

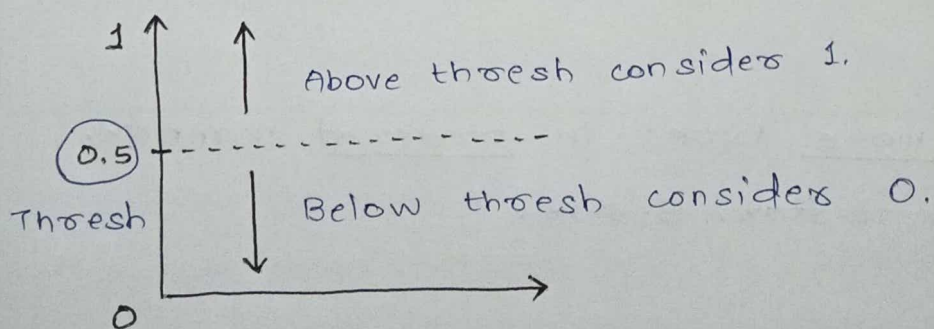
* Logistic Regression *

① Logistic Regression :-

- ↳ Supervised ML algorithm.
- ↳ Used for classification task, predict probability.
- ↳ Statistical algo, analyze relationship betⁿ set of independent var & dependent binary var.
- ↳ Powerful for decision-making.
- ↳ Example - Email prediction - spam or not.
- ↳ Since, uses output of Linear Regression funⁿ ~~as~~
• So referred as Regression, uses sigmoid funⁿ to estimate probability for given class.

• Logistic Function (Sigmoid Function) —

- ↳ Math funⁿ used to map predicted values to probabilities.
- ↳ Logistic Regⁿ must be between 0 & 1, cannot go beyond 1, form curve like "S" form.
- ↳ S-form curve called sigmoid / Logistic Funⁿ.
- ↳ Use concept of threshold value, define probability either 0 or 1. (set to 0.5).

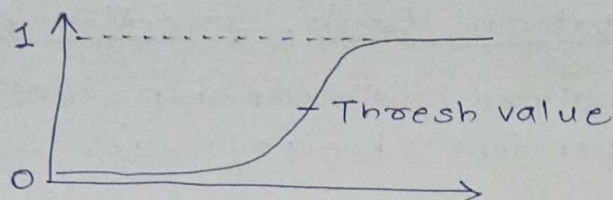


- ↳ Can set thresh as per requirement.

• Sigmoid Function:-

$$S_y = \frac{1}{1 + e^{-(y)}}$$

Where, $y = mx + c$ (Eqⁿ of Linear Line).



① Assumptions of Logistic Regression -

- ① Dependent variable must be categorical.
- ② Independent variable must not multi-collinear.
- ③ Resultant var must be binary.

② Types of Logistic Regression -

① Binary / Binomial:

↳ Dependent variable have only 2 types - 1 or 0.

↳ Ex: 0 or 1, Pass or Fail, Success or Fail.

② Multinomial: 3 or more possible types (unordered).

↳ Ex: low, high, medium.

③ Ordinal: 3/more types in ordered manner.

↳ Ex: 1-star, 2-star, 3-star.

<u>Linear Regression</u>	<u>Logistic Regression</u>
① Predict <u>continuous</u> var (Numeric data).	① Predict <u>categorical</u> var.
② Solve <u>Regression</u> problem.	② <u>Classification</u> problem.
③ Find <u>best fit line</u> .	③ Find <u>S-curve</u> .
④ Accuracy: <u>Least square</u> estimation.	④ <u>Max Likelihood</u> Estima ⁿ .
⑤ Output must be <u>continuous</u> (numeric) value. Ex. price, age.	⑤ <u>categorical</u> value. Ex. 0 or 1, Yes or No.
⑥ Required to have the <u>Linear relationship</u> bet ⁿ dependent & independent variables.	⑥ Linear relationship <u>not required</u> .
⑦ May be <u>collinearity</u> bet ⁿ independent variables.	⑦ should <u>no collinearity</u> bet ⁿ independent variables.

• Terminologies involved in Logistic Regression -

- ① Independent Variables - Input characters / predictors factors used to find outputs.
- ② Dependent Variable - Target variables, is to be predicted.
- ③ Logistic Funⁿ - Formula used to represent how independent & dependent relate to each other.
↳ Transform i/p variable to probability value betⁿ 0 & 1.
- ④ Odds - Ratio of something occurring to something not occurring.
- ⑤ Log-odds - Also k/a Logit funⁿ, is natural algo of odds.
↳ In LR, log odds of dependent var modeled as linear combinaⁿ of independent var & intercept.
- ⑥ Coefficient - LR model's estimated parameters, show how dep & indep var relate to one another.
- ⑦ Intercept - Constant term in LR model, represent log odds when all indep var equal to zero.
- ⑧ Maximum ~~likelihood~~ likelihood estimation - Method used to estimate coefficients of LR model, which maximizes likelihood of observing data from model.

① Logistic Regression Implementation steps:-

① Import Libraries - `pd, np, plt.`



② Import Dataset - `df = pd.read_csv('___')`



③ Pre-processing -

- Null value treatment
- Remove duplicates
- Handle categorical data.



`from sklearn.model_selection import train_test_split`
④ Split Dataset - `x = df.drop('target', axis=1)`
`y = df['target']`

`x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.20, random_state=42)`



`from sklearn.linear_model import LogisticRegression`
⑤ Model Training - `regressor = LogisticRegression()`
`regressor.fit(x_train, y_train)`



⑥ Result Prediction - `y_pred = regressor.predict(x_test)`



`sklearn.metrics => accuracy_score, classification_report`
⑦ Model Evaluation -

`score = accuracy_score(y_pred, y_test)`
`rp = classification_report(y_pred, y_test)`



`sklearn.metrics import roc_curve, auc`
⑧ AUC-ROC Curve -

`fpr, tpr, thresh = roc_curve(y_test,`
`regressor.predict_proba(x_test)[:,1])`
`area = auc(fpr, tpr)`

`plt.plot(fpr, tpr)`

`plt.plot([0,1], [0,1], 'k--')`

`plt.show()`