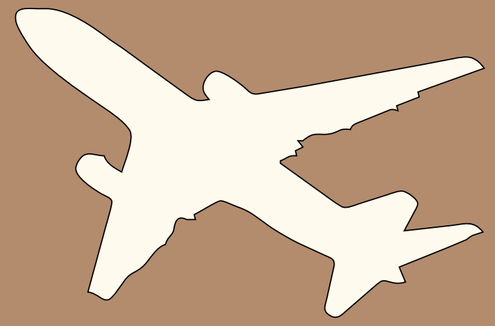
FLIGHT PRICE PREDICTION USING AUTOML

JOVITA - V

Data Science Intern -Hexaware Company



PROBLEM STATEMENT

- Flight ticket prices can be something hard to guess, today we might see a price, check out the price of the same flight tomorrow, and it will be a different story.
- To solve this problem, we have been provided with prices of flight tickets for various airlines between the months of March and June of 2019 and between various cities, using which we aim to build a model which predicts the prices of the flights using various input features

	Airline	Date_of_Journey	Source	Destination	Route	Dep_Time	Arrival_Time	Duration	Total_Stops	Additional_Info	Price
0	IndiGo	24/03/2019	Banglore	New Delhi	BLR → DEL	22:20	01:10 22 Mar	2h 50m	non-stop	No info	3897
1	Air India	1/05/2019	Kolkata	Banglore	CCU → IXR → BBI → BLR	05:50	13:15	7h 25m	2 stops	No info	7662
2	Jet Airways	9/06/2019	Delhi	Cochin	DEL → LKO → BOM → COK	09:25	04:25 10 Jun	19h	2 stops	No info	13882
3	IndiGo	12/05/2019	Kolkata	Banglore	CCU → NAG → BLR	18:05	23:30	5h 25m	1 stop	No info	6218
4	IndiGo	01/03/2019	Banglore	New Delhi	BLR → NAG → DEL	16:50	21:35	4h 45m	1 stop	No info	13302

Dataiku

Data Preparation

Python(Jupyter Notebook)

Build a Model and PIckle file

Streamlit and Heroku

Deployed

AIRLINE

SOURCE

DESTINATION

TOTAL_STOPS

EDA

UNIVARIATE & BIVARIATE ANALYSIS

dataiku

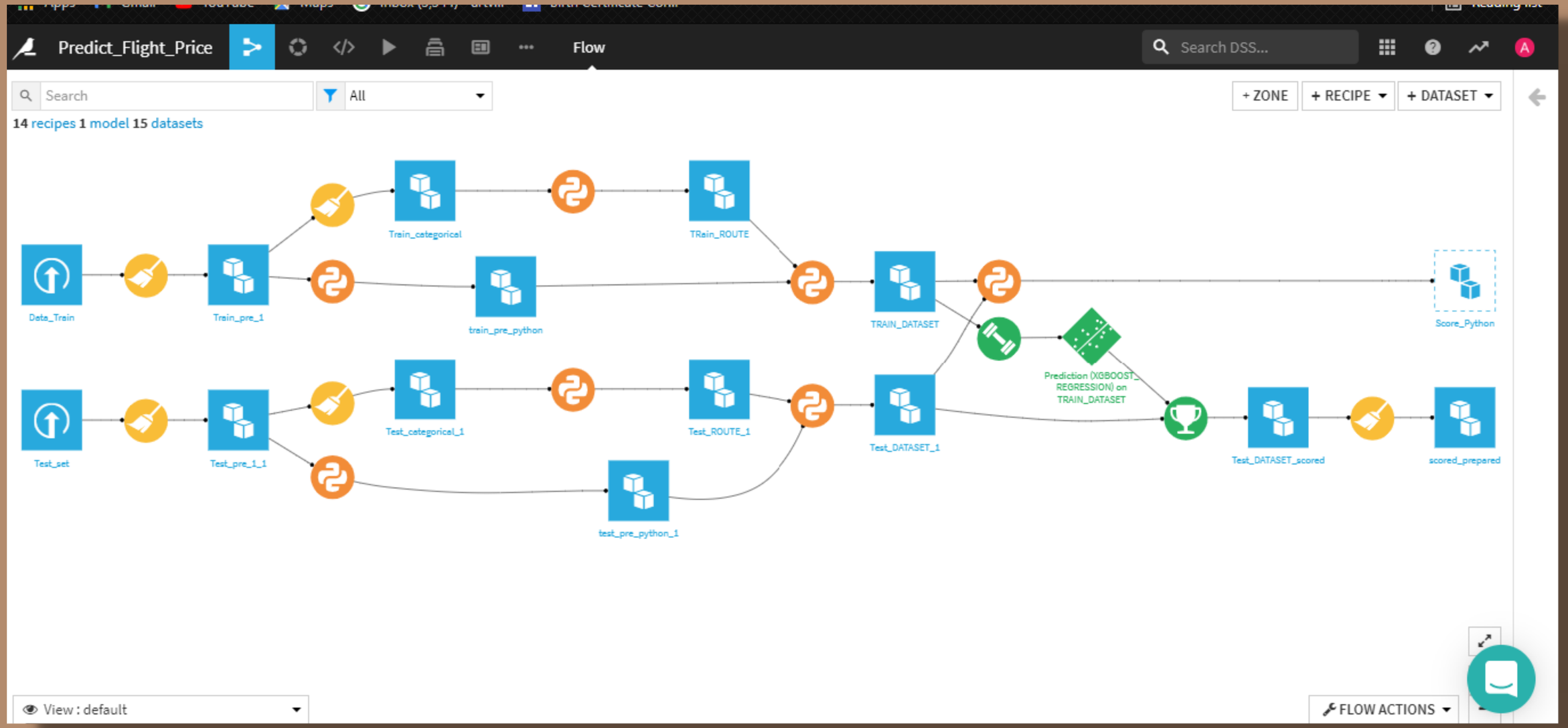
DATA PRE-PROCESSING

- MISSING VALUES
- SPLIT
- ENCODING
- REMOVING OUTLIERS

CORRELATION & HEAT MAP

FEATURE SELECTION

dataiku

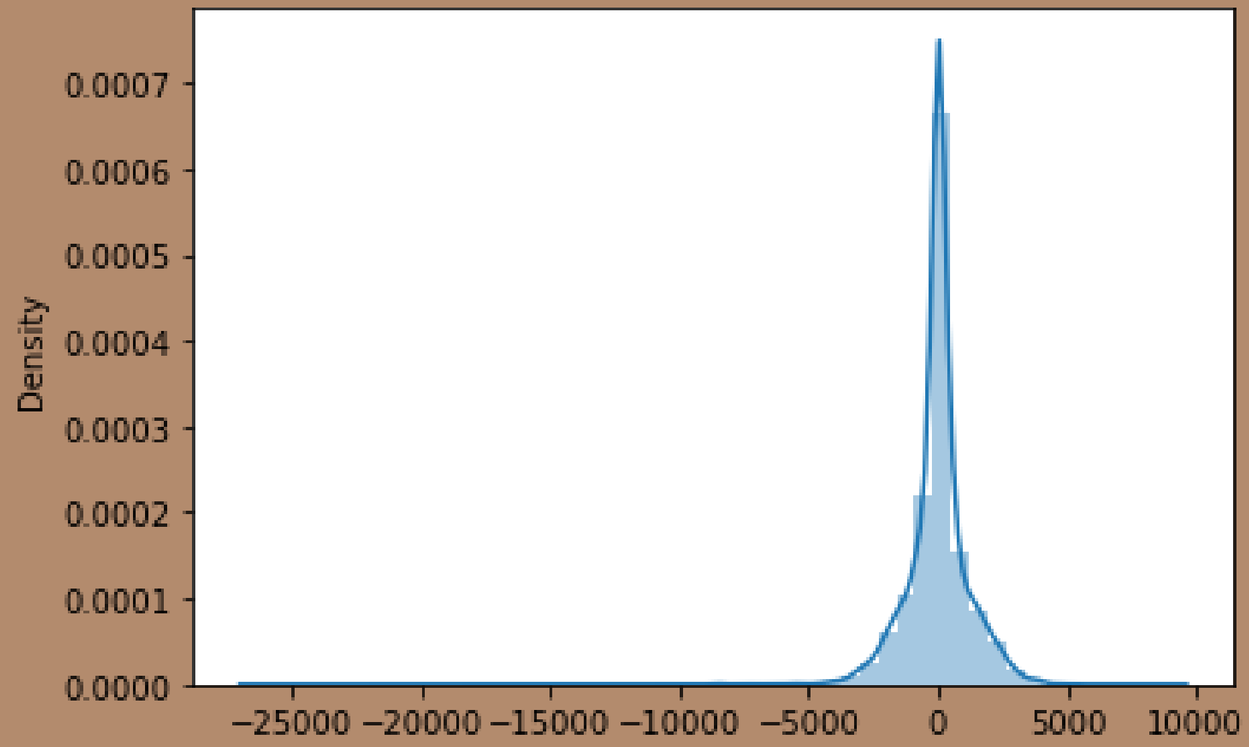

Predict_Flight_Price
Flow
Search DSS...
Search
All
+ ZONE
+ RECIPE
+ DATASET
14 recipes 1 model 15 datasets
Data_Train
Train_pre_1
Train_categorical
TRain_ROUTE
train_pre_python
TRAIN_DATASET
Score_Python
Prediction (XGBOOST_REGRESSION) on TRAIN_DATASET
Test_set
Test_pre_1_1
Test_categorical_1
Test_ROUTE_1
test_pre_python_1
Test_DATASET_1
Test_DATASET_scored
scored_prepared
View : default
FLOW ACTIONS


```
#tpot
tpot = TPOTRegressor(verbosity=2,random_state=25,generations=10, population_size=100,cv=5,scoring='r2')
tpot.fit(X_train,y_train)
```

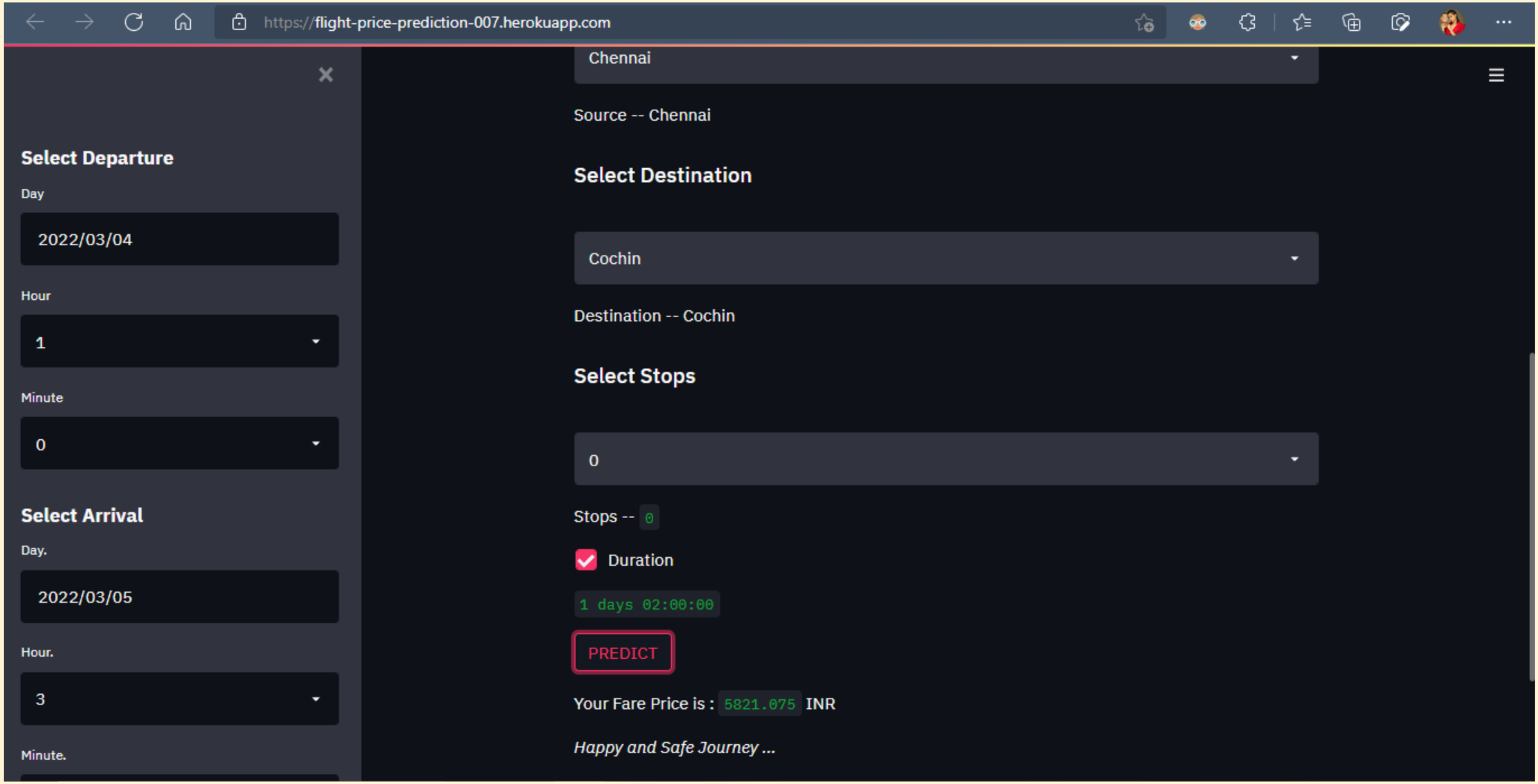
```
HBox(children=(HTML(value='Optimization Progress'), FloatProgress(value=0.0, max=1100.0), HTML(value='')))
Generation 1 - Current best internal CV score: 0.7936157462935419

Generation 2 - Current best internal CV score: 0.8312078964908446

Generation 3 - Current best internal CV score: 0.8329787103215971

Generation 4 - Current best internal CV score: 0.8359282146747379

Generation 5 - Current best internal CV score: 0.8359282146747379

Generation 6 - Current best internal CV score: 0.8396294347884963

Generation 7 - Current best internal CV score: 0.8409451942605746

Generation 8 - Current best internal CV score: 0.8409451942605746

Generation 9 - Current best internal CV score: 0.8421393077143066

Generation 10 - Current best internal CV score: 0.8423255803146166

Best pipeline: ExtraTreesRegressor(input_matrix, bootstrap=False, max_features=1.0, min_samples_leaf=1, min_samples_split=10, n_estimators=100)
TPOTRegressor(generations=10, random_state=25, scoring='r2', verbosity=2)
```

CONCLUSION

- Hence, at the end, we were successfully able to train our regression model ‘Extra tree Regressor’ to predict the flights of prices with an r2_score of 84%, and have achieved the required task successfully.



DEPLOYMENT



HTTPS://FLIGHT-PRICE-PREDICTION-007.HEROKUAPP.COM/

THANK YOU