Logistic regression..

Logistic regression is used to predict classification problems there are 2 categories.

① Binary logistic regression (binary classification)
② ~~Multiclass~~ Multivariable logistic regression (multi class classification)

⊙ Logistic Regression is used when the dependent variable (target) is categorical.
⊙ It is the go-to method for binary classification problems (problems with two class values).
⊙ Logistic regression is a statistical method for analyzing a dataset in which there are one or more independent variables that determine an outcome. The outcome is measured with a dichotomous variable (in which there are only two possible outcomes).
⊙ In logistic regression, the dependent variable is binary or dichotomous, i.e. it only contains data coded as 1 (TRUE, success, pregnant, etc) or 0 (FALSE, failure, non-pregnant, etc).
⊙ The goal of logistic regression is to find the best fitting (yet biologically reasonable) model to describe the relationship between the dichotomous characteristic of interest (dependent variable = response or outcome variable) and a set of independent (predictor or explanatory) variables.

In simple words, it predicts the probability of occurrence of an event by fitting data to a logit function.

Hence, it is also known as logit regression. Since, it predicts the probability, its output values lies between 0 and 1 (as expected).

A logistic function or logistic curve is a common 'S' shape (sigmoid curve) with equation

$$f(x) = \frac{1}{1+e^{-K(x-x_0)}} \text{ or } \frac{1}{1+e^{-x}}$$

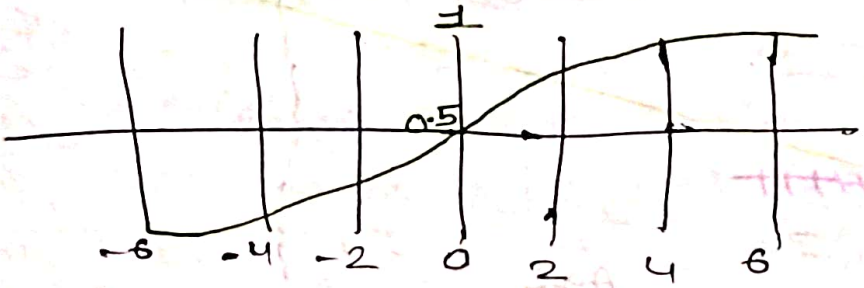
where,

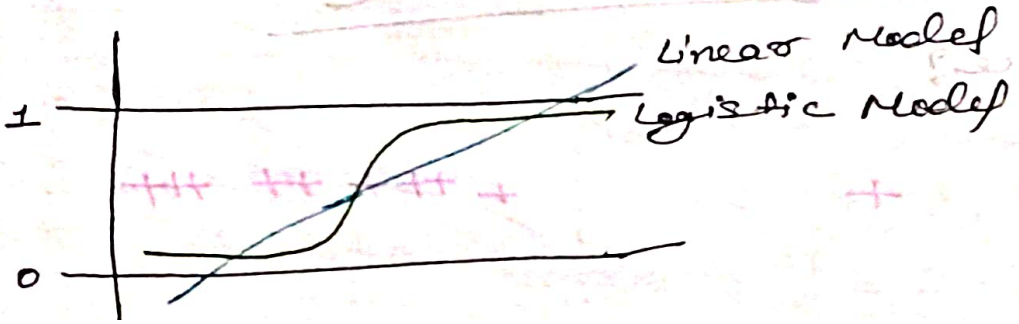
e = the natural logarithm base (also known as Euler's number).

x_0 = the x-value of the sigmoid's midpoint.

~~L = the curve's maximum value and~~

K = the logistic growth rate of steepness of the curve.

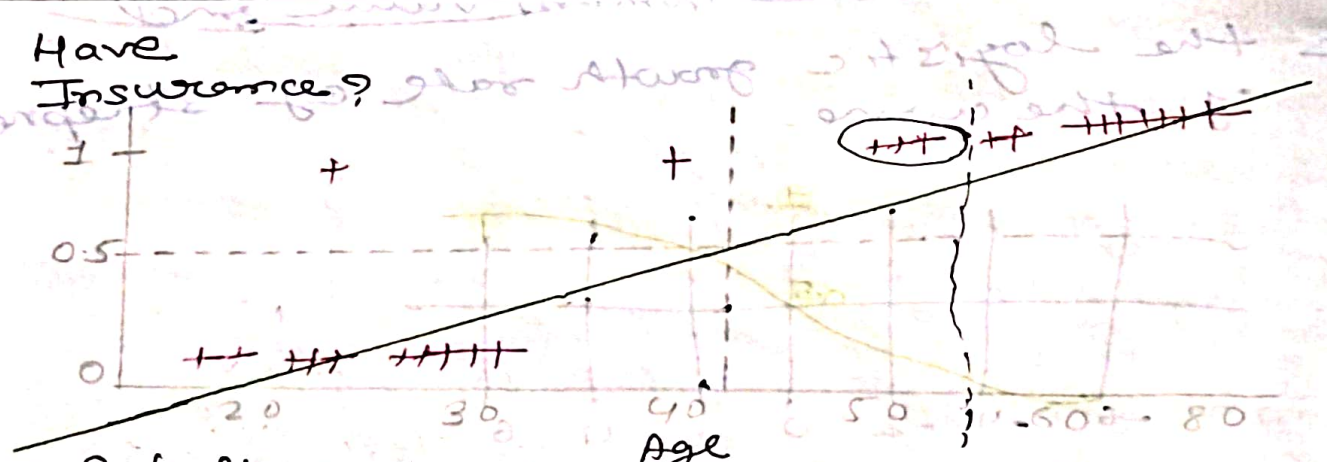




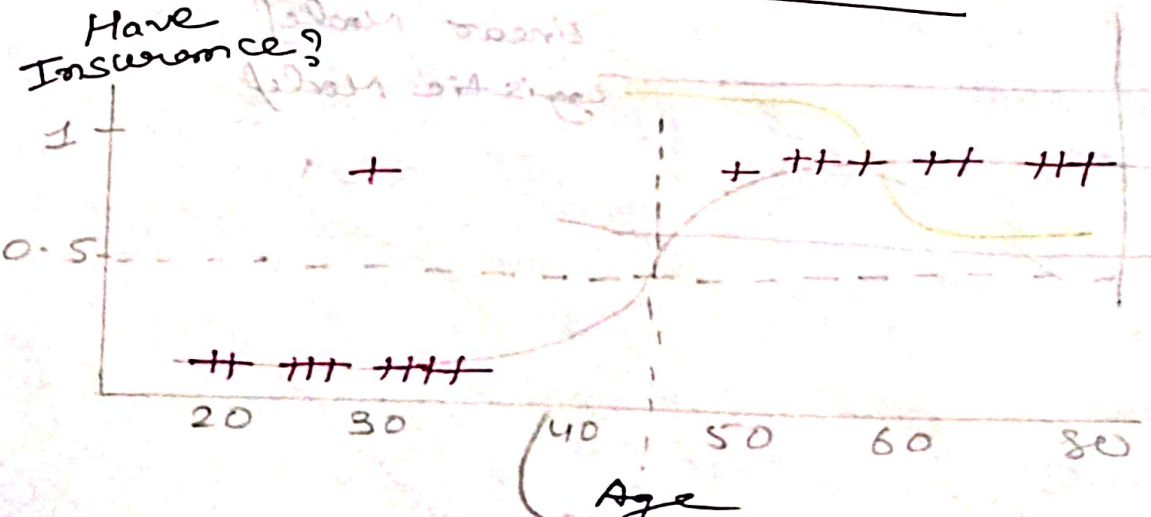
Prediction – Regression

- Prediction is similar to classification
  - ↳ First, construct a model
  - ↳ Second, use model to predict unknown value.
    - Major method for predict unknown value.
      - ⊙ Linear and multiple regression.
      - ⊙ Non-linear regression.
- Prediction is different from classification
  - ⊙ Classification refers to predict categorical class label.
  - ⊙ Prediction models continuous-valued functions.

Classification problem using linear regression



Solution.. Logit function



Confusion matrix for accuracy

- A confusion matrix is a summary of prediction results on a classification problem.
- The no. of correct and incorrect predictions are summarized with count values and broken down by each class. This is the key to the confusion matrix.
- The confusion matrix shows the ways in which your classification model is confused when it makes predictions.
- It gives us insight not only the errors being made by a classifier but more importantly the types of errors that are being made.

Actual	Prediction	
1	1	→ TP
1	0	→ FN
0	1	→ FP
0	0	→ FN

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

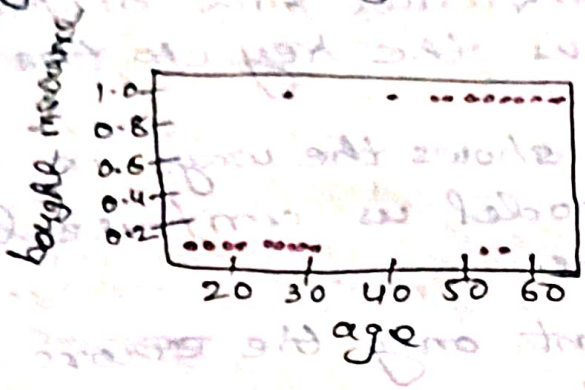
```
import pandas as pd
import numpy as np
from matplotlib import pyplot as plt
from sklear.model_selection import train_test_split
from sklear.linear_model import LogisticRegression
```

Data fetch and train_test_split

```
data = pd.read_csv('insurance_data.csv')
x = data[['age']]
y = data['bought_insurance']
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=.2)
```

Plot it..

```
plt.scatter(x,y, color='orange', marker='s')
plt.xlabel('age')
plt.ylabel('bought insurance')
```



Model it and predict..

```
model = LogisticRegression
model.fit(x_train, y_train)
model.predict(x_test)
y_test
model.score(x_test, y_test)
```

Assignment

(1) Now do some exploratory data analysis to figure out which variables have direct and clear impact on employee retention (i.e. whether they leave the company or continue to work).

(4)
```
com = pd.read_csv(r'path\filenamewithextension')
com
newdep = {
    'sale':1,
    'support':2,
    'accounting':3,
    'hr':4,
    'technical':5,
    'management':6,
    'IT':7,
    'product_mng':8,
    'marketing':9,
```

```
com['Department'] = com['Department'].replace(newdep)

com
com.Department.unique()
com.salary.unique()
newsal = {
    'low': 0,
    'medium': 0.5,
    'high': 1
}
com['salary'] = com['salary'].replace(newsal)
com
x = com[['statisfaction_level', 'last_evaluation',
    'number_project', 'average_monthly_hours', 'time_spend_company', 'Work_accident', 'promotion_last_5years', 'Departmat',
    'salary']]
y = com['left']
from sklearn.model_selection import train_test_split.
from sklearn.linear_model import LogicalRegression.
xtrain, xtest, ytrain, ytest = train_test_split(x, y, test_size = .2)

model = LogisticRegression()
model.fit(xtrain, ytrain)
model.predict(xtest)
ytest
model.score(xtest, ytest)
(Ex) groupeed = com.groupby('left')['salary'].mean().reset_index.rename(columns={'salary': 'avg_salary'}).round(2)
```

grouped

left	avg_salary
0	0.33

(2) Plot bar chart showing impact of employee salaries on retention.

```
grouped = df.grouped('salary', 'left')['left'].mean().
                                        round(2)
colors = ['c', 'b']
plt.figure(figsize=(12,6))
grouped.plot(kind='bar', color=colors)
plt.xlabel("Not working")
plt.ylabel("Average Salary")
plt.show()
```

(3) Plot bar charts showing correlation between department and employee retention.

```
retention_by_department = df.groupby('Department')
                          ['left'].mean().reset_index()
plt.figure(figsize=(10,6))
sns.barplot(data=retention_by_department,
            x='Department', y='left', palette=
                                    'viridis').
plt.title('Employee Retention by Department')
plt.xlabel('Department')
plt.ylabel('Retention Rate')
plt.legend()
plt.xticks(rotation=45)
plt.show()
```

(4) Now build logistic regression model using variables that were narrowed down in step

(5) Measure the accuracy of the model.

```
model.score(x,y)
```

Logistic regression with multi class classification

This kind of problem will have multiple classifications.

Like which team will win the world cup? --> its having multiple options.

We are going to identify handwritten digit recognition.

```
import matplotlib.pyplot as plt
from sklearn.datasets import load_digits
from sklearn.model_selection import train_test_split
from sklearn.metrics import confusion_matrix
from sklearn.linear_model import LogisticRegression
```

```
digits = load_digits()
len(digits)
dir(digits)
digits.data[100]
digits.target[0:5]
digits.images[0]
plt.matshow(digits.images[0])
```

Model creation.

```
xtrain, xtest, ytrain, ytest = train_test_split(digits.data, digits.target, test_size = .2)
model = LogisticRegression()
model.fit(xtrain, ytrain)
model.score(xtest, ytest)
```

Random prediction..

Random sample prediction means using predict with random

```
plt.matshow(digits.images[888])
digits.target[888]
model.predict([digits.data[888]])
```