Seazone

Junior Data Scientist Challenge Report

Pedro Perdigão Drapier
Data Scientist

Fortaleza, CE
25/03/2022

1. Data Cleaning

All the work was done in Python using mostly Pandas, Numpy and Scikit-Learn, since most of the work was based on forecasting, and Python is great for both data transformation, EDA and building Machine Learning models.

Since the data cleaning is a common step for all of the coming questions, it will have its own section. In this section, I will describe the steps taken to clean the data (a data cleaning log, of sorts) and try to explain my decision making process.

Revenue Table

Cleaning this table was rather simple. All that was needed to do was casting the correct dtypes to the columns, filtering out rows where the creation date was bigger than the date field and deleting any duplicate rows.

Listings Table

Dealing with this table, there were some problems that needed attention. They will be listed below, together with an explanation on how they were dealt with.

1. Missing “Number of rooms” values.

   154 listings did not have information about their number of rooms. Those values will be left as NaNs so they can be dealt with later on.

2. ‘Tipo’ column not matching “Category” suffix.

   Entries 319, 327, 411 and 429 had conflicting information between their “Categoria” column suffix and their “Tipo” column. Given that the resources to confirm which column holds true were not available and that these listings only appear in 2696 out of 288971 rows in the daily_revenue table, these 4 entries will be discarded from the analysis.

Apart from that, the “Categoria" column was split in two columns: a “Categoria” column that was meant only for the listing’s tier (SIM<JR<SUP<TOP<MASTER) and a “Qtde_Quartos” column that was meant to hold the number of rooms in a listing. After that, some general data cleaning needed to be done to remove typos - for example, TOPM entry in “Categoria" column was corrected to TOP - and entries that did not match the column context - for example, there text entries in numeric columns. Finally, the correct dtypes were cast to the columns and duplicates were removed.

To finalize, the revenue and listings DataFrames were joined using pd.merge with a left join on the “listing” column. The result was verified with a SQL query in Google Big Query.

2. Questions

a. What is the expected price and revenue for a listing tagged as JUR MASTER 2Q in march?

This is a **Regression** problem, in which we need to predict a value that we do not have any data about - since there were no JUR MASTER 2Q listings in the data.

Performance Metrics

For such a problem, the metrics we will use to evaluate my model are the **R-squared** and the **Mean Squared Error**, for their suitability to regression problems and for their ease of understanding.

Baseline

Now, we need to set a baseline to evaluate our model. Since we are trying to predict the price and revenue for a 2 bedroom listing, our baseline will be the average between the averages of 1 and 3 bedrooms listings. We will evaluate this baseline with our performance metrics against the average of 2 bedroom listings across all the combinations of Locations and Categories.

Feature Engineering

After that, we need to define and prepare the features we will use to train our model. Since we need to predict the price and revenue based on Location, Category, Number of Rooms and Month, these features need to be passed to the model. We will also assume that the JUR MASTER 2Q will be sold, so its occupancy and blocked state will be 1 and 0, respectively, and they also will be features in our model.

Imputing Values

Since the Number of Rooms column has missing values, we will have to fill those. We opted for MICE imputing, which uses a Linear Regression model to fill the missing values of a DataFrame based on already existing values in other columns.

For that, we need to define which columns we need to use to train the imputer. They are: Total Number of Beds, Number of Pillows, Number of Bathrooms and Total Capacity. The Total Number of Beds was used because it has a higher correlation to Number of Rooms than any of the individual bed type totals.

Preprocessing

The selected features need to be preprocessed before we feed them to our model. The categorical features - Category and Location - will be One-Hot Encoded, while the numerical features will be scaled with Scikit's StandardScaler.

Model Selection

The next step is selecting a machine learning model. We will evaluate some models with a K-fold with 10 folds cross validation method against our performance metrics and choose the best performing one. The models we will evaluate are: Linear Regression, Ridge Regression, Lasso Regression, Elastic Net and Decision Tree Regressor.

The model with the best performance was the Decision Tree Regressor, for both price and revenue.

Hyperparameter Tuning

To keep things simple, it was decided to only tune the max_depth parameter of the Decision Tree, and that was made using a Grid Search Cross Validation that evaluated the model with a range of max_depth going from 1 to 25. The best depth for the revenue model was 23 and for the price model was 24.

Final Testing

With the tuning of the hyperparameters, the models were evaluated against test data that wasn't fed to the previous cross validation steps, in order to decrease overfitting. Both the price and revenue models had satisfactory results here, easily surpassing the baseline.

```
------------------------------------------------------------
revenue model
------------------------------------------------------------


Baseline results
R-Squared: -0.3904
Mean Squared Error: 23231.0536
```

```
------------------------------------------------------------


Final validation of model
RMSE: 9894.478916667273
R-Squared: 0.8304483205600598


------------------------------------------------------------
```

```
------------------------------------------------------------
last_offered_price model
------------------------------------------------------------


Baseline results
R-Squared: -0.3612
Mean Squared Error: 21676.2220
```

```
------------------------------------------------------------


Final validation of model
RMSE: 18073.867495212824
R-Squared: 0.7234963061150448


------------------------------------------------------------
```

Forecasting

Finally, the predicted values for both price and revenue for a MASTER listing in Jurerê with two bedrooms in March are 592,06. The predictions are the same due to the price and revenue distributions being rather similar throughout the dataset and due the Decision Tree model's nature.

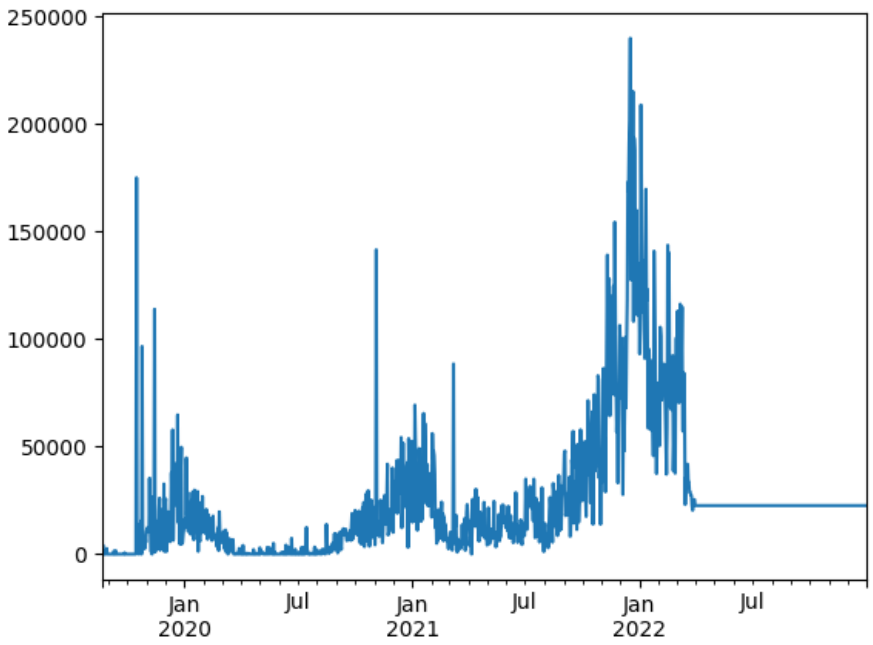
b. What is Seazone's expected revenue for 2022? Why?

For this question, a Time-Series forecasting model was built. The model was rather similar to the previous question model, since it was a Decision Tree that was optimized with GridSearchCV. What changed was:

The features: For time-series models, it is common to utilize auto regressive features. So, the features for this model were **yesterday's revenue** and the **difference between today's and yesterday's revenue**.

The cross validation: K-fold cross validation should not be used in time series models, so TimeSeriesSplit was used for cross validation.

The model was made to predict one step (one day) ahead, and then using this prediction to predict the next day, and so on. Given that, the model predicted a revenue of 30,976,029.30.



It is noted through the graph above that the model stabilized its predictions shortly after beginning them. For future improvements, it would be ideal to create a model that would not replicate this behavior.

c. How many reservations should we expect to sell per day? Why?

The model used here is the same as the one used on the previous problem, except that it is not needed to predict n-steps into the future.

The calculation of reservations sold a day was made based on the difference between the “occupancy” and “blocked” columns. This value was used to train a ML model to predict future demand daily. After training, this model predicted 39 daily sales.

3. Feedback

This was a great challenge! It was great to work with real-world data and to develop well-performing models to make forecasts on that data. Unfortunately, I couldn’t complete the whole challenge, but I hope that what I managed to do impresses you, and if it doesn’t, I would love some feedback so I can improve myself as a Data Scientist.

Thanks for the opportunity!