Linear Regression

A Relook at Supervised learning

Training a ML task for every input with a corresponding target, is called Supervised learning.

Supervised learning for price prediction of used Cars

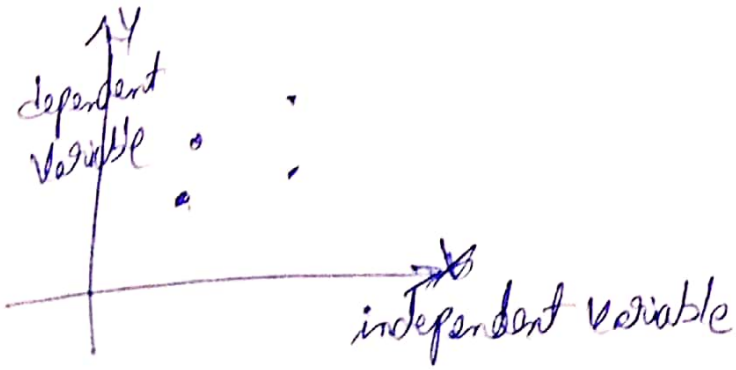
↳ Predict the (price) of an used car.

↳ Has (predictor) (independent) variables and an (outcome) (dependent) variable.

What is Regression

Predicts values of a (Continous dependent variable)

Using independent explanatory variable(s)



Scanned by CamScanner

Regression Use Cases

① Real estate : model house prices as a function of home's living area, no of bedrooms, bathrooms & lot size.

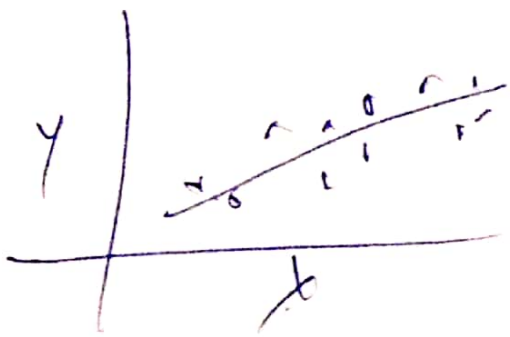
② Marketing : model relationship b/w online advertising costs and monthly e-commerce sales.

③ Medicine : Forecast medical expenses for insured population based on attributes like age, gender, smoker, BMI,

④ Weather : Model relationship b/w crop yeild based on amount of rainfall.

What is linear regression?

A form of regression that models linear relationship b/w dependent & independent variables.



X & Y are linearly related.

Scanned by CamScanner

What makes linear regression popular?

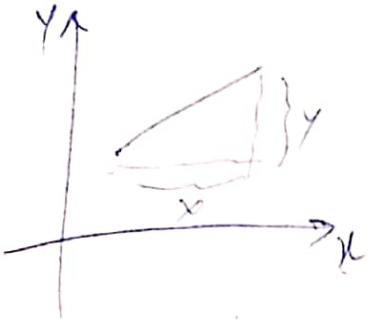
→ Ease of implementation =

→ powerful =

→ Easily Scalable =

ML = needing more data = DL = require tons of Data.

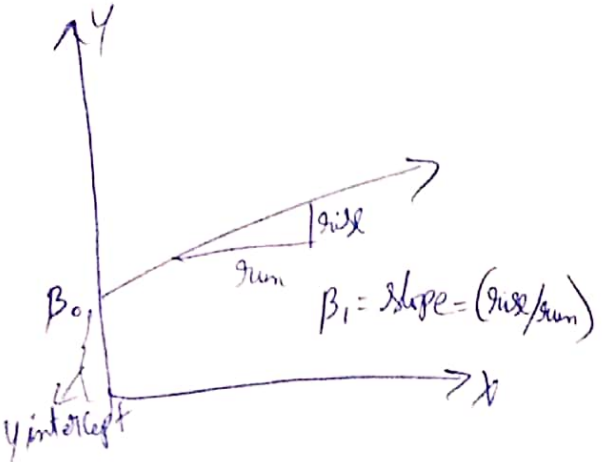
Describe equation of a line



dependent variable — Independent variable — where line crosses the Y-axis

$Y = mx + b$

Coefficient, rate & slope of line — Y-intercept



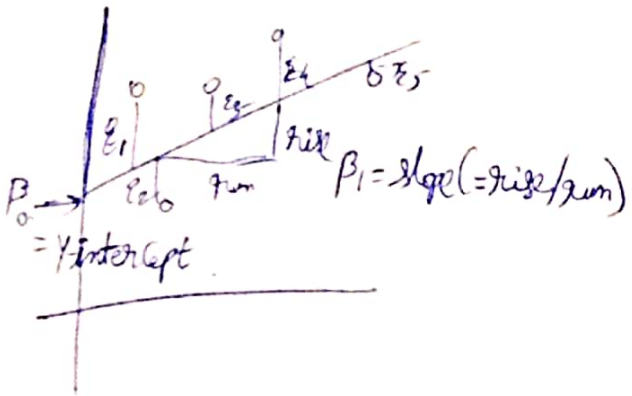
predicted value — Input variable — Intercept

$\hat{y} = \beta_1 x + \beta_0 + \varepsilon$

Error term

Slope of regression line

Slope has great impact on your model (β_1)



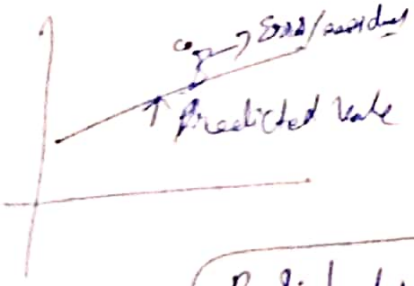
Errors are not directly added

$\hat{Y} = \beta_1 X + \beta_0 + \varepsilon$

β_1 ↓

Slope of regression line

Errors or residuals



→ Residual is the difference blw observed value of dependent variable and predicted value.

Residual = observed values − predicted values

Metrics used to Evaluate L.R. model

Accuracy = Closeness of actual predicted → as high as possible.

Loss function = Actual − Predicted (Error & Residual) → as low as possible.

What is the best fitting line? Line of Best fit.

Goal: find the best slope and intercept that fits data points

Soln: Minimize residual Errors.

Scanned by CamScanner

The following Metrics are used to measure model performance

MAE = Mean absolute Error.

MSE = Mean Squared Error.

RMSE = Root mean Squared Error.

MAE is the average absolute difference between actual and predicted values.

$$MAE = \frac{1}{n}\sum_{j=1}^{n} |y_i - \hat{y}_j|$$

Characteristics

→ Neutral to outliers.

→ Error is in same units as that of the data points.

MSE is the average Squared difference between actual and predicted values.

$$MSE = \frac{1}{n}\sum_{i=1}^{n} (y_i - \tilde{y}_i)^2$$

Characteristics :

→ Accounts impact of outliers.

→ Error is not in same unit as that of the datapoints

Scanned by CamScanner

RMSE is the square root of the average squared difference between actual and predicted values.

$$RMSE = \sqrt{\frac{1}{n}\sum_{j=1}^{n}(y_j - y_j)^2}$$

Characteristics :-

→ Accounts impact of outliers
→ Error is in same unit as that of the data points.

RMSE & MAE = which metrics is better without outliers

↳ Both RMSE & MAE are used to measure the model's accuracy.

RMSE & MAE = which metrics is better with outliers

↳ In RMSE, since errors are squared before they are averaged, RMSE gives relatively high weight to large error.

↳ Thus RMSE is more useful when large errors occur.

Scanned by CamScanner

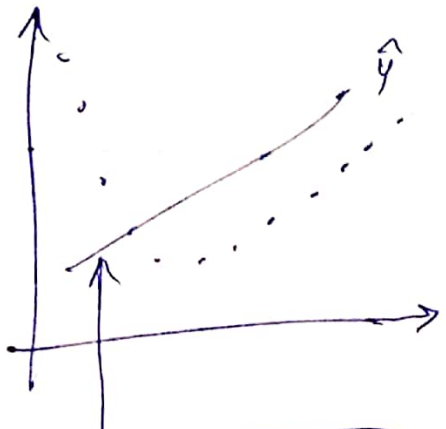
Assumptions in Linear Regression

→ A **Linear Relation** between dependent & independent variables.

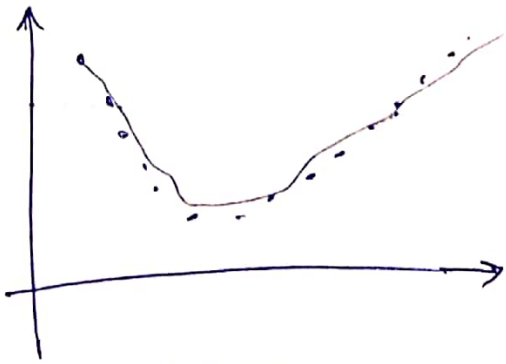
→ No **MultiCollinearity** between independent variables.

→ Residuals (Errors) are **homoscedastic**.

→ Residuals are **normally distributed**.

① 

Regression **line doesn't capture** much information due to distribution of data points



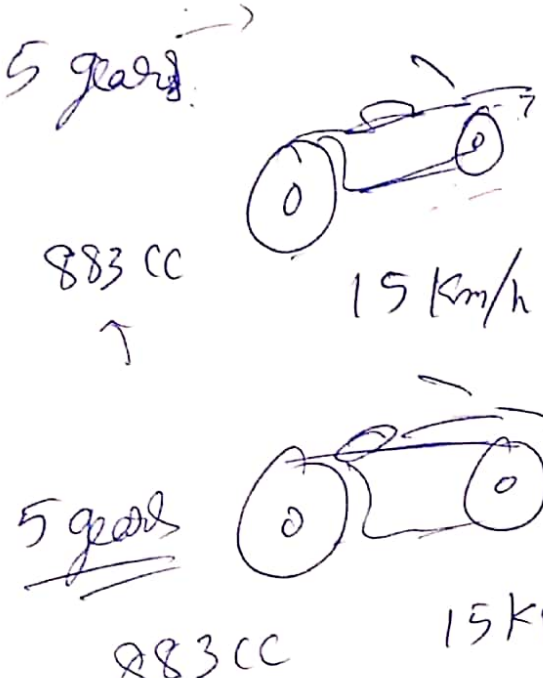
A **non-linear** approach works better in this scenario

② A statistical phenomenon where,

→ Independent explanatory variables are highly positive correlated with one another.

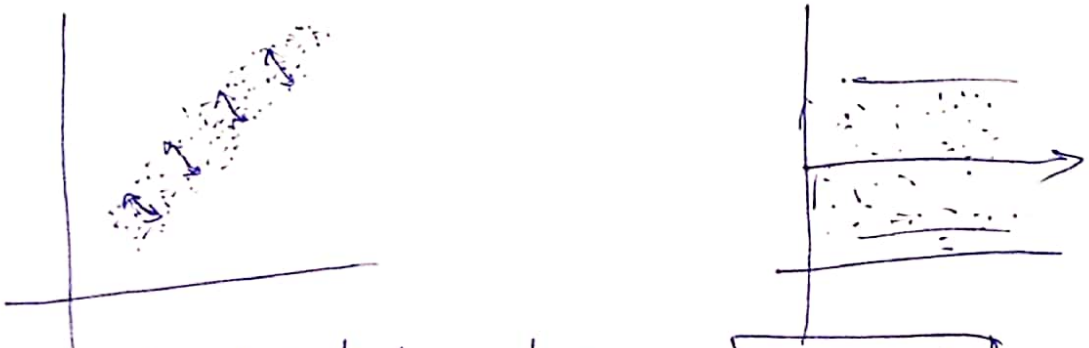
→ Highly Correlated independent variables explain the same variance about dependent variable. Thus introducing **Redundancy**.

Scanned by CamScanner



(3) Homoscedasticity ?

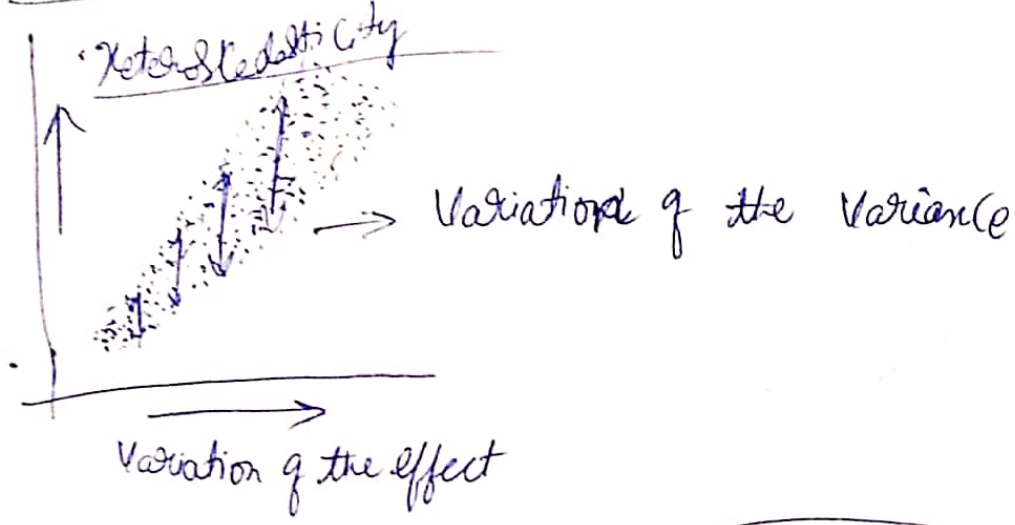
Homo = equal + Scedastic = dispersion = Homoscedastic implies Constant Variance.



Spread of residual values is **Constant** as we move along x-axis..

Scanned by CamScanner

Heteroskedastic looks like this

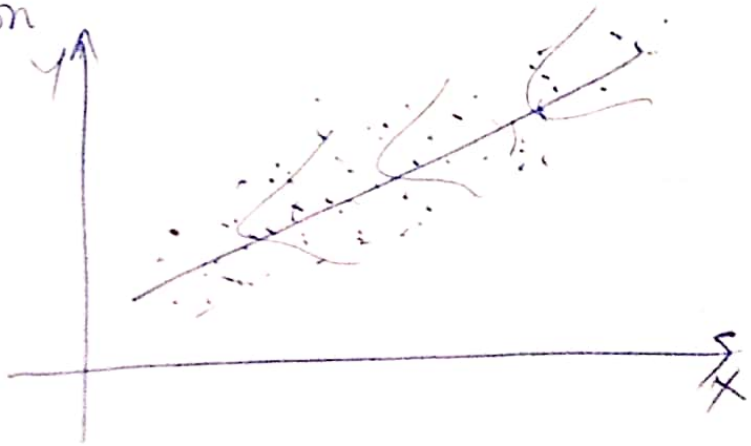


Spread of residual values increases as we move along ~~x~~ x-axis.

Bigger residuals imply greater error in Prediction

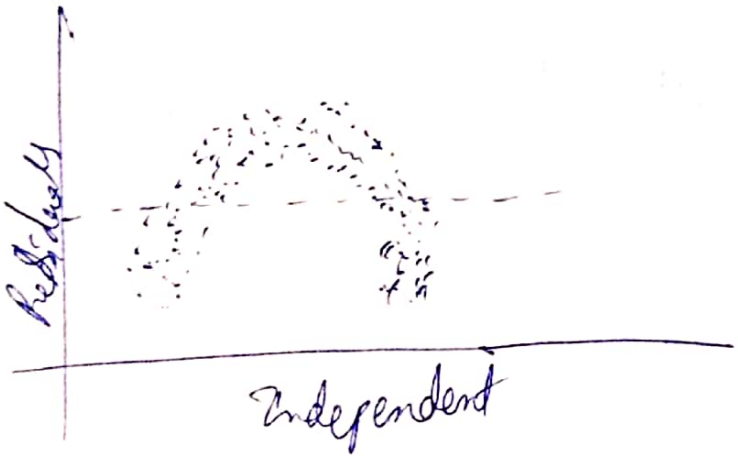
(4) Normal Distribution of residuals.

Data points should be random and follow a normal distribution



Scanned by CamScanner

Errors follow a curved pattern thus introducing bias



Exceptions to normal distribution assumption

↳ For larger samples, residuals may follow any pattern & biasness but still linear regression

↳ Normality is the (least important) assumption of linear regression.

Terminologies associated with error

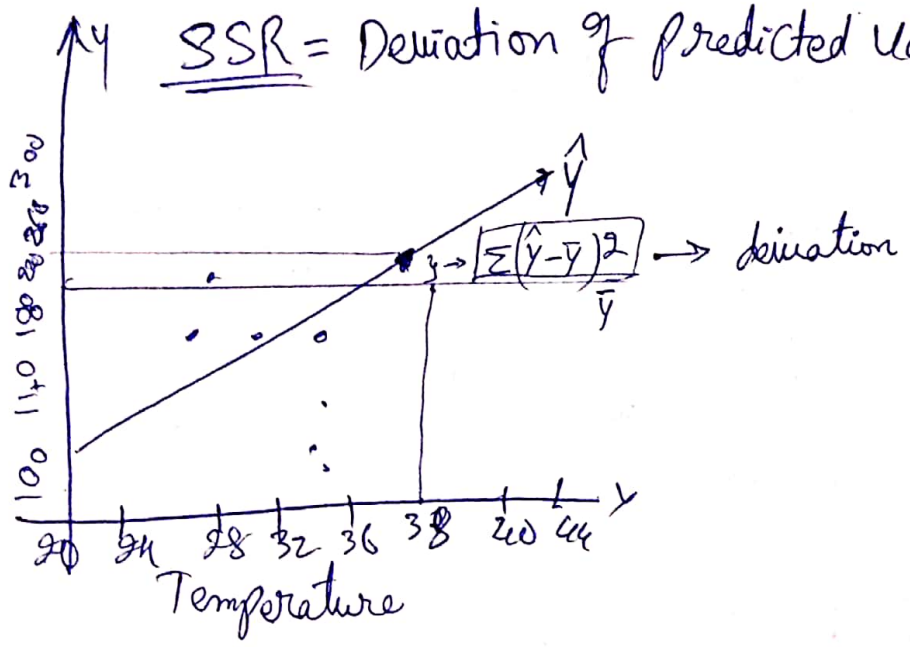
↳ SSR = Sum square due to regression.
↳ Explained deviation from mean.

↳ SSE = Sum square due to Error.
↳ Unexplained deviation from mean.

↳ SST = Sum square Total (SSE + SSR)

Scanned by CamScanner

SSR = Deviation of Predicted values from mean

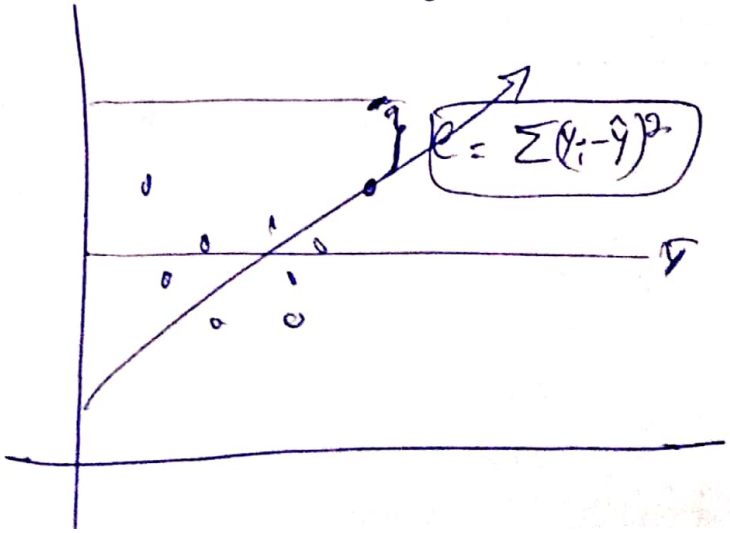


→ At 38°C on a particular day, we predicted 260 ice Cream to be sold which is higher than mean $\bar{y}$.

→ This deviation collectively for all predictions is termed as SSR (Sum Square due to Regression) or ESS (Explained Sum of Square)

Inference : we allept this deviation & use it to calulate R^2.

SSE = Deviation of Predicted values from actual value



Scanned by CamScanner

↳ At 38°C on a particular day, Actual no of ice Cream Sold was 300 which is higher than our Prediction (260 ice Cream)

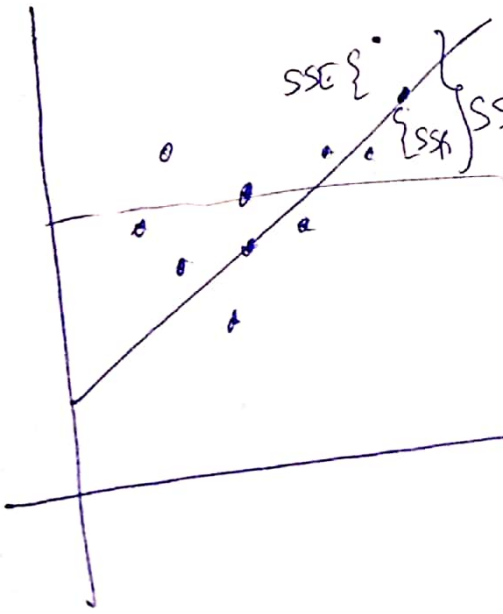
↳ This deviation or difference is called SSE (Sum Squared Error). The deviation is Unexpected or is Unexplained from mean $\bar{y}$.

Inference

↳ This difference can't be explained by the difference in independent variable.

↳ We want this as small as possible.

SST = Deviation of actual values from mean



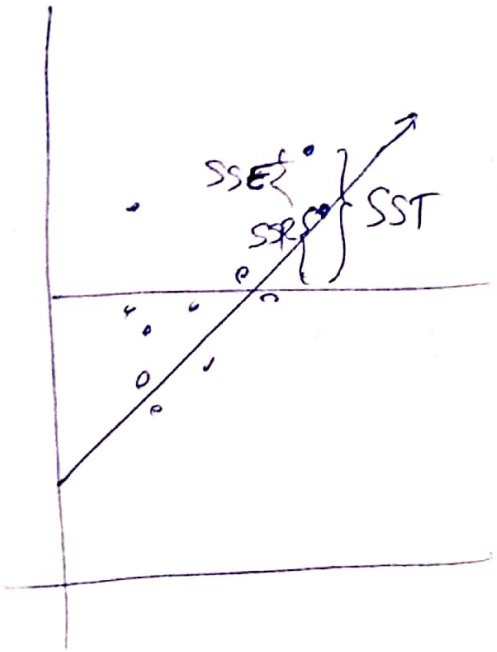
↳ Sum square total (SST) is the deviation of actual data point (y_i) from mean ($\bar{y}$).

↳ SST is composed of two parts, SSE & SSR

$$SST = SSE + SSR$$

Scanned by CamScanner

R - Squared



$$R^2 = \frac{SSR}{SST} \text{ or } 1 - \frac{SSE}{SST}$$

↳ most popular metric to measure performance of regression model.

↳ Tells about Variability or difference in Y Variable explained by difference in X Variable(s).

R^2 = Coefficient of determination

↳ R^2 is square of Correlation Coefficient (r) b/w dependent (Y) & independent (X) Variables.

↳ r tells us the degree of association b/w X & Y.

→ For Example, $R^2 = 0.81$ so $r = 0.9$, implies 90% change in Y is known to X & vice versa.

↳ Remaining 10% association b/w X & Y is Unknown & this is why X can't Explain remaining 19% difference (SSE) in Y.

Scanned by CamScanner

Adjusted R Squared

Adjusted R Squared is more powerful metric than R Squared to measure the linear regression model.

↳ It measures the proportion of variation explained by only those independent variables that really help in explaining the dependent variable.

↳ It penalizes you for adding independent variable that do not help in predicting the independent variable.

Variables	R-Squared	Adjusted R-Squared
1	67.5	67.1
2	85.9	84.2
3	88.9	81.7

↳ Adding 2nd variable adj. R^2 has increased along with R^2 because the variable has real impact on the dependent variable

Scanned by CamScanner

→ But 3rd Variable increased R^2 & Reduced Adj. R^2 implies that the Variable is not impactfull at all.

R-Squared	Adjusted R-Squared
Every time an independent Variable added to the model, the R-Square increases, even if the independent Variable is insignificant. R^2 never decrease.	Adjusted R-squared increases only when independent Variable is significant and affects dependent Variable
R^2 Can be zero	Adj R^2 Can be negative

Scanned by CamScanner

Machine Learning Algo's

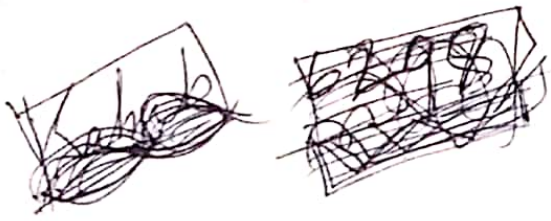
Steps involved in ML are

1. Load DataSet
2. EDA ⇒ An exhaustive & Extensive one

} PREML

3. Create Features & label
4. Split, Train and Test Data and Cross Validation
5. Create Instant of the model
6. Fit the Model.
7. PREDICT test data using trained model
8. Evaluate model Performance using Evaluation metrics

} Sub-Steps need to be Explored by Me



Scanned by CamScanner

Linear Regression

All steps with intermediary steps ~~[illegible]~~ involved in the algo is as below. [No template is necessary, But to ~~[illegible]~~ remember all steps are essential -]

(1) Import the ~~[illegible]~~packages (Load the data)

(2) Import Packages

(3) Explore data
Bikes.info
Bikes.head
} Clean Data ⇒ Pandas
Visualize Data ⇒ Seaborn & matplotlib

Step 1 - Creat features & label

~~[illegible]~~

```
X = bikes.drop(['count'], axis=1, inplace=False)
y = bikes['count']
X.shape
y.shape
```

P.T.O.

Scanned by CamScanner

Step 2 - Split • train & Test Data ⇒ only for Supervised

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=999)

→ any no

→ To get the same output

from sklearn.model_selection import train_test_split

→ Need to verify

X_train.shape
y_train.shape
X_test.shape
y_test.shape

Step 3 - Create Instant of the model.

from sklearn.linear_model import LinearRegression

lm = LinearRegression().

Step 4 - Fit model

lm.fit(X_train, y_train)

Step 5 Predict using the trained model.

predicted = lm.predict(X_test)

Predicted.shape

Step 6 Evaluate model performance

Scanned by CamScanner

metrics.mean_squared_error(y_test, predicted)

metrics.mean_absolute_error(y_test, predicted)

metrics.median_absolute_error(y_test, predicted)

→ lm.coef_

→ X.columns → just to see the attributes/variables

→ lm.intercept_

↳ Additional lines of codes to be entertained

→ In the last step we are appending the predicted house prices into the original data and computing Error in estimation for the test data.

```
fdf = pd.concat([test_x, test_y], 1)

fdf['Predicted'] = np.round(Predict_test, 1)

fdf['Prediction_Error'] = fdf['House_price'] - fdf['Predicted']
```

↳ P.T.O.

Fitting Using different methods

Scanned by CamScanner

Using Stats models to build linear Regression

```
import statsmodels.formula.api as Smf
model = Smf.ols('House_price ~ CRIM+ZN+ ..... ', data=Boston_df).fit()
```

(all features)

```
Print(model.summary())
```

Remove Insignificant Variables → Based on P- Coefficients

Print Heteroscedasticity Robust Std Errors

```
robust_model = model.get_robustcov_results()
Print(robust_model.summary())
```

Scanned by CamScanner