B. Tech. III (CSE) Semester – V

MACHINE LEARNING (CORE - 11)

CS305

1. Implement classification and regression techniques.

Program 1: Linear Regression

Dataset: iris.csv

The iris(the flower) dataset attached contains five variables namely,

i. SepalLength(Cm)
ii. SepalWidth(Cm)
iii. PetalLength(Cm)
iv. PetalWidth(Cm)
v. Species

All you need to do is performing Linear Regression in python on this dataset taking Sepal Length as Response or dependent variable and rest of the variables as independent ones.

Before performing Linear Regression, please check

1) whether there exists any multicollinearity in the independent variables with correlation matrix and suitable scatter plots.
2) Find the correlation between dependent variable and each independent variable.
3) Find if there is any outlier in the variables given with suitable boxplots.

Program 2: Logistic Regression

Dataset: Abalone data (abalone.csv)

- *Sources*:

(a) Original owners of database:

Marine Resources Division

Marine Research Laboratories - Taroona

Department of Primary Industry and Fisheries, Tasmania

GPO Box 619F, Hobart, Tasmania 7001, Australia

(contact: Warwick Nash +61 02 277277, wnash@dpi.tas.gov.au)

(b) Donor of database:

Sam Waugh (Sam.Waugh@cs.utas.edu.au)

Department of Computer Science, University of Tasmania

GPO Box 252C, Hobart, Tasmania 7001, Australia

(c) Date received: December 1995

- *Number of Instances*: 4177
- *Number of Attributes*: 8
- *Attribute information*:

Given is the attribute name, attribute type, the measurement unit and a brief description

Name	Data Type	Meas.	Description
Sex	nominal		M, F, and I (infant)
Length	continuous	mm	Longest shell measurement
Diameter	continuous	mm	perpendicular to length
Height	continuous	mm	with meat in shell
Whole weight	continuous	grams	whole abalone
Shucked weight	continuous	grams	weight of meat
Viscera weight	continuous	grams	gut weight (after bleeding)
Shell weight	continuous	grams	after being dried
Rings	integer		+1.5 gives the age in years

6. From the above description of the dataset predict the Sex of abalone using Logistic Regression Classifier and make notebook.

Program 3: Classification using Decision Tree

Dataset: Diabetes data.csv

Please perform Classification Analysis using Decision Tree & Random Forest Classifier on the diabetes dataset attached.

- Split the data into train & test set.
- The data set may contain missing values so check before diving into applying the algos.
- Please visualize the classification report using ROC & AUC plot

Program 4: Classification using Random Forest Tree

Please perform Classification Analysis using Random Forest Classifier on the diabetes dataset attached. Also follow the same objective given in program 3.

2. Implement clustering and statistical modeling methods.

Statistical modeling methods

A. Parametric Model

Program 4: Parametric Model: Naïve Bayes' classifier

Dataset: sms_spam.csv

Train a Naive Bayes model to classify future SMS messages as either spam or ham.

Steps:

1. Convert the words ham and spam to a binary indicator variable (0/1)
2. Convert the txt to a sparse matrix of TFIDF vectors
3. Fit a Naive Bayes Classifier
4. Measure your success using roc_auc_score

B. Non-Parametric Model

3. Implement various dimensionality reduction techniques

Program 5: Non-Parametric Model: Principal Component Analysis (PCA)

Dataset: SPECTF.test (To be fetched online)

The dataset describes diagnosing of cardiac Single Proton Emission Computed Tomography (SPECT) images. Each of the patients is classified into two categories: normal and abnormal. The database of 267 SPECT image sets (patients) was processed to extract features that summarize the original SPECT images. As a result, 44 continuous feature pattern was created for each patient.

Steps:

1. Read the **SPECTF** dataset from UCI machine learning repository (use Read SPECTF.txt)
2. Check the correlation between the variables and draw heatmap.
3. Train a Logistic Regression model and record the time taken to train before applying PCA
4. standardizing the variables
5. Apply PCA and depict number of significant variables for logistic regression.

6. Train a Logistic Regression model and record the time taken to train using significant variables.
7. Write your inferences.

Program 6: Non-Parametric Model: Decision Tree – CART (Classification and Regression Tree)

Datasets: titanic_train.csv and titanic_test.csv

1) Read the data.
2) Remove the impurity from the data by removing the missing values
3) Build the Decision Tree Classifier model with criterion="entropy", max_depth=3
4) Visualize the tree by visiting http://webgraphviz.com/
5) Print the score of the decision tree
6) Print Confusion matrix and depict it too.
7) Explore the changes in step 4 to 6 when we change the tree depth by Setting "max_depth" to 10 and "min_samples_split" to 5.

Program 7: Non-Parametric Model: Random Forest (Ensemble Learning)

Datasets: titanic_train.csv and titanic_test.csv

1) Read the data.
2) Remove the impurity from the data by removing the missing values
3) Build the Random Forest Classifier model with max_depth=3
4) Print the score of the fitted random forest
5) Apply following code on your model:
   print(my_forest.feature_importances_)
   list(zip(columns,my_forest.feature_importances_))
6) Print Confusion matrix and depict it too.
7) Make ROC curve on Predicted probabilities

Program 8: Clustering using K-Means algorithm

Dataset: iris.csv

Clustering Iris flowers to analyze characteristics exhibited by each cluster. Based on the features given we've to cluster possible species to understand their characteristics.

1) Split the data into train & test set.
2) Loop through each cluster and fit the model to the train set

3) generate the predicted cluster assignment and append the mean distance my taking the sum divided by the shape
4) Use the Elbow Method to identify number of clusters to choose
5) Pick the fewest number of clusters that reduces the average distance
6) Plot clusters: Do Canonical Discriminant Analysis for variable reduction

Program 9: Clustering using K-Means algorithm

Datasets: poker_train.csv.csv and poker_test.csv

1) Read the data.
2) Do the exploratory data analysis (EDA)
3) Standardize clustering variables to have mean=0 and sd=1 so that card suit and rank are on the same scale as to have the variables equally contribute to the analysis
4) k-means cluster analysis for 1-10 clusters due to the 10 possible class outcomes for poker hands
5) Plot average distance from observations from the cluster centroid to use the Elbow Method to identify number of clusters to choose
6) Perform the Canonical Discriminant Analysis for variable reduction:
   a) creates a smaller number of variables
   b) linear combination of clustering variables
   c) Canonical variables are ordered by proportion of variance accounted for
   d) most of the variance will be accounted for in the first few canonical variables

4. Implement neural networks and non-parametric techniques.

Program 10: Non-Parametric Model: Support Vector Machine (SVM)

Dataset: Abalone data (abalone.csv)

1) Read the data.
2) Do the exploratory data analysis (EDA)
3) Standardize the input
4) Build the Logistic Regression model then do prediction on test dataset.
5) Print the *accuracy_score, confusion_matrix, classification_report* for Logistic Regression model
6) Now, build the Support Vector Classification model then do prediction on test dataset.
7) Print the *accuracy_score, confusion_matrix, classification_report* for Support Vector Classification model
8) Perform PCA then use *print (pca_model.explained_variance_ratio_)* to significant variables
9) Visualize coefficients with heat map using the Principal components

10) Now with Principal Components build and test Logistic Regression and Support Vector Classification
11) Print the *accuracy_score, confusion_matrix, classification_report* for built model in step 10.

Program 11: Implement Neural Networks

Dataset: keras.datasets.mnist (Online)

1) Read the data MNIST from Keras
2) Visualize any random data as an image
3) Split the data (Use mnist.txt file)
4) Put 60k images in the training set and 10k images in the testing set.
5) Do the Feature Scaling [xtrain = xtrain/255, xtest = xtest/255]
6) Build the model [model = tf.keras.models.Sequential()]
7) Add the layers (Input, hidden and output layer)
8) Compile the model with setting: optimizer='adam', loss='sparse_categorical_crossentropy', metrics=['accuracy']
9) Train the model
10) Perform the predictions on testing dataset
11) Visualize confusion matrix with heat map
12) Save the model as *model* of *pickle*.
13) Open the saved model to predict random MNIST data.