Flight Fare Prediction

1. **Objective:** This analysis aims to predict flight prices given the various parameters. Data used for this analysis is taken from Kaggle. This will be a regression problem since the target or dependent variable is the price (continuous numeric value).
2. **Dataset Details:** This dataset mainly contains 14 columns:
    - *Unnamed*: drop this column (it's a duplicate index column)
    - *ItinID & MktID*: vaguely demonstrates the order in which tickets were ordered (lower ID #'s being ordered first)
    - *MktCoupons*: the number of coupons in the market for that flight
    - *Quarter*: 1, 2, 3, or 4, all of which are in 2018
    - *Origin*: the city out of which the flight begins
    - *OriginWac:* USA State/Territory World Area Code
    - *Dest*: the city out of which the flight begins
    - *DestWac*: USA State/Territory World Area Code
    - *Miles*: the number of miles traveled
    - *ContiguousUSA*: binary column -- (2) meaning flight is in the contiguous (48) USA states, and (1) meaning it is not (ie: Hawaii, Alaska, off-shore territories)
    - *NumTicketsOrdered*: number of tickets that the user purchased
    - *Airline Company*: the two-letter airline company code that the user used from start to finish (key codes below)
    - *PricePerTicket*: target prediction column
3. **Data Loading**: Data loading is the process of loading data sets from a source file or folder. The goal is to comprehend the dataset. Henceforth the details we obtained here:
    - Number of records present in the dataset: 9534417
    - There is no duplicated row.
    - All the columns contain equal data.
    - Categorical attributes: Origin, Dest, AirlineCompany
    - Float attributes: Miles, TicketperPrice
    - Numerical attributes: All the other attributes.
4. **Data Preprocessing**:
    - There are no missing values present.
    - I decided to remove the first three columns because they provide no useful insights for our analysis.
    - Perform Encoding for the categorical attribute: Although they are nominal attributes, I use label encoding instead of OneHotEncoder because the categories associated with each feature are too large, resulting in a large number of columns.
    - To avoid information loss, I decided to leave the float attributes.

5. **Feature Engineering**: Identifying the best feature that will contribute and have a good relationship with the target variable.
    - *Heatmap*: Rectangular representation of data as a color encoded matrix. This helps us to understand the correlation present between the attributes, the attributes which are highly correlated are considered redundancy.
    - *Feature Importance*: Using XGBRegressor to depict the important features. The reasons for using XGBRegressor are it's a huge dataset and a regression problem.
    - *SelectKbest*: SelectKBest is used to select the K best features. I am specifically selecting 6 features.
6. **Data Preparation**: After performing feature engineering I change the feature attributes with the new feature attributes selected for the feature engineering. A total of 9 features are selected. Furthermore, I split the dataset into train and test sets in 80% and 20%.
7. Modeling :
    - RandomForestRegressor
    - XGBRegressor
    - KNeighborsRegressor
    - DecisionTreeRegressor

Model	Training Score	Test Score	MAE	MSE	RMSE
RandomForestRegressor	0.3362	0.3267	83.33	13858.58	117.72
XGBRegressor	0.2775	0.2773	87.27	14875.39	121.96
KNeighborsRegressor	0.2424	0.2351	88.79	15743.87	125.47
DecisionTreeRegressor	0.3364	0.3259	83.38	13874.82	117.79

8. **Few Additional points:**
    a. I decided to use label encoding otherwise the number of features would have increased greatly.
    b. The float attributes are not rounded off in order to avoid the loss of information.
    c. During the time of data preparation, we could have separated our dataset into train, Val, test. The val set can be used later for tuning hyperparameters.

d. I decided to use these models because of the nature of the problem, i.e regression.
e. Hyperparameter tuning is missing because of the huge dataset each time I tried the RAM collapsed. But there can be various approaches like *randomsearchcv or selecting the best k for KneighborsRegressor.* We can discuss them in the interview.
f. Among the 4 models, RandomForestRegressor shows a better RSME.