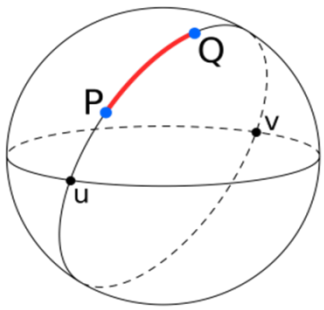
Assignment B-1

Problem Statement:

Predict the price of the Uber ride from a given pickup point to the agreed drop-off location

Haversine formula:

To calculate distance between two location on earth Haversine formula is used, which is developed to find the arc on sphere.



First, convert the latitude and longitude values from decimal degrees to radians. For this divide the values of longitude and latitude of both the points by 180/pi. The value of pi is 22/7. The value of 180/pi is approximately 57.29577951. If we want to calculate the distance between two places in kilometres, use the value 6, 378.8, which is the radius of Earth.

Value of Latitude ( ) in Radians, lat = Latitude / (180/pi) OR

Value of Latitude in Radians, lat = Latitude / 57.29577951

Value of Longitude ( )in Radians, long = Longitude / (180/pi) OR

Value of Longitude in Radians, long = Longitude / 57.29577951

To get the distance between point P and point Q use the following formula:

Database Used:

https://www.kaggle.com/datasets/yasserh/uber-fares-dataset

Python: Colab, spider or similar platform

YT Ref: --------

Code (As attached) & Graphs (wherever applicable): ------

Metrics used for performance measurement: _____

Conclusion: Uber data set gives distances of locations in longitude and latitude. The Haversine distance formula is used to find the distance in KM. the linear regression model developed using distance and fare, is used to predict the next fare if distance is provided.

Assignment B-2

Problem Statement:

Classify the email using the binary classification method. Email Spam detection has two states: a) Normal State – Not Spam, b) Abnormal State – Spam. Use K-Nearest Neighbours.

K-Nearest Neighbours.

The k-nearest neighbors algorithm, also known as KNN or k-NN, is a non-parametric, supervised learning classifier, which uses proximity to make classifications or predictions about the grouping of an individual data point. For classification problems, a class label is assigned on the basis of a majority vote—i.e. the label that is most frequently represented around a given data point is used.

- **Lazy learning algorithm** − KNN is a lazy learning algorithm because it does not have a specialized training phase and uses all the data for training while classification.
- **Non-parametric learning algorithm** − KNN is also a non-parametric learning algorithm because it doesn't assume anything about the underlying data.

Step 1 − For implementing any algorithm, we need dataset. So during the first step of KNN, we must load the training as well as test data.

Step 2 − Next, we need to choose the value of K i.e. the nearest data points. K can be any integer.

Step 3 − For each point in the test data do the following −

- **3.1** − Calculate the distance between test data and each row of training data with the help of any of the method namely: Euclidean, Manhattan or Hamming distance. The most commonly used method to calculate distance is Euclidean.
- **3.2** − Now, based on the distance value, sort them in ascending order.
- **3.3** − Next, it will choose the top K rows from the sorted array.
- **3.4** − Now, it will assign a class to the test point based on most frequent class of these rows.

Step 4 − End

Email classification as Spam/non-spam:

The Natural Language Toolkit (NLTK) is a platform used for building Python programs that work with human language data for applying in statistical natural language processing (NLP). It contains text processing libraries for tokenization, parsing, classification, stemming, tagging and semantic reasoning. Natural language processing is used for building applications such as Text classification.

Tokenization is the process by which a large quantity of text is divided into smaller parts called tokens. These tokens are very useful for finding patterns and are considered as a base step for stemming and lemmatization. We use the method word_tokenize() to split a sentence into words. The output of word tokenization can be converted to Data Frame for better text understanding in machine learning applications. It can also be provided as input for further text cleaning steps such as punctuation removal, numeric character removal or stemming. Machine learning models need numeric data to be trained and make a prediction. Word tokenization becomes a crucial part of the text (string) to numeric data conversion. Tokenized words are used for classification of spam and non-spam emails.

Database Used:

https://www.kaggle.com/code/ayhampar/spam-ham-dataset/data

Python: Colab, spider or similar platform

YT Ref: https://www.youtube.com/watch?v=VLBaKLHjJ7w

Code (As attached) & Graphs (wherever applicable) ---------

Metrics used for performance measurement: _____

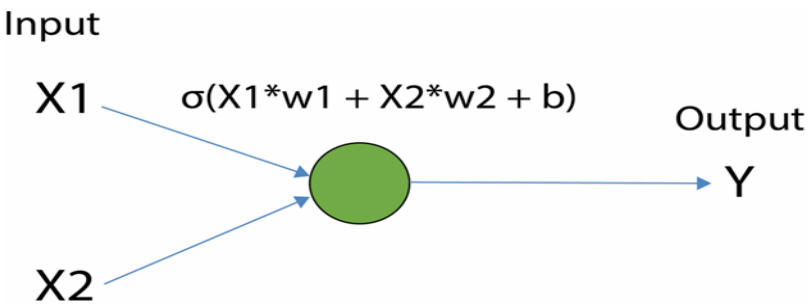
Conclusion: In this experiment we classified emails as spam and non-spam emails based on the words used in emails. For text processing NLTK package is used and words are collected for classification. K-nn algorithm finds the similarity of text with test text patterns and based on majority votes, email is classified as spam or non-spam.

Assignment B-3

Problem Statement:

Given a bank customer, build a neural network-based classifier that can determine whether they will leave or not in the next 6 months. (Customer Churning)

Neural Network Concept:



The image above is called one **node** or **unit** in the deep learning world, which is represented by the green circle. We call the parameters of the model as **coefficients** and **intercepts**. In deep learning models, the parameters are referred to as **weights** (w) and **biases** (b):

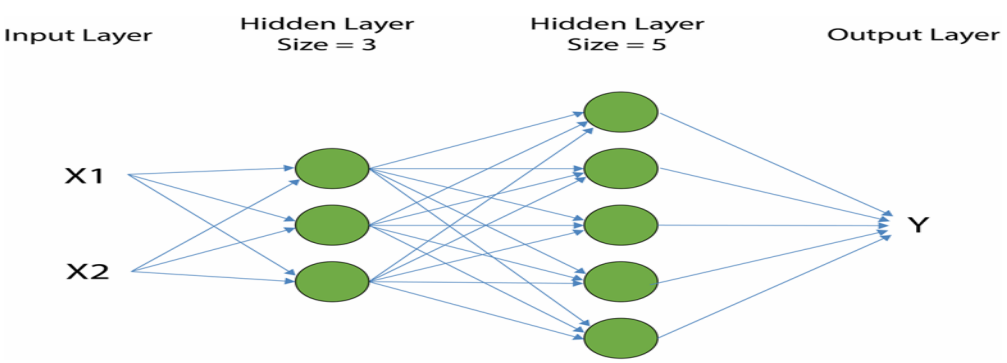
Forward propagation—the propagation of data through the network, multiplying the input values by the weight of each connection for every node, and

Backpropagation—the calculation of the gradient of the loss function with respect to the weights in the matrix,

Gradient descent—the optimization algorithm that's used to find the minimum of the loss functions. An overview of the logistic regression model with two-dimensional input can be seen in the following image.

Activation Function - a nonlinear function is applied to the sum of the weighted inputs and the bias term is used to compute the final output of the node. Ex Sigmoid, tanh, ReLU

It is also possible to build multi-layer neural networks by stacking multiple layers of processing nodes after one another, as shown in the following image.



Hyper parameters- The parameters that are required to be selected by the developer are called hyperparameters and include parameters such as the number of layers and the number of nodes in each layer.

Loss Function- When learning the optimal parameters (weights and biases) of a model, we need to define a function to measure error. This function is called the **loss function** and it provides us with a measure of how different network-predicted outputs are from the real outputs in the dataset.

Ex.- • mean_squared_error., mean_absolute_error., mean_absolute_percentage_error., binary_crossentropy, categorical_crossentropy

During the training process, we keep changing the model parameters until the minimum difference between the model-predicted outputs and the real outputs is reached. This is called an **optimization process**.

batch_size –this argument determines the number of data examples to be included at each iteration of the optimization algorithm. batch_size=None is equivalent to the standard version of gradient descent, which uses the entire dataset in each iteration.

epoch – this argument determines how many times the optimization algorithm passes through the entire training dataset before it stops.

Keras : Keras runs on top of open source machine libraries like TensorFlow or Cognitive Toolkit (CNTK). TensorFlow is the most famous symbolic math library used for creating neural networks and deep learning models. TensorFlow is very flexible and the primary benefit is distributed computing. CNTK is deep learning framework developed by Microsoft. It uses libraries such as Python, C#, C++ or standalone machine learning toolkits. TensorFlow is very powerful library but difficult to understand for creating neural networks.

Keras is based on minimal structure that provides a clean and easy way to create deep learning models based on TensorFlow or Theano. Keras is designed to quickly define deep learning models

Database Used:

https://www.kaggle.com/datasets/barelydedicated/bank-customer-churn-modeling

Python : Colab, spider or similar platform

YT Ref: https://www.youtube.com/watch?v=p5FydQTHtP0

Code (As attached) & Graphs (wherever applicable) ----------

Metrics used for performance measurement: _____

Conclusion: NN are self-learning models based on input data. NN model can be single layer or multi-layered. By using Keras libraries multilayer NN can be developed easily. Once trained NN can produce accurate result with partial data. NN are used to solve complicated engineering problems. Their training require more time as compared to other models of ML.

Assignment B-4

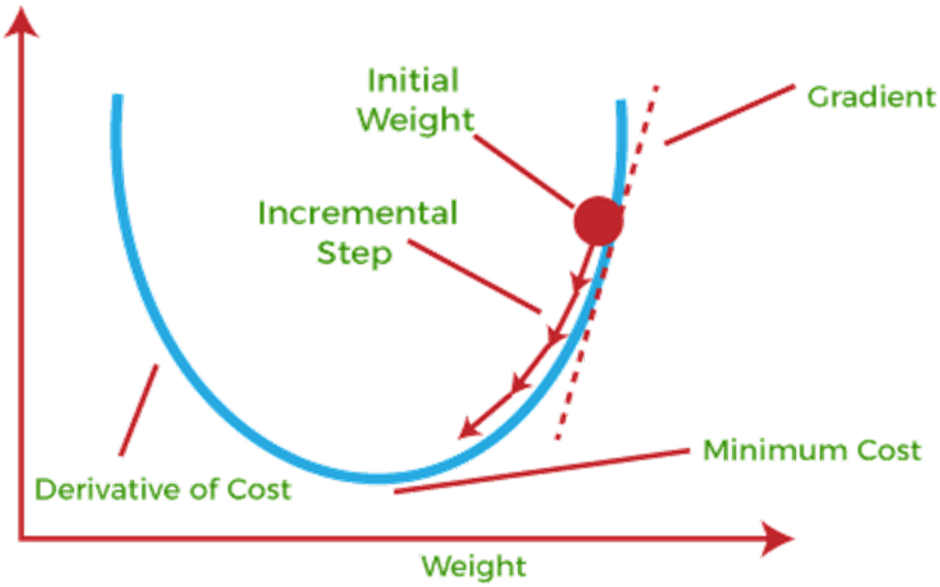
Problem Statement:

Implement Gradient Descent Algorithm to find the local minima of a function.

For example, find the local minima of the function $y=(x+3)^2$ starting from the point $x=2$.

Gradient Descent:

Gradient Descent is an iterative algorithm that is used to minimize a function by finding the optimal parameters. Gradient Descent is an optimization algorithm used for minimizing the cost function in various machine learning algorithms. It is basically used for updating the parameters of the learning model. We start with a random point on the function and move in the **negative direction** of the **gradient of the function** to reach the **local/global minima**.



Gradient Descent algorithm

Step 1 : Initialize X0 =2. Then, find the gradient of the function, dy/dx = 2*(x+3).

Step 2 : Move in the direction of the negative of the gradient . the step size is determined by learning rate. Let us assume the learning rate → 0.01

Step 3 : Let's perform 2 iterations of gradient descent

Iteration 1

X1= X0 –(learning rate)* dy/dx;

X1= X0 –(0.01)* (2(X0+3))

X1=2-(0.01)*(2(2+3))

X1=2-0.1

X1=1.9

Iteration 2

X2= X1 –(learning rate)* dy/dx;

X2= X1 –(0.01)* (2(X1+3))

X2=1.9-(0.01)*(2(1.9+3))

X2=1.9-0.098

X2=1.802

Step 4 : We can observe that the X value is slowly decreasing and should converge to -3 (the local minima) for the function given.

To stop the iterations two conditions are set

1. Select a precision variable in algorithm which calculates the difference between two consecutive "x" values. If the difference between x values from 2 consecutive iterations is lesser than the precision set, then stop the algorithm!
2. Set maximum count of iteration and exit the iterations.

Pseudo code for Gradient decent algorithm

```
while previous_step_size > precision and iters < max_iters:
    prev_x = cur_x #Store current x value in prev_x
    cur_x = cur_x - rate * df(prev_x) #Grad descent
    previous_step_size = abs(cur_x - prev_x) #Change in x
    iters = iters+1 #iteration count
```

Database Used: Inbuilt

Python: Colab, spider or similar flatform

Code (As attached) & Graphs (wherever applicable) _____

You tube link: https://www.youtube.com/watch?v=kEKwJ14uEbI

Metrics used for performance measurement: _____

Conclusion

Gradient Descent is an optimization algorithm used for minimizing the cost function in various machine learning algorithms. The steps of gradients are controlled by learning rate, gradient descent and precision. It used in ANN and deep learning.

Assignment B-5

Problem Statement:

Implement K-Nearest Neighbours algorithm on diabetes.csv dataset. Compute confusion matrix, accuracy, error rate, precision and recall on the given dataset.

K-Nearest Neighbours.

The k-nearest neighbours' algorithm, also known as KNN or k-NN, is a non-parametric, supervised learning classifier, which uses proximity to make classifications or predictions about the grouping of an individual data point. For classification problems, a class label is assigned on the basis of a majority vote—i.e. the label that is most frequently represented around a given data point is used.

- **Lazy learning algorithm** − KNN is a lazy learning algorithm because it does not have a specialized training phase and uses all the data for training while classification.
- **Non-parametric learning algorithm** − KNN is also a non-parametric learning algorithm because it doesn't assume anything about the underlying data.

Step 1 − For implementing any algorithm, we need dataset. So during the first step of KNN, we must load the training as well as test data.

Step 2 − Next, we need to choose the value of K i.e. the nearest data points. K can be any integer.

Step 3 − For each point in the test data do the following −

- **3.1** − Calculate the distance between test data and each row of training data with the help of any of the method namely: Euclidean, Manhattan or Hamming distance. The most commonly used method to calculate distance is Euclidean.
- **3.2** − Now, based on the distance value, sort them in ascending order.
- **3.3** − Next, it will choose the top K rows from the sorted array.
- **3.4** − Now, it will assign a class to the test point based on most frequent class of these rows.

Step 4 − End

Database Used:

https://www.kaggle.com/datasets/saurabh00007/diabetes.csv

Python: Colab, spider or similar platform _____

YT Ref: https://www.youtube.com/watch?v=11E390tiZXU

Code (As attached) & Graphs (wherever applicable) _____

Metrics used for performance measurement: _____

Conclusion: In this experiment we classified patients into diabetic and non-diabetics based on age, Glucose, Insulin, BMI. K-nn algorithm finds the similarity distance based on number of features used for classification. The distance array collected is sorted by ascending order and top N samples decide the class of the patient.

K-NN algorithm for classification of diabetic patients.

This video demonstrates classification of patients as diabetic and non-diabetic. In this K-NN algorithm is used on daibetes.csv data from Kaggle. It is group B-5 assignment as per SPPU B.E. Computer -2019- LP-III. Write-up and code is available if requested through my email pp.halkarnikar@gmail.com

Assignment B-6

Problem Statement:

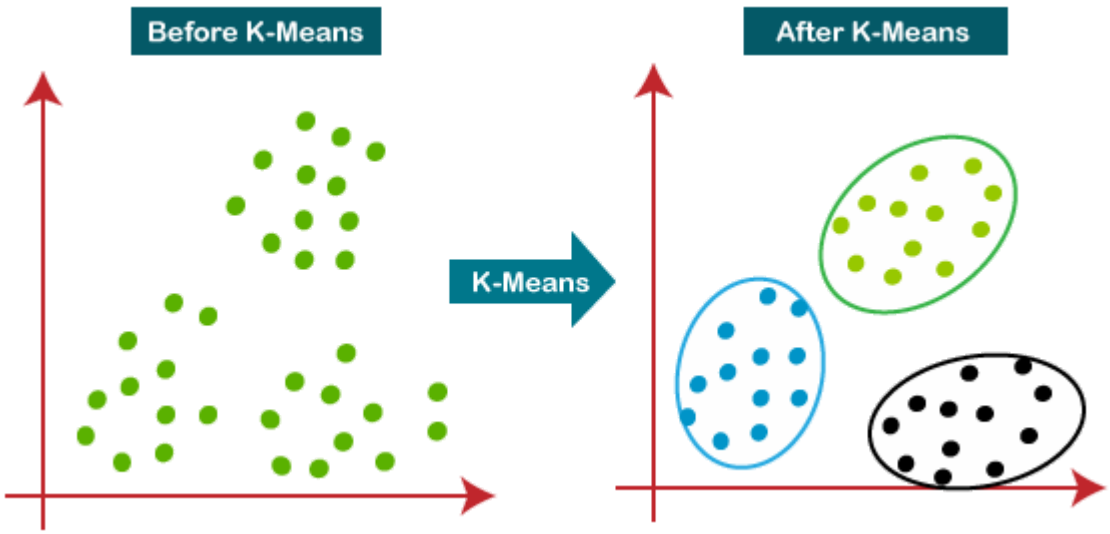
Implement K-Means clustering/ hierarchical clustering on sales_data_sample.csv dataset. Determine the number of clusters using the elbow method.

K-Means Clustering Algorithm

K-Means Clustering is an Unsupervised Learning algorithm, which groups the unlabeled dataset into different clusters. Here K defines the number of pre-defined clusters that need to be created in the process, as if K=2, there will be two clusters, and for K=3, there will be three clusters, and so on. It allows us to cluster the data into different groups and a convenient way to discover the categories of groups in the unlabeled dataset on its own without the need for any training. The algorithm takes the unlabeled dataset as input, divides the dataset into k-number of clusters, and repeats the process until it does not find the best clusters. The value of k should be predetermined in this algorithm.

The k-means clustering algorithm mainly performs two tasks:

- Determines the best value for K center points or centroids by an iterative process.
- Assigns each data point to its closest k-center. Those data points which are near to the particular k-center, create a cluster.



The way kmeans algorithm works is as follows:

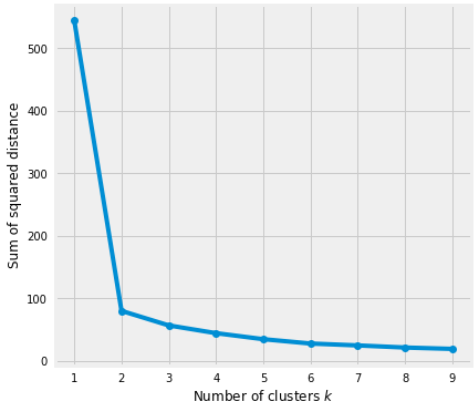
1. Specify number of clusters *K*.
2. Initialize centroids by first shuffling the dataset and then randomly selecting *K* data points for the centroids without replacement.

3. Keep iterating until there is no change to the centroids. i.e assignment of data points to clusters isn't changing.

- Compute the sum of the squared distance between data points and all centroids.
- Assign each data point to the closest cluster (centroid).
- Compute the centroids for the clusters by taking the average of the all data points that belong to each cluster.

Elbow Method

Elbow method gives us an idea on what a good k number of clusters would be based on the sum of squared distance (SSE) between data points and their assigned clusters' centroids. We pick k at the spot where SSE starts to flatten out and forming an elbow.



Database Used:

Dataset link : https://www.kaggle.com/datasets/kyanyoga/sample-sales-data

https://www.kaggle.com/code/victorngeno/mall-customers/data

Python : Colab, spider or similar platform --------

YT Ref: https://www.youtube.com/watch?v=H27rlggXIsk&t=35s

Code (As attached) & Graphs (wherever applicable) --------

Metrics used for performance measurement: _____

Conclusion

Kmeans clustering is one of the most popular clustering algorithms and usually the first thing practitioners apply when solving clustering tasks to get an idea of the structure of the dataset. The goal of kmeans is to group data points into distinct non-overlapping subgroups. It does a very good job when the clusters have a kind of spherical shapes. However, it suffers as the geometric shapes of clusters deviates from spherical shapes. It also doesn't learn the number of clusters from the data and requires it to be pre-defined.