Assignment-1

Based on NumPy

Q1: Questions on Basic NumPy Array

(a) Reverse the NumPy array: arr = np.array([1, 2, 3, 6, 4, 5])

(b) Flatten the NumPy arr: array1 = np.array([[1, 2, 3], [2, 4, 5], [1, 2, 3]]) using any two NumPy in-built methods

(c) Compare the following numpy arrays:
arr1 = np.array([[1, 2], [3, 4]])
arr2 = np.array([[1, 2], [3, 4]])

(d) Find the most frequent value and their indice(s) in the following arrays:
i. x = np.array([1,2,3,4,5,1,2,1,1,1])
ii. y = np.array([1, 1, 1, 2, 3, 4, 2, 4, 3, 3,])

(e) For the array gfg = np.matrix('[4, 1, 9; 12, 3, 1; 4, 5, 6]'), find
i. Sum of all elements
ii. Sum of all elements row-wise
iii. Sum of all elements column-wise

(f) For the matrix: n_array = np.array([[55, 25, 15],[30, 44, 2],[11, 45, 77]]), find
i. Sum of diagonal elements
ii. Eigen values of matrix
iii. Eigen vectors of matrix
iv. Inverse of matrix
v. Determinant of matrix

(g) Multiply the following matrices and also find covariance between matrices using NumPy:
i. p = [[1, 2], [2, 3]]
q = [[4, 5], [6, 7]]
ii. p = [[1, 2], [2, 3], [4, 5]]
q = [[4, 5, 1], [6, 7, 2]]

(h) For the matrices: x = np.array([[2, 3, 4], [3, 2, 9]]); y = np.array([[1, 5, 0], [5, 10, 3]]), find inner, outer and cartesian product?

Q2: Based on NumPy Mathematics and Statistics

(a) For the array: array = np.array([[1, -2, 3],[-4, 5, -6]])
i. Find element-wise absolute value
ii. Find the 25th, 50th, and 75th percentile of flattened array, for each column, for each row.
iii. Mean, Median and Standard Deviation of flattened array, of each column, and each row

(b) For the array: a = np.array([-1.8, -1.6, -0.5, 0.5,1.6, 1.8, 3.0]). Find floor, ceiling and truncated value, rounded values

Q3: Based on Searching and Sorting

(a) For the array: array = np.array([10, 52, 62, 16, 16, 54, 453]), find

i. Sorted array

ii. Indices of sorted array

iii. 4 smallest elements

iv. 5 largest elements

(b) For the array: array = np.array([1.0, 1.2, 2.2, 2.0, 3.0, 2.0]), find

i. Integer elements only

ii. Float elements only

Q4:

(a) Write a function named img_to_array(path) that reads an image from a specified *path* and save it as text file on local machine? (Note: use separate cases for RGB and Grey Scale images)

(b) Load the saved file into jupyter notebook?

Download the dataset (in csv format) from the following link:

https://www.kaggle.com/jahias/microsoft-adventure-works-cycles-customer-data

The Marketing department of Adventure Works Cycles wants to increase sales by targeting specific customers for a mailing campaign. The company's database contains a list of past customers and a list of potential new customers. By investigating the attributes of previous bike buyers, the company hopes to discover patterns that they can then apply to potential customers. They hope to use the discovered patterns to predict which potential customers are most likely to purchase a bike from Adventure Works Cycles.

Part I: Based on Feature Selection, Cleaning, and Preprocessing to Construct an Input from Data Source

(a) Examine the values of each attribute and Select a set of attributes only that would affect to predict future bike buyers to create your input for data mining algorithms. Remove all the unnecessary attributes. (Select features just by analysis).
(b) Create a new Data Frame with the selected attributes only.
(c) Determine a Data value type (Discrete, or Continuous, then Nominal, Ordinal, Interval, Ratio) of each attribute in your selection to identify preprocessing tasks to create input for your data mining.

Part II: Data Preprocessing and Transformation

Depending on the data type of each attribute, transform each object from your preprocessed data.

Use all the data rows (~= 18000 rows) with the selected features as input to apply all the tasks below, do not perform each task on the smaller data set that you got from your random sampling result.

(a) Handling Null values
(b) Normalization
(c) Discretization (Binning) on Continuous attributes or Categorical Attributes with too many different values
(d) Standardization/Normalization
(e) Binarization (One Hot Encoding)

Part III: Calculating Proximity /Correlation Analysis of two features

Make sure each attribute is transformed in a same scale for numeric attributes and Binarization for each nominal attribute, and each discretized numeric attribute to standardization. Make sure to apply a correct similarity measure for nominal (one hot encoding)/binary attributes and numeric attributes respectively.

(a) Calculate Similarity in Simple Matching, Jaccard Similarity, and Cosine Similarity between two following objects of your transformed input data.
(b) Calculate Correlation between two features Commute Distance and Yearly Income

Lab Assignment 3

Machine Learning (UML501)

Q 1	K-Fold Cross Validation for Multiple Linear Regression (Least Square Error Fit) Download the dataset regarding USA House Price Prediction from the following link: https://drive.google.com/file/d/1O_NwpJT-8xGfU_-3llUl2sgPu0xllOrX/view?usp=sharing Load the dataset and Implement 5- fold cross validation for multiple linear regression (using least square error fit). Steps: a) Divide the dataset into input features (all columns except price) and output variable (price) b) Scale the values of input features. c) Divide input and output features into five folds. d) Run five iterations, in each iteration consider one-fold as test set and remaining four sets as training set. Find the beta (β) matrix, predicted values, and R2_score for each iteration using least square error fit. e) Use the best value of (β) matrix (for which R2_score is maximum), to train the regressor for 70% of data and test the performance for remaining 30% data.
Q 2	Concept of Validation set for Multiple Linear Regression (Gradient Descent Optimization) Consider the same dataset of Q1, rather than dividing the dataset into five folds, divide the dataset into training set (56%), validation set (14%), and test set (30%). Consider four different values of learning rate i.e. {0.001,0.01,0.1,1}. Compute the values of regression coefficients for each value of learning rate after 1000 iterations. For each set of regression coefficients, compute R2_score for validation and test set and find the best value of regression coefficients.
Q 3	Pre-processing and Multiple Linear Regression Download the dataset regarding Car Price Prediction from the following link: https://archive.ics.uci.edu/ml/machine-learning-databases/autos/imports-85.data 1. Load the dataset with following column names ["symboling", "normalized_losses", "make", "fuel_type", "aspiration","num_doors", "body_style", "drive_wheels", "engine_location", "wheel_base", "length", "width", "height", "curb_weight", "engine_type", "num_cylinders", "engine_size", "fuel_system", "bore", "stroke", "compression_ratio", "horsepower", "peak_rpm", "city_mpg", "highway_mpg", "price"] and replace all ? values with NaN 2. Replace all NaN values with central tendency imputation. Drop the rows with NaN values in price column 3. There are 10 columns in the dataset with non-numeric values. Convert these values to numeric values using following scheme: (i) For “num_doors” and “num_cylinders”: convert words (number names) to figures for e.g., two to 2 (ii) For "body_style", "drive_wheels": use dummy encoding scheme (iii) For “make”, “aspiration”, “engine_location”,fuel_type: use label encoding scheme (iv) For fuel_system: replace values containing string pfi to 1 else all values to 0. (v) For engine_type: replace values containing string ohc to 1 else all values to 0. 4. Divide the dataset into input features (all columns except price) and output variable (price). Scale all input features. 5. Train a linear regressor on 70% of data (using inbuilt linear regression function of Python) and test its performance on remaining 30% of data. 6. Reduce the dimensionality of the feature set using inbuilt PCA decomposition and then again train a linear regressor on 70% of reduced data (using inbuilt linear regression function of Python). Does it lead to any performance improvement on test set?

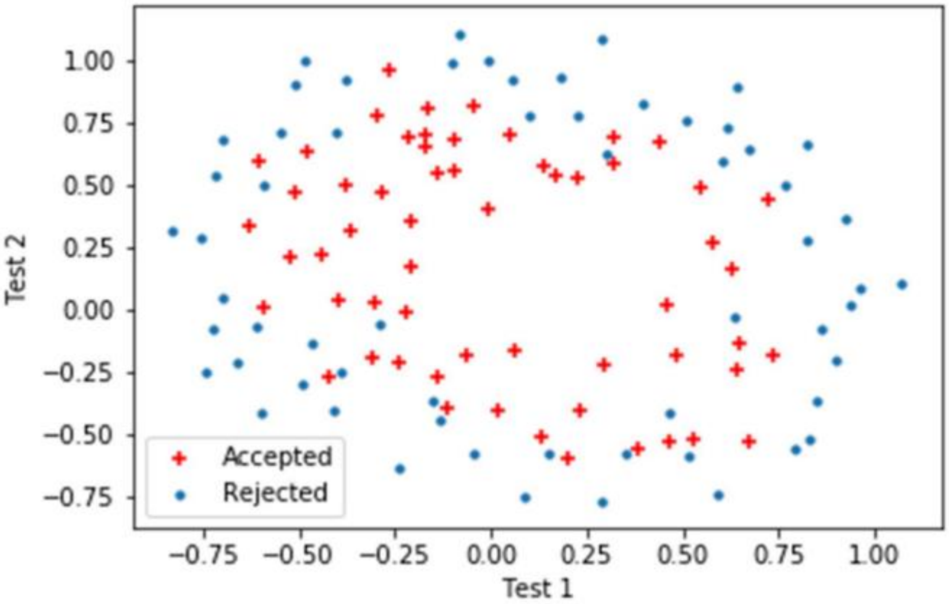
Lab Assignment 4

Machine Learning (UML501)

Q 1	(**Based on Step-by-Step Implementation of Ridge Regression using Gradient Descent Optimization**) Generate a dataset with atleast seven highly correlated columns and a target variable. Implement Ridge Regression using Gradient Descent Optimization. Take different values of learning rate (such as 0.0001,0.001,0.01,0.1,1,10) and regularization parameter (10^{-15},10^{-10},10^{-5},10^{-3},0,1,10,20). Choose the best parameters for which ridge regression cost function is minimum and R2_score is maximum.
Q 2	Load the Hitters dataset from the following link https://drive.google.com/file/d/1qzCKF6JKKMB0p7ul_lLy8tdmRk3vE_bG/view?usp=sharing (a) Pre-process the data (null values, noise, categorical to numerical encoding) (b) Separate input and output features and perform scaling (c) Fit a Linear, Ridge (use regularization parameter as 0.5748), and LASSO (use regularization parameter as 0.5748) regression function on the dataset. (d) Evaluate the performance of each trained model on test set. Which model performs the best and Why?
Q 3	**Cross Validation for Ridge and Lasso Regression** Explore Ridge Cross Validation (RidgeCV) and Lasso Cross Validation (LassoCV) function of Python. Implement both on Boston House Prediction Dataset (load_boston dataset from sklearn.datasets).

Lab Assignment 5

Machine Learning (UML501)

Q 1	Multiclass Logistic Regression Implement Multiclass Logistic Regression (step-by-step) on Iris dataset using one vs. rest strategy?
Q 2	Ridge Logistic Regression Download the exam dataset from the following link: https://drive.google.com/file/d/1wH6ofvNGPmORFlCLt72WGhJYPZiXstYh/view?usp=sharing The dataset labels that whether or not the student will get admission on the basis of the two exam scores. The plot of the data against exam1 and exam2  As clear from the figure, a linear decision boundary does not fit well. So, fit a Logistic Regression Classifier with polynomial function of test1 and test2 scores upto degree 6 using i. Step-by-Step Logistic Regression (with no regularization; alpha=10; number of iterations=1000) ii. Step-by-Step Logistic Regression (with ridge regularization; alpha=10; number of iterations=1000; lambda=0.2)

Lab Assignment 6

Machine Learning (UML501)

Q 1	Download the dataset from the following link. It has 5572 messages labeled as spam or ham: https://raw.githubusercontent.com/justmarkham/pycon-2016-tutorial/master/data/sms.tsv Implement Multinomial Naïve Bayes classifier (in-built) for detection of messages as spam or ham?
Q 2	(Gaussian Naïve Bayes Classifier) Implement Gaussian Naïve Bayes Classifier on the Iris dataset from sklearn.datasets using (i) Step-by-step implementation (ii) (ii) In-built function
Q3	Explore about GridSearchCV toot in scikit-learn. This is a tool that is often used for tuning hyperparameters of machine learning models. Use this tool to find the best value of K for K-NN Classifier using any dataset.

Lab Assignment 7

Machine Learning (UML501)

Note: Required datasets are on LMS

Q 1	Use the weather dataset to implement the decision tree. Try different available parameters of the inbuilt method.
Q 2	Implement SVM by taking Boston dataset by dropping (using correlation) the least significant feature and tune the values of C and gamma parameters. Write your own function to find the correlation between an input feature and out feature.
Q3	Implement K means clustering by using Mall Customers dataset, by making different clusters. Save the model evaluation parameters in CSV file, for each of the cluster.
Q4	Implement Hierarchical clustering means clustering by using Mall Customers dataset, by making different clusters. Save the model evaluation parameters in CSV file, for each of the cluster.
Q5	Implement ANN on Diabetes dataset by taking different hidden layers, Relu as activation function in the hidden layer and sigmoid as output. Store the weights in a CSV file.
Q6	Implement ANN on Breast Cancer dataset by taking different hidden layers, Relu as activation function in the hidden layer and sigmoid as output. Store the weights in a CSV file.