ROHAN DAYAL
19C0060

Page:
Date:

ML EXPERIMENT 1

=> AIM:

Predict the price of the Uber ride from a given pickup point to the agreed drop-off location.

Perform following tasks:

a) Pre process the dataset
b) Identify Outliers
c) Check for correlation
d) Implement linear regression & random forest regression model.
e) Evaluate the model & compare their respective scores like R2, RMSE etc.

=> OUTCOME:

Apply preprocessing techniques on dataset.

=> SYSTEM REQUIREMENT:

- 64 bit Open Source Linux or its derivatives
- Python, Jupyter Notebook

=> THEORY:

- Linear Regression -

i) Linear Regression is a linear approach for modelling the relationship between a scalar response & one or more explanatory variable.

ii) It constructs a straight line in the data plane that maps independent variables to dependent variable.

iii) A general equation for linear regression is -

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots \beta_n x_n + \varepsilon.$$

Page:
Date: / /

where $y \rightarrow$ dependent variable

$\beta_1 \cdots \beta_n \rightarrow$ Coefficient of regression

$\beta_0 \rightarrow$ Intercept of regression

$\varepsilon \rightarrow$ Noise

$x_1 \cdots x_n \rightarrow$ Independent variable

iv) One of the major drawbacks of linear regression is that it assumes a linear relationship between independent & dependent variables.

- Random Forest Regression :-

=> Every decision tree has high variance, but when we combine all of them together in parallel then the resultant variance is low as each decision tree gets perfectly trained on that sample data, & hence the output ~~does~~ depends on multiple decision trees.

=> In case of regression problems, the final output is the mean of all the outputs.

- R2 Score :-

$$R_2 = 1 - \frac{\sum_{i=1}^{N} (y_i - \hat{y}_i)^2}{\sum_{i=1}^{N} (y_i - \bar{y}_i)^2}, \quad \bar{y} = \frac{1}{N} \sum_{i=1}^{N} y_i$$

- RMSE score :-

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

Page :
Date : / /

=> CONCLUSION :-

Hence, two regression models using linear regression & random forests has been implemented & evaluated.

ROHAN VIKAL
19C0060

Page:
Date: / /

ML EXPERIMENT 2

=> **AIM :-**

Classify the email using binary classification method. Email spam detection has two states - as Spam & Not Spam.

Use k-NN & SVM for classification. Analyze the performance.

-> **OUTCOME :-**

Apply & evaluate classification & clustering techniques.

=> **SYSTEM REQUIREMENT :-**

- 64 bit Open Source linux or its derivatives.
- Python, Jupyter Notebook.

=> **THEORY :-**

- **K-Nearest Neighbors :-**

>> K nearest neighbors is one of the most basic yet essential classification algorithm in Machine learning.

>>> It belongs to the supervised learning domain & finds intense application in pattern recognition, data mining & intrusion ~~detect~~ detection.

>>> It is widely disposable in real life scenarios since it is non-parametric meaning, it does not make any underlying assumptions about the distribution of data.

>> Let m be number of training data samples. Let p be an unknown point

a) Store the training samples in an array of data points a[]. This means each element of this array represents tuple (x, y).
b) Make a set S of k smallest euclidean distances. Each of these corresponds to an already classified data points.
c) Return the majority label among S.

- Support Vector Machines:

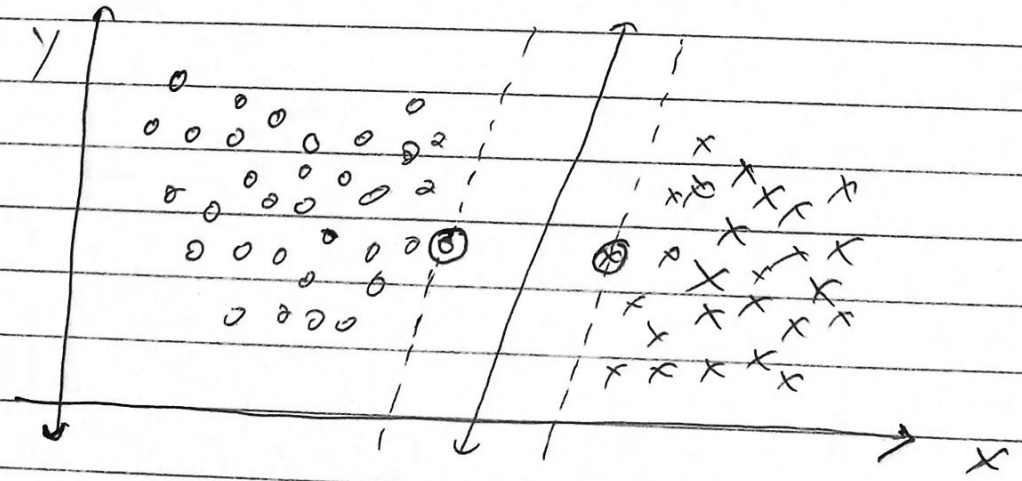
i) Support Vector machines is a relatively simple Supervised Machine learning Algorithm used for classification.

ii) If is more preffered for classification, it finds hyper-planes that create a boundary between the types of data.

iii) In 2D space, this hyper plane is a line. We plot each data item in an N dimensional plane.

iv) Next, we find optimal hyperplane to separate the data. So by this, ~~you must know~~ SVM can only perform binary classification.

v) However, there are various techniques to use for multi-class problems.



Page :
Date : / /

=> Conclusion :-

Hence a classification model using k-nearest neighbor & Support Vector Machine has been implemented & evaluated.

ROHAN DAYAL
1900060

ML EXPERIMENT 3

=> Aim :-

Given a bank customer, build a neural network based classifier that can determine they will leave or not in next 6 months.

Perform the following operations -

a) Read the dataset

b) Distinguish the feature & target set & divide the dataset into training & testing sets.

c) Normalize the train & test data

d) Initialize & build the model. Identify the points of improvement & implement the same.

e) Print the accuracy score & confusion matrix.

=> Outcome :-

Analyze & evaluate classification & clustering techniques.

Apply preprocessing techniques on datasets.

=> System Requirement :-

- 64 bit Open Source Linux or its derivatives
- Python, Jupyter Notebook.

=> Theory :-

- Neural Networks :-

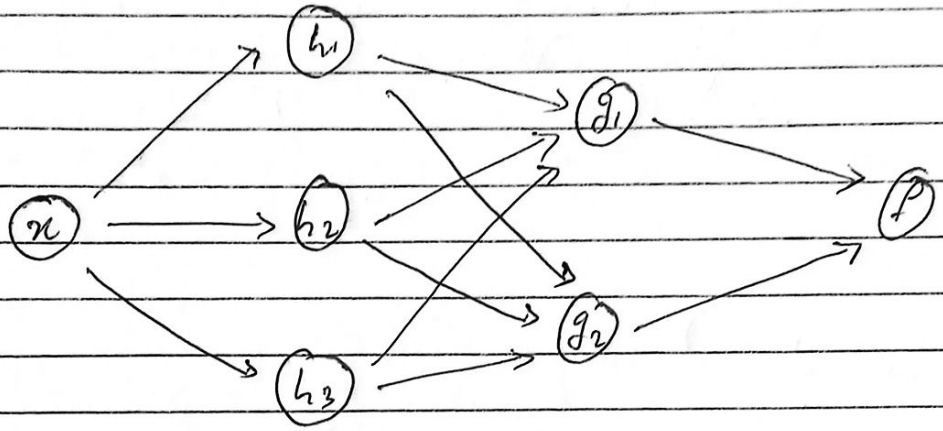
>> ~~Neural~~ Neural networks ~~are~~ consists of artificial networks of functions, called parameters which allows the computer to learn, & to ~~find~~ fine tune itself, by analyzing new data.

>> Each parameter, sometimes also refered to as neurons, is a function which produces an output after recieving one or

multiple inputs.

iii) Those outputs are then passed on to the next layer of neurons neurons, & so it continues until every layer of neuron have been considered, & the terminal neurons have received their input.

iv) Those ~~from~~ terminal neurons then output the final result for the model.



- Neural Network for Classification:-

i) A neural network can be used for classification modelling by using the correct activation functions, and correct loss function.

ii) For a model to be used as classifier, it should use sigmoid activation function for binary classification or softmax ——— " ——— for multiclass classification.

iii) The loss function & metrics for compiling a classification model must be "binary_crossentropy" & ["accuracy"] respectively.

Rahul Dawn

=> Conclusion :-

Hence a classification model using Neural Networks has been implemented & optimized.

ROHAN DAYAL
19C00060

ML EXPERIMENT 4

=> AIM :-
Implement k-nearest neighbors algorithm on diabetes.csv dataset. Compute Confusion Matrix, accuracy, error rate, precision, recall on the given dataset.

=> ~~Objet~~ OUTCOME :-
Analyze & evaluate classification & clustering techniques.

=> SYSTEM REQUIREMENT :-
- 64 bit Open Source Linux or its derivatives.
- Python, Jupyter Notebook

=> THEORY :-

- K Nearest Neighbors :-

i) K Nearest Neighbors is one of the most basic yet essential classification algorithm in Machine learning.

ii) It belongs to the supervised learning domain & finds intense application in pattern ~~regn~~ recognition, data mining & intrusion detection.

iii) It is widely disposable in real life scenarios since it is non-parametric meaning, it does not make any underlying assumptions about the distribution of data.

iv) Let m be number of training data samples. Let p be an unknown point.

a) Store the training samples in an array of data points a[]. This means each element of this array represents tuple (x, y).

b) Make a set S of k smallest euclidean distances. Each of these corresponds to an already classified data points.

c) Return the majority label among S.

- Confusion Matrix :

		Predicted: Negative	Predicted: Positive
Actual	Negative	True Negative (TN)	False Positive (FP)
	Positive	False Negative (FN)	(TP) True Positive

- ~~Error~~ Accuracy :

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

- Precision =

$$\text{Precision} = \frac{TP}{TP + FP}$$

- Recall -

$$\text{Recall} = \frac{TP}{TP + FN}$$

- Error Rate :-

$$\text{Error Type 1} = \frac{FN}{TN + FN + FP + TP} \qquad \text{Error Type 2} = \frac{FP}{\text{Total}}$$

=> Conclusion :-

Hence, a classification algorithm ~~beer~~ using a K-Nearest Neighbors algorithm has been implemented and evaluated.

ROHAN DAYAL
19C0060

ML EXPERIMENT 5

=> **AIM :-**

Implement k-means clustering / hierarchical clustering on sales-data-sample.csv dataset. Determine the number of clusters using the elbow method.

=> **OUTCOME :-**

Apply & evaluate classification & clustering techniques

=> **SYSTEM REQUIREMENT :-**

- 64bit Open Source Linux or its derivatives
- Python, Jupyter Notebook.

=> **THEORY :-**

- **K-means Clustering :-**

:) Clustering is one of the most common exploratory data analysis technique used to get an intuition about the structure of the data

:)) K means algorithm is an iterative algorithm that tries to partition the dataset into K pre-defined distinct non-overlapping subgroups (clusters) where each data point belongs to only one group.

:))) It tries to make the intra-cluster data points as similar as possible while also keeping the clusters as different (far) as possible.

i) The way kmeans algorithm works is as follows:
a) Specify number of clusters k.
b) Initialize centroids by first shuffling the dataset & then randomly selecting k data points for the centroids without replacement.
c) Keep iterating until there is no change to the centroids ie assignment of data points to clusters isn't changing.

- Elbow Plot :-

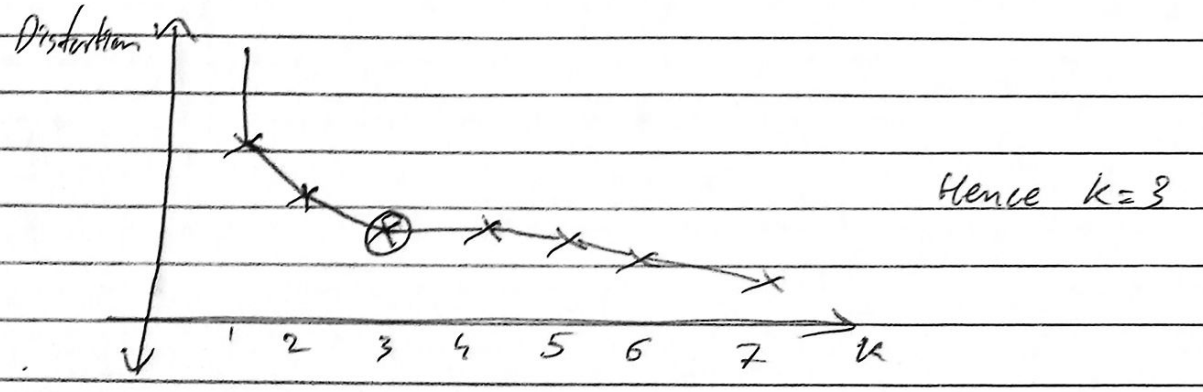
i) A fundamental step for any unsupervised algorithm is to determine the optimal number of clusters into which the data may be clustered.

ii) The elbow method is one of the most popular methods to determine this optimal value of k.

iii) An elbow plot is generated by ~~plotting~~ iterating the k-value from 1 to 10 and against distortion.

iv) Distortion is calculated as the average of the squared distances from the cluster centres of the respective clusters.

v) The k-value with ~~the~~ the minimal distortion & an ~~de~~ elbow like bent is chosen.



=> **CONCLUSION :-**

Hence a clustering algorithm model using a K-means clustering algorithm has been implemented.