**Acropolis Institute of Technology and**

**Research,Indore**  
 

**Department of Computer Science and Engineering**



**B. Tech. VI Semester(CI)**

**Jan - June 2022**

**Lab Manual**

**On**

**Machine Learning [CS-602]**

**Submitted to Submitted By**

Dr.Shilpa Bhalerao Aaditya Dubey

Professor , Head ,CSIT 0827CI191001

*Assignments*

1. ***Explain the 10 function of Numpy***

*🡪 Numpy is a python package for scientific computing that provides high-performance multidimensional arrays objects.*

# *linspace - The numpy.linspace(start, stop, num=50, endpoint=True, retstep=False, dtype=None, axis=0) function returns evenly spaced numbers over a specified interval defined by the first two arguments of the function (start and stop — required arguments).*

# *digitize - Maybe you have never heard about this function, but it can be really useful working with continuous spaces in reinforcement learning. The numpy.digitize(x, bins, right=False) function has two arguments: (1) an input array x, and (2) an array of bins.*

# *repeat - The numpy.repeat(a, repeats, axis=None) function repeats the elements of an array. The number of repetitions is specified by the second argument repeats.*

# *random - The numpy.random.randint(low, high=None, size=None, dtype=’l’) function returns random integers from the interval [low,high).*

# *polyfit - The numpy.polyfit(x, y, deg, rcond=None, full=False, w=None, cov=False) function outputs a polynomial of degree deg that fits the points (x,y), minimizing the square error.*

# *polyval - The numpy.polyval(p, x) function evaluates a polynomial at specific values. Previously, we have obtained a linear model to predict the weight of a man (weight=5.96\*height-224.50) by using the numpy.polyfit function.*

# *nan - Numpy library includes several constants such as not a number (Nan), infinity (inf) or pi. In computing, not a number is a numeric data type that can be interpreted as a value that is undefined.*

# *argmax - The numpy.argmax(a, axis=None, out=None) function returns the indices of the maximum values along an axis.*

# *squeeze - The numpy.squeeze(a, axis=None) removes single-dimensional entries from the shape of an array. The argument axis specifies the axis we want to squeeze out.*

# *histogram - The numpy.histogram(a, bins=10, range=None, normed=None, weights=None, density=None) computes the histogram of a set of data.*

# *Explain concept of Data frame in ML and write 10 commands used in ML from Pandas*

# *A Dataframe is a two-dimensional data structure, i.e., data is aligned in a tabular fashion in rows and columns. In dataframe datasets arrange in rows and columns, we can store any number of datasets in a dataframe.*

# *Commands in pandas*

# *read\_csv() : read\_csv() function helps read a comma-separated values (csv) file into a Pandas DataFrame.*

# *head() : head(n) is used to return the first n rows of a dataset.*

# *describe() : describe() is used to generate descriptive statistics of the data in a Pandas DataFrame or Series.*

# *memory\_usage() : memory\_usage() returns a Pandas Series having the memory usage of each column (in bytes) in a Pandas DataFrame.*

# *astype() : astype() is used to cast a Python object to a particular data type.*

# *loc[:]: loc[:] helps to access a group of rows and columns in a dataset, a slice of the dataset, as per our requirement.*

# *to\_datetime() : to\_datetime() converts a Python object to datetime format.*

# *value\_counts() : value\_counts() returns a Pandas Series containing the counts of unique values.*

# *drop\_duplicates() : drop\_duplicates() returns a Pandas DataFrame with duplicate rows removed.*

# *groupby() : groupby() is used to group a Pandas DataFrame by 1 or more columns.*

# *Explain command dataset splitting in python.*

# *To train any machine learning model irrespective what type of dataset is being used you have to split the dataset into training data and testing data.*

# *Scikit-learn alias sklearn is the most useful and robust library for machine learning in Python. The scikit-learn library provides us with the model\_selection module in which we have the splitter function train\_test\_split().*

# *Python Code*

# *# import modules*

# *import pandas as pd*

# *from sklearn.linear\_model import LinearRegression*

# *# read the dataset*

# *df = pd.read\_csv('Real estate.csv')*

# *# get the locations*

# *X = df.iloc[:, :-1]*

# *y = df.iloc[:, -1]*

# *# split the dataset*

# *X\_train, X\_test, y\_train, y\_test = train\_test\_split(*

# *X, y, test\_size=0.05, random\_state=0)*

# *Write the command for model creation in linear regression with detail of each parameter*

# *🡪*

# *Linear regression follows the linear mathematical model for determining the value of one dependent variable from value of one given independent variable.*

# *Step 1: Loading dataset*

# *import numpy as np*

# *import pandas as pd*

# *Step 2: Loading dataset*

# *dataset = pd.read\_csv(“weather.csv”)*

# *Step 3: Split to independent and dependent variables*

# *x = dataset.iloc[:,2].values*

# *y = dataset.iloc[:,3].values*

# *Step 4: Splitting data into training and testing data*

# *Step 5: Choosing the Model*

# *Step 6: Fit our model*

# *Step 7: Predict the output*

# *Write command for model creation with explanation*

# *🡪 The basic idea of any machine learning model is that it is exposed to a large number of inputs and also supplied the output applicable for them. On analysing more and more data, it tries to figure out the relationship between input and the result.*

# *The ML Model -Linear Regression*

# *Working with linear regression model is simple. Create a model, train it and then use it.*

# *Model building*

# *from sklearn.linear\_model import LinearRegression*

# *lr = LinearRegression()*

# *lr.fit(X\_train, y\_train)*

# *The first line imports the LinearRegression() function from the sklearn.linear\_model sub-module. Next, the LinearRegression() function is assigned to the lr variable and the .fit() function performs the actual model training on the input data X\_train and y\_train.*

# *Now that the model is built, we’re going to apply it to make predictions on the training set and test set as follows:*

# *y\_lr\_train\_pred = lr.predict(X\_train)*

# *y\_lr\_test\_pred = lr.predict(X\_test)*

# *What is model.fit() and model.predict()? Explain with example.*

# *🡪*

# *fit() method will fit the model to the input training instances while predict() will perform predictions on the testing instances, based on the learned parameters during fit.*

# *Now as an example, let’s consider a classification problem where we need to train a SVC model to recognise hand-written images. In the code below, we first load our data and then split it into training and testing sets. Then we instantiate a SVC classifier and finally call fit() to train the model using the input training and data.*

# *Now that we have trained our model, the next step typically involves predictions over the testing set. To do so, we need to call the method predict() that will essentially use the learned parameters by fit() in order to perform predictions on new, unseen test data points. Essentially, predict() will perform a prediction for each test instance and it usually accepts only a single input (X).*

# *What are commands used for Data preprocessing?*

# *🡪 Data Preprocessing refers to the steps applied to make data more suitable for data mining. The steps used for Data Preprocessing usually fall into two categories:*

1. *selecting data objects and attributes for the analysis.*
2. *creating/changing the attributes.*

*Categorical data refers to the information that has specific categories within the dataset. In the dataset cited above, there are two categorical variables – country and purchased.*

*#Catgorical data*

*#for Country Variable*

*from sklearn.preprocessing import LabelEncoder*

*label\_encoder\_x= LabelEncoder()*

*x[:, 0]= label\_encoder\_x.fit\_transform(x[:, 0])*

# *Splitting the dataset is the next step in data preprocessing in machine learning. Every dataset for Machine Learning model must be split into two separate sets – training set and test set.*

# *Explain any pre-trained model and how it is useful in transfer learning*

# *🡪 A pre-trained model is a saved network that was previously trained on a large dataset, typically on a large-scale image-classification task. You either use the pretrained model as is or use transfer learning to customize this model to a given task.*

# *Transfer Learning is a machine learning method where we reuse a pre-trained model as the starting point for a model on a new task. To put it simply—a model trained on one task is repurposed on a second, related task as an optimization that allows rapid progress when modeling the second task.*

# *By using pre-trained models which have been previously trained on large datasets, we can directly use the weights and architecture obtained and apply the learning on our problem statement. This is known as transfer learning. We “transfer the learning” of the pre-trained model to our specific problem statement.0*

# 