

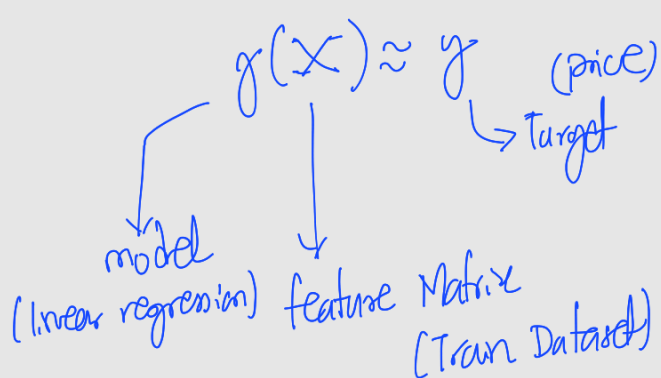
Car Price Prediction Project.

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



$$g(x_i)$$

$$\{x_{i1}, x_{i2}, \dots\},$$

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

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

$n = \text{no. of features/columns considered}$

$$Xw = y$$

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

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

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

\rightarrow But X is not a square matrix anytime

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

(*)

$$Xw = y$$

$$\text{or, } X^T X w = X^T y$$

$$\text{or, } (X^T X)^{-1} (X^T X) = (X^T X)^{-1} X^T y$$

$$\text{or, } \boxed{w = (X^T X)^{-1} X^T y}$$

Gram Matrix: $n = X^T X$

$$w = \text{np.linalg.pinv}(X.T \cdot \text{dot}(X)).\text{dot}(X.T).\text{dot}(y)$$

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

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

⇒ engine_hp, highway_mpg, city_mpg
 ↓ ↓ ↓
 x_1 x_2 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

$$= \sqrt{\frac{1}{m} \sum_{i=1}^m (\underbrace{g(x_i)}_{\text{prediction for } x_i} - \underbrace{y_i}_{\text{Actual value}})^2}$$

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

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

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

On regularization:

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

↓

can be 0.001 or 0.0001

* Model Usage

