INDEX

Name Abhinav Ishan Sub.

Std. Div. Roll No. 1BM21CS273

Telephone No. E-mail ID.

Blood Group. Birth Day.

Sr.No.	Title	Page No.	Sign./Remarks
1	Import & Export		
2	Preprocessing Technique		
3	ID3 tree		
4	Linear Regression		
5	Multiple Regression		
6	KNN		
7	Logistic Regression		
8	SVM model		
9	K-Means algorithm		
10	PCA		
11	CNN		
12	Random Forest		
13	Ada Boost		

PAGE NO :
DATE : 5/4/24

LAB-1

1) Write a python program to import and export data using Pandas library functions.

```
import pandas as pd
airbnb_data = pd.read_csv("/content/sample_data/california_housing_test.csv")
airbnb_data.head()
```

longitude latitude ~~median_age total_rooms total_bedr. pop. hou~~

	long.	lat.	median_age	total_rooms	total_bedr.	pop.	households	income	valu
0	-122.05	37.37	27	3885	661	1537	606	6.60	344700
1	-118.30	34.26	43	1510	310	809	277	3.59	176500
2	-117.81	33.78	27	3589	507	1484	495	5.79	270500
3	-118.36	33.82	28	67	15	49	11	6.135	330000
4	-119.67	36.33	19	1241	244	850	237	2.937	81700

PAGE NO
DATE

(2) Demonstrate various data preprocessing techniques for a given dataset.

Algorithm:

Import the dataset using read_csv()
Identify and handle the missing values:
dataset.isnull().sum()
- This gives the no. of null values in each column.

Solution to handle null values:
(1) Use Dropna is drop columns having high number of null values.

(2) Use Fillna to replace a NULL values with a specified value.

Encoding categorical data using pd.get_dummies which converts categorical data into dummy or indicator variables.

PAGE NO :
DATE : 12/04/24

LAB - 2

→ Use an appropriate data set for building the decision tree (ID3) and apply this knowledge to classify a new sample.

data set :

Outlook	Temperature	Humidity	Windy	Play
Sunny	Hot	High	Weak	No
Sunny	Hot	High	Strong	No
Overcast	Hot	High	Weak	Yes
Rain	Mild	High	Weak	Yes
Rain	Cool	Normal	Weak	Yes
Rain	Cool	Normal	Strong	No
Overcast	Cool	Normal	Strong	Yes
Sunny	Mild	High	Weak	No
Sunny	Cool	Normal	Weak	Yes
Rain	Mild	Normal	Weak	Yes
Sunny	Mild	Normal	Strong	Yes
Overcast	Mild	High	Strong	Yes
Overcast	Hot	Normal	Weak	Yes
Rain	Mild	High	Strong	No

To apply the ID3 algorithm:

1. Calculate the entropy of the target variable (Type).
2. For each attribute, calculate the entropy and information gain.

$$\text{Entropy}(S) = -P_{\oplus} \log_2 P_{\oplus} - P_{\ominus} \log_2 P_{\ominus}$$

(3) Select the attribute with the highest information gain as the root of the tree.

$$Gain(S, A) = Entropy(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} Entropy(S_v)$$

(4) Split the dataset based on the selected attribute and repeat the process recursively for each subset until all data is classified.

This dataset represents weather conditions (Outlook, Temperature, Humidity, Windy) and whether to play a game outside (Play).

We can use this dataset with the provided Decision Tree class to build a decision tree and classify new samples.

Output:

Predicted class: Yes

12/4/2024

PAGE NO :
DATE :

Implement Linear Regression & Multiple Regression.

-> Algorithm

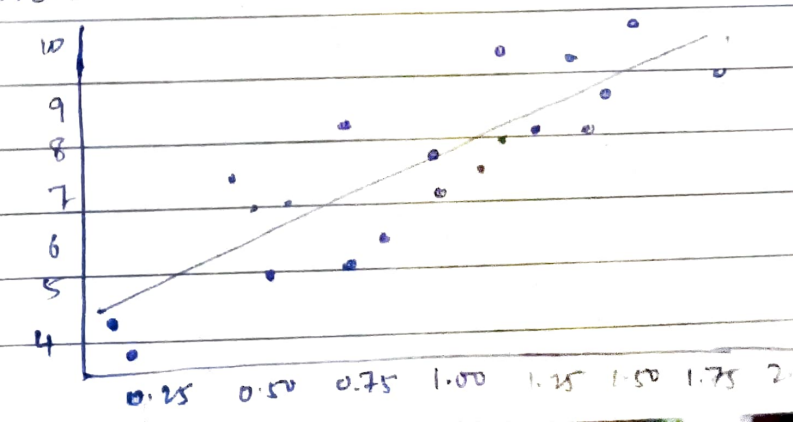
(1) Import necessary libraries
(2) Import dataset
(3) Visualization of dataset using different plots like heatmap, distribution plot, etc.
(4) Process the data, convert or encode categorical data
(5) Split the dataset into training set and
from sk learn, model_selection
import train_test_split

x_train, x_test, y_train, y_test =
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.



Mean squared error: 0.9177

PAGE NO
DATE

Implement Multiple Regression

1) Import necessary libraries like linear Regression
2) Import Dataset
3) Visualize the dataset using
4) Encode categorical data

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

5) Split dataset to training and test dataset
6) We can see multiple independent variables
7) Create regressor model

```
regressor = LinearRegression()
```

8) Fit the train set.
9) Test the model using test set
10) Compare the actual value and predicted value.

Output

Multiple

coefficients: [37.90 -241.96 542.42
347.70 -931.48 518.06
163.41 275.31 736.19
48.67]

Intercept : 151.3456

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

```
df = KNeighbourClassifier(n_neighbours = 5,
                          weights = 'uniform')
df.fit(x_train, y_train)
df.predict(test[:, :-1])
```

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.
```

4. Predict value

```
model.predict(df.drop(outcome))
```

output

Enter the no. of samples in dataset : 3
Enter the no. of features in your dataset : 5
Enter the features for sample 1 : 10.00 2.00
6.00 4.00 1.00
Enter the features for sample 2 : 5.00
8.00 4.00 12.00
4.00
Enter the features for sample 3 : 3.00 9.00
6.00 4.00
Enter the label : 0

Build Support vector Machine model

steps :-

1. Import the dataset to be trained
2. Convert the dataset into dataframe Add the target column to the dataframe.
3. Print the first few rows of dataframe to import the data. Create a scatter plot to visualize the relationship b/w sepal length and sepal width.
4. split the dataset into training and testing sets. Use 80% for training 10% for testing.
5. Create and train the SVM classification initialize an SVM classifier with a linear kernel. Train the dataset using training data.
6. Predict test labels : Use trained classifier to predict labels for the test set.
7. Evaluate model Calculate accuracy of model by comparing prediction labels with actual test labels.

output :- Accuracy of SVM classifier is 1.0.

y_pred : array([1, 0, 2, 1, 1, 0, 1, 2, 1, 1, 2, 0, 0, 0, 1, 2, 1, 1, 2, 0, 2, 0, 2, 2, 2, 2, 0, 0, 1, 0])

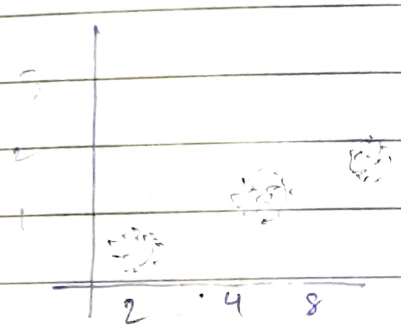
PAGE NO
DATE

Build k-Means algorithm

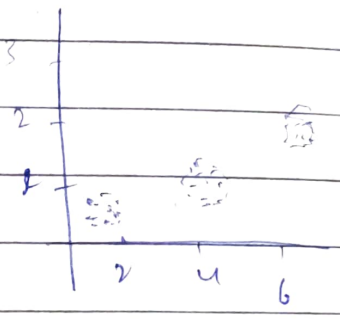
1. load the iris dataset
2. Initialize and fit k-means model
3. visualize the ~~resulting~~ results.
4. Plot original classification
5. Plot k-means clustering results

output

Real clusters



k-means clusters



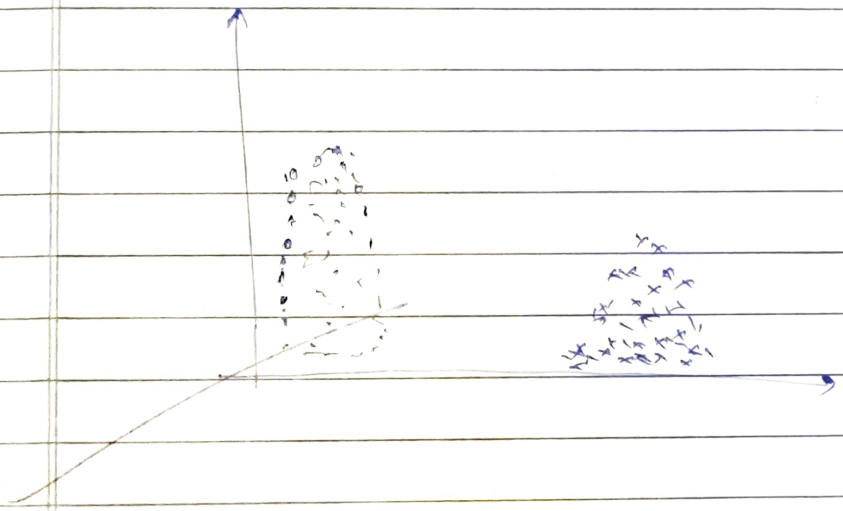
PAGE NO :
DATE :

Implement dimensionality reduction using principle component analysis method.

Algorithm:

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.

output:



Build an artificial neural network model with back propagation.

(1) Initialize parameters
- Normalize input features.
- Normalize the output
- Set hyperparameters

(2) Define activation functions.

(3) Training the network.
Forward propagation
* compute i/p to hidden layer
* Add bias.
* Apply activation function

(4) * Compute error
* Compute gradient
* Compute data

(5) update weights and biases

output :- I/p [[0.6667 1]
[0.373 0.556]
[0.1 0.667]]

Actual output: [[0.92] [0.86] [0.89]]

Predicted output : [[0.30056875]
[0.79393839]
[0.80112347]]

DATE :

Implement Random Forest ensemble method

1) Import necessary libraries
2) Load and inspect data
3) Pre process the data as in separating features and strength.
4) Split the data to training and test samples.
5) Initialize random predict classifier and train it using 'fit' method.
6) Make predictions on test sample using method predict.
7) Evaluate the model.

Output :- Accuracy = 0.98

Confusion Matrix :- [[23 0 0
0 19 0
0 1 17]]

PAGE NO :
DATE :

Implement Boosting method

1. Import libraries
2. Load the dataset
3. Data preprocessing involves separation of features and dataset.
4. Split the dataset to train and test samples.
5. Initialize the dataset classifier with specified no. of estimates and base estimators.
6. Train the model using the training dataset.
7. Make predictions for test sample using trained model.
8. Evaluate the model.

output

Metrics accuracy score: 0.9833