5/4/2024

Lab 1

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

Importing

```
import pandas as pd
data= pd.read_csv('/content/austinHousing.csv')
data.head()
```

Output

```
   zpid     city          streetAddress   zipcode   description
0  1113734  pflugerville  Victor Dr       78660     Lake Victor Dr
1  1209004  pflugerville  Strickling Dr   78660     Absolutely Gorg
2  208449   pflugerville  Dessau Rd       78660     Under Construction
3  1209013  pflugerville  Strickling Dr   78660     One Story home
4  601348   pflugerville  Jane Loop       78660     Brimming with
                                                    appeal
 5x47 columns
```

Exporting

```
url = "archive.ics.uci.edu/ml/machine-learning-database"
col_names = ["sepal-length", "sepal-width", "petal length",
             "petal-width", "class"]

iris_data = pd.read-csv(url, names=col-names)
iris_data.head()
```

```
    sepal_length_in_cm   sepal_width   petal_length
0   5.1                  3.5           1.4
1   4.9                  3             1.4           ....
2   4.7                  3.2           1.3
3   4.6                  3.1           1.5
4   5.0                  3.6           1.4
```

```
iris_data.to_csv("exported.csv")
```

Lab 2

5/4/2024

Demonstrate various data preprocessing techniques for a given dataset

Algorithm

1) Import dataset using pandas
2) Perform dataset.shape() to analyse shape of dataset
3) use is_null() function from pandas to analyse missing values.
4) drop or ~~ff~~ fill missing values according to your usecase. Example drop_na() and ffill().
5) use libraries like Scikit learn etc to perform more preprocessing if required

Lab3

12/4/2024

use an appropriate dataset for building ID3 and apply this knowledge to classify a new sample

Algorithm

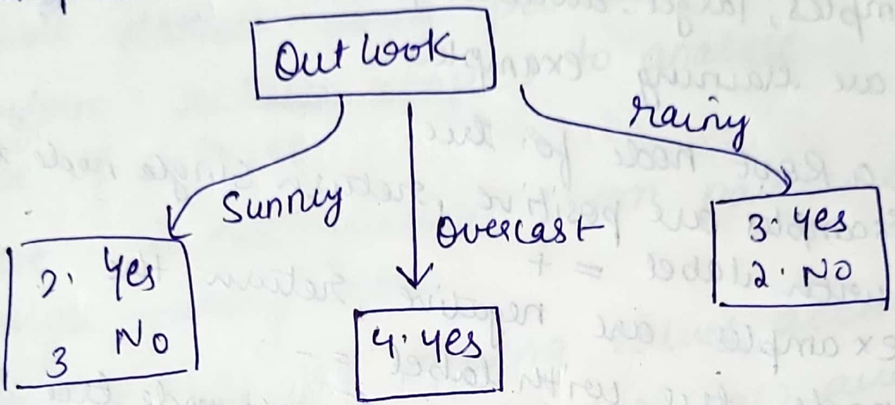
```
ID3( Examples, Target_attribute, Attributes)
Examples are training example.
• Create a Root node for tree
• If all Examples are positive, return single node tree
   Root with label = +
• If all examples are negative, return the
  Single node tree with label = -
• If attributes empty, return single node tree root
with label = most common value of Target_attribute
in Examples.
Otherwise Begin
    • A ← 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 below Root,
          corresponding to test A = vi
        → Let Examples vi, be subset of Examples
          that have 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
        ~~→ If E~~
                • Else below this new branch add
                  the subtree ID3.
  End
  Return Root
```

Output

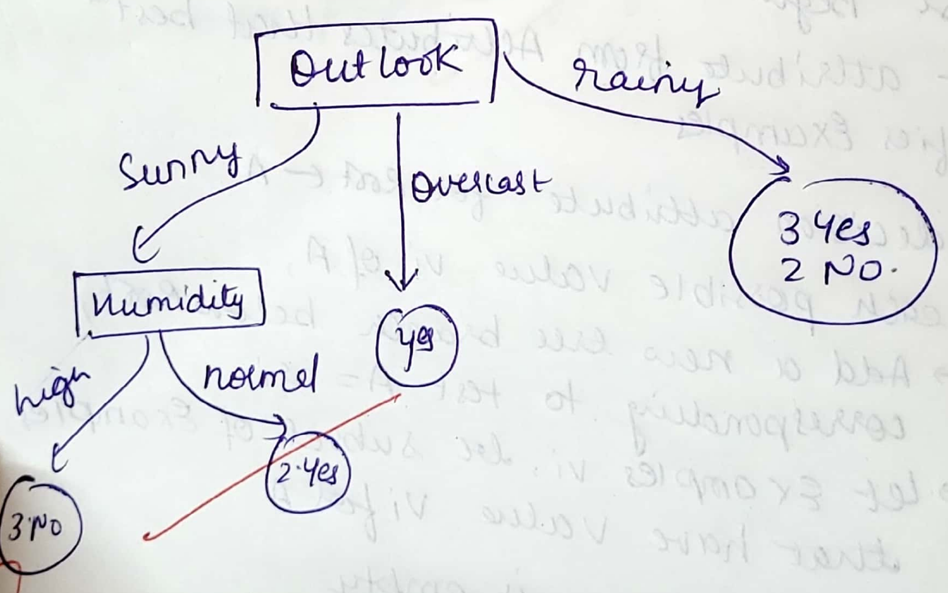
wind has highest information gain of 0.971

Humididay - 0.02

Temperatue - 0.02



Making a decision tree node using feature with maxm Information Gain (wind)



12/11/2024

Lab 4 19/04/2024

Implementation of Linear and Multi Linear Regression Algorithm

Step1 : Data Preprocessing using Pandas, numpy.

Step2 : Extract independent & dependent variable.

Step3: Split dataset into training and test set usually 70-30%..

Step 4 : Fitting the model into Training set using Scikit learn and create object of class regressor.

Step5 Predict test result

Stepc use performance metrics like MSE, MAE.

Multiple Linear Regression

Step1 : Data Preprocessing

Step2 : Importing the dataset

Step3 : Extracting dependent and independent Variables.

Step5 : Encoding dumming variables.

Step6 : fitting model

Step7 : performing prediction and using Metrics like MAE, MSE

Output

Intercep = 34.216

19/4/24

Lab 5 KNN Classification

KNN algorithm

Step1: Select optimal value of K

Step2: Calculate Euclidean distance

Step 3: find nearest neighbore

Step4 : vote for classification (majority voting) distance $(x, x_i) = \sqrt{\sum_{j=1}^{d}(x_j - x_{ij})^2}$

Output

Training model accuracy : 0.9714

3/5/2024

3/5/2024 Lab 6

Implementation Of Logistic Regression

Algorithm

① Initialization of Parameters

- initialize weights 'w' and bias 'b' to zeroes
- parameters are updated starting with neutral point

② Forward Propagation

Apply sigmoid activation function to get predicted output probability

③ Compute cost

using binary cross entropy loss function

④ Backward Propagation and Parameter update

⑤ Repeat 2-5

⑥ Prediction

Output

Training accuracy of 80% in first iteration

3/5/2024

Lab7

24/05/2024

PCA

1. Calculate mean
2. Calculate covariance matrix
3. Eigenvalues of covariance matrix
4. Computation of eigenvectors - unit
5. first principal components
6. Geomatrical meaning of first principal

Output :

```
pca.explained_variance_ratio
  array([0.98377428, 0.01620498])
```

Lab 8

24/05/2024

SVM

1. define Kernel function
   Eg. $K(x_1 x_2) = x_1 \cdot x_2$
2. Solve quadratic programming
3. compute weight and bias
4. identify support vectors
5. Make predictions

Output

```
Model = svm()
  model.fit(X_train, y_train)
  predictions = model.predict(X_test)
    accuracy =(y_test, predictions)
```

$\boxed{0.9823088}$

arrayles .

24/05/2024

Lab 9

K Means custering algorithm

① Select number K to decide the numbler of clusters

② Select random k points or centroids

③ Assign each point to closest centrod for predefind cluster

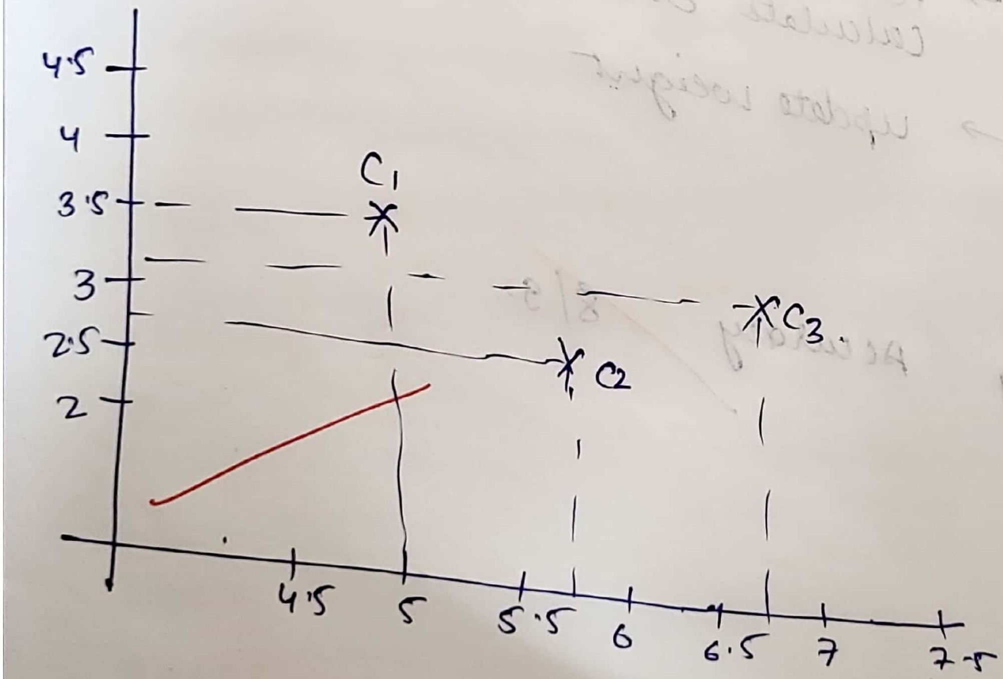
④ Calculate the variance and place a new centroid of each cluster.

⑤ Repeat 3., reassign the centroid

⑥ If any reassignment occurs, goto step 4, else go to finish.

⑦ The model is ready.

Output



31-05-2024

Lab 10

Build ANN with backpropagation

```
→ Create feed forward network with $n_i$ inputs,
  $n_{hidden}$ units, $n_{out}$ outputs.
→ Initialize all network weights to small
  random numbers.
→ Until the termination condition is met,
    Do
      → For each $(\vec{x}, \vec{t})$ in training examples
          → Propagate input forward
          → Propagate errors backward
          → For each hidden unit h,
            calculate error
          → update weight
```

Output

Testing Accuracy 8/9.

31/05/2024

Random Forest:

① Import libraries
② Load and prepare dataset
③ train the data before that do train test split.
④ Initialize Random forest regressor.
⑤ Train model.
⑥ Make predictions
⑦ Evaluate using MSE (Mean Square Error)

Output

accuracy = 0.93

31/05/2024

Adaboost Algorithm

① Import libraries

② load & prepare data

③ Initialize AdaBoost model – (learning-rate, n-estimators

④ make the model train

⑤ make predictions

⑥ Evaluate model on metrics like Mean Absolute Error

Output

accuracy : 0.9467

31/5/2024