INDEX

Name Goutham. S. Pujar

Sub. ML-LAB Batch 4

Std.: VI th sem

Div. " B "

Roll No. 1BM21CS277

Telephone No.

E-mail ID.

Blood Group.

Birth Day.

Sr.No.	Title	Page No.	Sign./Remarks
1.)	Explaning Pandas		
2.)	Data Pre-processing		
3.)	ID3 Decision tree		
4.)	Linear Regression		
5.)	Multi-Linear regression		
6.)	KNN classification		
7.)	Support vector Machine		30/5/2024
8.)	Artificial Neural Network		
9.)	Random forest Ensemble		
10.)	Boosting Ensemble		
11.)	K-Means clustering		
12.)	Dimentionality reduction using PCA		

CS

5/7/2024

SURYA Gold

Date______ Page______

LAB-1

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

```
import pandas as pd
data = pd.read_csv("/content/austinHousingData(1).csv")
data.head()

data['propertyTaxRate'] = 0
data.head(10)

data.to_csv("/content/austinHousingData(1).csv")
data.head(10)

#Reading from URL:
URL = "https://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.data"

col_names = ["sepal_length_in_cm", "sepal_width_in_cm",
 "petal_length_in_cm", "petal_width_in_cm", "class"]

iris_data = pd.read_csv(URL, names = col_names)
iris_data.head()

#exporting
iris_data = pd.read_csv("/content/cleaned_iris_data.csv")
iris_data.head(10)

irisdata.to_csv("cleaned_iris_data.csv")
```

5/4/2024

SURYA Gold

Date Page

Lab-2

2.> Demonstrate various data pre-processing techniques.

→ Algorithm:-

1.> Downloading the ~~ab~~ data set.
2.> Reading the csv file
df = pd.read.csv(datapath)

3.> Observing dataset using head(), info() and describe().
4.> Visualization of dataset using matplot lib.
5.> Creating test set and train set by splitting the data set
6.> Visualizing the data to gain the insights
7.> Finding the correlation between the categories
8.> Data cleaning by dropping Nan-values
housing.drop na()
9.> Imputation of missing values
10.> inputed = simple-impute (strategy = median)
11.> Encode categorical values with numbers.
12.> Scaling data using standardization or min-max strategy
13.> Training the linear regression model
14.> Calculating the root mean square error
15.> Training decision Tree
16.> Cross Validation using mean and standard
17.> Fitting test abstract and calculating the accuracy.

12/4/2024

12/4/2024

SURYA Gold

Date______ Page______

Lab-3

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

→ <u>ID3-Algorithm:-</u>

ID3 (Examples, Target Attribute, Attributes)

- Create a Root node for the tree
- If all Examples are positive, Return the single node tree Root, with label = +
- If All the Examples are negative, Return the single node tree Root with label = -
- If Attributes is empty, Return the single-node tree Root, with label = most common value of Target Attribute in Examples.

- Otherwise Begin
  - A ← the attribute from Attributes that best classifies Examples.
  - The decision attribute for Root ← A
  - For each possible value, vi of A
    - Add a new tree branch Root, corresponding to the test A = vi
    - Let Examples~~vi~~ be the subset of Examples that ~~value~~ vi for A
    - If Examples vi is empty
      - Then below this new branch add a leaf node with label = most common value of Target Attribute in Examples.
      - Else below this new branch add the subtree ID3 (Examples vi, Target Attributes - {A})

END

- Return Root.

8

12/4/2024

3/5/2024

SURYA Gold

Date______ Page______

Lab-4

Import linear regression algorithm using appropri[ate] dataset.

i.) Import necessary libraries.

ii.) Import data set

iii.) Visualization of dataset using different plots l[ike] heatmap, distributing plot etc.

iv.) Preprocess the data, convert or encode categori[cal] data.

v.) Split the dataset into training and testing se[t] from sklearn.model_selection import train_split.

vi.) Build model into training and testing set from sklearn.linear_model import LinearR[egression] lin_reg = LinearRegression()

vii.) Fit dataset to model by train it lin_reg.fit(x_t[rain], y_t[rain])

viii.) Calculate the accuracy using mean square er[ror].

3/5/2024

SURYA Gold

Date______ Page______

- Implement Multilinear Regression:-

i.> Encode Categorical data

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

ii.> Split dataset into training and testing set.

iii.> We can see multiple independent variables.

iv.> Create regression model.

regression = LinearRegression().

v.> Fix train set.

vi.> Test model using test set.

vii.> Compare actual value & predicted value.

3/5/2024

SURYA Gold

4a) Lab - 5 Date Page

=> Build Logistic Regression Model for given data

1.) Import all required ~~libstated~~ libraries

2.) Import given dataset

3.) Preprocess dataset to standard scale.

4.) Split dataset into test and train

5.) Code Logistic Regression Model.

lr = LogisticRegression(c=0.01, solver='liblinear')
.fit(x_train, y_train)

6.) Predict test set using model.

yhat = lr.predict(x_test)
yhat_probability = lr.predict_proba(x_test)

7.) Calculate the performance of accuracy of the model.

OK

Probability of insurance for age is 30.02%

The customer will get insurance

SURYA Gold

Lab-6

Date____ Page____

Build KNN classification Based for given dataset:

Get your Data :-

Collect your data set with features (like height, weight) and labels (like "tall" or "short").

Prepare your Data:-

Clean it up and split it into two sets, one for training and one for testing

Choose K :-

Decide how many neighbours you want to consider This is k.

Find close Neighbours:-

For each point in the testing set, find the K nearest points in the training set.

Vote for the class:-

Let these neighbours vote on the label for the test point.

Pick the winning class:-

The most common label among the neighbours is the predicted label for the test point.

Check ~~the~~ Your Model :-

Finally, see how well your model did by comparing its predictions to the actual labels in the test set.

o/p: Predication = [0 1 0 2]

SURYA Gold

Date______ Page______

Lab-7

=> Build Support vector machine model for ge[...] dataset

1.> Collect your data:- Start with a data set containing features and labels.

2.> Clean and Split your data:- Tidy up your and split it into two parts: one for train[...] one for testi[...]

3.> Pick a Kernel:- Select a method to transform data, such as • linear • radial basis • polynomial

4.> Model Training:

Use training data to teach the SVM to se[...] different classes by finding the best hyperpl[...]

5.> Parameter Tuning:

Adjust parameters like regularizations (C[...] Kernal settings to enhance model ~~pe~~ perfo[...]

6.> Model Evaluation:-

Assess your model's performan[...] on testing data, measuring metrics like a[...] precision, recall and f1-score. fine-tune if needed for better results.

O/p:-

Accuracy of Svm classifier => 1.0

SURYA Gold

Date____ Page____

Lab-8

=> Build Artificial Neural Network model with backpropogation

Algorithm:

1> Initialize parameters
- Normalize i/p feature matrix 'x';
- Normalize the output y;
- Set hyper parameters: no of epochs, lr, not of neurons in i/p layer, hidden layer and o/p layer.
- Optimize the weights and biases for hidden layer and output layer with random values

2-> Define Activation functions:-
- Signonial for $\alpha(x) = 1/1+e^{x}$
- Derivatives of Sigmoids = $\alpha'(x)$ is $\alpha(x).(1-\alpha(x))$

-> Training Network:-

-> Data preparatation:- Gather and clean data for training

-> Model Design:- Decide on the structure of the neural network.

-> Training: Teach the network to make accurate predications by adjusting its parameters based on data.

SURYA Gold

Date______ Page______

=> Back propagation:-

1> Forward Pass:- Input data, get output

2> Backward pass:- Compare output with targ adjust.

3) Update weights:- Make small changes to i accuracy. Repeat until satisfied.

i/p:-

- Size of house (sq. ft)
- Price of house ($)

1000 sq. ft -> $
1500 sq. ft -> $
2000 sq. ft ->

o/p:

Predicted price of house based on siz

3.1

SURYA Gold

Date_____ Page_____

Lab-9

Implement Random forest Ansenble Method

Algorithm:-

Random Sampling:- Randomly select a subset of data from the training set.

Decision Trees:- Construct multiple decision trees using the randomly selected subset of data.

Feature Randomness:- At each node of the tree, randomly select a subset of features of to consider for splitting.

Voting:- Each decision tree "votes" on the prediction for a new data point.

Aggregation:- Combine the predications from all the trees to make a final prediction.

~~For~~ Ensemble:- Random forest reduces overfitting and boosts accuracy by aggregating predictions for multiple decision trees.

o/p:-

Accuracy = 0.98

Confusion Matrix:-

[[23 0 0
0 19 0
0 1 17]]

SURYA Gold

Date ______ Page ______

31/5/2024

24

Implement boosting Ensemble method

1.> Train Weak Learner: Train a simple model on the data.

2.> Focus on Mistakes:- Give more attention to wrongly predicated examples.

3.> Repeat and Improve:- Train more models, each correcting the mistakes of the previous ones.

4.> Combine Predictions:- Combine the predictions of all models to make a final, stronger prediction.

Results:-

Metrics accuracy score => 0.9833.

5/6

24/5/24

Lab-10

Build K-means Algorithms clusters a set of data

-> Algorithm:-

1.) Choose Initial Centers:- Randomly select K points as initial cluster centers.

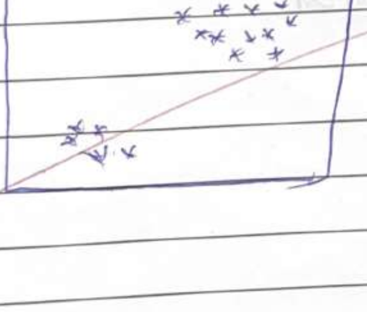
2.) Assign Data points:- Assign each data point to the nearest cluster center.

3.) Update centers:- Recalculate the cluster centers as the mean of all data points assigned to each cluster.

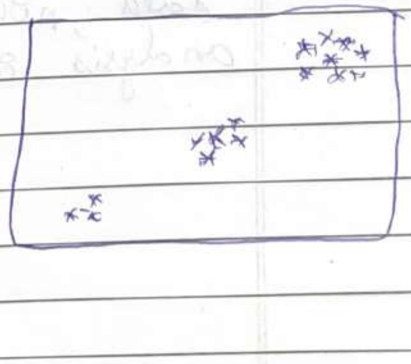
4.) Repeat: Repeat steps 2-3 until cluster centers stop changing or a maximum no of iteration is reached.

O/P:-

Real clusters:-



K-Means clustering



24/5/2024

Lab-11

⇒ Implement Dimensionality Reduction using Principle Component Analysis

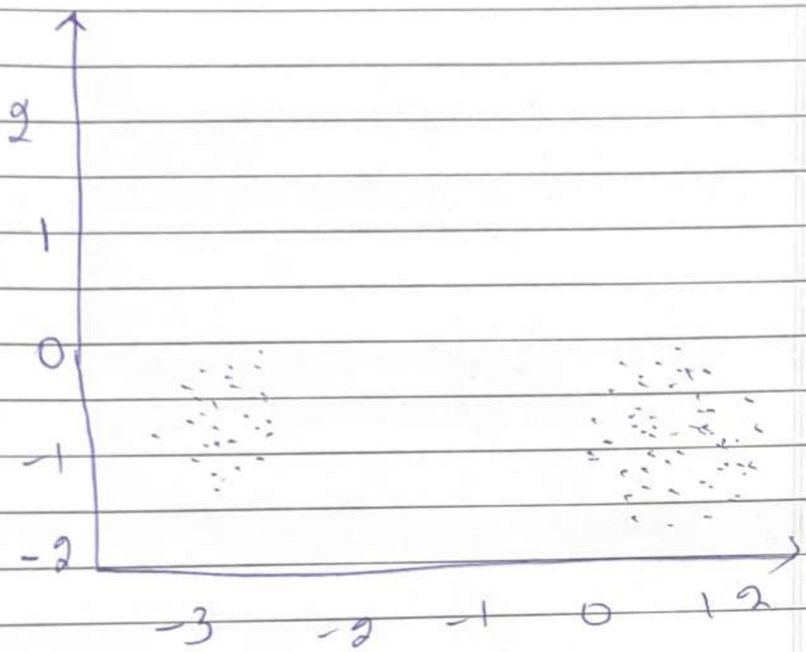
1.> Compute Covariance:-
Find the Covariance matrix of the i/p data.

2.> Eigen Decomposition:-
Break down the covariance matrix into eigenvectors and eigen values.

3.> Select Components:-
Choose the top eigenvectors based on their eigen values.

4.> Transform data:- Project the original data onto these selected eigenvectors.

5.> Reduce Dimensionality:- Use the transformed data now with fewer dimensions, for analysis or visulization.


2
1
0
-1
-2
-3
-2
-1
0
1 2