ASSIGNMENT DATA MINING
[MTL782]

Que1] Choose a data set from UCI Machine Learning Repository (or any other Source) for Multi class classification problems.

1. Your first task is characterize the data set. Answer the following questions about the data:

   a. What the data is about ?
   Ans] Consider a Car Dealer who buys Second-Hand-Cars from an auction and sells it to other consumers. Cars from such an auction are often in a very poor condition and hence the dealer has to kick it out. Ofcourse once the car is bought by the dealer the money lost cannot be recovered. But even after buying the car the dealer transports the car to his/her workshop tries to do some repairing over it and then finds out that the car is beyond recovery and cannot be sold. This class of cars are called **Kicks**. The cost incurred(after buying) is also a lot. All of this money goes for nothing. So we want to classify cars as **Kicks** or **Not-Kicks**. Hence we try to solve this problem statement by using the previous data of the cars, bought from different auctions/sellers.

   b. What type of benefit you might hope to get from data mining?
   Ans] Our aim is to make a model that can efficiently discover valid, previously unknown, potentially useful and understandable patterns in this large dataset. Since the data has **67,211 instances** and **30 attributes**, it is a large dataset. After the dealer has bought his car, our model comes into work. We can predict if the car is a **Kick** or **Not-Kick**. This can help save the dealer a lot of money.

   c. Discuss data quality issues: For each attribute,

      i. Are there problems with the data?
      Ans] The following are the problems with data :

      - **Missing Values** : The dataset contained missing values for certain data features namely,*MMRAcquisitionAuctionCleanPrice*, *MMRAcquisitionAuctionAveragePrice*, etc.
      - **Irrelevant Features** : The dataset contained irrelevant data features namely, *Colour of Car*.
      - **Correlated Features** : The dataset contained correlated/closely-related data features namely *Zip Code* and *the name of state(US)*.

      ii. What might be an appropriate response to the quality issues.
      Ans] The following are the problems with data :

      - **Data Reduction** : We eliminated irrelevant data features namely, *Colour of Car*, *Purchase Date* and *Zip Code*. The colour of the car won't affect the life or durability of the car, hence removed. The purchase date is not required, since we are considering the age of our vehicle. Again the Zip Code is not mandatory since we are considering the state where the vehicle was bought.
      - **Data Cleaning** : We had a few features for which the data records were empty. The missing data was only about *~1%* of the original data. Hence we removed those data instances.
      - **Data normalisation** : Since the data had features that were different by a few orders (example- *odometer reading* and *age of vehicle*) we had to normalise the data.
      - **Data Transformation** : Several features were of string dataype. Hence we mapped those strings to a Integers. Also, since nominal attributes have no natural ordering (but the numbers we have converted them into have) so we used the label encoder to implement it.

1. Implement (1) Decision Tree, (2) Random Forest, (3) Naïve Bayes Classifier (4) KNN classifier, (5) SVM, and (6) ANN and compare the performances using k-fold cross validation and other tuning techniques (grid search and parameter search, where ever applicable)
   Ans] Our Models and their performances are summarised as follows (We implemented the all classifier using ~14% of the data as our test data and the other part as our train data (except for SVMs).) :

   a. Decision Tree :
      - Hyper-Parameters Considered = max_features, min_samples_split, max_depth, criterion
      - Hyper-Parameter Tuning = Grid-Search
      - Best Hyper-Parameter Value Obtained
        - criterion = gini
        - max_features = log(2)
        - min_samples_split = 70
        - max_depth = None (meaning fixing max_depth decreases efficiency of model)
      - Corresponding Accuracy
        - Train Accuracy - 91.32%
        - Test Accuracy - 88.74%
        - F1 Score - 0.4963

   b. Random Forest :
      - Hyper-Parameters Considered = n_estimators, criterion, max_depth
      - Hyper-Parameter Tuning = Grid-Search
      - Best Hyper-Parameter Value Obtained =
        - n_estimators = 100
        - criterion = entropy
        - max_depth = 35
      - Corresponding Accuracy
        - Train Accuracy - 99.99%
        - Test Accuracy - 89.99%
        - F1 Score - 0.4819

   c. Naive Bayes Classifier :
      - Hyper-Parameters Considered = Priors, Var-Smoothing
      - Hyper-Parameter Tuning = Parameter-Search
      - Best Hyper-Parameter Value Obtained
        - Priors = None
        - Var-Smoothing = 0.5

- Corresponding Accuracy
  - Train Accuracy - 90.5%
  - Test Accuracy - 90.0%
  - F1 Score - 0.47

d. KNN Classifier :
- Hyper-Parameters Considered = n_neighbours, metric
- Hyper-Parameter Tuning = Parameter-Search
- Best Hyper-Parameter Value Obtained
  - n_neighbours = 4
  - metric = minkowski
- Corresponding Accuracy
  - Train Accuracy - 91.02%
  - Test Accuracy - 89.53%
  - F1 Score - 0.49

e. Support Vector Machine :
- Hyper-Parameters Considered = Kernel
- Hyper-Parameter Tuning = Grid-Search
- Best Hyper-Parameter Value Obtained
  - Kernel = polynomial
- Corresponding Accuracy
  - Train Accuracy - 90.49%
  - Test Accuracy - 90.01%
  - F1 Score - 0.473

f. Artificial Neural Network :
- Hyper-Parameters Considered = Hidden_layer_sizes, Activation_Functionn
- Hyper-Parameter Tuning = Grid-Search
- Best Hyper-Parameter Value Obtained
  - Activation Function = Logistic
  - (10,)
- Corresponding Accuracy
  - Train Accuracy - 90.54%
  - Test Accuracy - 90.01%
  - F1 Score - 0.47

Que_2] Use MNIST DATASET

1. Use the above classifiers to do multi-class classification where the idea is to classify the image.
   Ans] We have used the above six classifiers to classify the MNIST Dataset. Refer the attached files in for the same.
2. Exploration of Different Evaluation Metrics Evaluate your methods using different evaluation metrics. Tune the parameters using two powerful techniques of grid search and parameter search.
   Ans] Our Models and their performances are summarised as follows (We implemented the all classifier using ~14% of the data as our test data and the other part as our train data (except for SVMs).) :

   a. Decision Tree :
      - Hyper-Parameters Considered = max_features, min_samples_split, max_depth, criterion
      - Hyper-Parameter Tuning = Grid-Search
      - Best Hyper-Parameter Value Obtained
        - criterion = entropy
        - max_features = None
        - min_samples_split = 20
        - max_depth = None (meaning fixing max_depth decreases efficiency of model)
      - Corresponding Accuracy
        - Train Accuracy - 94.85%
        - Test Accuracy - 87.57%
        - F1 Score - 0.87
          3

   b. Random Forest :
      - Hyper-Parameters Considered = n_estimators, criterion, max_depth
      - Hyper-Parameter Tuning = Grid-Search
      - Best Hyper-Parameter Value Obtained =
        - n_estimators = 200
        - criterion = gini
        - max_depth = 45
      - Corresponding Accuracy
        - Train Accuracy - 100.00%
        - Test Accuracy - 97.02%
        - F1 Score - 0.96

   c. Naive Bayes Classifier :
      - Hyper-Parameters Considered = Priors, Var-Smoothing
      - Hyper-Parameter Tuning = Parameter-Search
      - Best Hyper-Parameter Value Obtained

- Priors = None
- Var-Smoothing = 0.13
- Corresponding Accuracy
  - Train Accuracy - 80.15%
  - Test Accuracy - 80.50%
  - F1 Score - 0.80

d. KNN Classifier :

- Hyper-Parameters Considered = n_neighbours, metric
- Hyper-Parameter Tuning = Parameter-Search
- Best Hyper-Parameter Value Obtained
  - n_neighbours = 3
  - metric = euclidean
- Corresponding Accuracy
  - Train Accuracy - 98.63%
  - Test Accuracy - 97.05%
  - F1 Score - 0.97

e. Support Vector Machine :

- Hyper-Parameters Considered = Kernel
- Hyper-Parameter Tuning = Grid-Search
- Best Hyper-Parameter Value Obtained
  - Kernel = radial-basis-function
- Corresponding Accuracy
  - Train Accuracy - 98.71%
  - Test Accuracy - 97.28%
  - F1 Score - 0.9727

f. Artificial Neural Network :

- Hyper-Parameters Considered = Hidden_layer_sizes, Activation_Functionn
- Hyper-Parameter Tuning = Grid-Search
- Best Hyper-Parameter Value Obtained
  - Activation Function = Logistic
  - (100, 100)
- Corresponding Accuracy
  - Train Accuracy - 97.67%
  - Test Accuracy - 96.55%
  - F1 Score - 0.96

Team Members

1. Harshvardhan Patel [2020MT10808]
2. Hanish Goyal [2020MT10805]
3. Divyansh Mohan Bansal [2020MT10800]

Last updated 2023-03-28 23:41:17 +0550