CARS24

ENGINE RATING PREDICTION

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

Ans.

1. The age of engine which is added as a new feature is calculated based on subtracting the inspection month and year from the year and month of the car registration (calculated in months).

2. EDA of the data is done using various visualizations like box and whisker plots, histograms, violin plots and pie chart for variables like rating, age, and odometer

3. The missing data is initially filled with None values and later dropped on.

4. Tried to follow the similar strategy based on https://medium.com/@atishj.nits/ml-deep-learning-into-used-car-pricing-cars24-part-2-878eb6bbe811
   For encoding categorical variables but instead of frequency of categorical variables, only the Boolean value (dummy variable) of categorical variable was considered for variables like engine_transmission_engineOil, gear shifting, clutch etc.
5. For few categorical variables which have low cardinality (For example Fuel _type] , dummy variables were created.
6. Rating 1.0 has variance/ IQR range with respect to eng age and the median is around 160 months which is higher than every other rating. Rating 4.0 are more frequent in the data set.
7. The Outliers in odometer reading > 220000 and outliers for age with engine rating 5.0 is 100 months.
8. The model is built using Gradient Boosting Regressor(estimators = 500), fine-tuned (up to some extent) and cross validated. It is also compared with other ensemble and parametric regression models. The metric used is mean squared error. The exact model with the features used are

   ```
   GradientBoostingRegressor(alpha=0.9, ccp_alpha=0.0, criterion='friedman_mse',
                             init=None, learning_rate=0.1, loss='ls', max_depth=3,
                             max_features=None, max_leaf_nodes=None,
                             min_impurity_decrease=0.0, min_impurity_split=None,
                             min_samples_leaf=1, min_samples_split=2,
                             min_weight_fraction_leaf=0.0, n_estimators=500,
                             n_iter_no_change=None, presort='deprecated',
                             random_state=7, subsample=1.0, tol=0.0001,
                             validation_fraction=0.1, verbose=0, warm_start=False)
   ```
9. All the models have scaled using standard scaler, 10 fold cross validation is used along with gridSearch CV for optimizing parameters used in gradient boosting regressor.
10. Finally model is saved as .pkl using joblib.

2. Basics:

a. How well does your model work?

The model works relatively well. It has MSE of `0.28705133`.

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

Need to deploy it on flask/AWS to test it on a larger scale to better test the performance. The K-fold cross validation is done and compared with other ML models. The model is also fine-tuned. So I am sure. it works with the similar features.

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

I used MSE (Mean squared Error) metric for the prediction. As it is a default regression metric used for most evaluations and we can observe the error clearly deceasing as the algorithm learns and after hyper tuning is done. Other Metrics like R2 score, RMSE, MAE etc. can also be used.

d. What issues did you encounter?

One hot encoding was the only issue I have encountered. There are large / high cardinal variables present. I have tried one hot encoding directly initially, that is when my laptop got stuck for a while: on these 7 sets of high cardinal variables.

The other alternative would be label encoder (but would ranging values from 0-7 for example instead of binary values.). Did apply label encoder though. After label encoding, I should have converted to one hot encoder as a nested array and then count the binary value 1s, in these nested arrays across high cardinal variables.

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

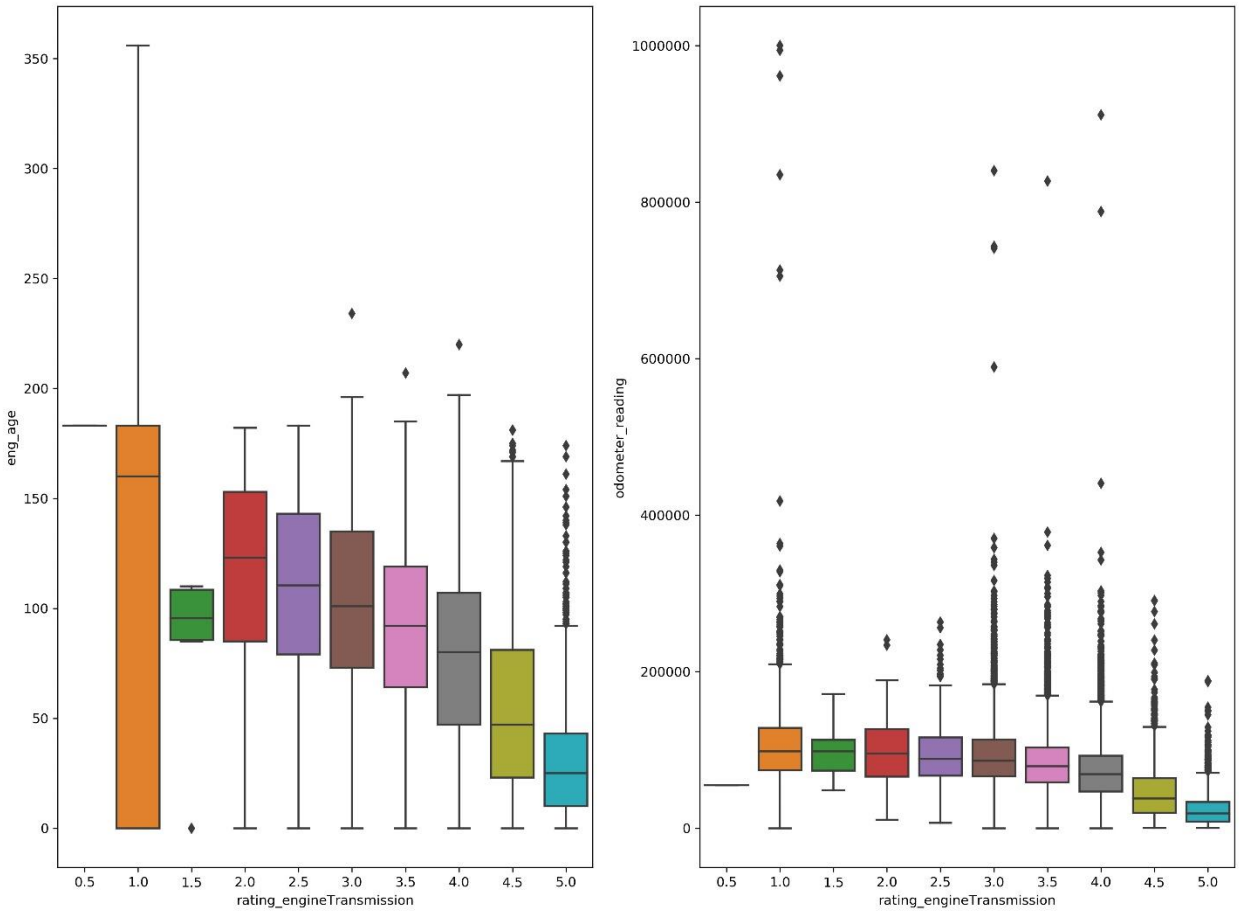
Few of the insights are already mention in 1. The box and whisker plots plotted are as shown below.

The heat map is plotted but, it difficult visualize.

To know the importance of each feature, we must calculate the correlation coefficient (Pearson's for example). Th
e `Pearson's correlation between odometer and engine rating: -0.363`

In [65]:

Box and Whisker Plot for rating & Engine Age



3. Next steps:

   a. What other data (if any) would have been useful?

   Removing zero correlation features while training the model would be helpful. Inclusion of day of the registration or registration date might be helpful.

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

   Would have calculated the frequency of categorical variables in high cardinal features.

   Used algorithms like XGBOOST.

Please send in all your code, model (jar, pickle, etc.) and a documentation answering the above questions.