FLIGHT DELAYS PROJECT REPORT

HARVARD UNIVERSITY
Professional Certificate in Data Science
Capstone Project # 1

Andrés Amaya Chaves
July 21st, 2021

INTRODUCTION

This is a hands-on project that requires the application of previous knowledge and expertise about Data Science, mainly those in Machine Learning.

The present project uses a dataset called “Flight Delay and Causes”, which contains flights trip details and multiple causes of delay; it is a public dataset available in https://www.kaggle.com/undersc0re/flight-delay-and-causes (https://www.kaggle.com/undersc0re/flight-delay-and-causes), so it is not licensed by anyone.

The following **Table 1** names the variables that compose the raw data of this public dataset, its meaning, and its units.

Table 1. Variables of the raw dataset “Flight Delay and Causes”

VARIABLE	DESCRIPTION
DayOfWeek	1 (Monday) - 7 (Sunday)
Date	Scheduled date
DepTime	Actual departure time (local, hhmm)
ArrTime	Actual arrival time (local, hhmm)
CRSArrTime	Scheduled arrival time (local, hhmm)
UniqueCarrier	Unique carrier code
Airline	Airline company
FlightNum	flight number
TailNum	plane tail number
ActualElapsedTime	Actual time an airplane spends in the air(in minutes) with TaxiIn/Out
CRSElapsedTime	CRSElapsedTime
AirTime	CRS Elapsed Time of Flight (estimated elapse time), in minutes
ArrDelay	Flight Time (in minutes)
DepDelay	Difference in minutes between scheduled and actual arrival time
Origin	Origin IATA(International Air Transport Association) airport code
Org_Airport	Origin Airport Name
Dest	Destination IATA code

VARIABLE	DESCRIPTION
Dest_Airport	Destination Airport Name
Distance	Distance between airports (miles)
TaxiIn	Wheels down and arrival at the destination airport gate, in minutes
TaxiOut	The time elapsed between departure from the origin airport gate and wheels off, in minutes
Cancelled	Was the flight canceled?
CancellationCode	Reason for cancellation
Diverted	1 = yes, 0 = no
CarrierDelay	Flight delay due to carrier(e.g. maintenance or crew problems, aircraft cleaning, fueling, etc), 0 = No, yes = (in minutes)
WeatherDelay	Flight delay due to weather, 0 = No, yes = (in minutes)
NASDelay	Flight delay by NSA(National Aviation System), 0 = No, yes = (in minutes)
SecurityDelay	Flight delay by this reason, 0 = No, yes = (in minutes)
LateAircraftDelay	Flight delay by this reason, 0 = No, yes = (in minutes)

The raw dataset, *"Flight_delay"*, includes **484551** entries that have a valid value for each of the **29** latter described in **Table 1**, which outputs a total of **14051979** flights that took place between **274** unique origin/destination cities, performed from **01-01-2019** to **31-05-2019**. The original file weighs about 89.1 MB.

The present project employs a set of six *K-Nearest Neighbors* models that differ in the k-value used to train the data, from which that yielding the lowest RMSE (Residual Mean Squared Error) error was selected to predict the 'Delay Range' over the previously created test set and then over the validation set.

For the training of the six *K-Nearest Neighbors* models, it was necessary to randomly sample a subset of 75% of the original entries (rows), due to a lack of enough computational power to accomplish the calculations within a reasonable time.

For that same cause, it was not predicted over the 'Delay' variable, but instead, it was predicted another variable mutated from the dataset, in which delays were defined to be within 15-minute ranges, called 'DelayRange'. This way there was a reduction in the number of levels of the training set and its corresponding predictions, decreasing the computational cost.

The *"Flight_delay"* dataset was split into a validation set, with a 10% size of the total entries, and a train-test dataset called *"subFlights"* with the remaining 90%, for the development, test, and application of the machine learning algorithm. The *"subFlights"* data set was also split, this time into training and test sets.

All K-nn models were built to make predictions of the Delay Range on a 15-minute basis, using the airline, flight distance, and the "weekSlot"[1] as predictors.

[1] This is a number that refers to one of the 168 time-spaces of 60 minutes that compose a week.

The goal of this capstone project is to reinforce, synthesize and apply the concepts learned all along with the **Professional Certificate Program**, as well as to have a rewarding experience with the development of a machine learning algorithm large real-world dataset.

METHODS

The gathering of the raw data was performed with a 'download.file' statement over a temp file, from a *Google Drive URL* that directly downloads a copy of the original dataset. The *Kaggle* webpage does not allow a direct download without previous authentication, so the copy was made to keep the project reproducible. Once downloaded the data, a 'fread' function with a nested 'readLines' function built the data frame 'flights_raw', containing the variables in **Table 1**, above.

After selecting some variables and mutating some others, the 'createDataPartition' function from the 'caret' library was used to split the data: taking 10% to the validation set, and the remaining 90% to the training-test set, called 'subFlights' in code. After the initial 90-10% splitting, the 'subFlights' data frame was also split between test and training sets in 20% and 80% of the data, respectively.

The code was developed in 4 stages (from stage 0 to stage 3); next, in **Table 2**, there is a summary of the structure and the 36 steps taken to accomplish the preparation, creation, test, and validation of the model.

Table 2. Structure and steps to develop the machine learning model.

Steps	Description
STAGE 0	DATA GATHERING
Step 0.1	Install needed packages
Step 0.2	Call needed libraries
Step 0.3	Gather raw DATA
STAGE 1	DATA CLEANSING AND PREPARATION
Step 1.1	Cleanse the data
Step 1.1.1	Select, format and mutate some columns
Step 1.1.2	Histogram of the departure time distribution over 60min slots over any day.
Step 1.1.3	Histogram of the departure distribution over the days of week.
Step 1.2	Select, format and mutate some columns
Step 1.2.1	'Build the 'Weekslot' column from 'DayOfWeek' and 'DepTimeSlot'
Step 1.2.2	Eliminate 'Day of week', 'DepTime' and 'DepTimeSlot'
Step 1.2.3	'DelayRange', rounding to the upper 15 minutes ranges
Step 1.2.4	Replace Airline character names for integers folowing the 'AirlineCodes'
Step 1.2.5	Histogram of the flight delay distribution, in minutes.
Step 1.3	Data splitting
Step 1.3.1	'flights' splitting into validation set and subFlight set
Step 1.3.2	'subFlights' splitting into training and test sets
Step 1.4	Sample 75% of Train and Test sets.
Step 1.4.1	Randomly sample 75% of the Train Set entries

Steps	Description
Step 1.4.2	Sample the Test Set in a tenth of the Train Set sample size.
Step 1.5	Define function for the later calculation of Residual Mean Squared Error
STAGE 2	TRAIN AND PREDICTION OF K.NEAREST NEIGHBORS MODELS
Step 2.1	Train over the 'flTrainSample' dataset with 6 values of k.
Step 2.2	Predict over 'flTestSample' (the TEST SET SAMPLE), with the just trained knn models and create a data frame with the predictions.
Step 2.3	Calculate RMSE for the 6 sets of predictions an build a data frame with them.
Step 2.4	Predict over the TEST SAMPLE SET
Step 2.4.1	Calculate prediction accuracies of all built models over the TEST SAMPLE SET.
Step 2.4.2	Determine the index of the max accuracy, and its corresponding k-value.
Step 2.4.3	Show the Accuracy and RMSE results as a Knitr table, for the built models so far.
Step 2.5	Show the results of the model that yields the highest accuracy, and that with the lowest RMSE.
Step 2.6	Predict over 'flightsTest' (the TEST SET) with the trained knn model that minimizes RMSE error.
Step 2.6.1	Calculate the corresponding RMSE and Accuracy.
Step 2.6.2	Report RMSE and Accuracy for the predictions over the TEST SET.
STAGE 3	PREDICT OVER VALIDATION SET
Step 3.1	Predict over 'validaton' set, with the best knn model (that minimizes RMSE).
Step 3.2	Calculate prediction RMSE error.
Step 3.3	Calculate prediction accuracies of all built models over the TEST SAMPLE SET.
Step 3.4	Report Accuracy and RMSE of the predictions over the validation set.
Step 3.5	For the predictions over the validation set, report Accuracy, RMSE, Data Range Size, and the proportion of RMSE relating to data range.)

The project process followed exactly the steps shown in **Table 2**. After the cleansing and preparation of the data, a 75% random sample of the training set was set apart to perform the training of the models. In Stage 2, a set of 6 *K-Nearest Neighbors* models built over the training set sample was used to evaluate the convenience of their usage to predict over the test set.

The six k-values used were: k = {3, 5, 7, 9, 11, 13}. Each model was tested over a 75% random sample of the test set.

Among the tested models, the **3-Nearest Neighbors** gave the lowest RMSE error, while also yielded the lowest accuracy, on the other hand, the **13-Nearest Neighbors** exhibited the highest RMSE error, while had the highest accuracy. It was taken the decision to go forward with the model getting the lowest RMSE error, namely, the **3-Nearest Neighbors** one.

Once selected this way, the **3-Nearest Neighbors** model was employed to make the proper prediction of the 'DelayRange' variable over the test set, and later over the validation set. In the next section, we can see some metrics that tell us how well the built model performs and other results.

RESULTS

As an overview, in **Table 3** there is a summary of the dimensions of each of the 9 datasets with which the code deals.

Table 3. Dimensions of datasets

Denomination	Dataset	Dimensions_Rows_x_Columns
Main	**flights_raw**	484551 x 29
1	**preFlights**	484551 x 6
2	**flights**	484548 x 5
2.1	**validation**	484548 x 5
2.2	**subFlights**	436110 x 5
2.2.1	**flightsTest**	436092 x 5
2.2.1.1	**flTestSample**	87223 x 5
2.2.2	**flightsTrain**	348877 x 5
2.2.2.1	**flTrainSample**	261665 x 5

From the nearly-raw data (there was not any data partition yet), was created the following two histograms (see **Figure 1**) that show the distribution of the flights' departure times over a 24-slot graph representing the 24 hours of a day (see **Figure 2**), and a 7-slot graph, related with the days of the week in which every flight was performed (in the whole data).

In **Figure 1**, every flight departing between 0:00 and 0:59, regardless of the date, is counted in slot '0', those within 1:00 and 1:59 belong to slot '1', and so on until those departing at 23:00 to 23:59.

In **Figure 2** we can see the departure distribution on the seven days of the week. From this we can conclude that most of the registered flights, from **01-01-2019** to **31-05-2019**, were done on Thursday and Friday, summing up together 163983 flights in the period, which means 33.8% of the total flights, while Friday flights alone represent 18.4 % of them.

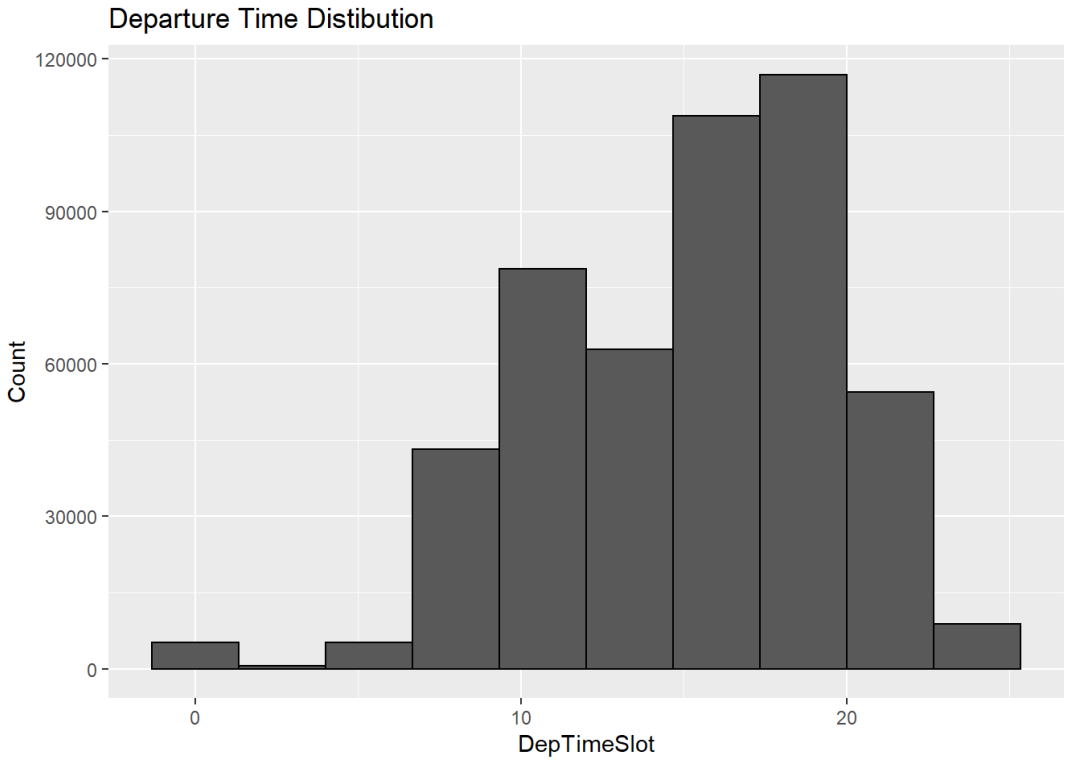

Figure 1. Departure time distribution in 24-hour slots.

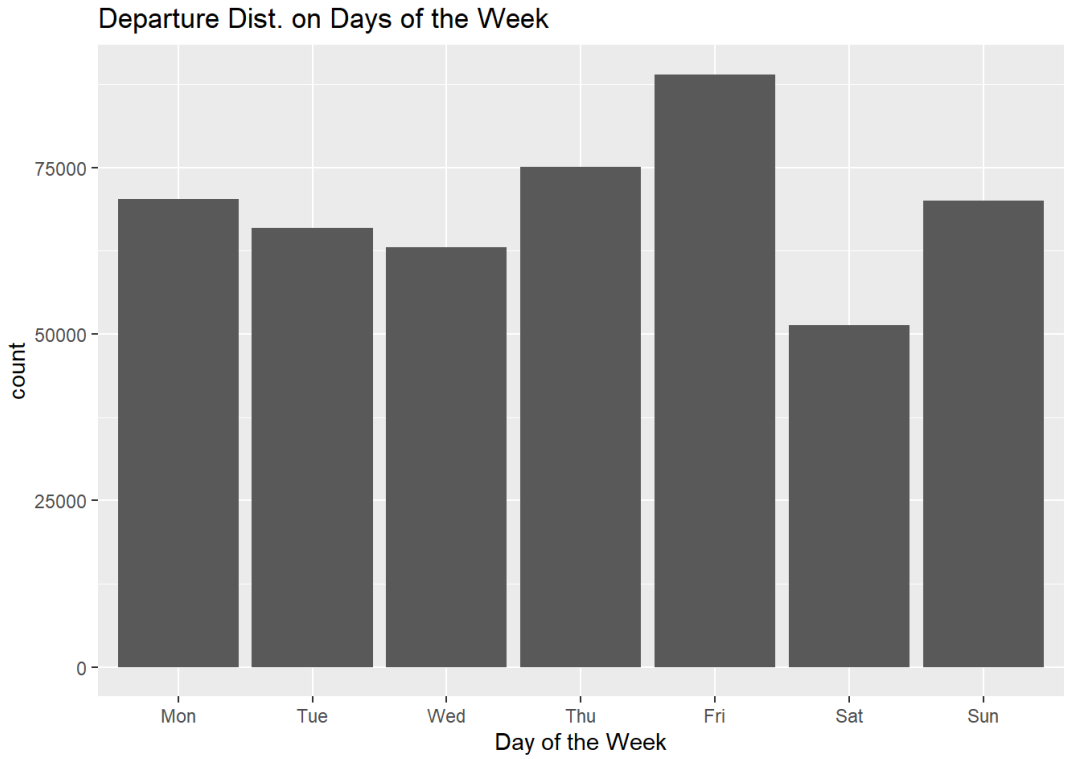

Figure 2. Departure time distribution on the 7 days of the week.

In **Table 4** we can see the count of flights by airline, following the codes given to the 12 airlines present in the dataset, arranged from the most common airline to that with the fewest number of flights registered.

Table 4. Count of flights by airline.

Code	Airline	Count	Percentage
1	Southwest Airlines Co.	119048	24.57%

Code	Airline	Count	Percentage
2	American Airlines Inc.	73053	15.08%
3	American Eagle Airlines Inc.	58698	12.11%
4	United Air Lines Inc.	56896	11.74%
5	Skywest Airlines Inc.	50384	10.4%
6	US Airways Inc.	31755	6.55%
7	Delta Air Lines Inc.	30220	6.24%
8	Atlantic Southeast Airlines	28678	5.92%
9	JetBlue Airways	15364	3.17%
10	Alaska Airlines Inc.	10000	2.06%
11	Frontier Airlines Inc.	9015	1.86%
12	Hawaiian Airlines Inc.	1440	0.3%

Next, we concentrate on the distribution of the most important variable in this project, 'Flight Delay', because it is the variable that all the built models will predict; in **Figure 3** we can see a histogram of its distribution. This variable corresponds to the time difference between the scheduled arrival time and the corresponding actual time for all flights.

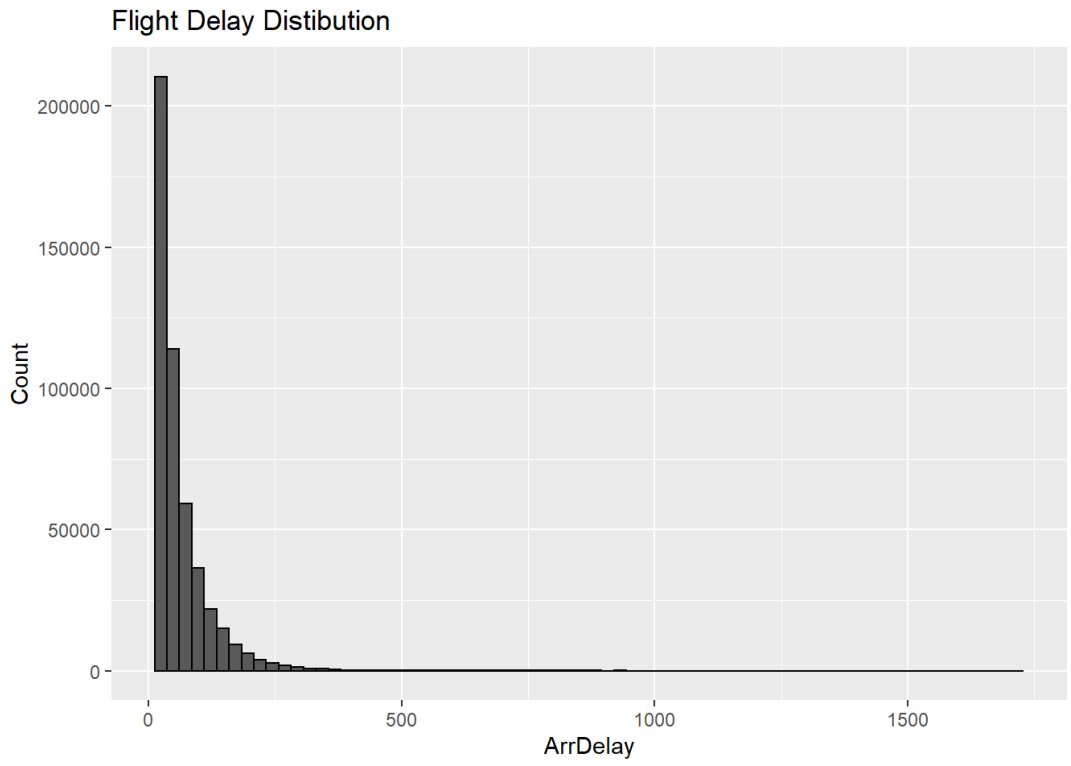

Figure 3. Flight delay Distribution.

After the mentioned split performed over the original dataset, the k-nn models, trained with a 75% random sample of the training set, were proven to perform predictions over a 75% random sample of the test set, giving the RMSE summarized in **Figure 4**, and the accuracies shown in **Figure 5**.

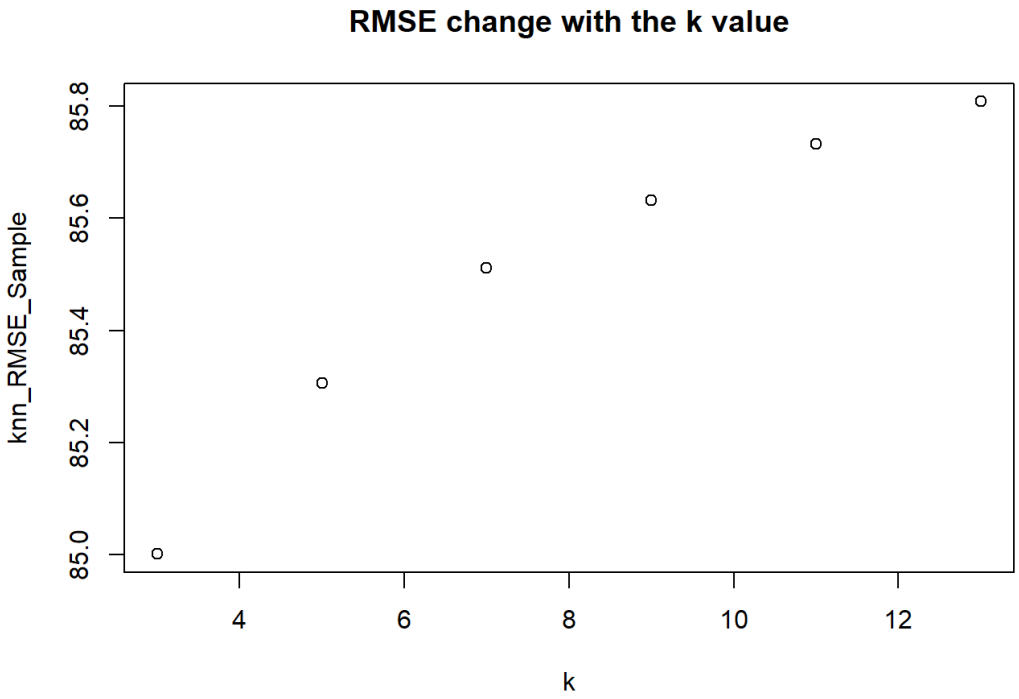

Figure 4. RMSE error of the six k-nn models, according to the k-value.

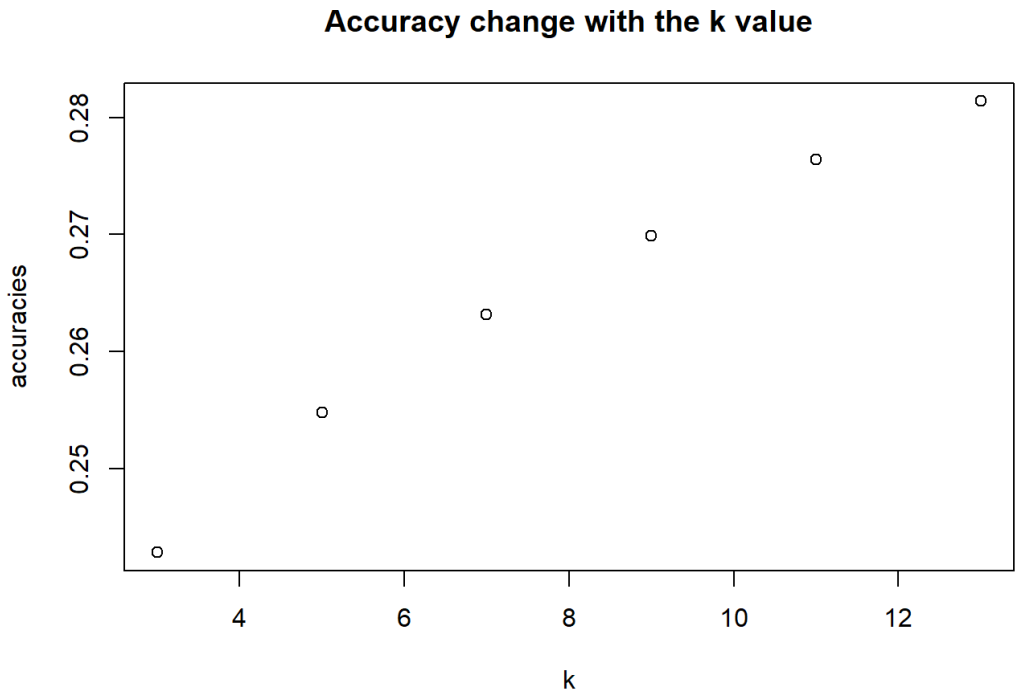

Figure 5. Accuracy of the six k-nn models, according to the k-value.

Table 5 summarizes the values of RMSE and accuracy for each of the applied machine learning models, over the sample test set, the test set, as well as over the validation set. Then, **Table 6** gives a summary of the best k-values found, which indicated the model to use over the (whole) test set.

Table 5. RMSE and accuracy values for each model.

Method	Accuracy	RMSE

Method	Accuracy	RMSE
3-nn Over Sample Test Set	0.2428	85.00153
5-nn Over Sample Test Set	0.2548	85.30602
7-nn Over Sample Test Set	0.2632	85.51130
9-nn Over Sample Test Set	0.2699	85.63245
11-nn Over Sample Test Set	0.2764	85.73238
13-nn Over Sample Test Set	0.2814	85.80806
3-nn Over TEST SET	0.2391	85.84650
3-nn Over VALIDATION SET	0.2409	84.35171

Table 6. RMSE and accuracy values for each model.

Best_Metric	Method	Accuracy	RMSE
Highest Accuracy	13-nn Over Sample Datset	0.2814	85.8081
Lowest RMSE	3-nn Over Sample Datset	0.2428	85.0015

As we can see in **Table 6**, in this problem a ***3-Nearest Neighbor*** model minimizes de RMSE error, while the trained ***13-Nearest Neighbor*** model maximizes the accuracy among the considered models. So, finally, the recently trained ***3-Nearest Neighbor*** model was used over the test set, and later over the validation set.

Note that the model was trained with a **75% random sample** of the training set, while it gave predictions over the whole test set, which just means that it was trained with less input information, therefore it could output slightly less accurate results, compared with hypothetically having trained with the whole training set.

So we can say that the built model makes predictions with an RMSE error of 85.00153 minutes over the validation set, which means that using this model there is a high probability that the error in single predictions of the *delay range* of flights is less or equal to +/- 85 minutes.

Since the ‘DelayRange’ variable ranges from 15 to 1140 minutes in the validation set, the latter RMSE error represents the 7.4979 % of that range, not a bad result for this machine learning model.

Finally, in **Table 7** we can see a summary of the metrics for the 3-Nearest Neighbors model applied over the test set and the validation set, to have an overview of its performance.

Table 7. Final performance summary.

Method	Accuracy	RMSE	Delay_min	Delay_max	Range_Size	Range_Percentage_RMSE
3nn Over TEST SET	0.2390	85.85	15	1155	1140	7.53 %
3nn Over VALIDATION SET	0.2409	84.35	15	1140	1125	7.5 %

Note that in the original raw data there is no negative value in the ‘ArrTime’ variable, its range is between 1 and 2400 minutes, therefore no registered flight arrived before the scheduled time. This does not imply that the prediction error cannot be negative, because any prediction can output a lower value than the actual delay range.

OBSERVATIONS

• From **Figure 1**, we can say that most of the flights are performed departing between 15:00 and 20:00, regardless of the airline, the distance of the flight, or the day of the week.

• From **Figure 2**, we can conclude that Thursday and Friday are the most common days of the week in which flights are done.

• From **Table 4**, it is observed that Southwest Airlines Co. operates most of the flights registered in the dataset, while American Airlines Inc. is second in the number of flights.

• From **Figure 3**, we can see a confidence interval between 0 and 250 minutes of flight delay, yielding a high probability. There is no flight with negative delay (arriving before the scheduled time).

• **Figures 4 and 5**, and **Table 5**, show us that with this dataset, increasing the number of neighbors of the k-nn model produces a higher RMSE error but a higher accuracy in the predictions.

CONCLUSIONS

• The built 3-Nearest Neighbors model was capable of making predictions over the test set and the validation set, with an RMSE error of 85.85 minutes and 84.35 minutes, respectively; which represents **7.53%**, and **7.50%** of the respective range sizes.

• Therefore, the trained 3-Nearest Neighbors model is capable of predicting the arrival delay range of flights, on a 15-minute basis, with an error below 90 minutes, provided the airline, the flight distance, and the time & day of the week.

• The model outputs predictions with higher accuracy and lower RMSE error when applied over the test set than when applied over the validation set.

• However, the RMSE is an absolute measure; when we calculate the proportion of RMSE error regarding the range size, we find that the model performs better in the validation set (commits less relative error), which has about 48450 more rows than its counterpart (test set is 10% smaller than validation set).

• The model was trained with a 75% sample of the training set, which barely represents 54% of the original number of rows (484551).

• While proving the K-values, an increment of them resulted consistently in more RMSE error but more accuracy, when predicting over the test set, as we can see in **Figures 4 and 5**. The lowest RMSE error was preferred.