Predicting Car Selling Prices Using Linear Regression and Principal Component Analysis

Rizwan Riaz (25k-7613)

Muhammad Mohib Qureshi (25k-7617)

Syed Owais Ali (25k-7621)

Usama Yousuf Khan (25k-7633)

Abstract

This project investigates the effectiveness of linear regression techniques in predicting car selling prices. We used a real-world dataset of used car listings. After standard preprocessing and feature engineering, we implement and compare three approaches: **Ordinary Least Squares (OLS)**, **SVD-based least squares solution**, and **batch Gradient Descent (GD)**. We also apply **Principal Component Analysis (PCA)** to study the structure of the feature space and visualize the data in a lower-dimensional representation. Our results show that **OLS, SVD,** and **well-tuned GD** achieve almost identical performance, with the best model reaching a **test R^2** of around **0.86** and a **test RMSE** of about **₹3 lakhs.** PCA further indicates that most variance can be captured by a relatively small number of components.

1. Introduction

Predicting prices is an important problem in the automotive market, crucial to both buyers and sellers for a fair valuation. It is influenced by several factors such as vehicle age, mileage, engine capacity, power, fuel type, transmission, and ownership history. To study this in a concrete setting, we use a real-world used car dataset containing thousands of listings, where each record describes a single vehicle through a mix of numerical attributes (e.g., odometer reading, engine size, mileage, power) and categorical attributes (e.g., fuel type, seller type, transmission, number of previous owners), along with its final selling price.

Formally, we describe the used car price estimation as a supervised regression problem. Given a collection of cars, each described by a feature vector of technical specifications and usage attributes, and an associated selling price, our goal is to learn a model that can predict a reasonable price for a new car based on its characteristics. In this project, we specifically aim to

1. Construct a linear mapping from the engineered features to the (log-transformed) selling price,
2. Evaluate how accurately this mapping explains the observed variation in prices on held-out test data, and
3. Investigate the underlying structure and redundancy of the feature space.

These objectives provide a framework for assessing different linear modelling choices and for understanding which aspects of the car descriptions are most informative for price prediction.

2. Methodology

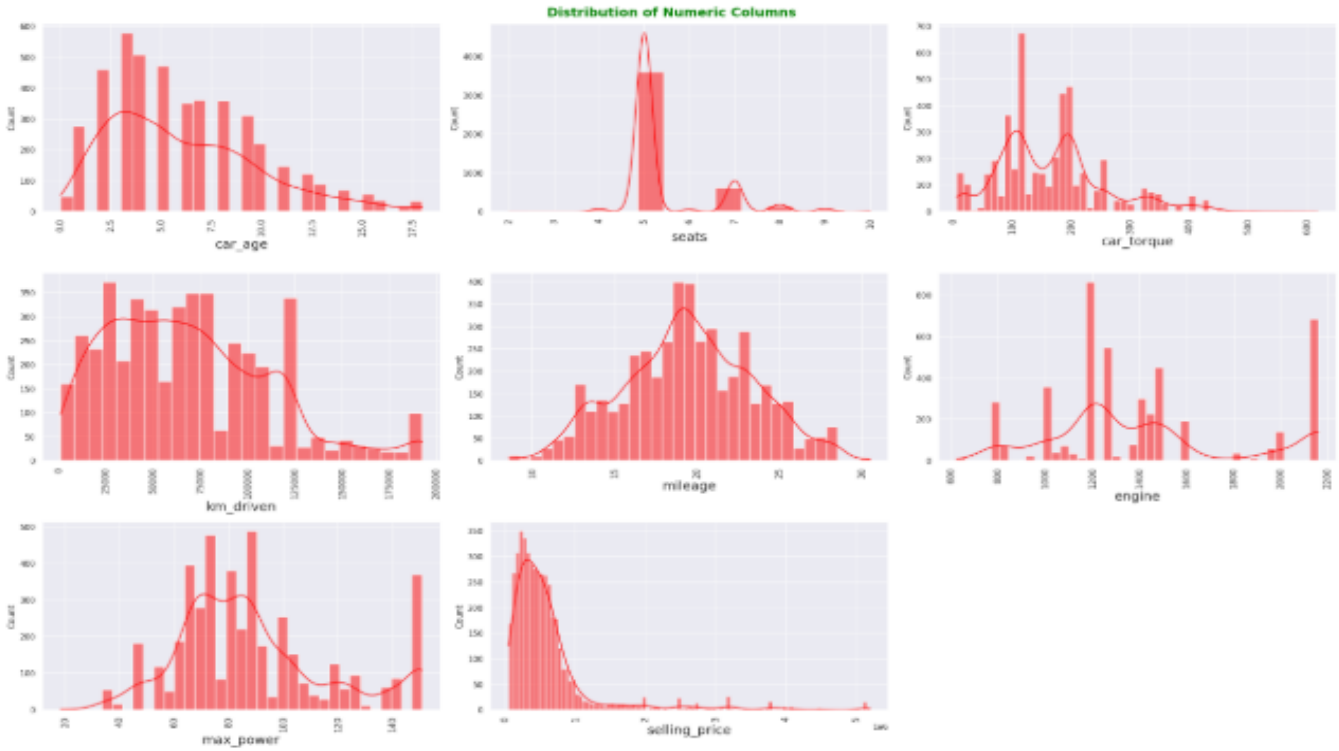
This section describes the steps used to transform the raw Car Dekho dataset into a form suitable for modelling and the methods applied for prediction and analysis. We first outline the preprocessing and feature engineering pipeline used to construct the final set of input variables. We then present the linear regression approaches based on Ordinary Least Squares (OLS), an SVD-based least squares solution, and batch Gradient Descent, followed using Principal Component Analysis (PCA) to study the structure of the feature space.

2.1. Dataset

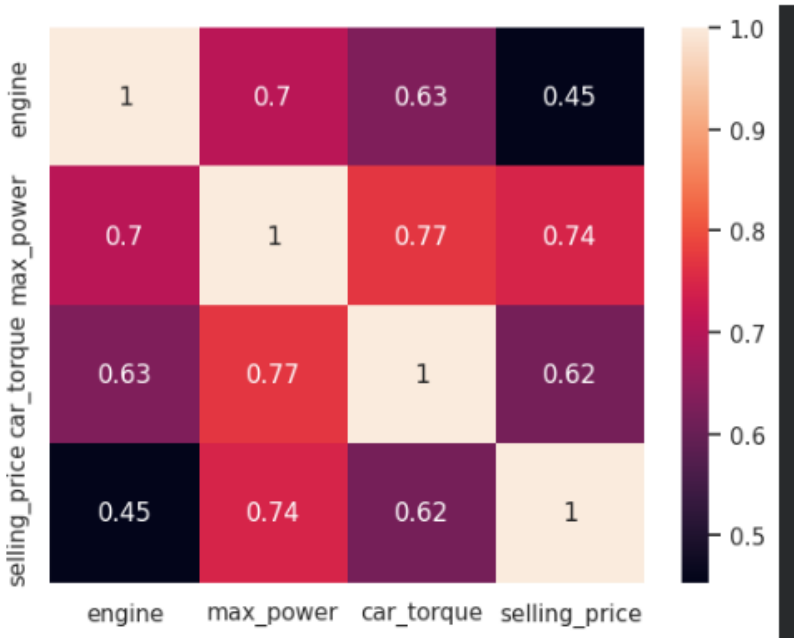
The dataset used is **CarDekho dataset[1]**, which consists of **8,128 car listings** and **13 original attributes**. The target variable is the car's selling price, while the input features include usage-related variables such as year of manufacture and kilometers driven, technical specifications such as mileage, engine displacement, maximum power, and torque, and several categorical descriptors. We perform basic preprocessing and feature engineering to make this data

suitable for modelling, including **deriving car age from the year**, **converting textual specifications into numeric form**, **encoding categorical variables**, **handling missing values**, **standardizing features**, and **applying a log transform to the selling price** to reduce skewness.

Figure showing the distribution of Numeric Columns:



Heatmap showing comparison of features against each other



2.2. Data Preprocessing and Feature Extraction

The original dataset contains 8,128 used car listings with 13 attributes, including the car name, year of manufacture, odometer reading, mileage, engine displacement, maximum power, torque, fuel type, transmission, seller type, ownership history, number of seats, and the selling price. Several of these fields are stored as text with units, and some entries contain missing values, so a sequence of preprocessing steps is required before model training.

We first cleaned the raw columns by standardizing column names and removing irrelevant or redundant fields. The name field was used to derive a brand and model identifier and then dropped from the final feature set. Similarly, the original year column was converted into a derived car_age feature, computed as the difference between the reference year-2025 and the year of manufacture, and the original year was subsequently removed. The torque field, which often contains multiple values and units within a single string, was parsed to extract a single numeric car_torque value in a consistent format. The inspiration of cleaning the dataset is taken from the kaggle [2].

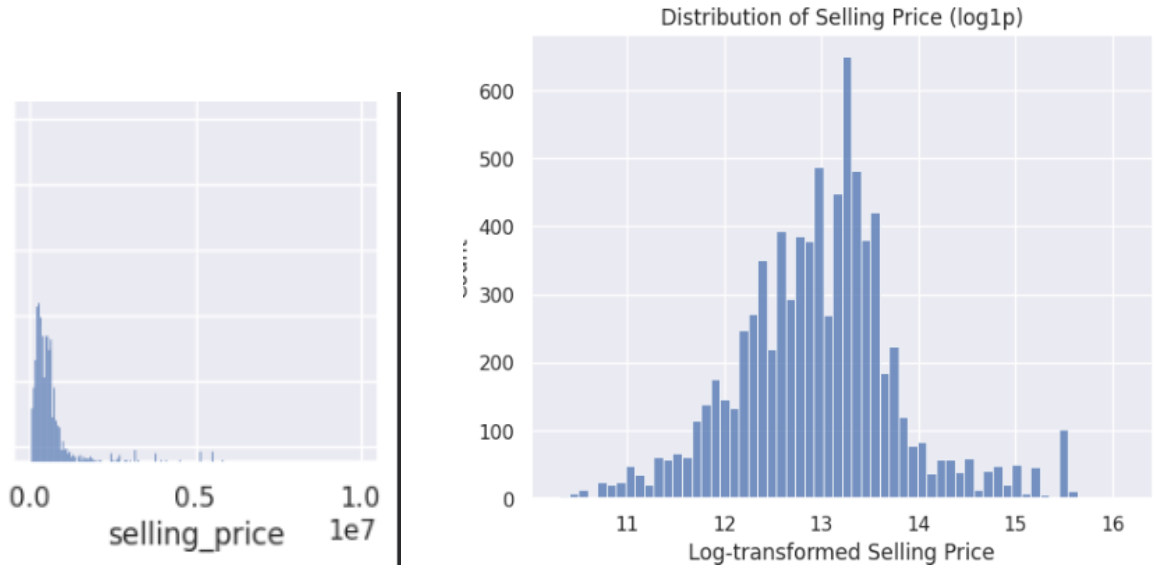
Numeric specifications stored as text were then converted into proper numerical features. For the mileage, engine, max_power, and torque columns, units and non-numeric characters (such as “kmpl”, “km/kg”, “CC”, “bhp”, or “Nm”) were stripped, leaving only the numeric component, which was cast to a number. This yielded cleaned continuous variables for the listed fields that could now be used in the regression models.

Missing values in the continuous features (mileage, engine, max_power, and car_torque) were handled using a K-nearest neighbours (KNN) with k=5, which estimates each missing entry based on similar records in the dataset. For the seats attribute, which has a small number of missing values and a limited set of typical values, we imputed the mode (most frequent value) and then cast the column to integer type.

Categorical attributes were encoded into numeric form to be compatible with linear models. The derived brand and model identifiers were transformed using label encoding, assigning a unique integer index to each distinct category. The remaining categorical features—fuel, seller_type, transmission, and owner—were converted using one-hot encoding

with a reference category dropped for each, resulting in a set of binary indicator variables such as fuel_Diesel, seller_type_Individual, and transmission_Manual. These indicator variables, together with the continuous features (car_age, km_driven, seats, mileage, engine, max_power, car_torque) and the encoded brand and model, form the final input feature vector for each car.

Before fitting the models, we applied two additional transformations. First, all input features were standardized using a z-score transformation so that each has approximately zero mean and unit variance, which is important for the stability of Gradient Descent and for interpreting PCA. Second, the target variable, selling_price, was transformed using the log(1+y) function to reduce skewness and compress the range of prices. Selling price distribution before and after log transformation.



After performing data cleaning and scaling, the dataset was converted into a NumPy array for efficient numerical computations. The features (X) and target variable (y) were then divided into training and testing subsets using an 80:20 split.This separation serves two purposes:

1. Model Evaluation: The testing set provides an unbiased assessment of the model's predictive performance on unseen data.
2. Model Improvement: By evaluating performance on the test set, hyperparameters (e.g., learning rate in gradient descent) and feature selection can be fine-tuned to achieve better generalization.

2.3. Linear Regression Approaches

Three regression methods were implemented to model the relationship between features and selling price:

1. Ordinary Least Squares (OLS) Regression:

OLS estimates coefficients θ by minimizing the sum of squared residuals between predicted and actual values:

$$\theta = (X^\top X)^{-1} X^\top y$$

Here y would equal to the y= θX which is equal to y= θ_0+ $\theta_1 X_1$+ $\theta_2 X_2$ … and so on

θ_0 define the bias, $\theta_{1...}$ θ_n defines the importance of each feature like weights.

Here, $X$is the feature matrix (with bias term) and $y$is the log-transformed target. OLS provides an analytical solution and allows straightforward interpretation of feature importance via coefficients.

2. SVD-Based Regression:

In this approach, the feature matrix $X$is factorized using Singular Value Decomposition (SVD):

$$X = U\Sigma V^\top$$

The coefficients are then computed as:

$$\theta = V\Sigma^{-1}U^\top y$$

Here y would equal to the y= θX which is equal to y= θ_0+ θ_1X_1+ θ_2X_2 … and so on

θ_0 defines the bias, $\theta_{1...}$ θ_n defines the importance of each feature like weights.

Advantage of SVD over OLS

2.1. SVD improves **numerical stability** in computations.
2.2. It can handle **ill-conditioned matrices** where X^TX is nearly singular.
2.3. It works well with **near-collinear or highly correlated features**.
2.4. Unlike the normal equation, SVD avoids directly inverting X^TX, reducing the risk of large numerical errors.

3. Gradient Descent (GD):

Gradient descent is an iterative optimization method that updates coefficients to minimize the mean squared error:

$$\theta \leftarrow \theta - \eta\nabla J(\theta)$$

Here, $J(\theta)$ is the loss function and η is the learning rate. The learning rate was carefully tuned (η ≈ 0.1) to ensure convergence within 1,000 iterations. GD is particularly useful when the dataset is large and matrix inversion (as in OLS) is computationally expensive.

4. Principal Component Analysis (PCA):

PCA was applied to explore feature correlations and reduce dimensionality. The steps include:

4.1. Centering the scaled feature matrix by subtracting the mean.

4.2. Performing SVD to decompose the matrix into orthogonal principal components.

4.3. Analyzing explained variance to determine how many components capture the majority of data variance.

4.4. Projecting the dataset onto the first two principal components for visualization and interpretability.

This methodology ensures robust modeling, stable coefficient estimation, and interpretability of the contributions of different features in predicting car prices.

3. Result

Regression Performance: The three regression methods Ordinary Least Squares (OLS), SVD-based regression, and Gradient Descent (GD) were evaluated using R^2 and RMSE metrics on both training and testing datasets. The results are summarized in the table below:

METHOD	TRAIN R^2	TEST R^2	TRAIN RMSE	TEST RMSE
OLS (NORMAL EQUATIONS)	0.840818	0.861792	321,314	300,988
SVD-BASED	0.840818	0.861792`	321,314	300,988
GRADIENT DESCENT (LR =0.1)	0.840868	0.861747	321,264	301,035

Coefficients values for OLS, SVD and GD are:

```
=== Summary ===
OLS Coefficients: [ 1.29735889e+01 -4.48076482e-01 -1.68881931e-02  2.33014543e-02
  3.37784258e-02  1.10446060e-01  3.43825361e-01 -5.77232190e-04
  1.11993560e-01  1.11525909e-02  1.93645295e-02 -4.33318315e-02
  2.52500978e-03 -7.23796658e-02 -1.84622545e-02 -3.56777563e-02
  1.30536878e-02 -2.89570503e-02 -1.21629125e-02  3.17937202e-03]
SVD Coefficients: [ 1.29735889e+01 -4.48076482e-01 -1.68881931e-02  2.33014543e-02
  3.37784258e-02  1.10446060e-01  3.43825361e-01 -5.77232190e-04
  1.11993560e-01  1.11525909e-02  1.93645295e-02 -4.33318315e-02
  2.52500978e-03 -7.23796658e-02 -1.84622545e-02 -3.56777563e-02
  1.30536878e-02 -2.89570503e-02 -1.21629125e-02  3.17937202e-03]
GD Coefficients : [ 1.29735869e+01 -4.47988388e-01 -1.68789924e-02  2.33080821e-02
  3.35173410e-02  1.10326965e-01  3.43781258e-01 -2.22126375e-04
  9.46727438e-02  8.77495719e-03  2.18247686e-03 -4.34186393e-02
  2.55421718e-03 -7.23283272e-02 -1.85116440e-02 -3.56870441e-02
  1.30502005e-02 -2.89544346e-02 -1.21811128e-02  3.21500663e-03]
```

Observations and Insights:

1. OLS vs SVD: The results of OLS and SVD-based regression are nearly same both in terms of R^2 and RMSE. This indicates that the dataset is well-conditioned and matrix inversion in OLS does not introduce significant numerical errors. The SVD method provides additional stability, particularly useful for datasets with multicollinearity or ill-conditioned feature matrices.

2. Gradient Descent Performance: Gradient Descent closely approximates the analytical solutions obtained by OLS and SVD. With a properly tuned learning rate ($\eta = 0.1$), it converged in 1,000 iterations. With Gradient descent the model is performing slightly better on traning data with R^2 0.840868 slighty better then OLS and SVD and training RMSE of 321,264 better then OLS and SVD but the Test RMSE is 301,035 higher then both indicating that model has overfitted slightly.

3. Error Magnitude: The RMSE values (~300k) reflect the average deviation of predicted car prices from actual values. Log-transforming the target variable helped stabilize variance and improved predictive accuracy across all methods.

Overall, all three regression approaches achieve comparable predictive performance, validating the robustness of linear modeling for this dataset . The choice of method can be guided by dataset size, conditioning, and computational efficiency:

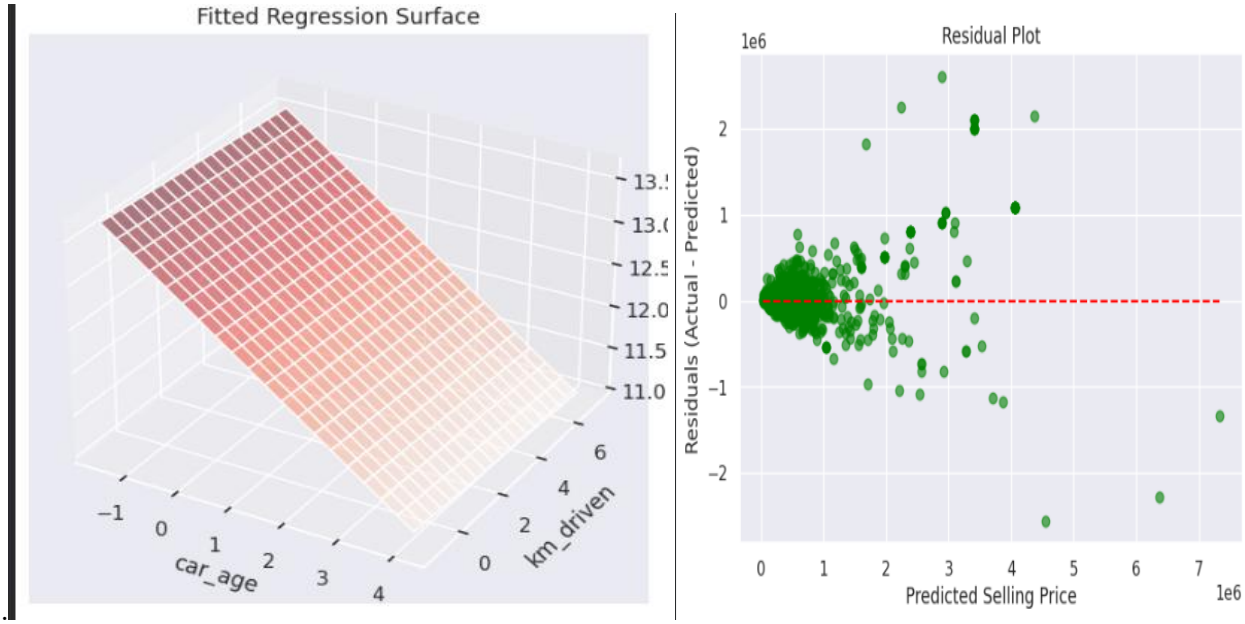
- OLS is straightforward for smaller, well-conditioned datasets.
- SVD ensures numerical stability and interpretability of components.
- Gradient Descent offers scalability for large datasets where matrix inversion is computationally expensive.

4. Discussion

4.1. OLS vs SVD Regression

The comparison between Ordinary Least Squares (OLS) and SVD-based regression shows nearly same predictive performance across training and testing datasets (Train $R^2 \approx 0.841$, Test $R^2 \approx 0.862$). The feature coefficients from both methods match almost exactly, confirming that the dataset is well-conditioned and does not suffer from multicollinearity severe enough to destabilize the normal equations.

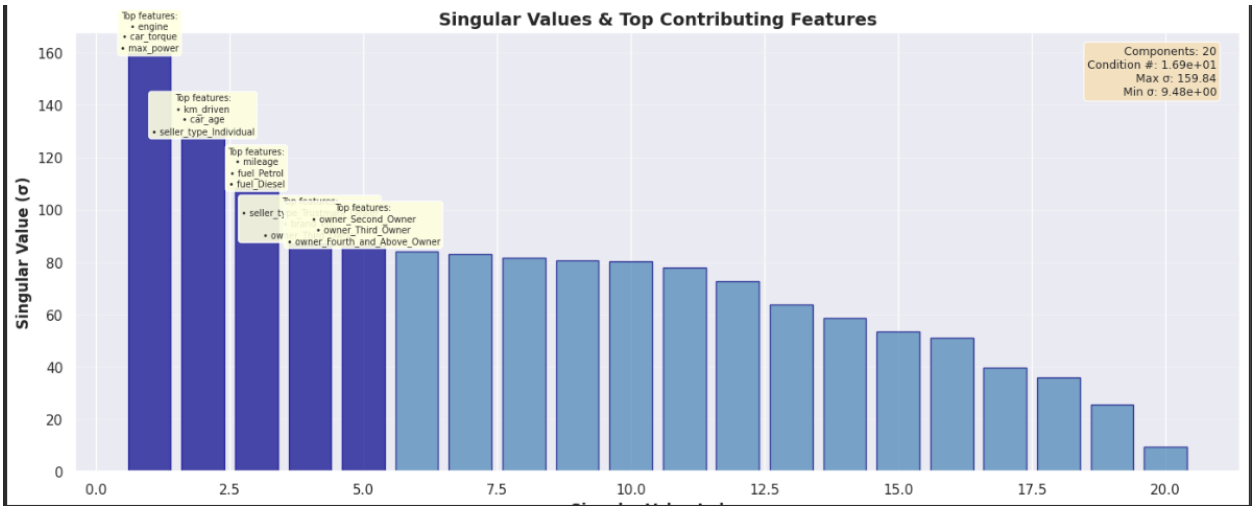
This OLS plot shows how the OLS creates a plane for the predicting the selling price based on car_age and km_driven The redisual plot for OLS is also plotted showing the predicted selling price.



This indicates that the models accurately predicted the selling prices for most cars, particularly those with typical price ranges, while a few extreme outliers may still be present in the dataset that were not addressed in this project

The SVD methods coffieince defines the importance of each features. The SVD method, however, offers additional numerical stability, especially in situations where $X^\top X$is nearly singular or features are highly correlated.

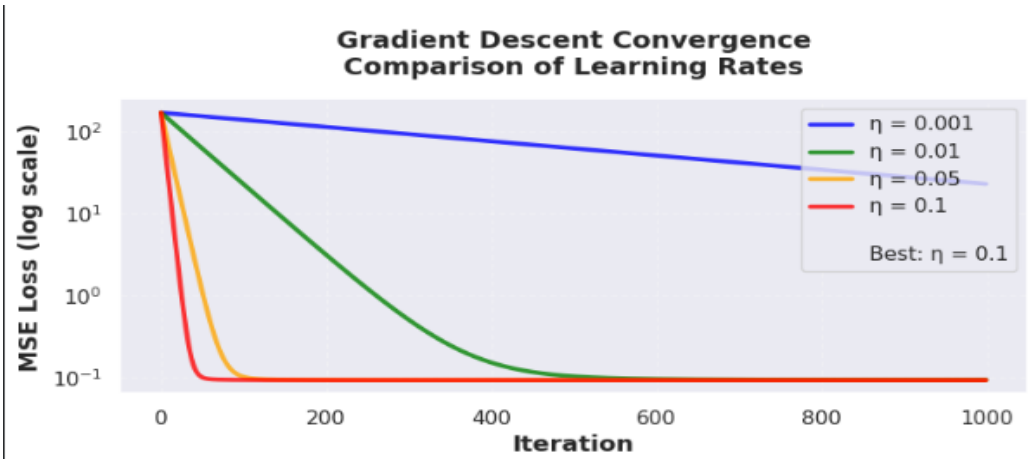
The plot highlights that while the top singular values dominate the variance, the smaller singular values are non-zero, indicating that no features are perfectly collinear. Annotated top contributing features show that engine,torque and power are the main drivers of variance in the data.The plot of this is given below:



4.2 Gradient Descent

Gradient Descent (GD) produced slightly better results than OLS and SVD on training data, demonstrating that iterative optimization can approximate analytical solutions effectively. The convergence plot (not shown here but in your notebook) confirms that with a learning rate of $\eta = 0.1$, the algorithm converges steadily within 1,000 iterations:

- Slight differences in RMSE and coefficient values.
- GD is advantageous for large datasets where matrix inversion in OLS becomes computationally expensive. The coverage graph for the GD for different Learning rate is given below:



MSE Loss:

4.3 Feature Importance

Across all regression methods, the most influential features are:

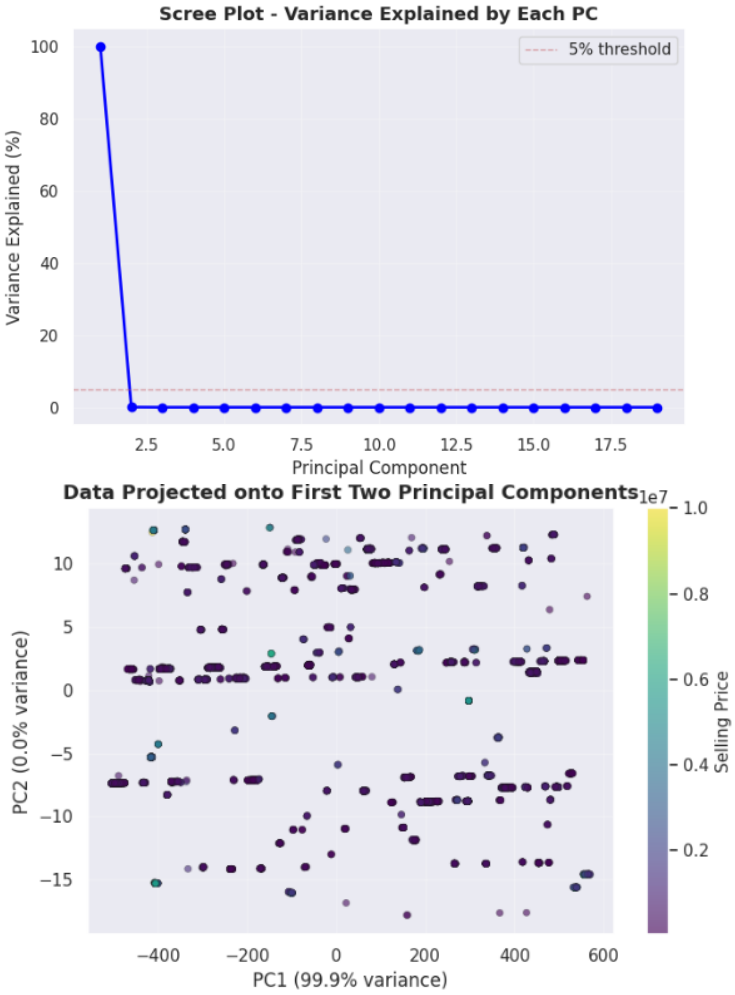
1. Car Age: Older cars predictably sell for lower prices.
2. Engine: Higher engine capacity increases selling price.
3. Max Power: Higher power generally increase price but can interact with age and model type.

The bar plot of |θ| coefficients confirms that categorical features such as fuel type and seller type also contribute but to a lesser extent.

4.4 PCA Insights

Principal Component Analysis (PCA) revealed significant collinearity among features, as the first few principal components explain most of the variance:

- The scree plot shows that the first 7 PCs capture ~95% of total variance.
- The 2D projection of data onto the first two PCs demonstrates the spread of selling prices and highlights feature correlations visually:



The Graph show that most of the dataset's variability can be captured using just one principal component, indicating high correlation among features. This also suggests that dimensionality reduction is feasible without losing much information.

This is mainly due to the fact the high engine mene high torque and good mileage and that is relatively new so most of the feature as strongly correlated

5. Conclusion

This analysis demonstrates the effectiveness of linear regression approaches in predicting car selling prices:

- All models achieved strong predictive performance (Test $R^2 \approx 0.86$).
- SVD-based regression ensures numerical stability and is preferred when features are highly correlated.
- OLS is sufficient for well-conditioned datasets.
- Gradient Descent provides flexibility for large datasets but requires careful tuning of learning rate and iterations.
- PCA confirms feature redundancy, allows dimensionality reduction, and aids visualization.

Key Takeaways:

1. Age, engine capacity, and max_power, Km_driven dominate price prediction.
2. Regularization or advanced models could further improve predictions.
3. Choice of regression method should consider dataset size, conditioning, and computational resource

References

Dataset is taken from the kaggle car dekho dataset [1] and Data cleaning and visualization inspiration is taken from the Kaggle (ABDO HASSAN) [2]:

[1] https://www.kaggle.com/datasets/nehalbirla/vehicle-dataset-from-cardekho

[2] https://www.kaggle.com/code/abdohassan01/vehicle-price-cleaning-analysis-prediction-sm