INDEX

(6 zeros)

Name Harsh Kumar Sub.

Std. Div. Roll No.

Telephone No. E-mail ID.

Blood Group. Birth Day.

Sr.No.	Title	Page No.	Sign./Remarks
1	import and export data using pandas		
2	build decision tree (D3) and build classifier		
3	linear regression		
4	multilinear regression		
5	KNN		
6	logistic regression		
7	support vector machine model		
8	K-means algorithm		
9	dimensionality reduction		
10	artificial neural network		
11	random forest ensemble technique.		

5/8/24

PAGE NO :
DATE :

Write a python program to import and export data using Pandas library function

→

temp T1P

import pandas as pd
# Read the CSV file
airbnb_data = pd.read_csv("/content/sample_data/california_housing_test.csv")
# View the first 5 rows
airbnb_data.head()

longitude	latitude	median age	total rooms	total bedroom
-122.05	3737	27	3885	661
-118.30	34.26	43	1510	310
-117.81	33.78	27	3589	507
-118.36	33.82	28	67	15
-119.67	36.33	19	1241	244

PAGE NO:
DATE:

2) demonstrate various data preprocessing technique for given dataset?

import the dataset using read_csv()

identify and handle the missing value:

dataset.isnull().sum()

this gives the no of null values in each column

various to handle null values:-

1) use dropna() to drop columns having high no of null values.

2) use fillna() to replace a NULL value with specified value.

encoding categorical data using pd.get_dummies() which converts categorical data into dummy or indicator variables.

LAB - 2.

Use an appropriate data set for building the decision (D3) and apply this to classify a new sample

ID3 (Iterative dichotomiser 3) is a classic algorithm for building decision trees.

Input :- Training Dataset 'D' with 'm' examples and set of attributes A.

O/P :- A decision Tree T

procedure :-

→ If all the examples in D belong to the same class C, return a leaf node labelled with class C.

→ If A is empty, return a leaf node labelled with a majority class in D.

→ Otherwise choose the best attribute A best from A to split D.

→ Create a decision tree N using A best.

→ for each possible value V of Abest.

PAGE NO.
DATE

1) Partition D into subset DV such that Examples with value V for Aboot have an subset Dv

2) If DV is empty add a leaf node to N labelled with majority in D

3) Otherwise add a subtree to N by recursively calling the ID3 algorithm with DV and A-Aboot

4) Return the decision tree.

Output

1) Highest info gain = 0.246 = Outlook

2) Highest info gain = 0.971 = Sunny

3) Next attribute is Humidity

12/11/2024

Q) implement linear Regression using appropriate dataset

A) Algo

1) import necessary libraries
2) import dataset
3) visualization of dataset using different plots like heatmap distribution scatterplot etc
4) preprocess the data, convert categorical data
5) split the dataset into training set and test set, from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.3, random_state = 23)

6) build model
from sklearn.linear_model import LinearRegression

lin_reg = LinearRegression()

7) fit the dataset to the model and train it.
lin_reg.fit(X_train, y_train)

8) calculate the accuracy using mean square error.

PAGE NO.
DATE

Mean Squared Error = 0.9177833691

a) Implement multilinear regression

1) import necessary libraries like numpy, pandas, matplotlib

2) import dataset

3) visualise the dataset using matplotlib.pyplot.

4) Encode categorical data

ct = ColumnTransformer(transformers=[('encode', OneHotEncoder(), [3])], remainder='passthrough')

5) split dataset to training and test dataset

6) We can use multiple independent variables

7) Create regression regression model
regression = LinearRegression()

8) fit the train set

9) test the model using test set

10) Compare the actual & value and predicted value.

PAGE NO:
DATE

Implement KNN

1) import libraries necessary for KNN
```
import pandas as pd
import numpy as np
from sklearn.neighbours import KNeighbourClassifier
```

2) Read dataset.
```
df = pd.read_csv('iris.csv')
test = df[0:1, :]
train = df[1:, :]
```

3) Create model and predict
```
clf = KNeighbours(n_neighbour = 5, weights = 'uniform')
clf.fit(X_train, y_train)
df.predict(test[:, :-1])
```

PAGE NO
DATE :

Implement Logistic Regression

1) import necessary libraries
import pandas as pd
import numpy as np
from sklearn.linear_model import LogisticRegression

2) Load the dataset
iris = load_iris
X = iris.data
y = iris.target

3) Create model and train
model = LogisticRegression()
model.fit(df.drop('outcome'),
df.outcome);

4) predict value
Model.predict(df.drop('outcome'));

O/P

Enter no of sample = 3
no of feature = 5
enter feature for label 0
enter feature for 1 : 16.00 2.00 8.00
4.00
enter label 0 1.00

PAGE NO :
DATE :

Enter sample for label 2 ? 5.00 8.00 -4.00 12.00 4.00

Enter index : 1

Enter feature ? 3.00 9.00 6.00 4.00

enter label

Accuracy of label 1.0

3.17

PAGE NO :
DATE :

Build support vector machine model for given dataset.

Step :-

I) Import dataset to be trained (iris dataset)

II) Convert the dataset into dataframe (pd.DataFrame) add the target column as the dataframe

III) Print the first few of dataframes to inspect the data. Create a scatter plot to visualize the relationship b/w sepal length and sepal width

IV) Split the dataset into training & testing sets. use I D y. for training I D y. for testing

V) Create and train the SVM classifier. initiate an SVM classifier with a linear kernel. train the dataset using training data.

VI) predict the labels using trained classifier to predict labels for the test set.

PAGE NO :
DATE :

VII) Evaluate model - Calculate accuracy of model by comparision prediction values with Actual test label

O/P → Accuracy of SVM classifier is 1.0

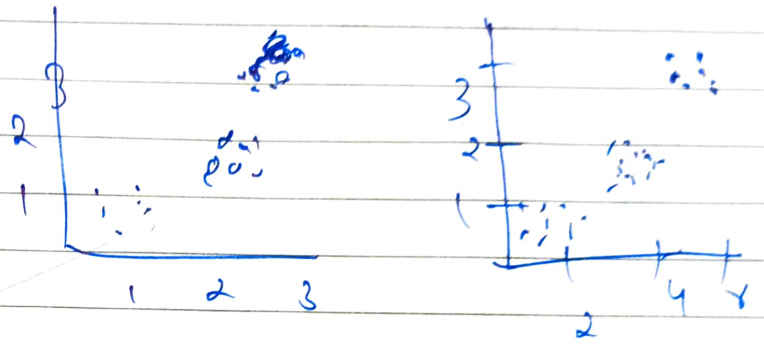
y predicted : array([1, 0, 2, 1, 0, 2, 0, 0, 0, 0, 1, 2, 1, 1, 2, 0, 2, 0, 2, 2, 2, 2, 2, 0, 0, 1, 0]);

Build K-means algorithm on iris dataset

Algorithm :-

1) Load the iris dataset
2) Initialize and fit K-means model
3) Visualize the clustering result
4) Plot original classification
5) Plot K-means clustering results

O/P

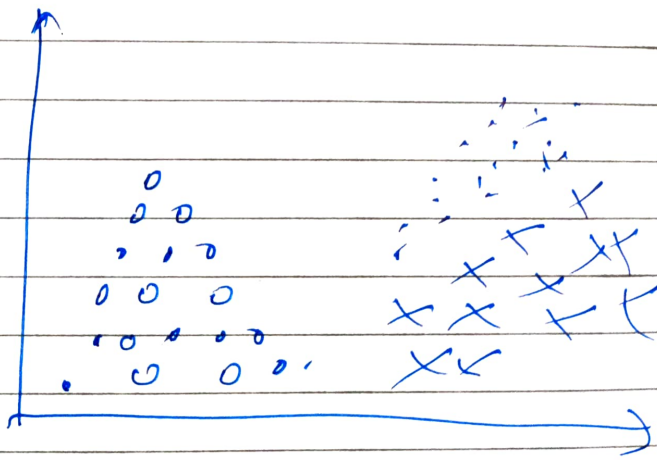


PAGE NO :
DATE :

Implement dimensionality reduction using principal component analysis method

Algorithm 2 -

1) import necessary libraries for data handling, standardization, PCA and plotting.
2) Load the iris dataset
3) Standardize the data
4) Apply PCA using the model
5) Convert PCA result to dataframe
6) Visualize the PCA result.



PAGE NO
DATE

Build an artificial neural network model with back propagation:

Algorithm :-

1) initialize parameters

- Normalize input features, matrix, 'O'
- Normalize the O/P, 'y'
- set hyperparameter, no of epochs, no of neurons.

2) define activation functions
- sigmoid function adjustments

3) Training the network
forward propagation
* compute i/p to hidden layer
* add bias
* apply activation function

4) ~~Bad~~ Backpropagation

* compute error
* compute gradient
* compute delta.

PAGE NO :
DATE :

5) update weights and biases

$$o/p = I/p \begin{bmatrix} [0.6667 & 1] \\ [0.333 & 0.556] \\ [0.1 & 0.667] \end{bmatrix}$$

Actual o/p $\begin{bmatrix} [0.92] & [0.86] & [0.79] \end{bmatrix}$

predicted o/p $\begin{bmatrix} [0.8605687] \\ [0.7898439] \\ [0.8011234] \end{bmatrix}$

PAGE NO :
DATE :

Implement Random Forest Ensemble method :

Algorithm :-

1) Import necessary libraries
2) Load and unspect data
3) Pre process the data as in Replacing features and missing
4) split the data Random forest classifier and train it using 'fit' method
5) split the data to training and test samples 0.4 to allocate 40% of data to testing and use rest for training
6) make predictions on test sample using method predict
7) evaluat the model

O/p Accuracy 0.8

confusion matrix

[[28 0 0
0 19 0
0 1 17]]

PAGE NO
DATE

implement boosting ensemble method

Algorithm :

1) import libraries
2) load the dataset
3) data preprocessing or cleaning separation features and dataset
4) split the dataset to train and test samples
5) initialize the adaboost classifier with specifying of estimators and base estimator
6) train the model using the training data
7) make predictions for the samples using trained model
8) evaluate the model

Result

Model accuracy score : 0.9833