1.Briefly describe your approach to this problem and the steps you took

By analysing the data, we get to know Dataset has number of missing values.

So I followed different steps to deal with missing value:

1.Identify missing data

- Count missing values in each column

2.Deal with missing data

- Drop Dataframe columns containing either 50% or more than 50% NaN values
- We have 3 features still left with missing data with one approx 50% missing data. So we will drop one feature with approx 50% missing data and fill rest 2 columns data with their most accurred value.
- After these steps, there is no any missing data in our dataframe and features column reduce from 73 to 21

3.Correct data format

- Check data type for each column
- Check unique value for each column
- we figure out that, in dataframe the columns which have 2 unique value are bassically "yes" or "No". so to bring it into proper data type we will change "yes" with "1" and "No" with "0"
- now we have stills some columns with multiclass classification. so we used labelencoder to bring it into correct data type.

Detecting Outliers

- Plotting a boxplot of odometer_reading vs rating_engineTransmission
- We detect some outliers above 500000 odometer_reading
- we deleted all odometer_reading value, which was above 500000

then after we check:

odometer_reading as potential predictor variable of rating_engineTransmission:-
we found that, As the odometer_reading goes down, the rating_engineTransmission goes up: this indicates a negative direct correlation between these two variables.

The correlation between 'odometer_reading' and 'rating_engineTransmission' is approximately: - 0.37

year as potential predictor variable of rating_engineTransmission:-
we found that, As the year goes up, the rating_engineTransmission goes up: this indicates a positive direct correlation between these two variables.
We can examine the correlation between 'year' and 'rating_engineTransmission' is approximately: 0.58

Pearson Correlation

The Pearson Correlation measures the linear dependence between two variables X and Y.
The resulting coefficient is a value between -1 and 1 inclusive, where:
1: Total positive linear correlation.
0: No linear correlation, the two variables most likely do not affect each other.
-1: Total negative linear correlation.

Data Preprocessing

- Normalising odometer_reading variable
- Encoding the target label using LabelEncoder
- Separating the features and target variable from the dataframe
- Splitting the data into training set & test set

Model Building

DECISION TREE CLASSIFIER

Accuracy = 0.4026615969581749
mean_squared_error = 2.6372623574144485
F1 Score = 0.4006055607213342

LOGISTIC REGRESSION
Efficiency = 0.48897338403041823
mean_squared_error = 2.2699619771863118
F1 Score = 0.4354797226130068

RANDOM FOREST

Efficiency = 0.43155893536121676
mean_squared_error = 2.314828897338403
F1 Score = 0.42075143693272843

KNN ALGORITHM
Efficiency = 0.4984790874524715
mean_squared_error = 2.2258555133079847
F1 Score = 0.4539112120836323

2. Basics:

a. How well does your model work?

My model worked well with **KNN ALGORITHM** , accuracy is 49.84.

b. How do you know for sure that’s how well it works?

I compared F1 score and mean_squared_error of different model and figure out that KNN ALGORITHM has highest F1 Score = 0.4539112120836323 and lowest mean_squared_error=2.2258555133079847, among all models.

The F1 score can be interpreted as a harmonic mean of the precision and recall, where an F1 score reaches its best value at 1 and worst score at 0.

c.What stats did you use to prove its predictive performance and why?

I used accuracy_score, that gives me idea about the efficiency of my model,

mean_squared_error, it gives a relatively high weight to large errors.

F1 Score, its best value at 1 and worst score at 0.

d.What issues did you encounter?

- Lots of missing values are there which needed to be handled for best fit model
- Handled categorical data using One- hot encoding for multivariable
- Highly imbalanced data

e.What insights did you obtain from this data? For example: What features are important? Why? What visualizations help you understand the data?

- As we have to handle lot of missing values here, we have to choose features that have enough data to train our model better
- negative direct correlation between odometer_reading and predictor variable of rating_engineTransmission
- positive direct correlation between “year” and predictor variable of “rating_engineTransmission”

3.Next steps:
a.What other data (if any) would have been useful?

Engine life numerical data, engine servicing time span, would have been useful with good correlation with predictor.

b.What are some other things you would have done if you had more time?

I would do more work to deal with missing data