Links for the Car Prediction Project

GitHub - https://github.com/Santhosh01161/Car-Price-Prediction-App/tree/main

Docker Command-

```
docker run -p 8080:5000 santy01161/carpriceprediction:latest
```

TASK 2

When trying to predict a car's price, some features naturally have a bigger impact than others. Based on the data, and the corelations matrix - we can see that the Max_power and Year show a clear relationship with selling_price. In this model I have used Max_power, year and km_driven. Other columns that appear during exploration (brand, fuel, seller_type, transmission, owner, mileage, engine, seats) are not included in the final training matrix as they did not show promising relationship to the dependant variable which is selling_price. This can be seen on the multivariable boxplots above.

On the modeling side, a non-linear, tree-based approach performs best because it captures the curved relationships between year, km_driven, and max_power (with the target modeled on the log scale), which shows up in the prediction tables where estimates track actual prices closely across both low and high ranges; in comparison, a plain Linear Regression would be too rigid since it assumes one global linear relationship and would likely underfit these three features even after the log transform, while SVR can model non-linearity but is very sensitive to scaling, making it easy to either under- or over-fit without careful tuning, and k-Nearest Neighbors tends to be unstable on these differently scaled numeric ranges and doesn’t extrapolate well. Overall, the tree-style model is the most natural and reliable fit for this setup, and this can be seen on under modeling under cross validation- Random-Forest Regressor has the highest mean (close to 0) compared to the rest of the models.