Car Price Prediction Project.

Project Plan:

1) Prepare data and do EDA
2) Use linear regression for predicting price
3) Understand the internals of linear regression
4) Evaluate the model with RMSE
5) Feature Engineering
6) Regularization
7) Using the model

EDA

Linear Regression

$g(X) \approx y$ (price)

y → Target

g → model (linear regression)

X → feature Matrix (Train Dataset)

$g(x_i) = w_0 + w_1 x_{i1} + w_2 x_{i2} + w_3 x_{i3} + \dots$

$g(x_i) = \{x_{i1}, x_{i2}, \dots\}$,

$= w_0 + \sum_{j=0}^{n} w_j x_{ij}$

n = no. of features/column considered

$Xw = y$

or, $X^{-1}Xw = X^{-1}y$

or, $Iw = X^{-1}y$

or $w = X^{-1}y$

↳ But X is not a square matrix everytime

$\therefore w = X^{-1}y$ cannot be used.

(*)

$Xw = y$

Gram Matrix = X^TX

or, $X^TXw = X^Ty$

or, $(X^TX)^{-1}(X^TX) = (X^TX)^{-1}X^Ty$

or, $\boxed{w = (X^TX)^{-1}X^Ty}$

$$w = np.linalg.inv(X.T.dot(X)).dot(X.T).dot(y)$$

```
# ones = np.ones(X.shape[0])
  X = np.column_stack([ones, X])

# def fast_linear_regression(X, y):
      # find the values of w.
```

⇒ engine_hp, highway-mpg, city-mpg

engine_hp → x_1, highway-mpg → x_2, city-mpg → x_3

$$w_0 + w_1 x_1 + w_2 x_2 + w_3 x_3$$

$$= w_0 x_0 + w_1 x_1 + w_2 x_2 + w_3 x_3$$

$x_0 = 1$

$$= \sum_{j=0}^{n} w_j x_j$$

#RMSE

$$= \sqrt{\frac{1}{m} \sum_{i=1}^{m} (g(x_i) - y_i)^2}$$

$g(x_i)$: prediction for x_i; y_i: Actual value

---

df.copy() → By default, Deep copy

df.copy(deep=False) → shallow copy

ndarray.copy() → always deep copy

$$w = (X^T X)^{-1} X^T \cdot y$$

if $X^T X$ is singular, $(X^T X)^{-1}$ does not exist

↑ used to avoid overfitting

$X_{m \times n}$ $X^T_{n \times m}$

$(X^T X)_{n \times n}$

On regularization:

$w = (X^T X + 0.01 * np.eye(n))^{-1} X^T . y$

↓

can be 0.001 or 0.0001

⊛ Model Usage

↓

find the weights from train dataset

↓

apply the w's on the val & test datasets

to predict y

↓

calculate RMSE to determine the score

Load Dataset

↓

EDA

↓

Setting up validation Framework

↓

Train Linear R. model

↓

RMSE

↓

Feature Engineering ↻

↓

Regularization

↓

Using the Model