Experiment
Assignment No. 03

Title of the Assignment:

Descriptive Statistics - Measures of Central Tendency & variability.

Perform the following operations on any open source dataset (eg. data.csv)

1. Provide summary statistics (mean, median, minimum, maximum, standard deviation) for a dataset (age, income, etc.) with numeric variables grouped by one of the qualitative (categorical) variables. For example, if your categorical variable is age groups & quantitative variable is income, then provide summary statistics of income grouped by the age groups. Create a list that contains a numeric value for each response to the categorical variable.
2. Write a python program to display some basic statistical details like percentile, mean, standard deviation etc. of the species of 'Iris-setosa', 'Iris-versicolor' & 'Iris-versicolor' of iris.csv dataset.

Provide the codes with o/p & explain everything that you do in this step.

Objective of the Assignment :→

Students should be able to perform the statistical operations using python on any open source dataset

Prerequisite :→

1. Basic of python progearoming
2. Concept of statistics such as mean, median, minimum, maximum, standard deviation etc.

Experiment No. 04

Title of the Assignment :→

Create a Linear Regression Model using python/R to predict home prices using Boston Housing Dataset (http://www.kaggle.com/c/boston-housing). The Boston housing dataset contains information about various houses in Boston through different parameters. There are 506 samples & 14 feature variables in this dataset.
The objective is to predict the value of prices of the house using the given features.

Objectives of the Assignment :→

Students should be able to data analysis using linear regression using python for any open source dataset.

Prerequisite :→

1. Basic of python programming.
2. Concept of Regression.

Theory :→

(1) Linear Regression :→

- It is a mlc learning algo. based on supervised learning. It targets predicts values on the basis of independent variables.

- It is preferred to find out the relationship betⁿ forecasting & variables.
- A linear relationship betⁿ a dependent variable (x) is continuous; while relationship should be available in betⁿ independent variable (Y). relationship may be continuous. A linear relationship should be available in betⁿ predictor & target variable & so known as Linear Regression.
- It is should as an eqⁿ of line like:

$$Y = m*X + b + e$$

where,

b ⇒ intercepted

m ⇒ slope of line

e ⇒ error term

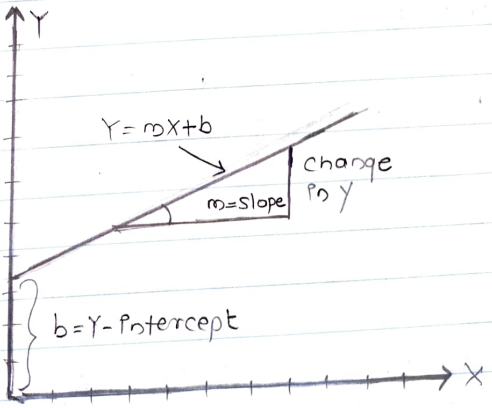



Fig. 01: Geometry of linear regression

- This eq$^n$ can be used to predict the value of target variable Y based on given predictor variables (X). as shown in fig.01.
- Fig.02 shown below is about the relation bet$^n$ weight (in kg) & height (in cm), a linear relation. It is an approach of studying in a statistical manner to summerise & learn the relationships among continuous (quantitative) variables.
- Here a variable, denoted by 'x', is considered as predictor, explanatory or independent value. & y is considered as the response, outcome or dependent variable.

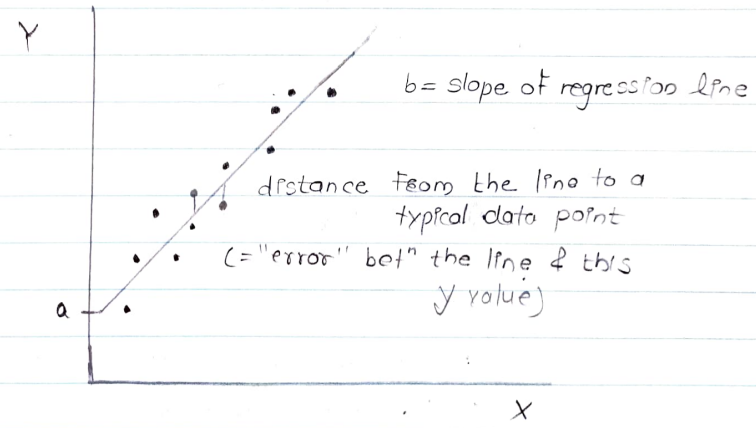



Fig.02: Relation bet$^n$ weight (in kg) & height (in cm)

- Multivariate Regression?
  - It concerns the study of two or more predictor variables.
  - A simple linear model $Y = a + bX$ is in original feature will be transformed into polynomial feature is transformed & further a linear regression applied to it & it will be something like

$$Y = a + bX + cX^2$$

(2) Least Square Method for linear Regression :→

- A simple linear model is the one which involves only one dependent & one independent variable. Regression Model are usually denoted in Matrix Notations.
- However for a simple univariate linear model it can be denoted by the regression eq?

$$\hat{Y} = \beta_0 + \beta_1 x \quad \text{---(1)}$$

where,

$\hat{Y}$ ⇒ dependent or response variable

$x$ ⇒ Independent or i/p variable

$\beta_0$ ⇒ value of $y$ when $x=0$ or $y$ intercept.

$\beta_1$ ⇒ value of slope of line $\varepsilon$ is the error or the noise.

- This linear reg eq$^n$ represents a line also known as the 'regression line'.
- This technique estimate parameters $\beta_0$ & $\beta_1$ of by trying to minimise the square of errors at all the point in the sample set. The error is the deviation of the actual sample data point from the regression line. The technique can be represented by the eq$^n$.

$$\min \sum_{i=0}^{n} (\hat{y} - y)^2 \quad \text{—— (2)}$$

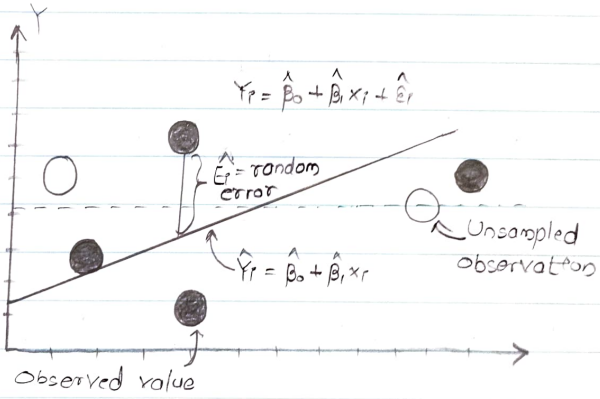
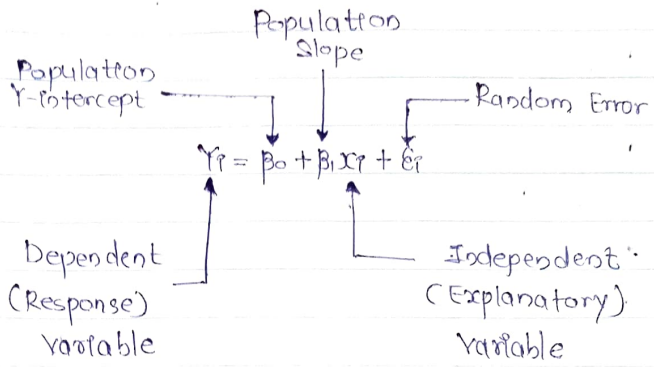



Fig. 03




Using differential calculus on eqn 1 we can find the values of $\beta_0$ that the sum of squares (that is equation 2) is minimum.

$$\beta_1 = \sum_{i=1}^{n} (x_i - \bar{x})(Y_i - \bar{y}) \Big/ \sum_{i=1}^{n} (x_i - \bar{x}^2) \quad \text{— (3)}$$

$$\beta_0 = \bar{y} - \beta_1 \bar{x} \quad \text{— (4)}$$

(3) Measuring Performance of linear regression :

Mean Square Error : (MSE)

It represents the error of the estimator or predictive model created based on the given set of observations in the sample.

$$MSE = \frac{1}{n} \Sigma \underbrace{(Y - \hat{Y})^2}$$

The square of difference bet$^n$ actual & predicted

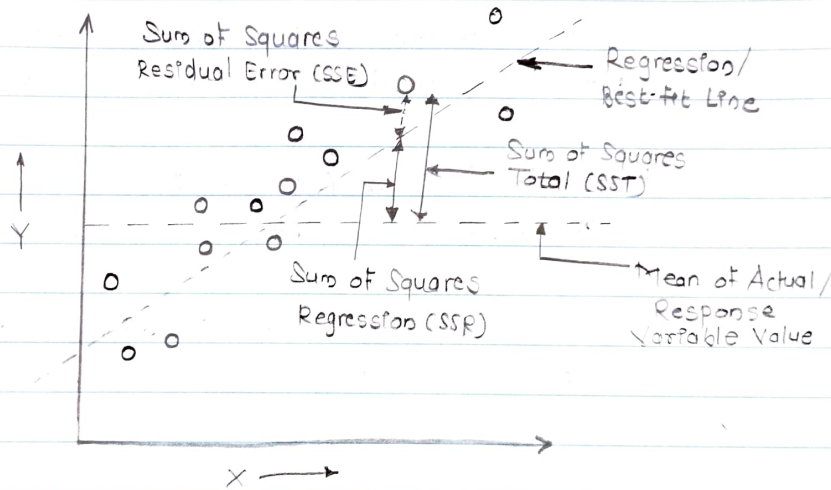
An MSE of zero(0) represents the fact that the predictor is a perfect predictor.

RMSE (Root Mean Squared Error):

This is a method that calculate the least-squares error & take root of summed values

$$RMSE = \sqrt{\sum_{i=1}^{n} \frac{1}{n} (\hat{y}_i - y_i)^2}$$

R-Squared:




R-Squared is the ratio of sum of squares regression (SSR) & the sum of squares total (SST).

$$SST = \sum_{i=1}^{n} (y_i - \bar{y})^2$$

$$SSR = \sum_{i=1}^{n} (\hat{y}_i - \bar{y}_i)^2$$

$$SSE = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

$$R^2 = \frac{SSR}{SST} = \frac{\sum (\hat{y}_i - \bar{y})^2}{\sum (y_i - \bar{y})^2}$$

A value of R-squared closer to 1 could mean that the regression model covers most part of the variance of the values of the response variable & can be termed as a good model.

(4) Example of Linear Regression :-

Consider following data for 5 student.

| Student | Score in X standard ($x_i$) | Score in XII standard ($y_i$) |
|---|---|---|
| 1 | 95 | 85 |
| 2 | 85 | 95 |
| 3 | 80 | 70 |
| 4 | 70 | 65 |
| 5 | 60 | 70 |

| $x$ | $Y$ | $x-\bar{x}$ | $y-\bar{y}$ | $(x-\bar{x})^2$ | $(x-\bar{x})(y-\bar{y})$ |
|---|---|---|---|---|---|
| 95 | 85 | 17 | 8 | 289 | 136 |
| 85 | 95 | 7 | 18 | 49 | 126 |
| 80 | 70 | 2 | -7 | 4 | -14 |
| 70 | 65 | -8 | -12 | 64 | 96 |
| 60 | 70 | -18 | -7 | 324 | 126 |
| $\bar{x}=78$ | $\bar{y}=77$ | | | $\sum(x-\bar{x})^2=730$ | $\sum(x-\bar{x})(y-\bar{y})=470$ |

(i) Linear regression eqn that best predicts standard XIIth score.

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

$$\beta_1 = \sum_{i=1}^{n}(x_i - x)(y_i - \bar{y}) \Big/ \sum_{i=1}^{n}(x_i - \bar{x})^2$$

$$\beta_1 = 470/730 = 0.644$$

$$\beta_0 = \bar{y} - \beta_1 \bar{x}$$

$$\beta_0 = 77 - (0.644 * 78) = 26.768$$

$$\hat{y} = 26.76 + 0.644\, x$$

(ii) Interpretation of regression line

Interpretation 1:

For an increase in value of $x$ by 0.644 units there is an increase in value of $y$ in one unit.

~~Iter~~ Interpretion 2

Even if $x=0$ value of independent variable it is expected that value of $y$ is 26.768.

(iii) If a student's score is 65 in std X, then b is expected score in XII standard is 78.288

for $x = 80$ the $y$ value will be

$\hat{y} = 26.76 + 0.64 * 65 = 68.38$

(5) Training data set & Testing data set

- M/c learning algo. has two phases -

1. Training
2. Testing

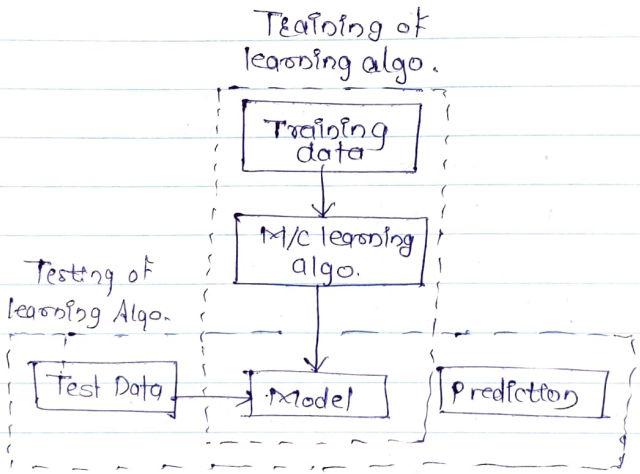



Fig. Training & testing phase in M/c learning.

(a) Training Phase :→

- Training dataset is provided as I/p to this phase
- Training dataset is a dataset having attributes & class labels & used for training mlc learning algo. to prepare models.

(b) Testing phase :→

- Testing dataset is provided as input to this phase.
- Test dataset is a dataset for which class label is unknown, It is tested using model.
- A test dataset used for assessment of the finally chosen model.

(c) Generalization :→

- Generalization is the prediction of the future based on the past system.
- It needs to generalize beyond the training data to some future data that it might not have seen yet.
- The ultimate aim of the mlc learning model is to minimize the generalization error.
- The generalization error is essentially the average error for data the model has never seen.

(a) Training Phase :→
- Training dataset is provided as i/p to this phase
- Training dataset is a dataset having attributes & class labels & used for training m/c learning algo. to prepare models.

(b) Testing phase :→
- Testing dataset is provided as input to this phase.
- Test dataset is a dataset for which class label is unknown, It is tested using model.
- A test dataset used for assessment of the finally chosen model.

(c) Generalization :→
- Generalization is the prediction of the future based on the past system.
- It needs to generalize beyond the training data to some future data that it might not have seen yet.
- The ultimate aim of the m/c learning model is to minimize the generalization error.
- The generalization error is essentially the average error for data the model has never seen.

Algorithm (Synthesis Dataset):

Step-01: Import libraries & create alias for pandas, Numpy & matplotlib.

Step-02: Create a Dataframe with Dependent Variable (x) & Independent variable y.

Step-03: Create Linear Regression Model using polyfit fun

Step-04: Observe the coeff. of the model.

Step-05: Predict the Y value for x & observe the o/p.

Step-06: Predict the y_pred for all values of x

Step-07: Evaluate the performance of Model (R-Square)

Step-08: Plotting the linear regression model

Algorithm (Boston Dataset)

Step-01:

Import libraries & create alias for pandas, numpy & Matplotlib.

Algorithm (Synthesis Dataset):

Step-01: Import libraries & create alias for pandas, Numpy & matplotlib.

Step-02: Create a Datafeame with Dependent Variable (x) & Independent variable y.

Step-03: Create Linear Regression Model using polyfit fu?

Step-04: Observe the coeff. of the model.

Step-05: Predict the Y value for x & observe the o/p.

Step-06: Predict the y_pred for all values of x

Step-07: Evaluate the performance of Model (R-Suare)

Step-08: Plotting the linear regression model

Algorithm (Boston Dataset)

Step-01:

Import libraries & create alias for pandas, numpy & Matplotlib.

Algorithm (Synthesis Dataset):

Step-01: Import libraries & create alias for pandas, Numpy & matplotlib.

Step-02: Create a Dataframe with Dependent Variable (x) & Independent variable y.

Step-03: Create Linear Regression Model using polyfit fun

Step-04: Observe the coeff. of the model.

Step-05: Predict the Y value for x & observe the o/p.

Step-06: Predict the y_pred for all values of x

Step-07: Evaluate the performance of Model (R-Suare)

Step-08: Plotting the linear regression model

Algorithm (Boston Dataset)

Step-01:

Import libraries & create alias for pandas, numpy & Matplotlib.

Algorithm (Synthesis Dataset):

Step-01: Import libraries & create alias for pandas, Numpy & matplotlib.

Step-02: Create a Dataframe with Dependent Variable (x) & Independent variable y.

Step-03: Create Linear Regression Model using polyfit fu^n

Step-04: Observe the coeff. of the model.

Step-05: Predict the Y value for x & observe the o/p.

Step-06: Predict the y_pred for all values of x

Step-07: Evaluate the performance of Model (R-Suare)

Step-08: Plotting the linear regression model

Algorithm (Boston Dataset)

Step-01:

Import libraries & create alias for pandas, numpy & Matplotlib.

Step-02 :
Import the Boston Housing dataset.

Step-03 :
Initialize the dataframe

Step-04 :
Add the feature names to dataframe

Step-05 :
Adding target variable to dataframe.

Step-06 :
Perform Data Preprocessing (check for missing values)

Step-07 :
Split dependent variable & independent variable.

Step-08 :
Splitting data to training & testing dataset.

Step-09 :
Use linear regression (Train the m/c) to create model.

Step-10 :
Predict the y_pred for all values of train_x & ~~train~~ test_x.

Step-11:
Evaluate the performance of Model for train_y & test_y

Step-12:
Calculate Mean Square paper for train_y & test_y

Step13:
Plotting the linear regression model.

Conclusion:
In this way we have done data analysis using linear regression. for Boston Dataset & predict the price of houses using the feature of the Boston Dataset.

## Experiment No. 05

### Title of the Assignment :→

1. Implement logistic regression using python / R to perform classification on Social_Network_Ads.csv dataset.
2. Compute Confusion matrix to find TP, FP, TN, FN, Accuracy, Error rate, Precision, Recall on the given dataset.

### Objective of the Assignment :→

Students should be able to data analysis using logistic regression using python for any open source dataset.

### Prerequisite :→

1. Basic of python Programming
2. Concept of Regression.

### Theory :→

(1) Logistic Regression :→

- There are lots of classification problems that are available, but logistic regression is common & is a useful regression method for solving the binary classification problem.

- Logistic Regression can be used for various classification problem such as spam detection.
- Logistic Regression is one of the most simple & commonly used Machine Learning algorithms for two-class classification.
- Logistic regression is a statistical method for predicting binary classes.
- In a Linear Regression Eq$^n$:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_n x_n$$

where, $y \Rightarrow$ dependent variable
$x_1, x_2, \ldots x_n \Rightarrow$ explanatory variables.

Sigmoid fu$^n$ :→

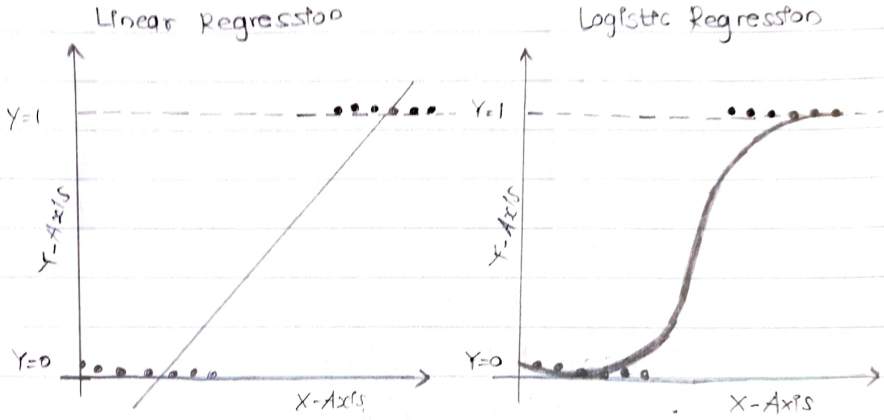
$$p = \frac{1}{1+e^{-y}}$$

Apply Sigmoid fu$^n$ on linear regression:

$$p = 1 / 1 + e^{-(\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_n x_n)}$$

(2) Differentiate bet$^n$ linear & logistic regression.

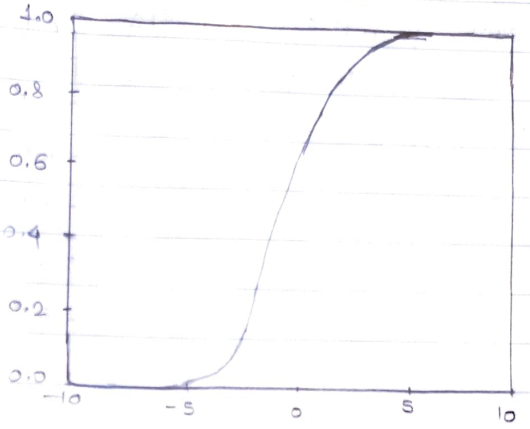
- Linear regression gives you a continuous o/p, but logistic regression provides a constant o/p.
- Linear regression is estimated using Ordinary Least Squares (OLS) while logistic regression is estimated using Maximum Likelihood Estimation (MLE) approach.




(3) Sigmoid fu$^n$ :→

- The sigmoid fu$^n$, also called logistic fu$^n$, gives an 'S' shaped curve that can take any real-valued number & map a value bet$^n$ of 1.
- If curve goes to +ve infinity, y predicted will become 1 & if the curve goes to -ve infinity, y predicted will become 0.

$$f(x) = \frac{1}{1+e^{-(x)}}$$



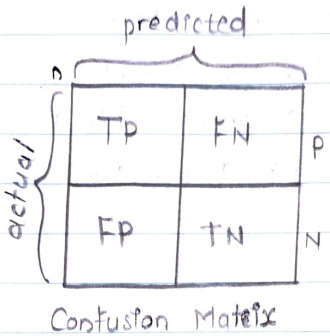

(4) Types of Logistic Regression :-

- Binary Logistic Regression
- Multinomial Logistic Regression
- Ordinal Logistic Regression

(5) Confusion Matrix Evaluation Metrics

- Contingency table or Confusion matrix is often used to measure the performance of classifiers.

The following table shows the confusion matrix for two class classifier.




Confusion Matrix

Some Important measures derived from confusion matrix are :

- Number of positive (Pos) :
  Total number instances which are labelled as +ve in a given dataset.
- Number of negative (Neg) :
  Total Number instances which are labelled as negative in a given dataset.
- No. of True +ve (TP) :
  No. of instances which are actually labelled as positive and the predicted class by classifier is also +ve.
- No. of True -ve (TN) :
  No. of instances which are actually labelled as -ve & the predicted class by classifier is also -ve.

- No. of False +ve (FP) :→
  No. of instances which are actually labelled as -ve & predicted class by classifier is +ve.
- No. of False -ve (FN) :→
  No. of instances which are actually labelled as +ve & predicted class by classifier is -ve.
- Accuracy :→

$$acc = \frac{TP+TN}{TP+FP+TN+FN} = \frac{TP+TN}{Pos+Neg}$$

- Error Rate :

$$err = \frac{FP+FN}{TP+FP+TN+FN} = \frac{FP+FN}{Pos+Neg} \quad \text{or}$$

$$err = 1 - acc$$

- Precision :

$$precision = \frac{TP}{TP+FP}$$

- Recall :

$$recall = \frac{TP}{TP+FN}$$

Algorithm (Boston Dataset):

Step-01: Import libraries & create alias for pandas, Numpy & Matplotlib.

Step-02: Import the Social_Media_Adv Dataset.

Step-03: Initialize the dataframe

Step-04: Perform Data Preprocessing.

- Convert Categorical to Numerical Values if applicable
- Check for Null values.
- Covariance Matrix to select the most promising features.
- Divide the dataset into Independent (X) Dependent (Y) variables.
- Split the dataset into training & testing datasets
- Scale the features if necessary.

Step-05: Use Logistic regression (Train the M/C) to create Model

Step-06: Predict the y_pred for all values of train_x & test_x.

Step-07: Evaluate the performance of Model for train_y & test_y.

Step-08: Calculate the required evaluation parameters.

Conclusion :→

In this way we have done data analysis using logistic regression for Social Media Adv. & evaluate the performance of model.

Experiment No. 06

Title of the Assignment :→

1. Implement Simple Naive Bayes classification algo. using python / R on Iris.csv dataset.
2. Compute Confusion matrix to find TP, FP, TN, FN, Accuracy, Error rate, Precision, Recall on the given dataset.

Objective :→

Students should be able to data analysis using Naive Bayes classification algo. using python for any open source dataset.

Prerequisite :→

1. Basic of python programming.
2. Concept of Join & Marginal Probability.

Theory :→

1) Concept used in Naive Bayes classifier.

- Naive Bayes classifier can be used for classification of categorical data.
  - Let there a 'j' number of classes, $c = \{1, 2, \dots j\}$
  - Let, i/p observation is specified by 'p' feature. Therefore i/p observation x is given, $x = \{F_1, F_2, \dots F_p\}$
  - Naive Bayes classifier depends on Bayes rules from probability theory.

- Prior probabilities: Probabilities which are calculated for some event based on no other information are called prior probabilities.
- For eg. P(A), P(B), P(C) are prior probabilities because while calculating P(A), occurrence of event B or C are not concerned i.e. no information about occurrence of any other event is used.

Conditional Probabilities:

$$P\left(\frac{A}{B}\right) = \frac{P(A \cap B)}{P(B)}, \text{ if } P(B) \neq 0 \quad \text{---(1)}$$

$$P\left(\frac{B}{A}\right) = \frac{P(B \cap A)}{P(A)} \quad \text{---(2)}$$

From eq$^n$ (1) & (2)

$$P(A \cap B) = P\left(\frac{A}{B}\right) \cdot P(B) = P\left(\frac{B}{A}\right) \cdot P(A)$$

$$\therefore P\left(\frac{A}{B}\right) = \frac{P\left(\frac{B}{A}\right) \cdot P(A)}{P(B)}$$

Is called Bayes Rule.

Conditional Probability :→

$$P(C_k|X) = \frac{P(X|C_k) * P(C_k)}{P(X)}$$

Now we have two classes & 4 features. so if we write this formula for class $C_1$, it will be,

$$P(C_1|x_1 \cap x_2 \cap x_3 \cap x_4) = \frac{P(x_1 \cap x_2 \cap x_3 \cap x_4|C_1) * P(C_1)}{P(x_1 \cap x_2 \cap x_3 \cap x_4)}$$

Here we replaced $C_k$ with $C_1$ & $x$ with the intersection $x_1, x_2, x_3, x_4$.

$$P(C_1|x_1 \cap x_2 \cap x_3 \cap x_4) = \frac{P(x_1|C_1) * P(x_2|C_1) * P(x_3|C_1) * P(x_4|C_1) * P(C_1)}{P(x_1) * P(x_2) * P(x_3) * P(x_4)}$$

Algorithm (Iris Dataset):

Step-01 : Import libraries & create alias for Pandas, Numpy & Matplotlib.

Step-02 : Import the Iris dataset by calling URL

Step-03 : Initialize the data frame

Step-04 : Perform Data preprocessing

Step-05 : Use Naive Bayes algorithm (Train the m/c) to create Model

Conditional Probability :→

$$P(C_k|X) = \frac{P(X|C_k) * P(C_k)}{P(X)}$$

Now we have two classes & 4 features, so if we write this formula for class $C_1$, it will be,

$$P(C_1|x_1 \cap x_2 \cap x_3 \cap x_4) = \frac{P(x_1 \cap x_2 \cap x_3 \cap x_4|C_1) * P(C_1)}{P(x_1 \cap x_2 \cap x_3 \cap x_4)}$$

Here we replaced $C_k$ with $C_1$ & $X$ with the intersection $x_1, x_2, x_3, x_4$.

$$P(C_1|x_1 \cap x_2 \cap x_3 \cap x_4) = \frac{P(x_1|C_1) * P(x_2|C_1) * P(x_3|C_1) * P(x_4|C_1) * P(C_1)}{P(x_1) * P(x_2) * P(x_3) * P(x_4)}$$

Algorithm (Iris Dataset):

Step-01 : Import libraries & create alias for Pandas, Numpy & Matplotlib.

Step-02 : Import the Iris dataset by calling URL

Step-03 : Initialize the data frame

Step-04 : Perform Data preprocessing

Step-05 : Use Naive Bayes algorithm (Train the m/c) to create Model

Step-06 : Predict the y-pred for all values of train_x & test_x.

Step-07 : Evaluate the performance of Model for train_y & test_y.

Step-08 : Calculate the required evaluation parameter

Conclusion :→

In this way we have done data analysis using Naive Bayes Algo. for Iris dataset & evaluate the performance of model.

Experiment No. 04

Title :→

1. Extract Sample document & apply following document preprocessing method :
   Tokenization, POS Tagging, stop words removal, stemming & Lemmatization.
2. Create representation of document by calculating Term Frequency & Inverse Document Frequency.

Objective :→

Students should be able to perform Text Analysis using TFIDF Algo.

Prerequisite :→

1. Basic of Python Programming.
2. Basic of English language.

Theory :→

(1) Basic concepts of Text Analysis :→

- Text mining is also referred to as text analytics.
- Text mining is a process of exploring sizable textual data & finding patterns.
- Text mining processes the text itself, while NLP processes with the underlying metadata.
- Finding frequency count of words, length of the sentence, presence / absence of specific words is known as text mining.

- Text mining is preprocessed data for text analysis.
- In text analytics, statistical & m/c learning algo. are used to classify information.

(2) Text Analysis Operations using natural language toolkit :

- NLTK is a leading platform for building python Progeams to coork with human langauge data.
- It provides easy-to-use interfaces & lexical resources such as WordNet, along with a suite of text processing libraries for classification, tokenization, stemming, tagging, parsing & semantic reasoning & many more.

(1) Tokenization :

Tokenization is the first step in text analysis The process of breaking down a text paragraph into smaller chunks such as words or sentences is called tokenization.

- Sentence tokenization : split a ~~gra~~ paragraph into list of sentences using sent_tokenize () method.
- Word tokenization : split a sentence into list of words using word_tokenize () method.

(2) Stop words removal :→

Stopwords considered as noise in the text. Text may contain stop words such as it is, am, are, this, a, an, the etc. In NLTK for removing stopwords, you need to create a list of stopwords & filter out your list of tokens from these words.

(3) Stemming & lemmatization :→

Stemming is a normalization technique where lists of tokenized words are converted into shortened root words to remove redundancy. Lemmatization in NLTK is the algo. process of finding the lemma of a word depending on its meaning & context.

(4) Pos Tagging :→

Pos (Parts of Speech) tell us about grammatical information of words of the sentence by assigning specific token (Determiner, noun, adjective, adverb, verb, personal Pronoun etc) as tag (DT, NN, JJ, RB, VB, PRP, etc) to each words.

② Stop words removal :→

Stopwords considered a [illegible] in text. Text may contain stop w[illegible]ch won't is, am, are, this, a, an, the. In NLTK for removing stopwords, you need to create a list of stopwords & filter out your list of tokens from these words.

③ Stemming & lemmatization :→

Stemming is a normalization technique where lists of tokenized words are converted into shortened root words to remove redundancy. Lemmatization in NLTK is the algo. process of finding the lemma of a word depending on its meaning & context.

④ Pos Tagging :→

Pos (Parts of Speech) tell us about grammatical information of words of the sentence by assigning specific token (Determiner, noun, adjective, adverb, verb, personal Pronoun etc) as tag (DT, NN, JJ, RB, VB, PRP, etc) to each words.

(8) Text Analysis Model using TF-IDF

Term Frequency - Inverse document Frequency (TFIDF), is a numerical statistic that is intended to reflect how important a word is to a document in a collection or corpus.

- Term Frequency (TF)

  It is a measure of the frequency of a word (w) in a document (d).

$$TF(w,d) = \frac{\text{occurences of } w \text{ in document } d}{\text{total no. of words in document } d}$$

- Inverse Document Frequency (IDF)

  It is the measure of the importance of a word.

$$IDF(w,D) = \ln\left(\frac{\text{Total No. of documents (N) in corpus } D}{\text{No. of documents containing } w}\right)$$

- Term Frequency - Inverse Document Frequency (TFIDF)

  It is the product of TF & IDF.

$$TFIDF(w,d,D) = TF(w,d) * IDF(w,D)$$

(4) Bag of Words (BoW)

- Machine learning algo. cannot work with raw text directly. Rather, the text must be converted into vectors of no.
- In natural language processing, a common technique for extracting features from text is to place all of the words that occure in the text in a bucket. This approach is called a bag of words model or BoW.

Algorithm for Tokenization, Pos Tagging, stop words removal, stemming & lemmatization:

Step-01: Download the required packeges.
Step-02: Initialize the text.
Step-03: Perform Tokenization
Step-04: ~~Perform Tokenization~~ Removing Punctuations & stop word.
Step-05: Perform stemming
Step-06: Perform Lemmatization
Step-07: Apply Pos Tagging to text.

Algorithm for Create representation of document by calculating TFIDF.

Step-01: Import necessary libraries.
Step-02: Initialize documents
Step-03: Create Bag of words (BoW) for document A & B.

Step-4: Create collection of unique words from Document A & B.

Step-5: Create a dictionary of words & their occurrence for each document in the corpus.

Step-06: Compute term frequency for each of our documents

Step-07: Compute the term Inverse Document frequency.

Step-08: Compute the term TF/IDF for all words.

Conclusion :→

- In this way we have done text data analysis using TF IDF algo.

Step-4: Create collection of unique words from Document A & B.

Step-5: Create a dictionary of words & their occurrence for each document in the corpus.

Step-06: Compute term frequency for each of our documents

Step-07: Compute the term Inverse Document frequency.

Step-08: Compute the term TF/IDF for all words.

Conclusion :→

- In this way we have done text data analysis using TF IDF algo.

Experiment No. 08

Title :→

Data Visualization - I

1. Use the inbuilt dataset 'titanic'. The dataset contains 891 rows & contains information about the passengers who ~~bord~~ boarded the unfortunate Titanic ship. Use the Seaborn library to see if we can find any patterns in the data.
2. Write a code to check how the price of the ticket (column name : 'fare') for each passengerts distributed by plotting a histogram.

Objective :→

Students should be able to perform the data visualization operation using python on any open source dataset.

Prerequisite :

1. Basic of python programming
2. Seaborn Library, Concept of Data Visualization

Experiment No. 09

Title :→

Data Visualization II

1. Use the inbuilt dataset 'titanic' as used in the above problem. Plot a box plot for distribution of age with respect to each gender along with the information about ~~wethe~~ whether they survived or not. (Column name : 'sex' & 'age')
2. Write observations on the inference from the above statistics.

Objective :→

Students should be able to perform the data visualization operation using python on any open source dataset.

Prerequisite :→

1. Basic of python programming
2. Seaborn Library, Concept of Data visualization.

Page No

Experiment No. 10

Title :→

Data Visualization III

Download the Iris flower dataset or any dataset into a DataFrame (e.g. http://archive.ics.uci.edu/ml/datasets/Iris). Scan the dataset & give the inference as:

1. List down the features & their types (eg. numerical nominal) available in the dataset.
2. Create a histogram for each feature in the dataset to illustrate the feature distributions.
3. Create a box plot for each feature in the dataset.
4. Compare distributions & identify outliers

Objective :→

Students should be able to perform the data visualization operation using python on any open source dataset

Prerequisite :→

1. Basic of python programming
2. Seaborn Library, Concept of Data Visualization
3. Types of variables.