‘?

The same point is valid for any note-type variable in the dataset. The issue boils down to the problem that there is **no universal “unit” of satisfaction**. It is just as non-trivial to measure it as to predict it - simply because everyone perceives satisfaction in a subjective way. Our problem has an additional layer of complexity since we don’t have information how precisely the survey questions have been described to the customers - so even if we *did* have some carefully designed satisfaction unit, we cannot be sure if all respondents referred to the same aspects of service when filling out the survey.

We aim to overcome this problem, by binning notes into wider classes, depending on how well they explain and affect the overall satisfaction. Since we have a lot of 5-leveled `Note` factors, there’s a strong suspicion that in such large set there must exist some adjacent levels such that overall satisfaction is invariant to displacements in that group of levels. In other words, we could collapse notes of $1, 2\&3$ to one group if they carried similar information. Hence we will again let `woe.binning` automatically select bins and then verify the result.

Since there are 12 `Note` variables, we will only display one of the WOE tables as an example. However for all it has been verified that **adjacent** levels have been binned, so the binning is plausible, and the IV of the newly binned features are above 0.1.

Final.Bin	Total.Count	Total.Distr.	1.Count	0.Count	1.Distr.	0.Distr.	0.Rate	WOE	IV
1	11713	9.8%	1561	10152	2.4%	18.6%	86.7%	-203.8	0.329
3 + 2	38010	31.9%	11063	26947	17.1%	49.3%	70.9%	-105.6	0.340
5 + 4	69532	58.3%	51945	17587	80.4%	32.2%	25.3%	91.7	0.443
Total	119255	100.0%	64569	54686	100.0%	100.0%	45.9%	NA	1.111

Next, we will check the information value for other categorical variables. They are binary, so binning has not been applied to them in the earlier step. The table 7 presents features and their IV. We see that there are features with IV smaller than 0.1 - we are going to drop those from the dataset.

Lastly, a brief look at the spearman rank correlation matrix (fig. 2) shows that there are no highly correlated "note" features among ordinal variables in the dataset.

Table 7: Information Value for all variables.

varName	IV
IsPersonalTravel	0.0532949
FlightDistance	0.1433763
Age	0.1769584
IsFemale	0.1893377
FoodNote	0.2330999
WifiNote	0.2920904
CheckInNote	0.3607163
Class	0.4001969
IsLoyal	0.4537176
CleanNote	0.4555277
BaggageNote	0.4718849
LegRoomNote	0.5714645
eBoardingNote	0.5948280
ServiceNote	0.6151573
eSupportNote	0.9223110
eBookingNote	1.1114520
SeatNote	1.4504018

varName	IV
EntertainmentNote	2.2947658

```
##  [1] "Class"             "IsSatisfied"       "IsFemale"         
##  [4] "IsLoyal"           "IsPersonalTravel"  "Age"              
##  [7] "FlightDistance"    "EntertainmentNote" "SeatNote"         
## [10] "eBookingNote"      "eSupportNote"      "ServiceNote"      
## [13] "eBoardingNote"     "LegRoomNote"       "BaggageNote"      
## [16] "CleanNote"         "CheckInNote"       "WifiNote"         
## [19] "FoodNote"
```

```
## [1] "IsPersonalTravel"
```

```
## [1] "character"
```

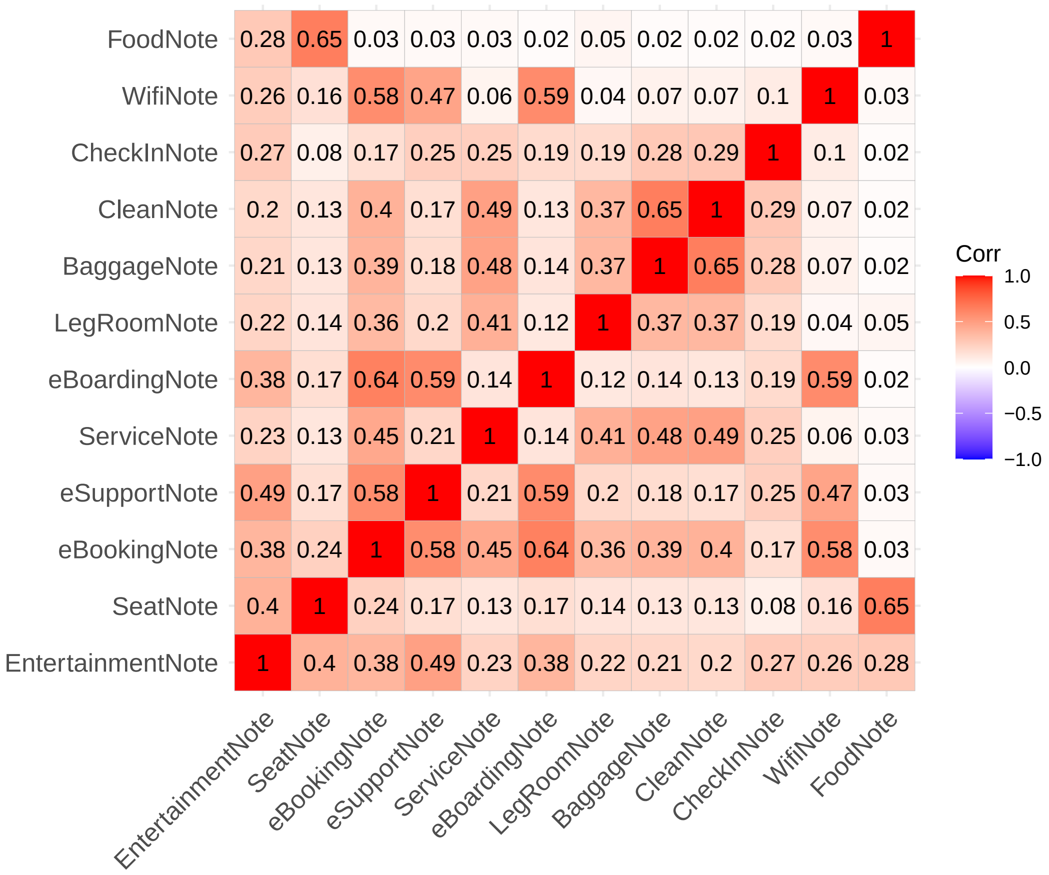
```
##  [1] "Class"             "IsSatisfied"       "IsFemale"         
##  [4] "IsLoyal"           "Age"               "FlightDistance"   
##  [7] "EntertainmentNote" "SeatNote"          "eBookingNote"     
## [10] "eSupportNote"      "ServiceNote"       "eBoardingNote"    
## [13] "LegRoomNote"       "BaggageNote"       "CleanNote"        
## [16] "CheckInNote"       "WifiNote"          "FoodNote"
```

2.2.3 Feature Encoding

Some machine learning algorithms require numerical data, so we considered **ordinal encoding** and **dummy encoding** to transform our data.

We ran the following thought experiment to determine which encoding to employ. Say we use ordinal encoding and assign numbers to each factor level. We could encode `Class` this way and assign a mapping like: {'Eco': 1, 'EcoPlus': 2, 'Business': 3}. This type of representation ensures the quality of the service is properly represented in the numeric data, but the question is whether this translates the same to the overall satisfaction? Yes, the business class is clearly more comfortable to travel in, but the *expectations* (the baseline) of business-class passengers will also be quite higher than the expectations of say, passengers in the economic class.

We chose to employ dummy encoding to encode `Class` - to avoid making assumptions about baseline satisfaction criteria across different passenger classes. For `Note` features however this problem is non-existent, since a higher



	EntertainmentNote	SeatNote	eBookingNote	eSupportNote	ServiceNote	eBoardingNote	LegRoomNote	BaggageNote	CleanNote	CheckInNote	WifiNote	FoodNote
FoodNote	0.28	0.65	0.03	0.03	0.03	0.02	0.05	0.02	0.02	0.02	0.03	1
WifiNote	0.26	0.16	0.58	0.47	0.06	0.59	0.04	0.07	0.07	0.1	1	0.03
CheckInNote	0.27	0.08	0.17	0.25	0.25	0.19	0.19	0.28	0.29	1	0.1	0.02
CleanNote	0.2	0.13	0.4	0.17	0.49	0.13	0.37	0.65	1	0.29	0.07	0.02
BaggageNote	0.21	0.13	0.39	0.18	0.48	0.14	0.37	1	0.65	0.28	0.07	0.02
LegRoomNote	0.22	0.14	0.36	0.2	0.41	0.12	1	0.37	0.37	0.19	0.04	0.05
eBoardingNote	0.38	0.17	0.64	0.59	0.14	1	0.12	0.14	0.13	0.19	0.59	0.02
ServiceNote	0.23	0.13	0.45	0.21	1	0.14	0.41	0.48	0.49	0.25	0.06	0.03
eSupportNote	0.49	0.17	0.58	1	0.21	0.59	0.2	0.18	0.17	0.25	0.47	0.03
eBookingNote	0.38	0.24	1	0.58	0.45	0.64	0.36	0.39	0.4	0.17	0.58	0.03
SeatNote	0.4	1	0.24	0.17	0.13	0.17	0.14	0.13	0.13	0.08	0.16	0.65
EntertainmentNote	1	0.4	0.38	0.49	0.23	0.38	0.22	0.21	0.2	0.27	0.26	0.28

Figure 2: Spearman correlation shows no significant colinear relationships in ordinal variables

note should correspond to higher satisfaction for any rational passenger. Here to avoid inflating the dataset with additional $4 \cdot 14 - 14 = 42$ sparse binary columns we will stick to the original ordinal encoding. This choice nonetheless should be revisited and controlled once we reach the stage of model choice and model fitting.

The resulting, encoded dataframe looks the following way:

```
## Rows: 119,255
## Columns: 34
## $ EntertainmentNote.M <fct> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~
## $ EntertainmentNote.H <fct> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~
## $ SeatNote.M          <fct> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~
## $ SeatNote.H          <fct> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~
## $ eBookingNote.H      <fct> 0, 1, 0, 0, 1, 1, 1, 0, 1, 0, 1, 1, 1, 0, 1, 0, 1,~
## $ eBookingNote.M      <fct> 0, 0, 0, 1, 0, 0, 0, 1, 0, 1, 0, 0, 0, 1, 0, 1, 0,~
## $ eSupportNote.M      <fct> 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 1, 0, 0,~
## $ eSupportNote.H      <fct> 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0,~
## $ ServiceNote.M       <fct> 0, 1, 1, 0, 0, 1, 1, 1, 1, 0, 0, 0, 1, 0, 0, 1, 0,~
## $ ServiceNote.L       <fct> 0, 0, 0, 1, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 1,~
## $ eBoardingNote.H     <fct> 0, 1, 0, 0, 1, 1, 1, 0, 1, 0, 1, 0, 1, 1, 1, 0, 1,~
## $ eBoardingNote.M     <fct> 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0,~
## $ LegRoomNote.M       <fct> 0, 1, 1, 0, 0, 1, 0, 0, 1, 0, 1, 0, 1, 1, 0, 0, 1,~
## $ LegRoomNote.H       <fct> 0, 0, 0, 1, 1, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 1, 0,~
## $ BaggageNote.M       <fct> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0,~
## $ BaggageNote.H       <fct> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~
## $ CleanNote.M         <fct> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0,~
## $ CleanNote.H         <fct> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~
## $ CheckInNote.L       <fct> 0, 0, 1, 1, 1, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0,~
## $ CheckInNote.H       <fct> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~
## $ WifiNote.H          <fct> 0, 1, 0, 0, 1, 1, 1, 0, 1, 0, 1, 1, 1, 0, 1, 0, 1,~
## $ WifiNote.M          <fct> 0, 0, 0, 1, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 1, 0,~
## $ FoodNote.M          <fct> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~
## $ FoodNote.H          <fct> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~
## $ Age.30s             <fct> 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0,~
## $ Age.40s50s          <fct> 1, 1, 0, 1, 0, 0, 1, 0, 0, 1, 1, 0, 1, 0, 0, 1, 1,~
## $ Age.60plus          <fct> 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0,~
## $ FlightDistance.M    <fct> 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 0, 0, 0,~
## $ FlightDistance.H    <fct> 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1,~
## $ Class.Business      <fct> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0,
## $ Class.EcoPlus       <fct> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0,
## $ IsSatisfied         <fct> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
## $ IsFemale            <fct> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
## $ IsLoyal             <fct> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
```

3 A train-test split of data

4 Baseline Model

Before jumping straight into cutting-edge mathematical models it can be very teaching to fit a simple one and analyze how well it can perform on a given problem. It sets a ground zero for any more complicated models to follow, and gives an idea about how complex is the problem at hand.

Therefore we will fit a simple adaptive linear neuron (Adaline) to our train set and evaluate it's performance on the test set. It is a binary classification model, designed to find a decision boundary for *linearly separable* datasets. However even if the data is not perfectly separable, we can still fit the algorithm and optimize a chosen loss function - *MSE* in our case.

4.1 Fitting and performance

Conceptually, Adaline is just a single layer neural network. It consists of two parts:

- A linear combination of inputs which is piped through an identity activation function - this is used for learning weights
- A unit step decision function - this part enables performing binary predictions.

Adaline optimizes the weights of that linear combination (a.k.a. the *net input*) to arrive at the best fit on the training set.

We will put our own spin on the adaline algorithm to avoid overfitting. Additional *L2* regularization during training will give the model an incentive to opt for smaller weights, and an *early stopping* callback will halt the training for us, should the validation loss start to plateau and stop decreasing for at least 3 consecutive epochs.

For the baseline model we won't be performing costly cross-validation, but rather only compare key model performance indicators across train and test sets. Figure 3 presents that there are no significant performance differences between training and test sets across a selection of metrics, therefore we conclude that the model is able to generalize the learned rules well for unseen data and is not overfitted.

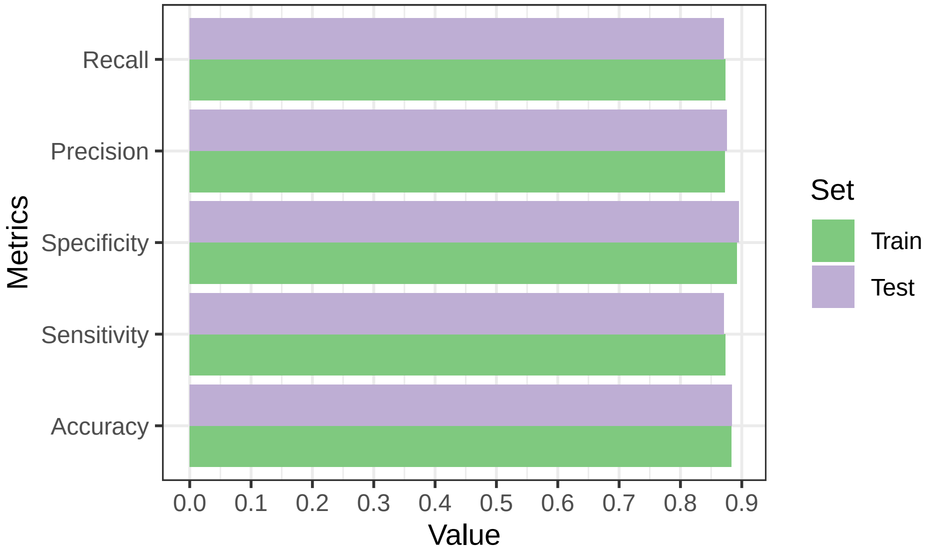


Figure 3: Performance metrics for the baseline Adaline model.

In the case of our dataset, since we're working with one-hot-encoded binary features, the magnitudes and signs of fitted adaline weights can be used as proxies for the directions and magnitudes of the influence on the overall satisfaction for a particular feature. More precisely, the weights tell us by how much will the *net input* increase if a particular feature goes from 0 to 1. Once the net input (corrected for bias) exceeds 0.5, this model will predict a satisfied passenger.

The top 3 influencial feature-note pairs (in a positive and negative sense) according to Adaline model are summarized in table 8. One should note that the coefficients translate to overall satisfaction **indirectly**, since e.g. `SeatNote.H=1` implies `SeatNote.M=0`, and these effects offset in the overall net input. However the coefficients should be sufficient to analyze at least the overall importance of particular features.

We see that solid `Entertaiment` and `Seat` notes can severely sway the satisfaction in the positive direction. On the other hand, we have low `Service` notes that moderately drag down the final satisfaction. The biggest negative coefficients that we see are corresponding to `Food`, but they don't seem make intuitive sense. Here a high food note would impact satisfaction more negatively that a low food note. However it is entirely possible that this is due to some missing variable interactions which Adaline is not designed to pick up.

Table 8: Top 3 positive and negative weights for the Adaline baseline model.

Rank	Feature	Weight
1	EntertainmentNote.H	0.169
2	EntertainmentNote.M	0.141
3	SeatNote.H	0.123
31	ServiceNote.L	-0.023
32	FoodNote.H	-0.026
33	FoodNote.M	-0.034

Since Adaline's accuracy on the test set is 0.884, we can infer that our data is not hard to separate, or at least it is relatively easy to reach a reasonable performance. For all challenger models we will be aiming to achieve better metrics, which must compensate for additional model complexity.

5 The Challenger Models

5.1 Random Forest

5.1.1 Fitting and performance

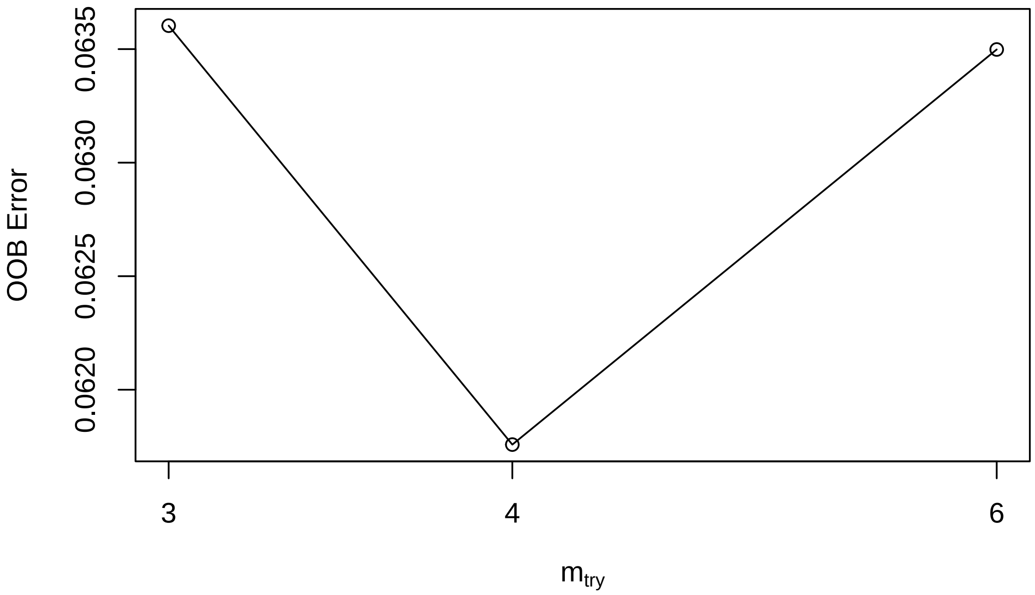
We will try out Random Forest method. Firstly, we consider model including all of the variables.

```
##
## Call:
##  randomForest(formula = frm, data = Airlines_binned_train, importance = TRUE)
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 4
##
##         OOB estimate of  error rate: 6.18%
## Confusion matrix:
##       0     1 class.error
## 0 40963  2777  0.06348880
## 1  3123 48541  0.06044828
```

After running Random Forest algorithm on basic model we can be quite pleased with the results. Only 6.18% of the observations were misqualified thus we obtained model with 93.82% accuracy.

We denote number of variables tried at each split as **mtry**. This parameter allows us to choose optimal number of generated decision trees. By optimal we mean such **mtry** that value of OOB estimate of error rate is the lowest.

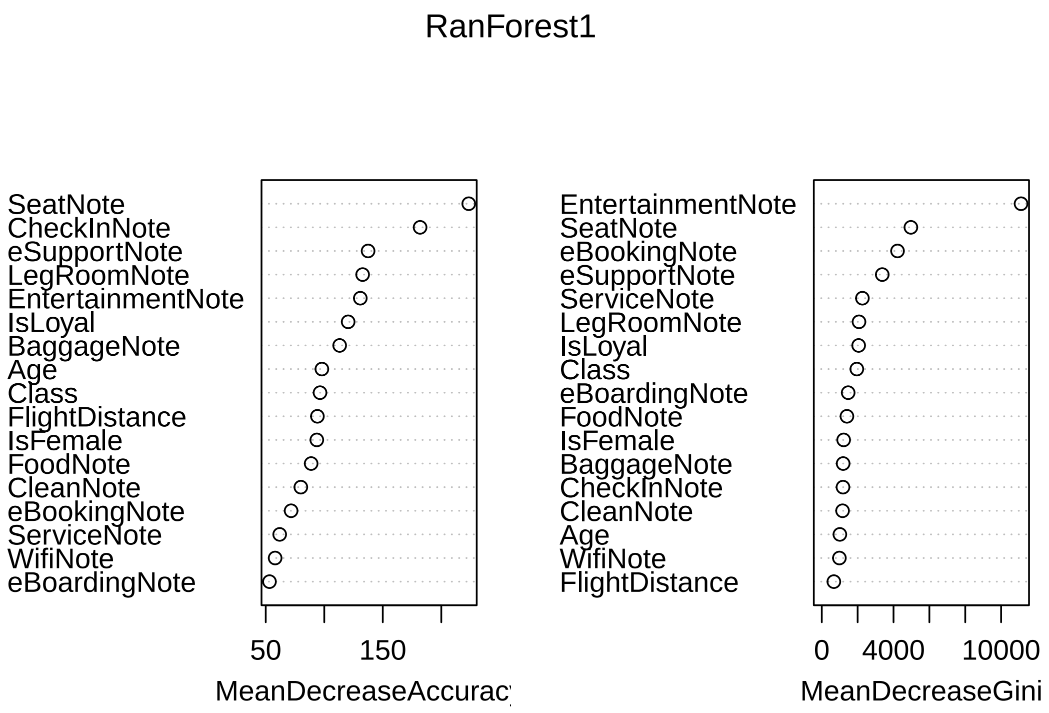
```
## mtry = 4  OOB error = 6.18%
## Searching left ...
## mtry = 3     OOB error = 6.36%
## -0.02987101 0.01
## Searching right ...
## mtry = 6     OOB error = 6.35%
## -0.02817379 0.01
```



```
##       mtry   OOBError
## 3.OOB    3 0.06360320
## 4.OOB    4 0.06175842
## 6.OOB    6 0.06349839
```

As we see, the value chosen by deafult is the best one.

Random Forest enables us to determine how important each variable is in the model, two criteria are used:
• MeanDecreaseAccuracy - expresses the accuracy lost by leaving particular variable out of the training set.
• MeanDecreaseGini - measures node impurity at each split, highest purity means that each node contains only elements of a single class.

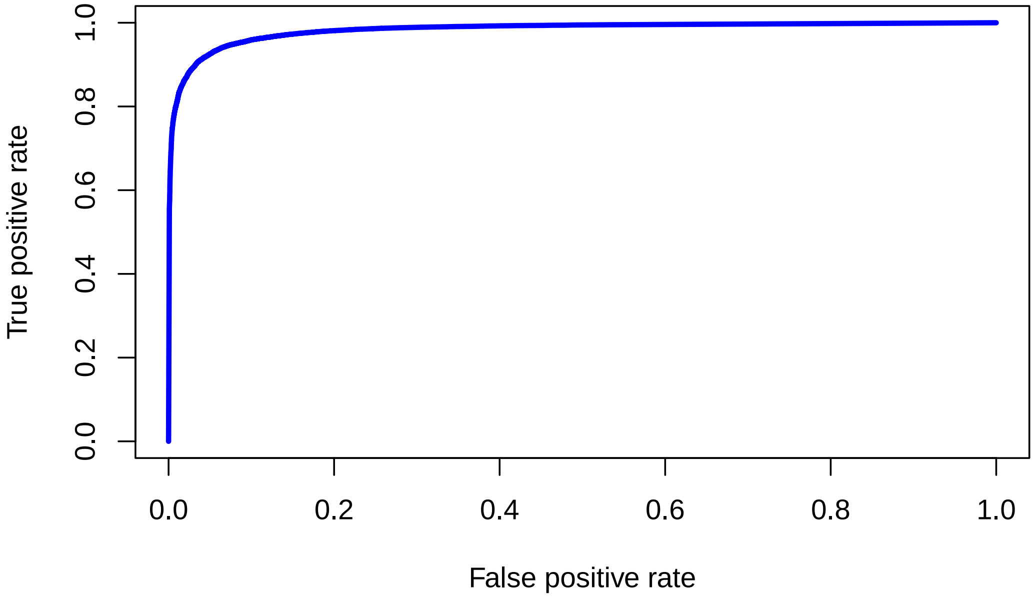


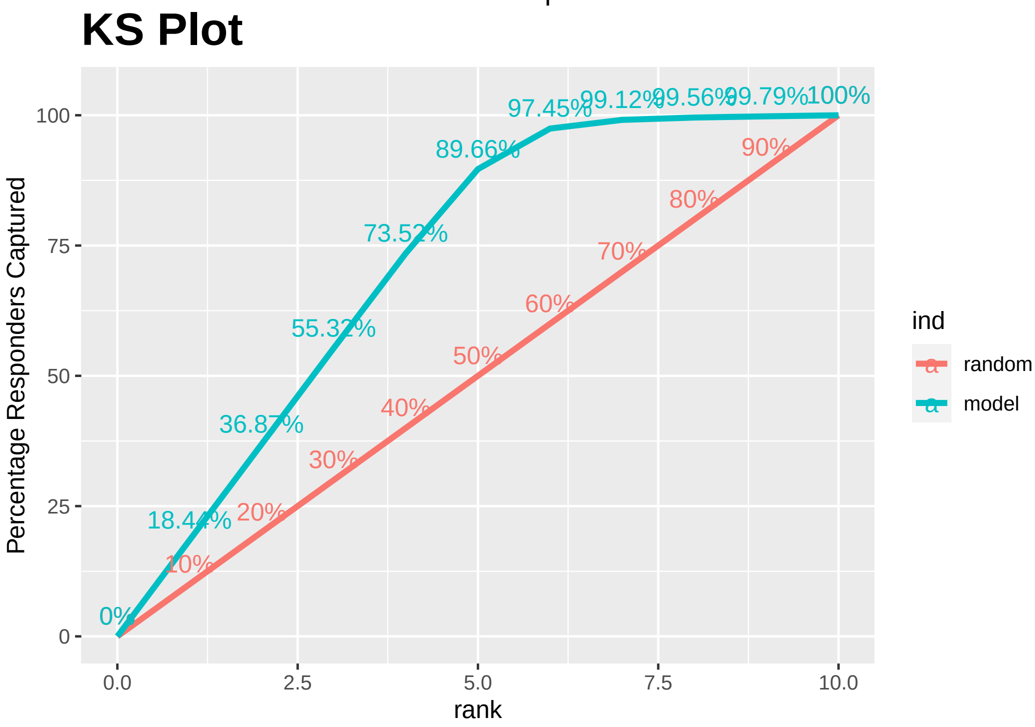
In terms of accuracy of the model, eBoardingNote, WifiNote and ServiceNote seem to be the least important. The most influential are SeatNote, CheckInNote and eSupportNote. The accuracy of the model could drop significantly if we left them out. As for Gini coefficient, the most important are EntertainmentNote, SeatNote and eBookingNote and the least ServiceNote, WifiNote and eBoardingNote.

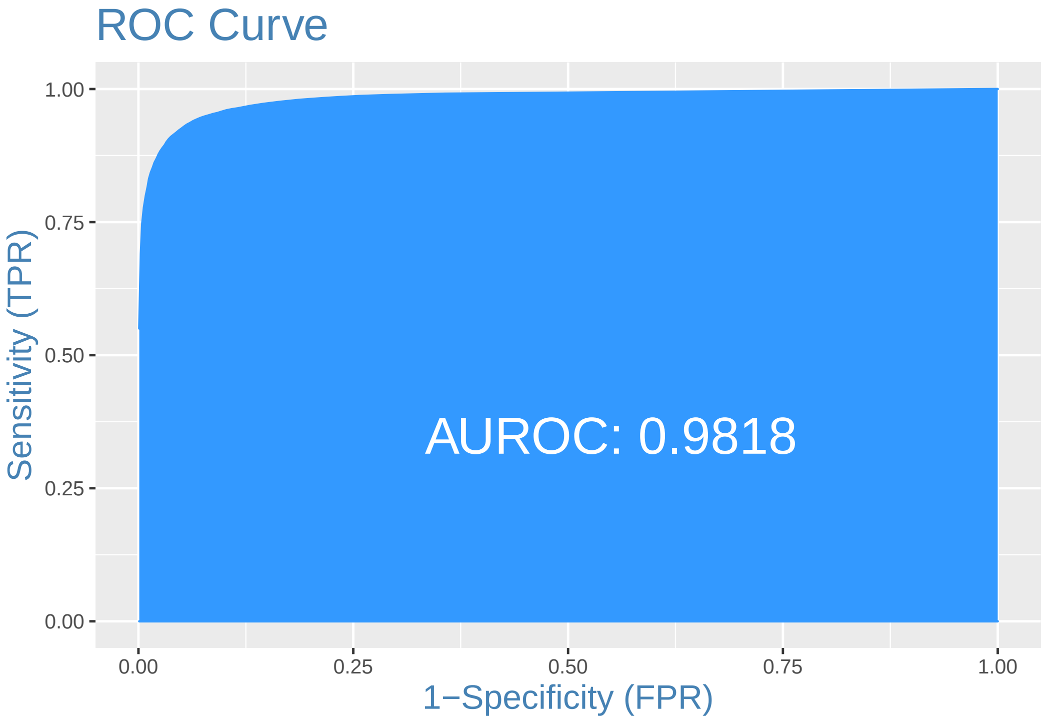
SeatNote is placed high on both plots, so we may assume that this variable has strong influence on satisfaction level. The same applies to eSupportNote and EntertainmentNote. While thinking of clients' satisfaction, the airline should definitely focus on providing them with comfortable seats and proper online customer support. Moreover, making sure that passengers are entertained throughout the flight (e.g. movies, headphones with music) will definitely help them look back at the flight with nothing but enjoyment.

While thinking of the least important variables we have one immediate candidate - WifiNote - it seems like availability and quality of WiFi service does not matter that much to the customers.

Now, with the help of the ROCurve, we will check the performance of this model.







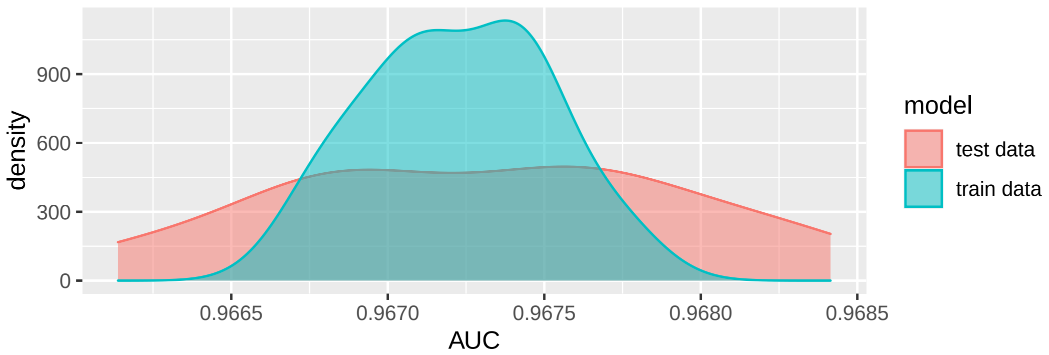
The model has an AUC of 0.9826568 on training data and 0.9370581 on testing data. As the difference between them is small we may presume that this model would be a good choice, plots also look satisfying.

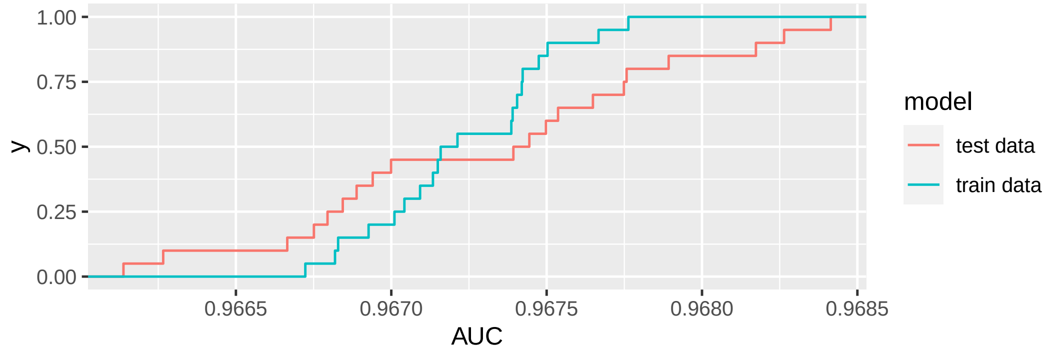
Just to make sure that we don't miss out on any better model, we decided to check some of the options for improvement but did not succeed. Excluding any variables resulted in making accuracy worse. Adding interactions didn't change accuracy or performance of the model at all. Our final choice is the basic model.

We can now approach validation part.

5.1.2 Validation

We will validate the model using the cross validation method and opt for the Monte Carlo Cross Validation. We will draw 100 times a training data-set containing 70% of observations and then study the AUC on the testing data-set.





Observed values for the AUC of the test data have the median equal to 0.9674186, the average equal to 0.9673029 and standard deviation equal to 6.4559167×10^{-4}. Given the plots and parameters we are content with the results, AUCs of training and test sets keep in line with each other. The model does not over-fit so we find it valid.

5.2 Logistic Regression

5.2.1 Fitting and performance

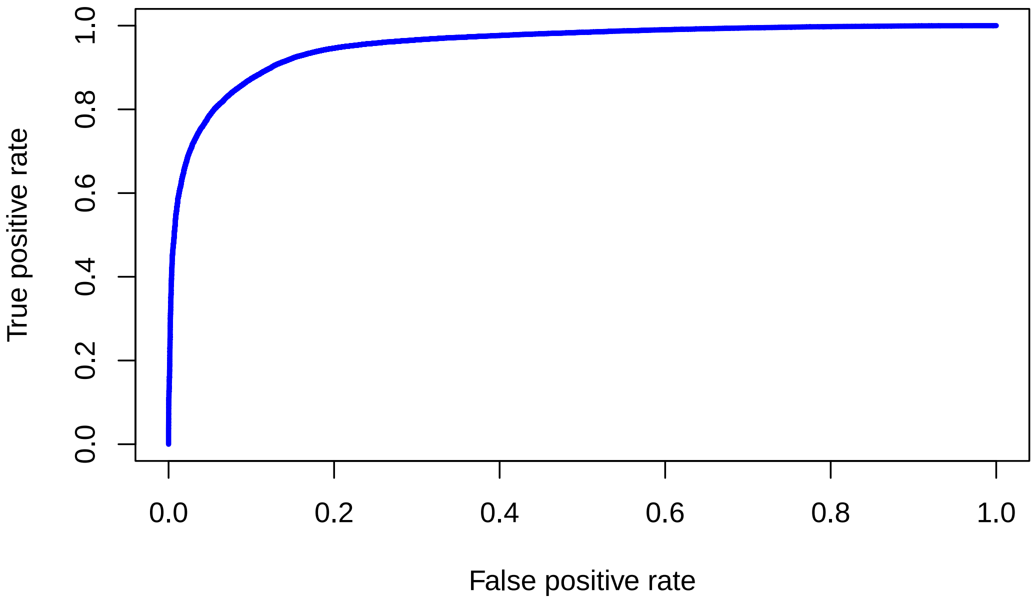
The next model that we will study is a logistic regression. We start from the most basic model which takes all the variables.

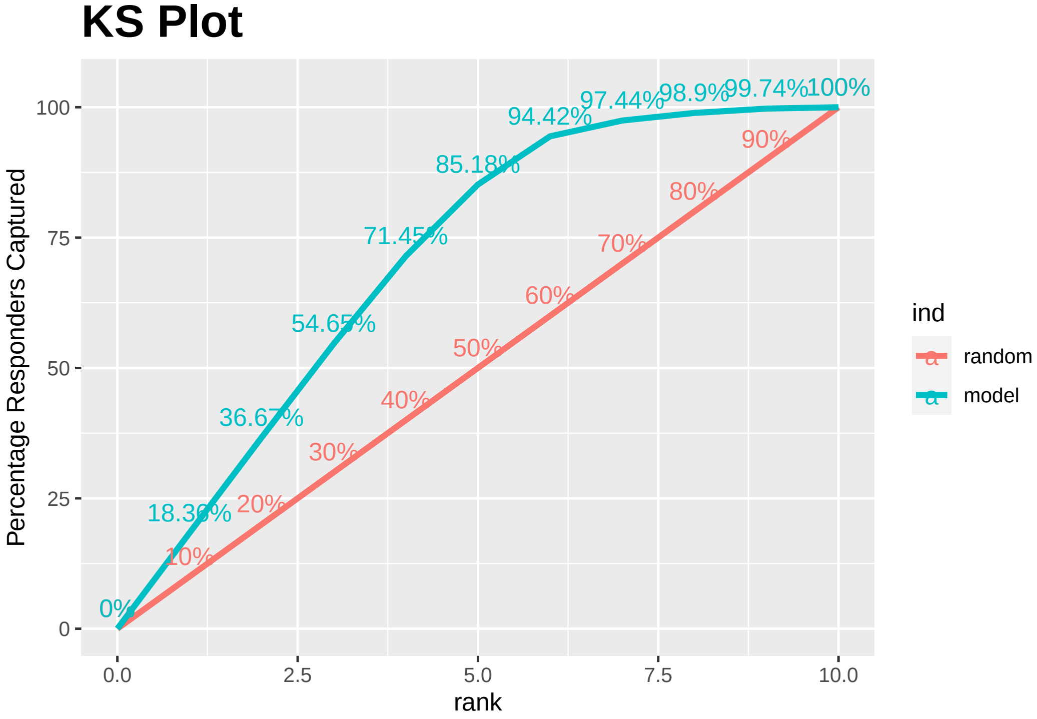
```
##
## Call:
## glm(formula = frm, family = "binomial", data = Airlines_binned_train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -4.1699  -0.3623   0.0313   0.3099   3.6301
```

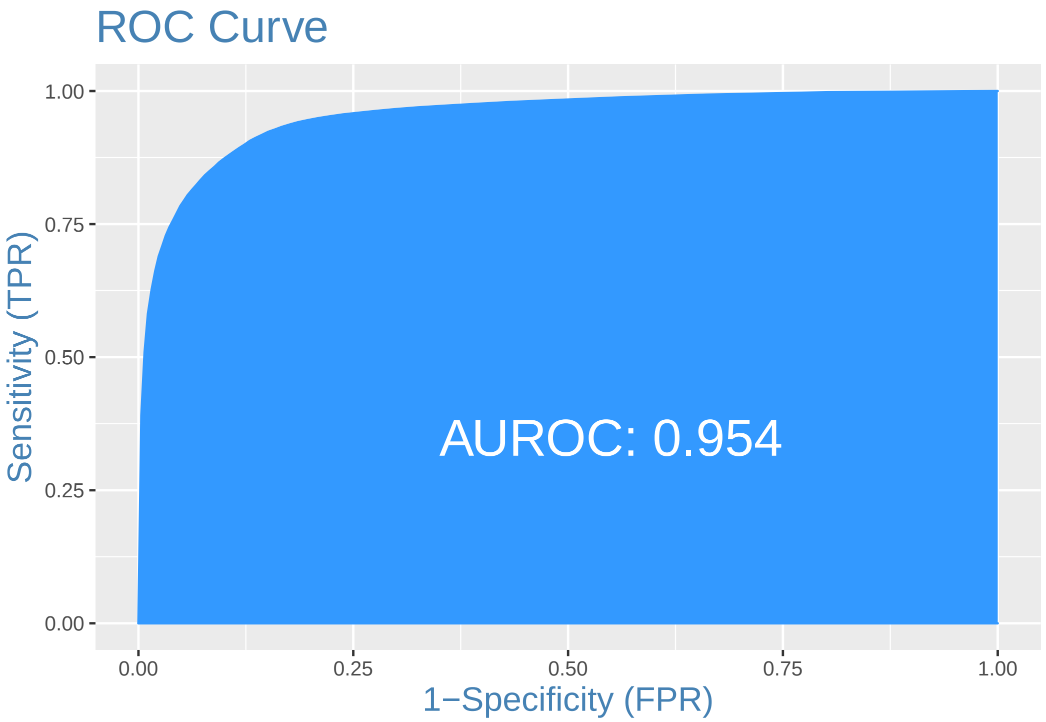
```
##
## Coefficients:
##                    Estimate Std. Error z value Pr(>|z|)
## (Intercept)        -7.21962    0.08599 -83.961  < 2e-16 ***
## ClassBusiness       1.34302    0.02605  51.550  < 2e-16 ***
## ClassEcoPlus        0.03494    0.04385   0.797 0.425568
## IsFemale1           0.97658    0.02355  41.466  < 2e-16 ***
## IsLoyal1            1.91910    0.03742  51.283  < 2e-16 ***
## Age30s             -0.02214    0.03196  -0.693 0.488560
## Age40s50s           0.21030    0.03105   6.774 1.25e-11 ***
## Age60plus          -0.37541    0.04313  -8.704  < 2e-16 ***
## FlightDistanceM    -0.19019    0.02919  -6.516 7.21e-11 ***
## FlightDistanceH    -0.13764    0.03939  -3.494 0.000475 ***
## EntertainmentNoteM  1.82628    0.02847  64.152  < 2e-16 ***
## EntertainmentNoteH  2.99102    0.04336  68.979  < 2e-16 ***
## SeatNoteM           0.85488    0.03330  25.670  < 2e-16 ***
## SeatNoteH           5.13222    0.11098  46.246  < 2e-16 ***
## eBookingNoteH       1.64792    0.06294  26.180  < 2e-16 ***
## eBookingNoteM       0.85178    0.06052  14.073  < 2e-16 ***
## eSupportNoteM       0.22928    0.03314   6.918 4.59e-12 ***
## eSupportNoteH       0.47526    0.03731  12.739  < 2e-16 ***
## ServiceNoteH        0.72232    0.04724  15.290  < 2e-16 ***
## ServiceNoteM        0.22391    0.04594   4.874 1.09e-06 ***
## eBoardingNoteH      0.19938    0.03785   5.267 1.38e-07 ***
## eBoardingNoteM      0.32782    0.03447   9.509  < 2e-16 ***
## LegRoomNoteM        0.03922    0.04776   0.821 0.411536
## LegRoomNoteH        0.75533    0.04772  15.827  < 2e-16 ***
## BaggageNoteM        0.10721    0.03359   3.192 0.001412 **
## BaggageNoteH        0.46100    0.03863  11.932  < 2e-16 ***
## CleanNoteM          0.06595    0.03497   1.886 0.059276 .
## CleanNoteH          0.48344    0.03958  12.214  < 2e-16 ***
## CheckInNoteM        0.33644    0.02874  11.706  < 2e-16 ***
## CheckInNoteH        0.90613    0.03728  24.307  < 2e-16 ***
## WifiNoteH          -0.13901    0.05062  -2.746 0.006025 **
## WifiNoteM           0.12726    0.04902   2.596 0.009435 **
## FoodNoteM          -0.48163    0.03440 -14.000  < 2e-16 ***
## FoodNoteH          -0.51322    0.04304 -11.924  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 131599  on 95403  degrees of freedom
## Residual deviance:  51904  on 95370  degrees of freedom
## AIC: 51972
##
## Number of Fisher Scoring iterations: 7
```

We notice that according to this model:\ • business class passengers have a much higher satisfaction level than eco plus class passengers, who in their turn have a higher satisfaction level than eco class ones,\ • satisfaction level is higher for females and loyal customers,\ • passengers at the age of 40s and 50s have a higher satisfaction level than the others,\ • satisfaction level is lower for long and medium flight distances than for the short ones,\ • the higher entertainment, seat, eBooking, eSupport, service, leg room, clean, baggage and check-in notes, the higher the satisfaction level.\

What we find weird is that taking into account wifi, food and eBoarding the satisfaction level is higher for medium notes than for the high ones. This may be a consequence of the fact that something went wrong and the airlines wanted to compensate for this for example with food. Another reason could be connected with the flight distance. Wifi and more food are usually available during long flights. However, the longer the flight, the more things can go wrong and the more people are tired, uncomfortable etc. Nevertheless, we don't have enough background information to make such conclusions. We decide to leave these variables in our model for now and check its performance, but later we will try do improve it.


True positive rate
0.0
0.2
0.4
0.6
0.8
1.0
0.0
0.2
0.4
0.6
0.8
1.0
False positive rate


KS Plot
Percentage Responders Captured
100
75
50
25
0
0.0
2.5
5.0
7.5
10.0
rank
0%
18.36%
36.67%
54.65%
71.45%
85.18%
94.42%
97.44%
98.9%
99.74%
100%
10%
20%
30%
40%
50%
60%
70%
80%
90%
ind
random
model



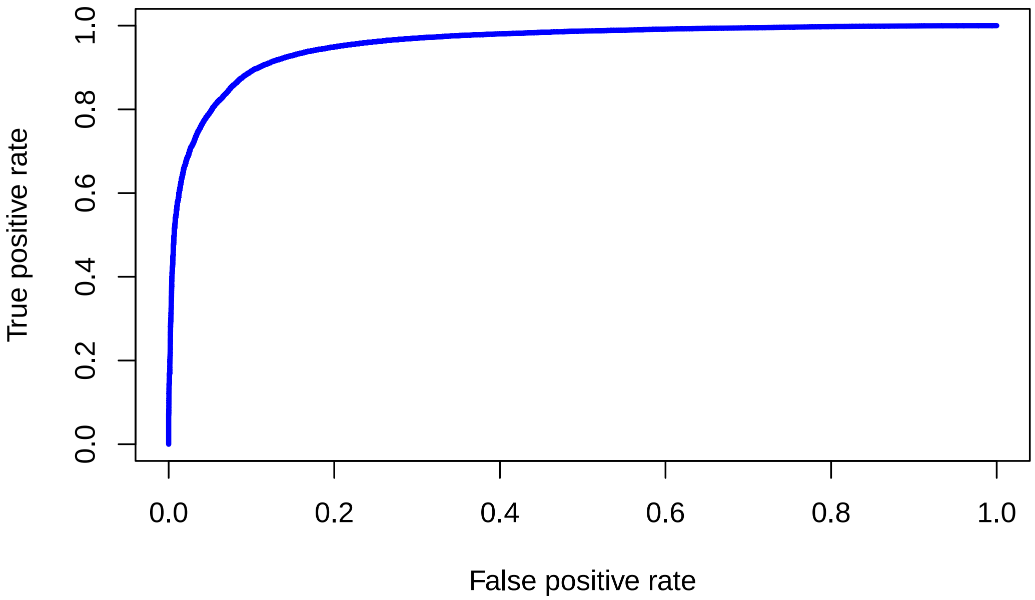
The AUC on the training data is 0.9542947 and on the testing data 0.9541048. The difference is small. KS plot and the ROC curve also looks great. This model is a good contender. However, we decide to try adding some interactions between variables as some of them seem natural. After many attempts we opt for including the interactions between gender&loyalty, gender&class, gender&age and class&flight distance. We tried also for example class&food or class&service, but they didn't improve our model. What is more, we decided not to include WifiNote as after adding interactions it was the least significant and the relationship was weird. Thus we get the final logistic regression model, which is summarized below.\

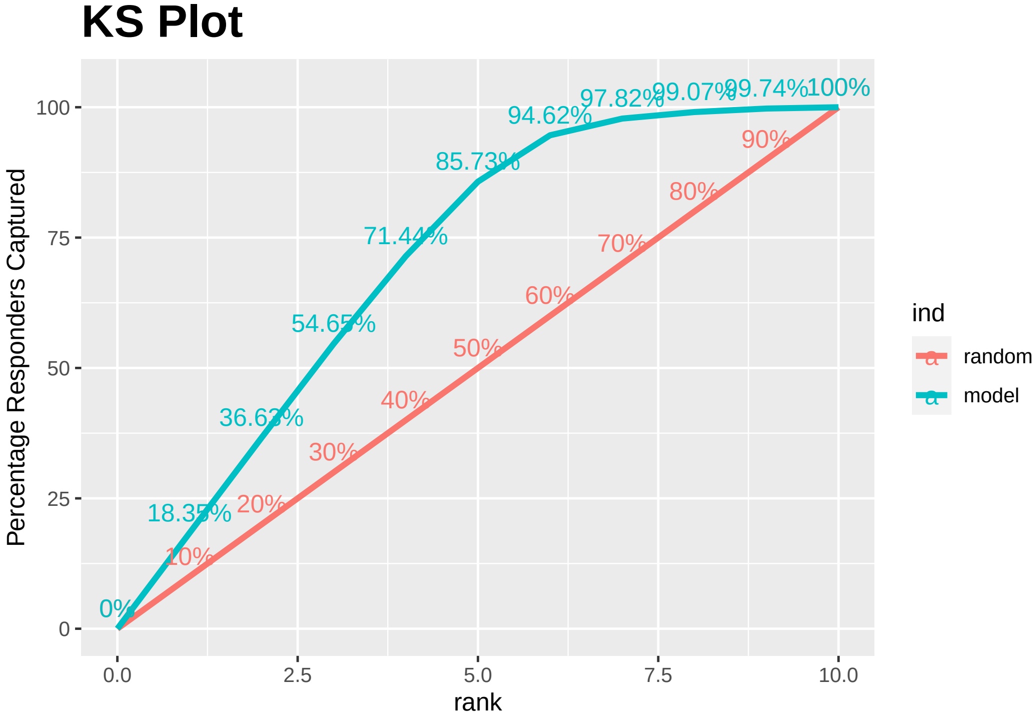
```
## 
## Call:
## glm(formula = frm2, family = "binomial", data = Airlines_binned_train)
## 
## Deviance Residuals: 
##     Min       1Q   Median       3Q      Max  
## -4.1219  -0.3482   0.0296   0.3028   3.5915  
## 
```

```
## Coefficients:
##                               Estimate Std. Error z value Pr(>|z|)    
## (Intercept)                   -7.23908    0.09822 -73.699  < 2e-16 ***
## ClassBusiness                  1.74811    0.06208  28.159  < 2e-16 ***
## ClassEcoPlus                   0.42848    0.11666   3.673 0.000240 ***
## IsFemale1                      1.15594    0.07710  14.992  < 2e-16 ***
## IsLoyal1                       1.40869    0.05355  26.307  < 2e-16 ***
## Age30s                         0.21245    0.04727   4.494 6.99e-06 ***
## Age40s50s                      0.55289    0.04442  12.447  < 2e-16 ***
## Age60plus                     -0.33849    0.06615  -5.117 3.10e-07 ***
## FlightDistanceM               -0.22820    0.04753  -4.801 1.58e-06 ***
## FlightDistanceH               -0.44938    0.08464  -5.310 1.10e-07 ***
## EntertainmentNoteM             1.75566    0.02894  60.658  < 2e-16 ***
## EntertainmentNoteH             2.84328    0.04440  64.040  < 2e-16 ***
## SeatNoteM                      0.84514    0.03383  24.985  < 2e-16 ***
## SeatNoteH                      5.15478    0.11218  45.949  < 2e-16 ***
## eBookingNoteH                  1.55519    0.05784  26.888  < 2e-16 ***
## eBookingNoteM                  0.92479    0.05355  17.270  < 2e-16 ***
## eSupportNoteM                  0.16166    0.03321   4.868 1.13e-06 ***
## eSupportNoteH                  0.42142    0.03776  11.161  < 2e-16 ***
## ServiceNoteH                   0.69248    0.04817  14.377  < 2e-16 ***
## ServiceNoteM                   0.17398    0.04644   3.747 0.000179 ***
## eBoardingNoteH                 0.22947    0.03721   6.168 6.94e-10 ***
## eBoardingNoteM                 0.33404    0.03464   9.644  < 2e-16 ***
## LegRoomNoteM                  -0.02540    0.04874  -0.521 0.602319    
## LegRoomNoteH                   0.66935    0.04891  13.686  < 2e-16 ***
## BaggageNoteM                   0.11801    0.03457   3.414 0.000640 ***
## BaggageNoteH                   0.50288    0.03998  12.577  < 2e-16 ***
## CleanNoteM                     0.10369    0.03610   2.872 0.004076 ** 
## CleanNoteH                     0.55564    0.04111  13.516  < 2e-16 ***
## CheckInNoteM                   0.34630    0.02908  11.909  < 2e-16 ***
## CheckInNoteH                   0.94897    0.03817  24.862  < 2e-16 ***
## FoodNoteM                     -0.47547    0.03481 -13.660  < 2e-16 ***
## FoodNoteH                     -0.51726    0.04323 -11.966  < 2e-16 ***
## IsFemale1:IsLoyal1             1.08540    0.07105  15.276  < 2e-16 ***
## IsFemale1:Age30s              -0.39527    0.06544  -6.041 1.54e-09 ***
## IsFemale1:Age40s50s           -0.57123    0.06321  -9.038  < 2e-16 ***
## IsFemale1:Age60plus           -0.06771    0.09023  -0.750 0.453013    
## ClassBusiness:IsFemale1       -1.53533    0.05220 -29.412  < 2e-16 ***
## ClassEcoPlus:IsFemale1        -0.48121    0.09760  -4.930 8.20e-07 ***
## ClassBusiness:FlightDistanceM  0.49425    0.06107   8.093 5.82e-16 ***
## ClassEcoPlus:FlightDistanceM  -0.19410    0.10774  -1.802 0.071615 .
## ClassBusiness:FlightDistanceH  0.73526    0.09554   7.696 1.41e-14 ***
## ClassEcoPlus:FlightDistanceH  -0.12560    0.20768  -0.605 0.545328
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 131599  on 95403  degrees of freedom
## Residual deviance:  50220  on 95362  degrees of freedom
## AIC: 50304
##
## Number of Fisher Scoring iterations: 7
```

We notice that according to our model:\ • business class passengers have a much higher satisfaction level than eco plus class passengers, who in their turn have a higher satisfaction level than eco class ones,\ • satisfaction level is higher for females and loyal customers,\ • passengers at the age of 40s and 50s have a higher satisfaction level than the others,\ • the higher entertainment, seat, eBooking, eSupport, service, leg room, clean, baggage and check-in notes, the higher the satisfaction level,\ • generally, the longer the flight distance, the less satisfied people are. However, this is not the case in business cass, where people are more satisfied on longer distances,\ • females travelling business and eco plus class are less satisfied than the ones from eco class and females over 60 are more satisfied than the younger ones.

Now we will check the performance of the model using few criteria.





The logistic regression model has an AUC of 0.9573007 on the training data and 0.956471on the testing data. The difference is very small. The KS is 0.7923236. Moreover, we can see (on the ROC curve graph) that our classifier

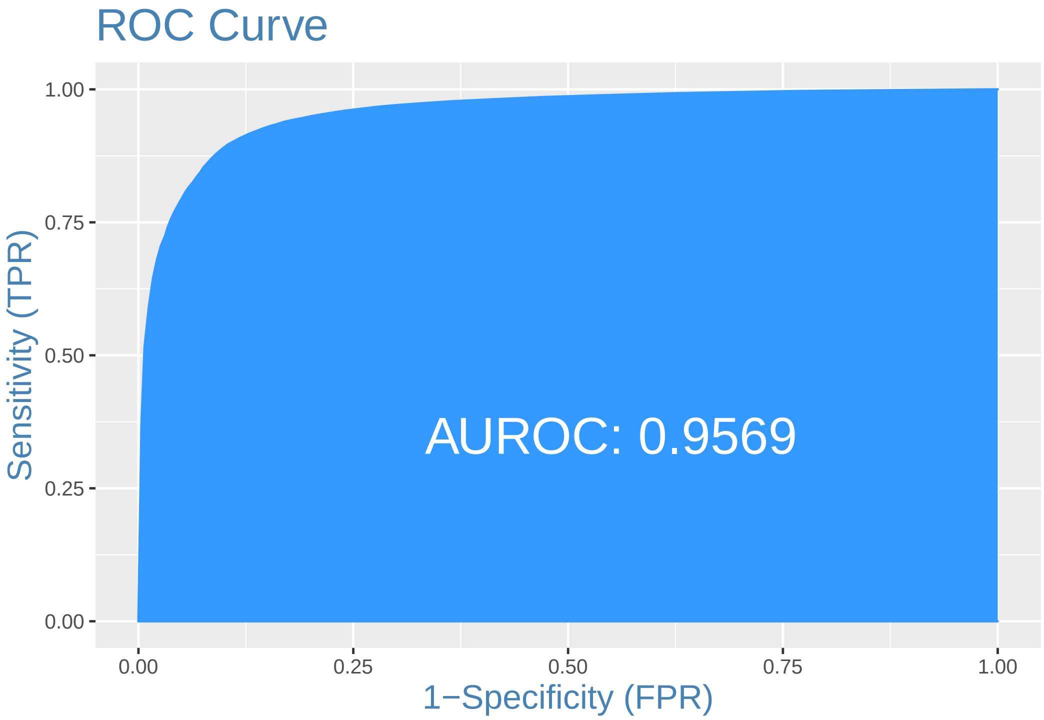


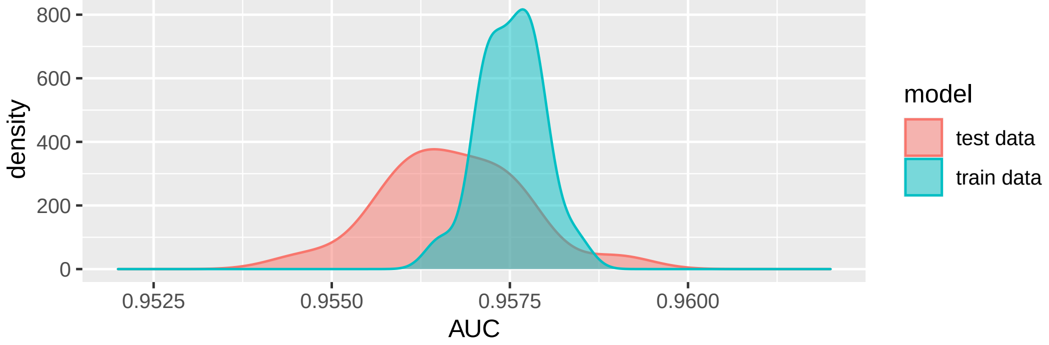
Figure 4: The ROC (receiver operating curve) for our model

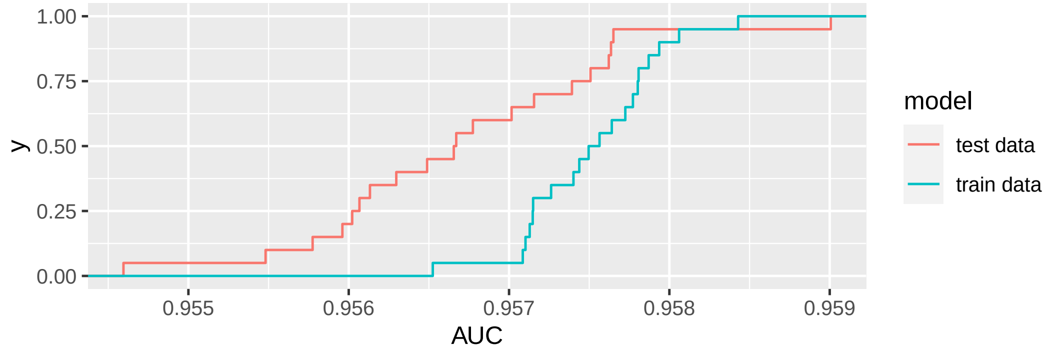
is not far from being perfect. Therefore, we will move forward to the validation part.

5.2.2 Validation

To validate the model we will use the cross validation method and opt for the Monte Carlo Cross Validation.

We use the Monte Carlo Cross Validation with a test data-set that spans 30% of our observations and 70% in the training data-set. We will draw 100 times a training data-set of 0.7 and study the AUC on the testing data-set.





```
## [1] 0.9566629
```

```
## [1] 0.9566953
```

```
## [1] 0.0009762646
```

We are satisfied with the performance of our model. The median of the observed values for the AUC of the test data is 0.9566629, the average is 0.9566953 with a standard deviation of 9.7626465×10^{-4}. We find these results great.

5.3 State-of-the-art: Neural Network model

We have also made an attempt to solve the problem with a neural network.

The network consists of the following layers:

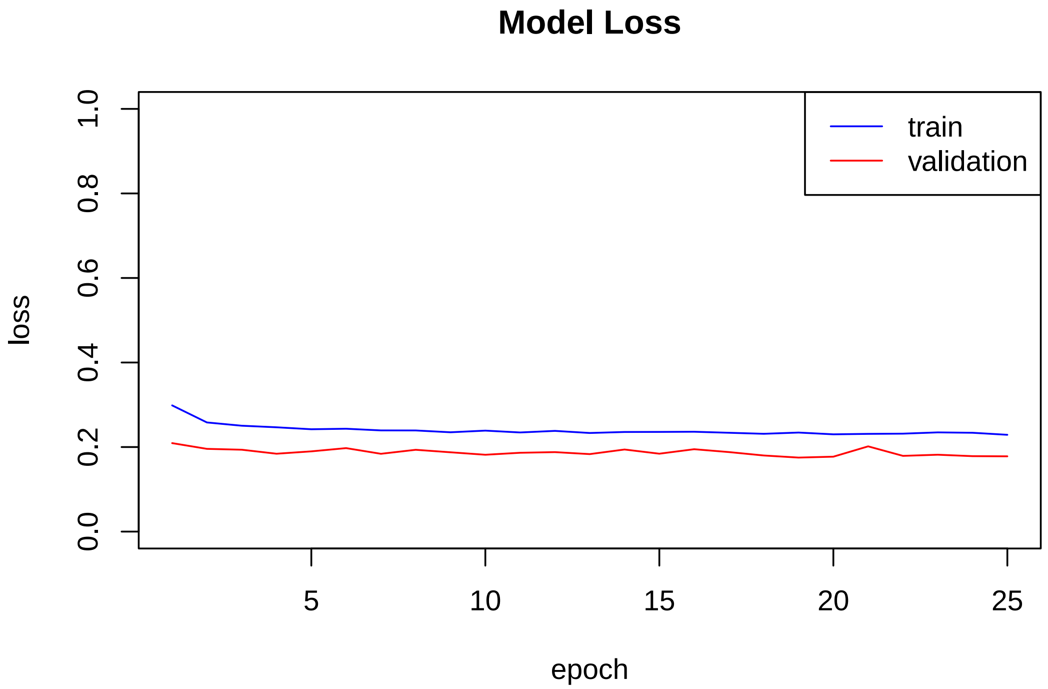
- input with 34 nodes,
- 3 hidden ones with 24, 16, 8 nodes respectively,
- output with 2 nodes.

In each layer we have used dropout = 0.3 and ReLU (Rectified Linear Unit) activation function.

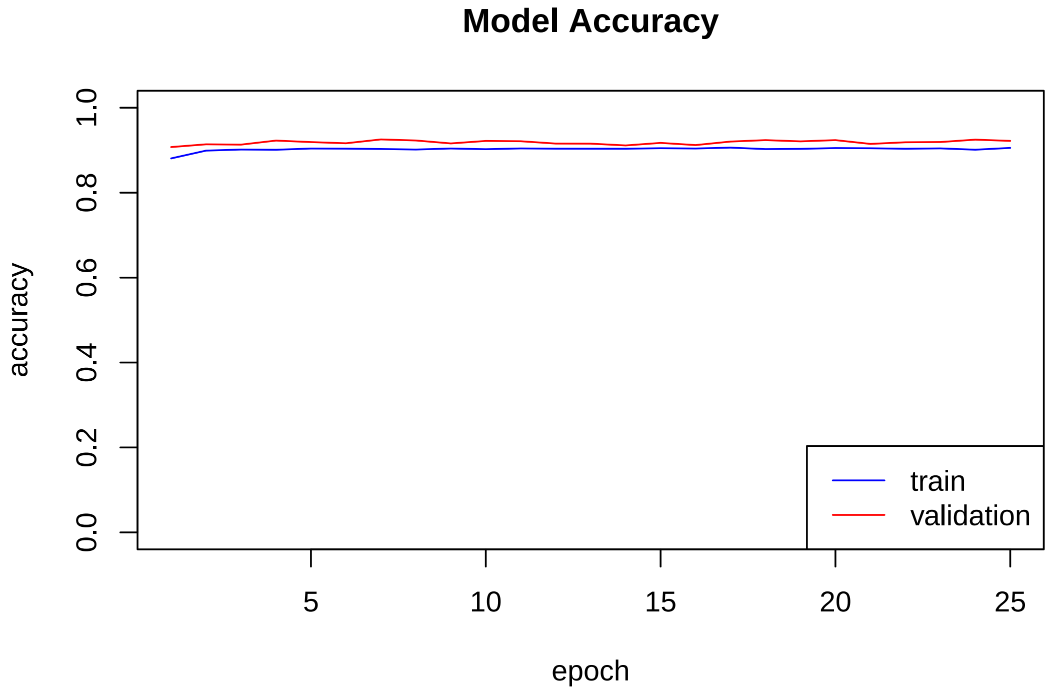
```
create_nn_model <- function() {
  nn_model <- keras_model_sequential() %>%
    layer_dense(units = 34, activation = 'relu',
                input_shape = dim(train_input_x)[2]) %>%
    layer_dropout(rate = 0.3) %>%
    layer_dense(units = 24, activation = 'relu') %>%
    layer_dropout(rate = 0.3) %>%
    layer_dense(units = 16, activation = 'relu') %>%
    layer_dropout(rate = 0.3) %>%
    layer_dense(units = 8, activation = 'relu') %>%
    layer_dropout(rate = 0.3) %>%
    layer_dense(units = 2, activation = 'softmax')
  nn_model %>% compile(
    loss = 'categorical_crossentropy',
    optimizer = optimizer_adam(lr = 0.01),
    metrics = c('accuracy')
  )
  nn_model
}
nn_model <- create_nn_model()
```

5.3.1 Fitting and performance

We performed model matching on binned and encoded data in 50 epochs.



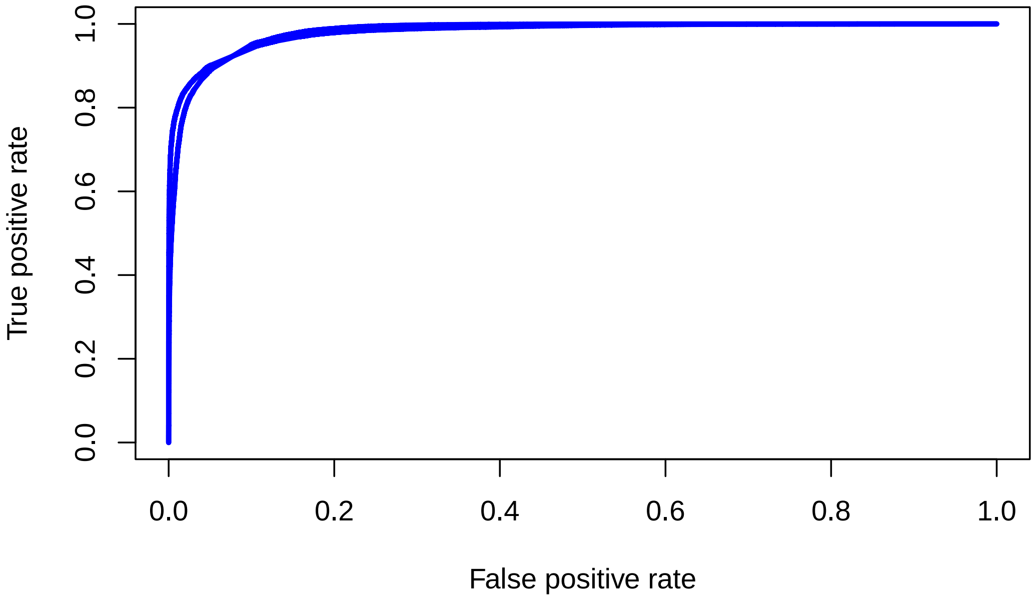
Model Loss

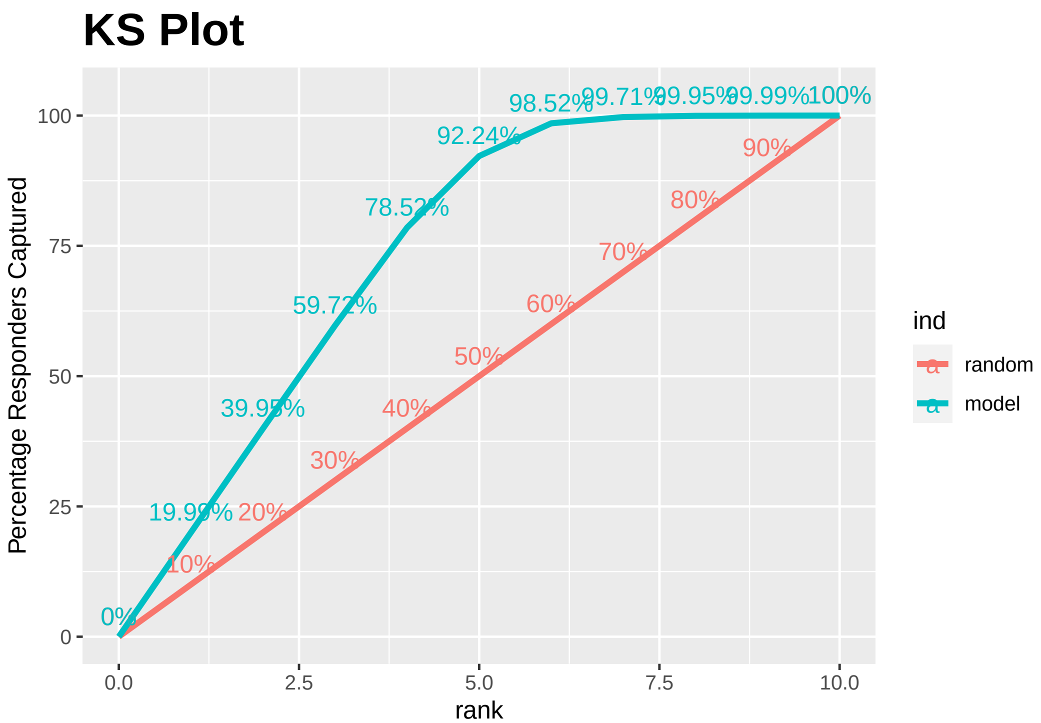


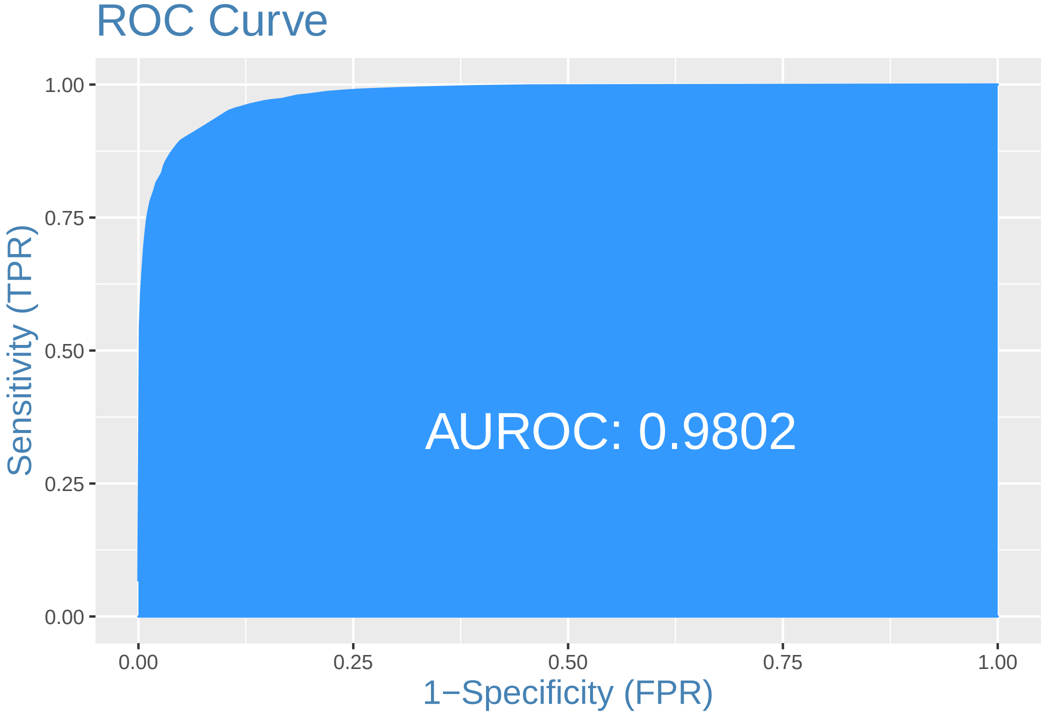
Model Accuracy

Surprisingly, the neural network has achieved most of its performance after only one epoch. This can be seen in the charts above.

```
##  [1] 0.9735321 0.9795128 0.9738887 0.9848778 0.9844361 0.9819136 0.9809298
##  [8] 0.9824050 0.9822437 0.9811294 0.9745668 0.9805752 0.9786453 0.9809332
## [15] 0.9779799 0.9749139 0.9852288 0.9762452 0.9832038 0.9790004
```

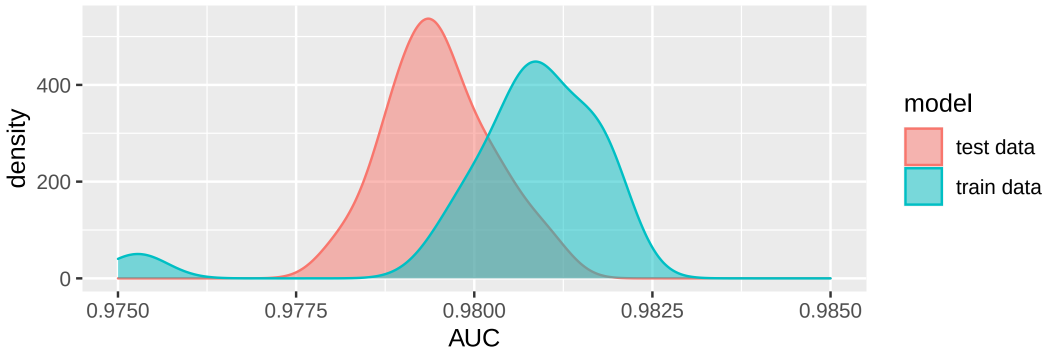


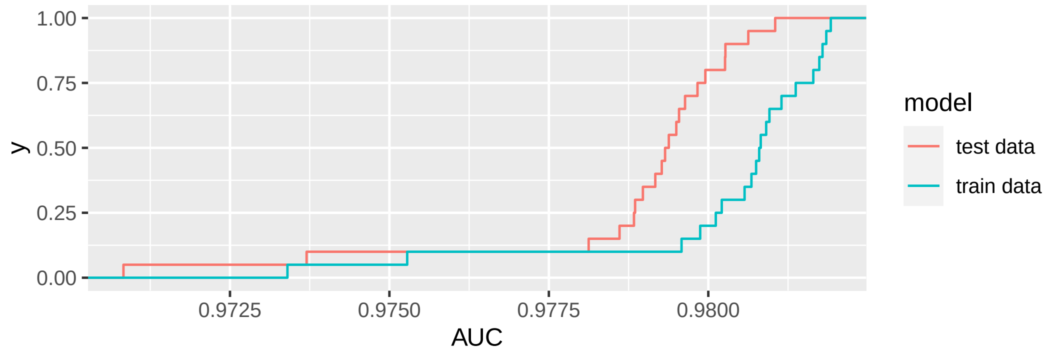

KS Plot
Percentage Responders Captured
100
75
50
25
0
0.0
2.5
5.0
7.5
10.0
rank
0%
19.99%
39.95%
59.72%
78.52%
92.24%
98.52%
99.71%
99.95%
99.99%
100%
10%
20%
30%
40%
50%
60%
70%
80%
90%
ind
random
model


ROC Curve
Sensitivity (TPR)
1.00
0.75
0.50
0.25
0.00
0.00
0.25
0.50
0.75
1.00
1−Specificity (FPR)
AUROC: 0.9802

5.3.2 Validation

To keep the convention we have also used Monte Carlo Cross Validation despite the high computational cost.





Validation showed that the model is highly effective.

The AUC for the neural network model on training data has: * the median equal to 0.9808101 for train data and 0.9793519 for test data, * the average equal to 0.9802706 for train data and 0.9787867 for test data, * the standard deviation equal to 0.0021555 for train data and 0.0023823 for test data.

6 Conclusion

In the plots, we showed the accuracy of models with the data they had not seen before. As we can see, all challangers models performed well. The neural network model is the most accurate, but we recommend logistic regression 2 for implementation - It is transparent - its operation can be easily explained to the customer. Moreover, the effects achieved by logistic regression are only

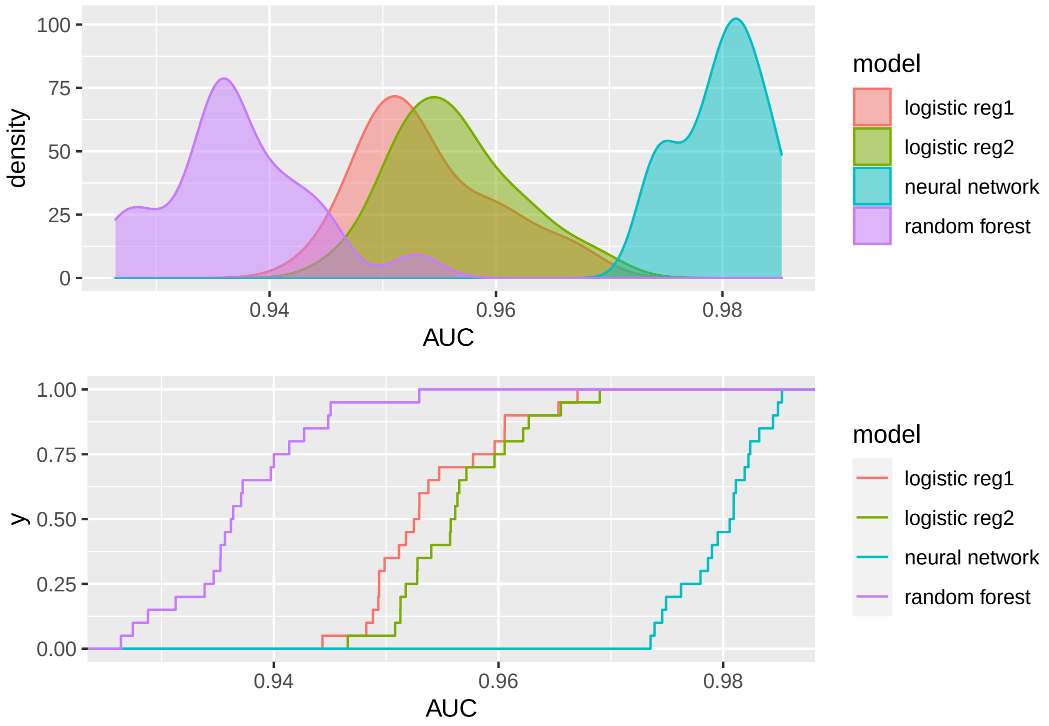


Figure 5: The kernel density for the observed areas under the curve (top) and the cumulative probability density functions (bottom)for the challenger models. All AUCs shown are for the test data only.

slightly worse.

In the following table we summarise these results:

Model	Mean AUC on Test Data	Mean AUC on Training Data
logistic regression 1	0.9539931	0.9542947
logistic regression 2	0.9564216	0.9571064
random forest	0.9371036	0.9672647
neural network	0.9798081	0.9795287

7 Bibliography

De Brouwer, Philippe J.S. 2020. The Big r-Book: From Data Science to Learning Machines and Big Data. New York: John Wiley & Sons, Ltd.