Abhi Reddy

Project 3 Proposal

Buying or selling a used car is something many people do, but deciding what a fair price should be is not obvious. The value of a car depends on several factors at once, such as its age, mileage, fuel type, transmission, seller type, and ownership history. Most people rely on a few online listings, rough rules of thumb, or personal experience, which can easily lead to prices that are too high or too low. In this project, I will use learned machine learning methods to predict the selling price of a used car from its basic attributes. I will treat this as a supervised regression problem, where the input is a set of features about the car and the output is a numeric price. My main goals are to see how accurately standard models can predict price using basic listing information, to compare simple linear models with tree-based models, and to understand which features matter most for price. I also want to check whether the patterns learned by the models match common sense, such as newer and lower-mileage cars tending to be more expensive. My project will use real-world data and include model comparison, evaluation, and interpretation.
I will use a public used-car dataset originally collected from the CarDekho website and shared on Kaggle. Each row in the dataset represents one used-car listing. The columns include the car's model name, the year associated with the car, the selling price, the total kilometers driven, the fuel type, the type of seller, the transmission type, and a description of ownership history. This dataset is a good fit for the project. It is easy to download and load into Python and it has a mix of numerical and categorical features that work well with the algorithms taught in class. At the same time, it is small enough that I can run experiments quickly without worrying about long training times. Before training any models, I will make sure to clean and prepare the data. I will check for missing values and remove or fix obviously invalid entries, such as rows with missing prices or impossible mileage values. I may consider extracting the brand or manufacturer from the model name, since brand often influences price. After cleaning and feature creation, I will split the dataset into training, validation, and test sets so that I can tune models on the validation set and report final performance on the test set.

This project will be implemented in Python. I will use pandas and NumPy for data handling, scikit-learn for preprocessing and modeling, and matplotlib or a similar library for visualizations. I will begin with a short exploratory data analysis. This will include computing basic statistics for variables like selling price, year, age, and miles driven, and plotting simple charts to see how these variables are distributed. I will look at how price changes with age and mileage and how it differs across fuel types, transmissions, and seller types. This step will help me spot outliers, see whether some variables are skewed, and build intuition about what the models should learn.
Next, I will set up a preprocessing pipeline. Numerical features such as age and miles driven will be used directly, and I could apply a simple transformation such as a logarithm if a variable is very skewed. Categorical features like fuel type, seller type, transmission, ownership category, and possibly brand will be converted into numerical form using one-hot encoding. I will

implement this pipeline using scikit-learn so the same steps are applied during training and testing. Now for modeling, I plan to compare several approaches. As a simple baseline, I will consider a basic model that always predicts the mean selling price from the training data. I will then train a standard linear regression model that uses all of the preprocessed features. If time allows, I will also try regularized linear models such as Ridge or Lasso regression to see whether they improve generalization. In addition to linear models, I will use tree-based methods that can capture non-linear relationships and interactions. I plan to train a random forest regressor and, if I have time, a gradient boosting regressor. For the tree-based models, I will perform light hyperparameter tuning, adjusting parameters such as the number of trees and maximum depth, guided by performance on the validation set. To evaluate the models, I will use standard regression metrics. I will compute mean absolute error to measure the average size of the prediction error in the same units as price, root mean squared error to penalize larger mistakes more strongly, and R squared to indicate how much of the variation in price is explained by the model. Interpretation is an important part of the project and so for the linear models, I will inspect the learned coefficients and check whether they make sense, for example whether the coefficient on age is negative. For the tree-based models, I will examine feature importance scores to see which features contribute most to accurate predictions. I will also create a scatter plot comparing predicted prices and actual prices for the best-performing model to get a visual sense of how well the model fits and where it tends to make errors.

The main products of this project will be a written report, the code, and a small set of figures and tables. The report will describe the problem I am addressing, the dataset I used, how I cleaned and preprocessed the data, which models I trained, how I evaluated them, and what I found. It will include a clear explanation of the main results and a short discussion of what the models reveal about used-car pricing. I will also comment on limitations, such as the fact that the data comes from a single market and does not include information about accidents, service history, or negotiation, and I want to suggest possible extensions, such as incorporating text descriptions or images from listings. The code will be organized in a Jupyter notebook. It will load the dataset, perform exploratory analysis, carry out preprocessing and feature engineering, train and evaluate the models, and generate the tables and plots used in the report. I will keep the code readable and well-documented. The figures and tables will include at least one table summarizing the performance of all models in terms of the chosen metrics, a few simple plots showing basic properties of the data, a scatter plot of predicted versus actual prices for the best model, and a visualization of feature importance for at least one tree-based model. All in all, these deliverables will demonstrate an end-to-end application of classical machine learning methods to a realistic used-car price prediction algorithm.