Ryanair Ancillary Revenue Model for Summer Flights

Kevin Sheahen

—

3/30/2023

—

Introduction

In this report I'll walk the reader through my analysis of Ryanair's 2022 flight data and the projection I developed for a sample of 2023 flights. The software used in this document is Python through the Jupyter terminal and the data is loaded from csv files. Comments can be seen throughout the code with the hope in guiding users in the process and reasoning for each action. The code is divided into three sections: 1) Importing the Data, Packages, Cleaning and Merging, 2) Exploratory Data Analysis/Visualization and Feature Engineering and 3) Training/Testing a Machine Learning Model and Applying it. Please use the code and csv of my projections alongside this report to help understand the submission. If there should be any confusion I'd be happy to clarify the question.

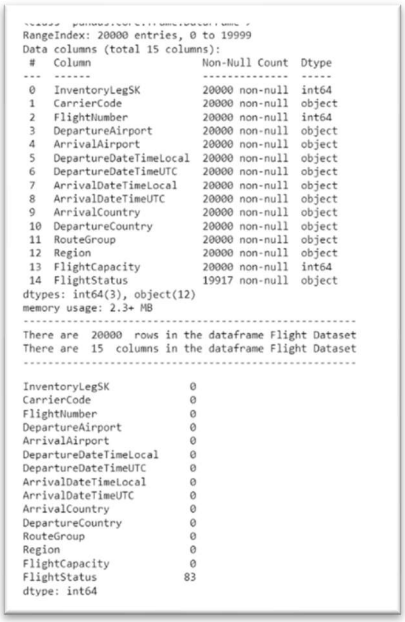
Importation

To begin building the model it requires a series of packages, and settings configured. To briefly explain the packages loaded include: pandas, numpy, matplotlib, os seaborn, scipy, warnings, and sklearn. Additionally, I chose to set the max column and row display to None as I personally would rather code with the .head() function and be able to easily read the column names. Warnings are ignored for improved reliability of myself and users.

Cleaning

To begin the data exploration and cleaning a series of steps must be performed. First, I import the three datasets, assign them names using the .name() function and examine the dimensionality of the dataframe. This gives me a frame of reference for how large of data we are dealing with and upon further examination the sales data frame seems to be transactional in relation to the flight data frame. So in later steps I'll need to aggregate the sales data in order to join it with the flight data set.

For now I run the same code on all three data frames that produce comparable results to the image below. From this I determine if there are duplicates, where our Null data is, and the data type of each column which will be helpful information when transforming data.



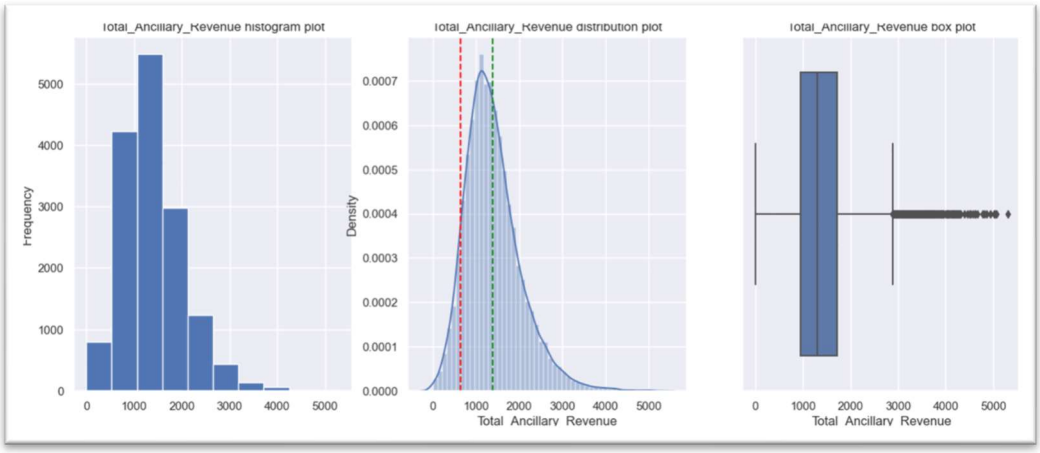
```
RangeIndex: 20000 entries, 0 to 19999
Data columns (total 15 columns):
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
 0   InventoryLegSK          20000 non-null  int64
 1   CarrierCode             20000 non-null  object
 2   FlightNumber            20000 non-null  int64
 3   DepartureAirport        20000 non-null  object
 4   ArrivalAirport          20000 non-null  object
 5   DepartureDateTimeLocal  20000 non-null  object
 6   DepartureDateTimeUTC    20000 non-null  object
 7   ArrivalDateTimeLocal    20000 non-null  object
 8   ArrivalDateTimeUTC      20000 non-null  object
 9   ArrivalCountry          20000 non-null  object
 10  DepartureCountry        20000 non-null  object
 11  RouteGroup              20000 non-null  object
 12  Region                  20000 non-null  object
 13  FlightCapacity          20000 non-null  int64
 14  FlightStatus            19917 non-null  object
dtypes: int64(3), object(12)
memory usage: 2.3+ MB
-------------------------------------------------------
There are  20000  rows in the dataframe Flight Dataset
There are  15  columns in the dataframe Flight Dataset
-------------------------------------------------------

InventoryLegSK            0
CarrierCode               0
FlightNumber              0
DepartureAirport          0
ArrivalAirport            0
DepartureDateTimeLocal    0
DepartureDateTimeUTC      0
ArrivalDateTimeLocal      0
ArrivalDateTimeUTC        0
ArrivalCountry            0
DepartureCountry          0
RouteGroup                0
Region                    0
FlightCapacity            0
FlightStatus             83
dtype: int64
```

The Flight dataset contained 83 null values that in the FlightStatus column that were removed along with flights that had a status of cancelled. For the sales dataset I removed all instances of null values across the Priority_Boarding_UnitsSold, Priority_Boarding_RevenueEUR, 20KG_Bag_UnitsSold, and 20KG_Bag_RevenueEUR columns. The reason behind this is that in total it represented just 13,109 rows out of 990,568. Additionally, if I was to replace the NaN's with a value of 0 or the mean I believe I might skew the data too much or give too much importance to the mean. Checking the Validation set we can see that there are no missing data values and does not need any cleaning done. There are two main items to complete before I can visualize and explore the data deeper. Creating a "Total Ancillary Revenue" column and merging the flights and sales data frame. I quickly do this on the sales data frame so a user could also see the total revenue there before it's merged. Once completed I'll aggregate the sales data by InventoryLegSK before merging it to flights. This is because the data here is more granular than the flights data frame so in order to merge them together I have to bring it to a higher level view (by flight).
Additionally, a last check is performed on the Flight Status of the flight data frame. I've chosen to remove the cancelled flights, I would note that if we wanted to perform a further analysis on cancellation predictions or factoring that in to my current model it could be done. I would not that 2.45% of flights were cancelled and I will apply this in a limited way to the predictions later in this document. From my understand cancellations are hard to predict as many factors such as natural, political, human resources and others play a part in them.
Once executed we have a merged_df which has the Priority_Boarding_UnitsSold, Priority_Boarding_RevenueEUR, 20KG_Bag_UnitsSold, and 20KG_Bag_RevenueEUR, Total Ancillary Revenue and others by flight. This can then also be modified to aggregate the flights by day, airports, countries, regions, etc. which is done in later steps for visualizations purposes.

Exploration

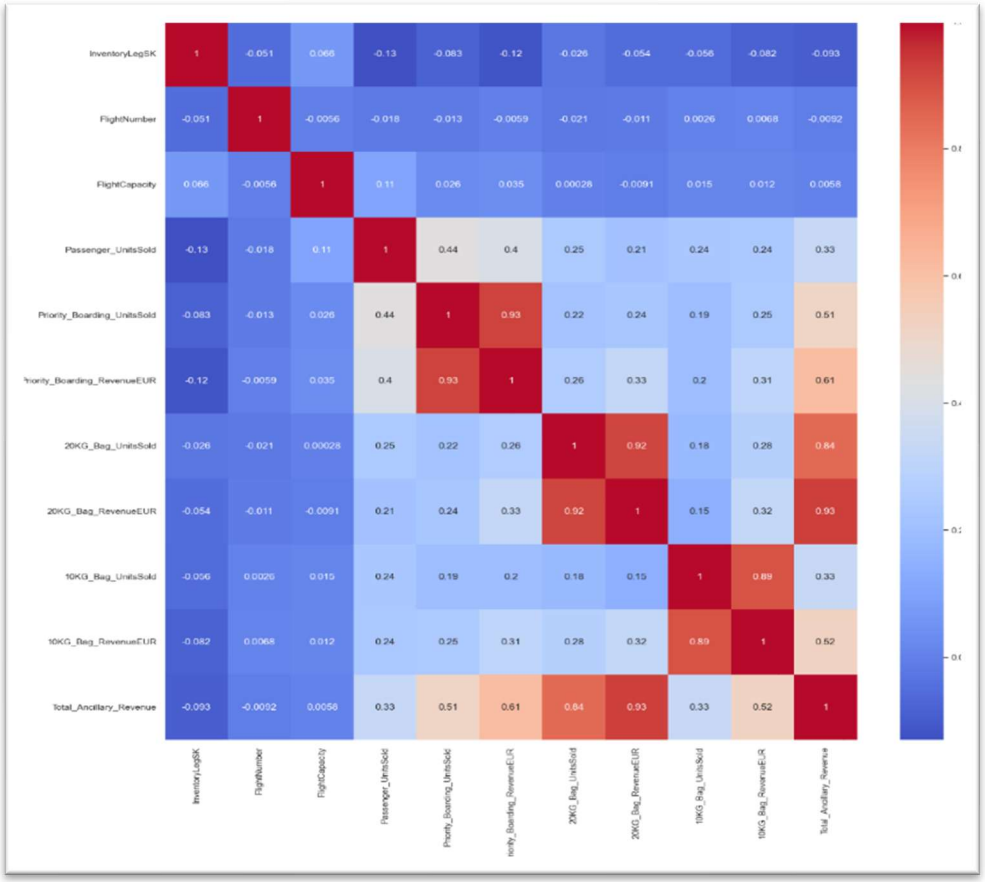
The various graphs below are intended to show the different information from the merged_df.



A)

a. Our histogram, distribution and box plots of the Total Ancillary Revenue give information on the data at a high level. The histogram and distribution plot indicate the data is skewed left. The red line is the mean and the green is our standard deviation respectively. Additional we learn from the box plot that the there seem to be a considerable amount of high level outliers past €3,000 for the Total Ancillary Revenue. It will be interesting to see if the model can predict these well.

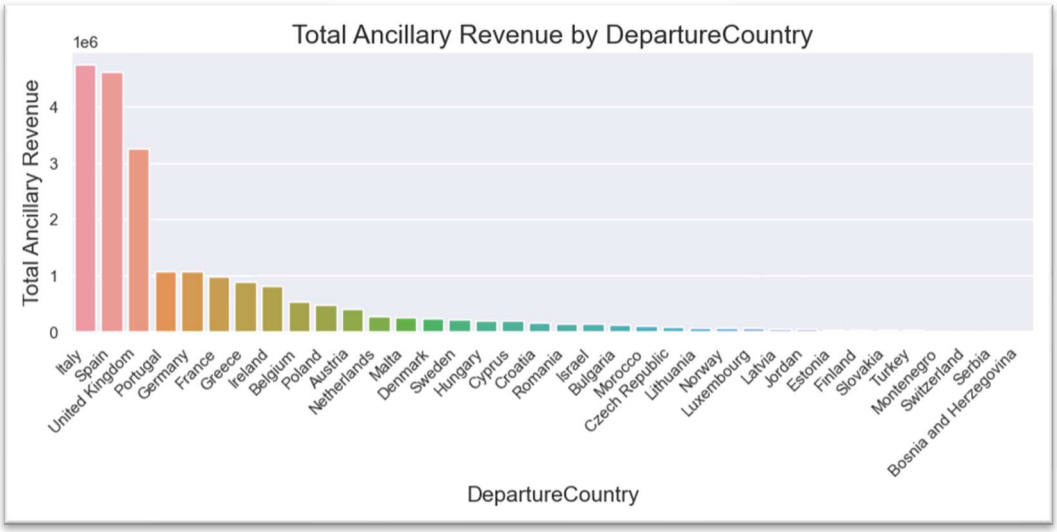
b. These visuals are available for the other columns (20KG, 10KG, Passenger Units Sold, etc.) in the code.



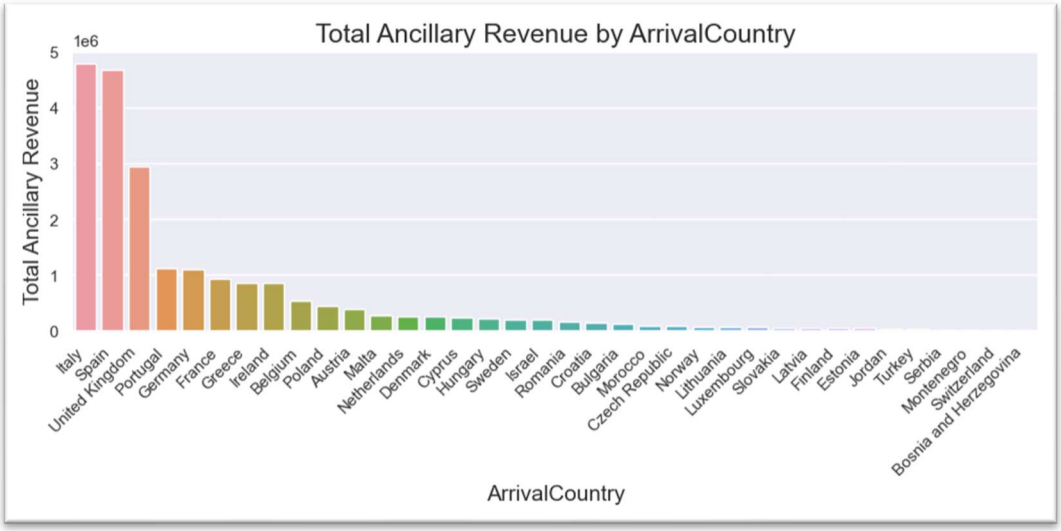
B)

a. From this heatmap we notice besides the obvious correlations there are some slight correlations between the ancillary items for purchase. For

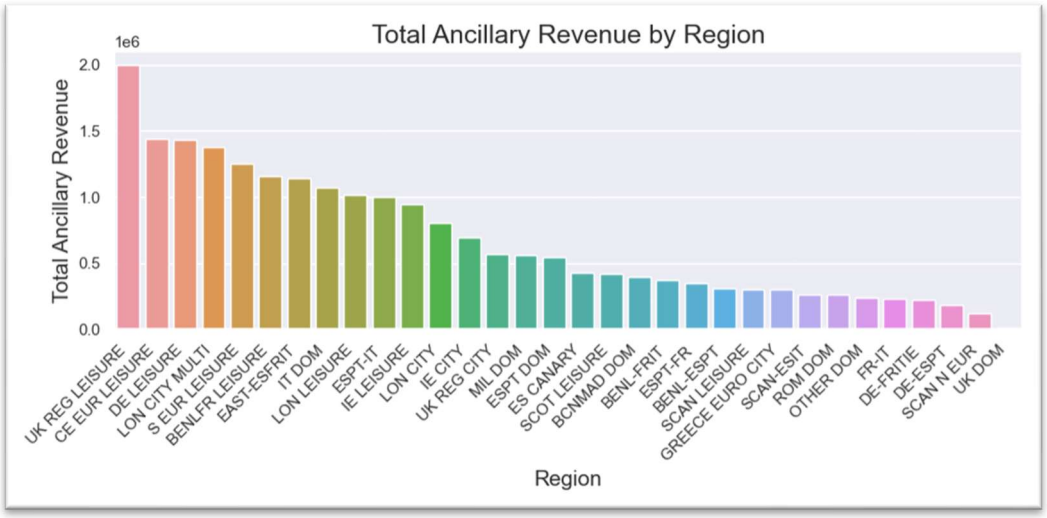
example, 20KG_Bag_UnitsSold has a positive relationship of 0.22 and 0.18 for Priority_Boarding_UnitsSold and 10KG_Bag_UnitsSold. These are relatively low but the takeaway that can be determined is that sales in one category can affect others in a positive manner.



C)

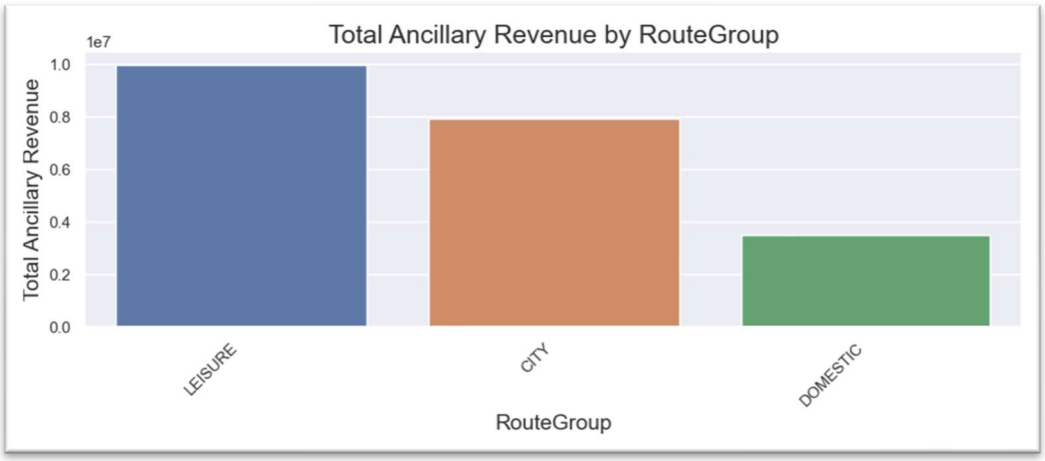


a. The two graphs above indicate the most common departure and arrival countries. From this it seems that countries a consistent rate of incoming and outgoing flights. No countries have massive deficits or surpluses between arrivals and departures. This makes sense because most likely Ryanair operates on the method of never flying flights back to hubs empty.



D)

a. Lastly, an important exploration and visual to display is the Total Ancillary Revenue by Region description.

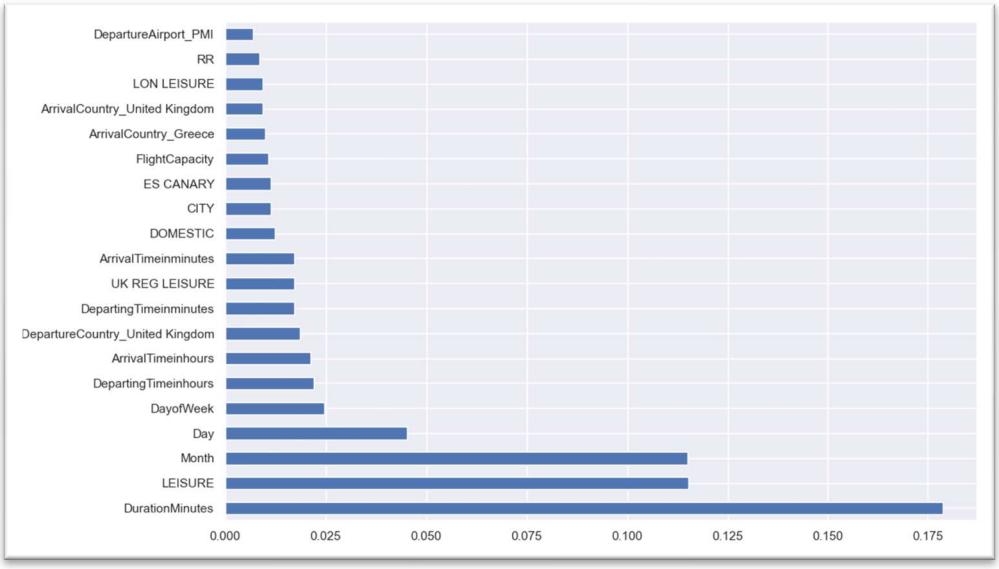


E)

a. Lastly, we can see that Leisure and City routes make up the most of the Ancillary Revenue with Domestic flights lagging. Hypothesizing, this might be because people are travelling home from within there own country. They wouldn't need clothes or other essentials with them if they have them waiting for them.

Features

As it stands the data can't be used in a machine learning so I performed hot-encoding to change the written data into numeric data. This caused a large split out to occur producing over 500 columns. These additional columns came from departure airport, arrival airport, departure country, arrival country, group, route, and carrier. My thinking behind hot-encoding vs label-encoding was that the data in question was not ordinal and had no inherent ranking. Thus I believed there was an advantage to using hot-encoding. So the feature selection included the split outs mentioned above, FlightCapacity, Year, Month, Day, DayofWeek, DepartingTimeinhours, DepartingTimeinminutes, ArrivalTimeinhours, ArrivalTimeinminutes and DurationMinutes. Below we can see the 20 most important features for the model.



I did note that keeping Days and the Minutes fields for arrival and departure might be unnecessary. However, I believed in the summer months there could be more importance when transitioning from early months to middle months. Some further analysis might be required if this is truly beneficial to keep in the model.
When I originally ran my model it would take up to 18 minutes for the model to complete, which is expected for a data frame that measures ~15,000 rows by ~500 columns. So in order to overcome this inefficiency I employed Principal Component Analysis. This produce a smaller subset of important features, by factoring in many features to create "new features". This proved effective as the run time dropped to only 2 ½ minutes. Most importantly it produced more accurate results leading to the idea that there was too much "noise" previously being fed into the model.

Justification and Reasoning of Model

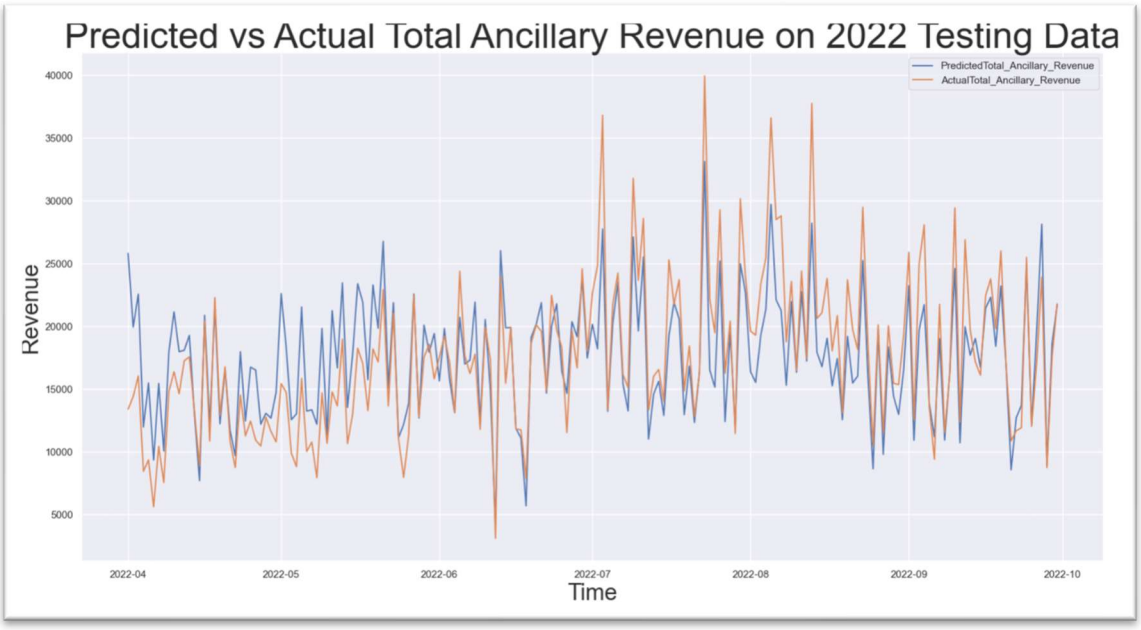
The algorithm I used for this model to help with finding the best answer was Random Forest Regression. Though I could use Linear Regression because it is advantageous for being simple and fast. However, it is limited because of its assumption of linearity meaning that if our dataset has a non-linear relationship it will fail to detect it. Therefore, I found it useful to employ Random Forest Regression which has the advantages of modelling non-linear relationships, robustness to overfitting, and an ensemble of decision trees. By using multiple decision tree predictions and taking the averages of them this improves accuracy and reduces overfitting.

Though Random Forest Regression uses a considerable amount of computational resources this was acceptable in my opinion as the predictions made in this model were not for a real time scenario. However, a completion time of 2 ½ minutes is fairly quick on such a large dataset using Random Forest Regression.

Training, Validation and General Performance

I performed the training and testing of the model on several y variables: 'Passenger_UnitsSold', 'Priority_Boarding_UnitsSold', 'Priority_Boarding_RevenueEUR', '20KG_Bag_UnitsSold', '20KG_Bag_RevenueEUR', '10KG_Bag_UnitsSold', '10KG_Bag_RevenueEUR', and 'Total_Ancillary_Revenue'. I thought this necessary as it would not only be helpful to see the total prediction of the Ancillary Revenue but the other fields might be of interest. I used an 85/15 train to test size ratio to create the appropriate split. I used 85/15 ratio instead of a standard 80/20 ratio because I wanted to give the model slightly more data to train on without compromising the size of the test dataset too much. The configuration of the Random Forest Regressor settings were: 500 n estimators, 7 maximum depths, and a minimum sample splits of 3. Though the n estimators which dictates the number of trees built before the average is accepted is high, I considered this acceptable as the predictions weren't needed in real time. Meaning that it would run for a considerable time but that is okay because the nature of the situation doesn't call for speedy response. Setting maximum depths at 7 meant that overfitting was limited. Lastly, having the minimum sample splits set to 3 meant that the model would produce many branches instead of a default of 20, the hope is that this would get more accurate results. Below is a table of an aggregated difference between my predicted values on the test set and the real values. As well as a timeseries that shows the predicted vs actual values over time. Generally, the model predicted lower values than the actual values. This can be expected as the Random Forest Algorithm will not project a number greater than the maximum in the training set.

Total_Ancillary_Revenue	Passenger_UnitsSold	Priority_Boarding_UnitsSold	Priority_Boarding_RevenueEUR	20KG_Bag_UnitsSold	20KG_Bag_RevenueEUR	10KG_Bag_UnitsSold	10KG_Bag_RevenueEUR
-26,809.83	-418.94	-246.95	-1,665.82	-234.10	-20,872.75	-192.71	-6,949.08



Additionally, iterating through the y variables produced print outs like the one below showing the important metrics of difference between the predicted and correct values. For example, from this I can determine the MAE is not to large considering the high revenue per flight, however where I think this model lacks is that it's R^2 is on the lower end at 35%. This might lead me to think that more relevant data is needed, and/or reduction in some features selected. On its own the MAPE is quite large and would mean that the predictions are off by enormous values. I think this might be the work of some large outliers as in the aggregate the projects are relatively close (mentioned in the next paragraph).

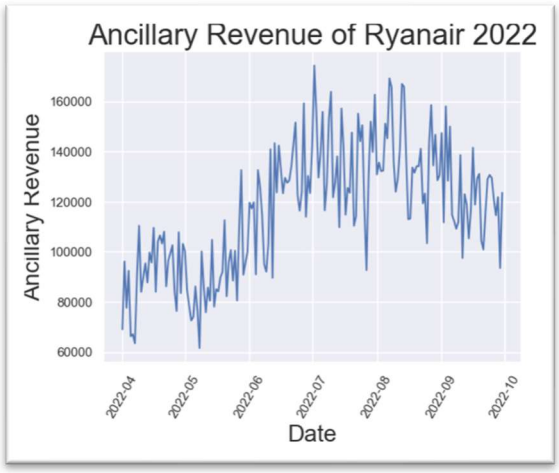
```
----------------------------
Mean Absolute Error (MAE): 387.23246735481666
Mean Squared Error (MSE): 270948.57668791956
Root Mean Squared Error (RMSE): 520.5272103242246
Mean Absolute Percentage Error (MAPE): 1.2732873671799988e+16
Explained Variance Score: 0.35049951311581484
Max Error: 3310.325112264037
Median Absolute Error: 300.3506985920553
R^2: 0.35014974685447686
Total_Ancillary_Revenue
----------------------------
```

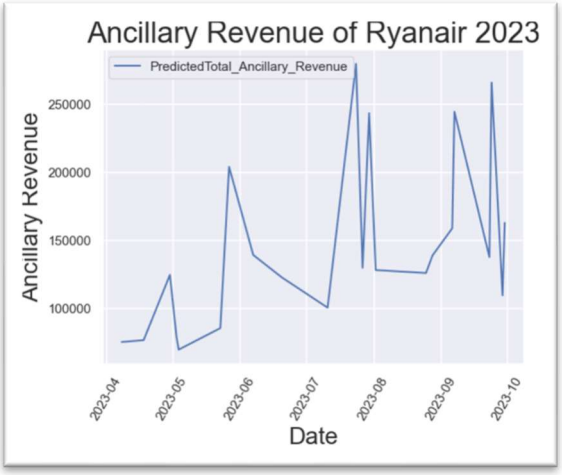
After this training and testing phase I moved forward and applied the model to the validation data set of planned summer flights. The predicted results are below in the table for their respective category. The table contains 2022 actuals, 2023 projected revenue, 2023 projected revenue considering the cancelation rate of 2.45%, and if the 2023 projected revenue was brought to scale of 2022 actuals (2022 sample size). To note there is a small difference in Total Ancillary Revenue and the aggregation of all the revenue categories. So, in total the model predicted that among the 1,924 flights planned this summer if they were to depart successfully it would generate ~€2.76 million in Ancillary Revenue and sell ~340,000 tickets. If cancelation was factored in then we would see ~€2.69 million in Ancillary Revenue and ~331,851 tickets sold.

Projection Table

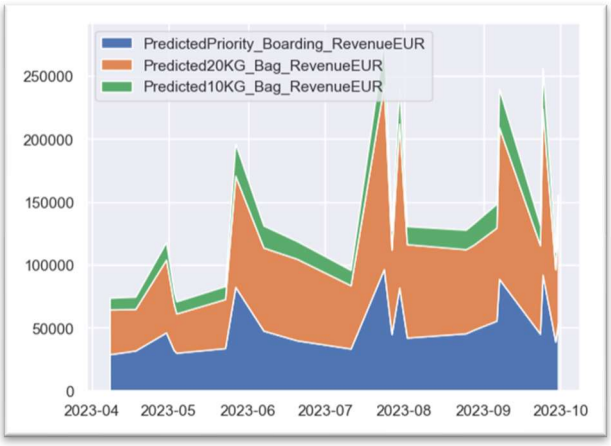
Scenario	Total_Ancillary_Revenue	Passenger_UnitsSold	Priority_Boarding_UnitsSold	Priority_Boarding_RevenueEUR	20KG_Bag_UnitsSold	20KG_Bag_RevenueEUR	10KG_Bag_UnitsSold	10KG_Bag_RevenueEUR	Sample Size (Flights)
2022 Actual	€21,372,874	2,748,030	508,341	€8,258,403	317,879	€10,443,264	129,085	€2,671,206	15,359
2023 Projection	€ 2,762,883	340,186	64,502	€1,063,747	42,051	€1,411,673	17,146	€362,241	1,924
2023 Projection (w/ Cancelations)	€ 2,695,192	331,851	62,921.	€ 1,037,685	41,021	€ 1,377,087	16,725	€353,366	1,876
2023 Projection on scale to 2022 Actuals	€ 22,055,675	2,715,657	514,910	€8,491,730	335,693	€11,269,171	136,875	€ 2,891,71	15,359

With further analysis we can see that on the scale of the 2022 dataset the models project expects there to be ~€22 million in Ancillary Revenue. It is important to note that this number can be skewed for a variety of reasons especially the sample of flights is biased. However, it is helpful to see what a theoretical number at full scale might be. Below is a side by side comparison of last years flight revenue by day compared to the projections of this year. There is a rough resemblance in performance, I believe if there was a larger validation set including more days then the two graphics would look even more similar. In conclusion, the model suggests similar earnings and metrics to the previous year.





Additionally attached is a plot of a stacked graph that shows the projected performance of the revenue generating categories (Below). The higher revenue generating mediums will be the add-ons of 10KG and 20KG bags. While Priority Boarding Revenue will be much lower but still contributing significantly.



The model completed both training and projections against the validation set in 2 ½ minutes. This is an improvement considering the size of the original dataset, the hot-encoding and the nature of the algorithm chosen. In total the program took a little over 7 minutes to complete.

Future Improvements

In order to improve the model I would incorporate more historical data beyond just one sample of summer flights. This would open up to even more accurate predictions and insights. Mainly because with more data offers the opportunity to use different algorithms that are designed for timeseries evaluations such as: ARIMA and Seasonal ARIMA. I believe these two would be extremely good at predicting the daily revenue but might not be the best for individual flight predictions. Additionally, more data to predict on would be beneficial as the predictions were only against a small set of upcoming flights this summer. Removing either departing and arriving airports or departing and arriving countries might be helpful as well. I've also mentioned the idea that trimming some of the date and time features might be helpful (day, departing/arrival minutes). This reduction could open the door to new features such as ones that give greater understanding to the date (i.e. national holidays, weather delays, etc.). Additionally, performing Pairwise Regression could be a good practice to include in the future. The practice could help to reduce the unnecessary features that overlap and are highly correlated. Therefore being another tool to improve accuracy and efficiency.

Lastly, I think external data could be helpful to paint a story. Either other data relating to cost of operations, revenue from tickets, etc. that could be gathered from Ryanair itself or data related to oil prices, geopolitics, and macro-economics that would typically come from 3rd party would be interesting. For other Ryanair specific data that would be helpful could be age, one-way/return tickets, and time of purchase. This might lead to new insights such as flights to the Mediterranean are purchases farther in advance or similarly age is helpful indicator of likelihood to purchase ancillary products.

Additional Visualizations