Logistic Regression - Theory

What is Classification?

Predict Categories of dependent Variable (output) Using independent (Input) Variable(s).

↳ Will it rain tomorrow?
↳ Will the flight arrive in time?

Log. Regression :- Type of classification technique while Underlying Concept is based on Linear Regression to predict Categories.

Eg:- given the past data, should we give loan to this Customer?

Use Cases of Logistic Regression

Finance => Can Credit be approved?
HealthCare => Dr. am I Diabetic?
Engineering => Will the machine work after 6 months?

Benefits of Logistic Regression

→ Dependent Variable doesn't need to be Correlated with independent Variables.

→ No Normality assumption of dependent Variable

→ There is no homogeneity of Variance assumption.

→ Normal distribution of Errors is not assumed.

→ Tree based & other algorithms may have higher accuracy, but Logistic Regression is used preferably in Probability Based Solution.

Why not use linear Regression for Classification?

$\hat{Y} = \beta_1 x + \beta_0$ - This equation has solution to the gn.

If we use Linear Regression the line incorrectly divides a single Category into two.

→ No Linear Regression can't Predict value of Categorical Variable.

→ The line drawn doesn't Capture any information about outcome variable.

Challenges with LR:-

↳ output becomes greater than 1 & less than 0.

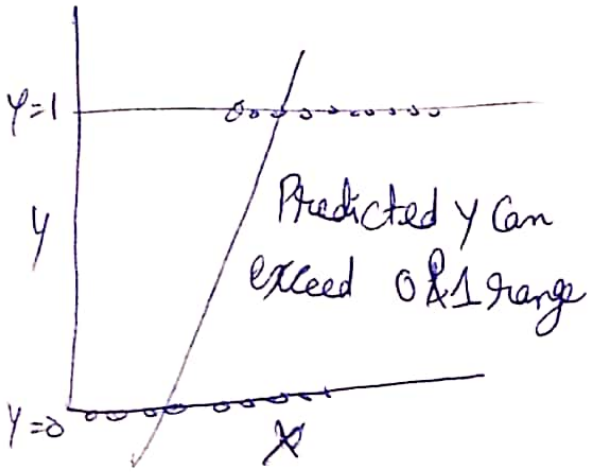
↳ This violates the rule of Probability.

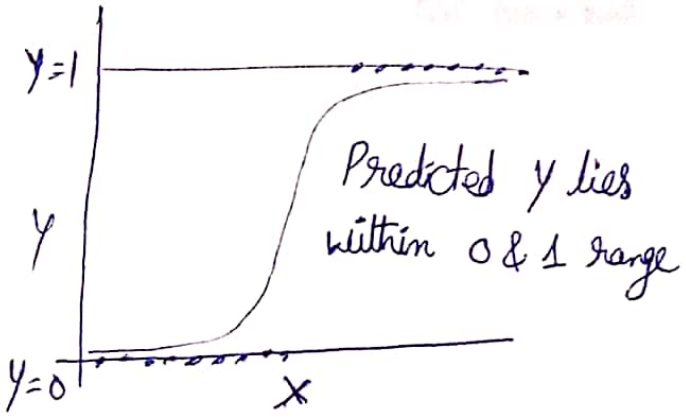
$$P(Y) = \beta_0 + \beta_1 X$$

→ Logistic Regression helps us when output Variable is Categorical.

⇒ Transforms Linear function to a function that

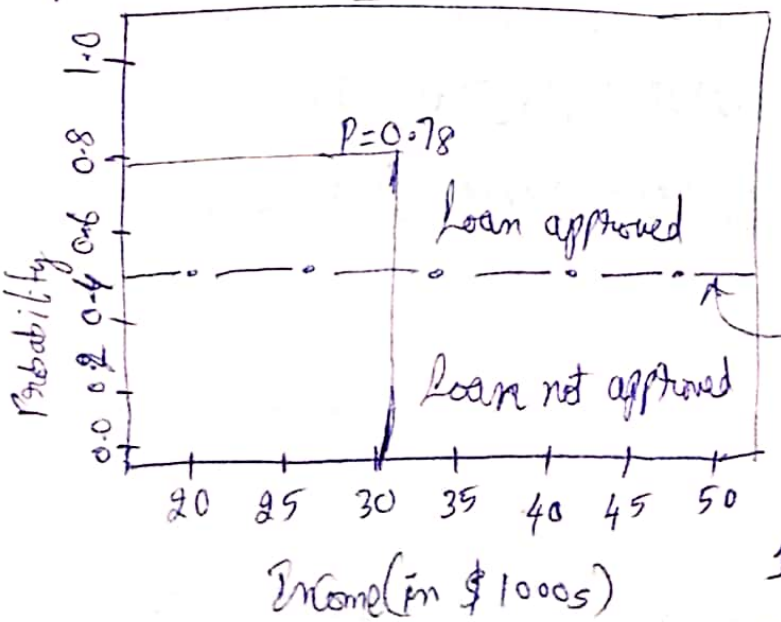
Can Predict Categories with Probabilities.





Logistic Regression always outputs a Probability Value to assign observations to classes.

Assigning observations to Classes (Yes(1)/No(0)) depends on threshold Probability.



threshold Probability (0.5)

1 (approved), because 0.78 > 0.5

0.78 > 0.5 (default threshold Probability)

Odds Ratio

Logistic Regression Can be written in terms of odds Ratio

$$\text{Odds Ratio} = \frac{\text{Probability of event occuring}}{\text{Probability of event not occuring}}$$

$$\text{Odds Ratio} = \frac{P}{1-P}$$

applying odds ratio on loan approved example,

Odds of loan being approved is

$$\frac{0.78}{0.22} = 3.54$$

Logit Function / Sigmoid Function

Transformation of linear function into non linear function that predicts Categories.

$$\text{logit}(P) = \ln \frac{P}{1-P} = \beta_0 + \beta_1 x$$

Let's Break logit function into Probability.

$$\ln\left(\frac{P(Y)}{1-P(Y)}\right) = \beta_0 + \beta_1 X$$

$$P(Y) = \frac{e^{\beta_0+\beta_1 X}}{1+e^{\beta_0+\beta_1 X}}$$

This is also called Sigmoid function that is used to Predict probability to assign observations to Categories.

$$0 \leq \frac{e^{\beta_0+\beta_1 X}}{1+e^{\beta_0+\beta_1 X}} < 1$$

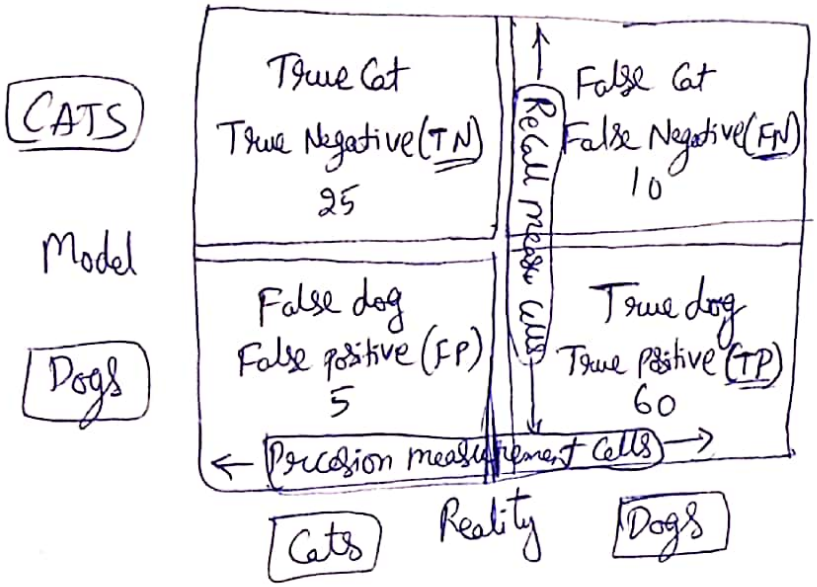
↳ Equation above gives probability b/w 0 & 1

↳ As a result, it fulfills the criteria of assigning observations to classes with probability.

Classification metrics

metrics used to evaluate Classification Algo

Confusion matrix, Accuracy Precision and Recall



Reality => Total Animal = 100
Total Dogs = 70
Total Cats = 30

Model Prediction => objective → identify Dogs

True positive = 60
↳ Dogs Correctly identified

False Negative = 10
↳ Dogs not Correctly identified

True Negative = 25
↳ Cats Correctly identified.

False Positive = 5
↳ Cats not Correctly identified.

Model Performance

To identify dogs

$$\text{Accuracy} = (TP + TN)/\text{Total} = 85\%$$

⇒ How many Cats and Dogs are correctly identified by model.

$$\text{Precision} = TP/(TP+FP) = 92.3\%$$

⇒ When model identifies an animal as Dog, how many times model is right.

$$\text{Recall} = TP/(TP+FN) = 85.7\%$$

⇒ out of all Dogs (Total dogs), how many dogs model is able to identify.

$$\text{F-1 score} = \frac{(\text{Precision} * \text{Recall}) \times 2}{(\text{Precision} + \text{Recall})}$$

(arithmetic mean) $\frac{x+y}{2}$

Harmonic mean of Precision & Recall $\rightarrow \frac{2xy}{x+y}$

Closer the F score to 1, the better the model

Logistic Regression

1. Import packages (load the data)
2. EDA → Boxplot, Hist, Corr, Pairplot,
3. Data cleaning
   - missing values
   - outlier treatment

Step 1 — Substeps

4. Feature Engineering.

Converting Categorical to Numerical for Column

```
default_dummies = pd.get_dummies(Cred_df.default, prefix='default',
                                 drop_first=True)
Cred_df = pd.concat([Cred_df, default_dummies], axis=1)
```

```
Student_dummies = pd.get_dummies(Cred_df.student, prefix='student',
                                 drop_first=True)
Cred_df = pd.concat([Cred_df, student_dummies], axis=1)
```

Label encoding is used for more than two-classes

Remember the Confusion matrix; very very imp

Step = 1

~~Labels &~~ ~~Splitting~~ & features & labels Splitting/Create

x = ...

y = ...

Step = 2

Splitting the data into train & test

from sklearn.model_selection import train_test_split

x_train, x_test, y_train, y_test

= train_test_split(x, y, test_size = 0.30, random_state = 0)

Step 3 Creating Instance of the model

```
from sklearn.linear_model import LogisticRegression
from sklearn import metrics
logreg = LogisticRegression()
```

Step 4 Fitting the model

~~logreg = LogisticRegression()~~

```
logreg.fit(x_train, y_train)
```

Step 5 Predict the model

```
y_pred_test = logreg.predict(x_test)
```

Step 6 Evaluation of model

```
print("Test accuracy:", metrics.accuracy_score
                        (y_test, y_pred_test))
```

~~Test accuracy~~

```
print("Train accuracy:", metrics.accuracy_score
                        (y_train, y_pred_train))
```

super 6 months!

Creating Confusion matrix

```
Conf = metrics.confusion_matrix(y_test, y_pred_test)
```

Classification Report

```
cr = metrics.classification_report(y_test, y_pred_test)
```