PAGE NO.:
DATE / /

NAME :- AFNAN ATTAR PRN :- F19112003 CLASS :- BE COMP II
SUBJECT :- ML EXPERIMENT NO :- 1

Q1) What are application of linear regression?

Ans Application of linear regression are as follows :-

I] Trend Line :- A trend line represents a trend, the long-term movement in time series data after other components have been accounted for.

II] Epidemology :- Evidence relating tobacco smoking to mortality came from observational studies employing regression analysis.

III] Finance : Capital assest pricing model uses linear regres.

IV] Economics : Linear regression is a predominant empirical tool in economics.

V] Machine learning : linear regression plays an important role in the subfield of artificial intelligence known as machine learning.

Q2) What are important function used for linear regression while program implementation and explain their purpose?

Ans I] Cost Function :

1. It evaluates model's predictions and tells us how accurate are the model's predictions.
2. Given as : $J(\theta) = \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2$

II] loss Function :-

1. Cost function is the sum of losses from each data point calculated with loss function.
2. Given as :- least square Error $= \frac{1}{2} (h_\theta(x^{(i)}) - y^{(i)})^2$

PAGE NO:
DATE / /

* Python implementation :-

```
class linear Regression:
    def __init__(self, learning_rate = le-3, n_steps = 1000):
        self.learning_rate = learning_rate
        self.n_steps = n_steps

    def fit (self, X, Y):
        Xtrain = np.c[np.ones(X.shape[0]), X]
        self.W = np.random.rand((Xtrain.shape[1]))
        for i in range (self.n_steps):
            self.w = self.W - self.learning_rate *
                     self.calcGradient (Xtrain, Y)

    def predict (self, X):
        Xpred = np.c_[np.ones(X.shape[0]), X]
        return np.dot(Xpred, self.w)

    def calcGradient(self, X, Y):
        return 2/X.shape[0] * np.dot(X.T,
            (np.dot(X, self.w) - Y))
```

Q3) How does a Random Forest work? What are the advantages of random forest methodology?

Ans 1. Random forest like its name implies consists of a large number of individual decision trees that operates as an ensemble.

2. Each individual tree in the random forest spits out a class prediction and the class with most votes becomes our model's prediction.

3. The fundamental concept behind random forest is a simple but powerful one :- the wisdom of crowds

(AKA Audience Poll in KBC).

4. Advantages of random forest methodology :-

i) More accurate than decision tree algorithm.

ii) Provides an effective way of handling missing data.

iii) Solves issue of overfitting in decision tree.

iv) Provides reasonable prediction without hyperparameter tuning.

PAGE NO.:
DATE / /

NAME: AFNAN ATTAR PRN:- F19112003 CLASS:- BE COMP II
SUBJECT:- ML EXPERIMENT No.:

Q1) Why should we not use the KNN algorithm for large datasets?

Ans 1. KNN works well with smaller datasets because it is a lazy learner.

2. It needs to store all the data and then make decision only at run time.

3. Another thing is that KNN is sensitive to noise.

Q2) Is Feature Scaling required for the KNN algorithm? Explain with proper justification.

Ans Yes, feature scaling is required to get better performance of the KNN algorithm.
For example, imagine a datasets having n number of instances and N number of features.
If one features has values ranging between 0 & 1 while other varies from -999 to 999, when these values used to calculate euclidean distance will affect performance by giving higher weightage.

Q3) What do you mean by Hinge loss?

Ans 1. In machine learning, the hinge loss is a loss function used for training classifiers.

2. It is used for "maximum-margin" classification most notably for support vector machines (SVMs).

3. For an intended output $t = \pm 1$ and a classifier score y, the hinge loss of prediction y is defined as-
$$l(y) = \max(0, 1 - t \cdot y).$$

PAGE NO:
DATE / /

Q4) What's the "Kernel trick" and how is it useful?

Ans 1. Kernel trick is to convert dot product of support vectors to the dot product of mapping function.

2. This trick avoids the need to explicitly map the input data to high-dimension feature space in order to train linear learning algorithms to learn a nonlinear function or decision boundary.

3. SVM Optimization problem :-

$$\sum_i \alpha_i - \frac{1}{2} \sum_i \sum_j \alpha_i \alpha_j y_i y_j (\vec{x_{sv_i}} \cdot \vec{x_{sv_j}})$$

we convert this part.