Name –Sarvesh Kale Roll no -193079024

A.)Observations on dataset

What I think is Age, Education, Environment Satisfaction ,years with current employee ,last promotion ,years in current Role ,and all the columns which have numbers in it are important .

The columns in which verbal data is given should be somehow converted into numbers by assigning some weight to the words .

Since the Data is **Multidimensional ,the visualization of data becomes difficult**

Which was easier for 2d or 3d features ,so if somehow you can project the 23d data to 3d or 2d then visualization can become easy .

I found that coming up **with assigning weights to words is pretty difficult** as you will be interfering with data with your own intuition.

Also I observed one thing that data preprocessing is important thing in machine learning or the situation is **Garbage in Garbage out**.

B.)Preprocessing on data

There are various preprocessing I learnt ,one hot encoding was used by me ,which is plain assigning 1's and 0's to the word that is present or absent for that particular row ,a more good method would have been assigning weights to words But I was not able to figure it out in programming .

Frequency encoding is also preprocessing but I thought that it won't be that effective as we are interested in the degree to which certain word has effect in a row .

C.)Approaches Used

The following approaches were tried out by me ,I used Scikit learn

1. Neural Network Classifier
   MLPClassifier(solver='adam', activation='relu',alpha=1e-4,early_stopping=True,hidden_layer_sizes=(20,15,10,5,3), random_state=1,learning_rate='adaptive')

   The parameters were tuned and the classifier was trained each time ,activation was changed ,hidden layer size was changed ,the value of alpha was changed .

2. SVC which is support vector classifier

   SVC(kernel='linear',degree=3,C=100,gamma='auto')
   So the kernel was changed to all possible :-linear,sigmoid,poly,rbf(which is default),the degree was changed each time for poly kernel to see which number is the best .Changing the value of C was like imposing a penalty .Svm with linear kernel outperforms others .

3. RandomForest

   RandomForestClassifier(criterion='gini',n_estimators=3000 ,random_state=20,max_features=10,min_samples_split=2)

   Here the estimators means number of decision trees .This approach works best for me followed by neural nets and then SVC.

D.)Results and Final Learning

RandomForest Classifier works best for me . I was able to explore the library of scikit learn which was interesting ,a lot of parameters that were taught in class were to be seen as trainable parameters ,

However training the classifiers to achieve the best result is a cumbersome task ,*I* Did some googling and found out that the parameters are first randomly selected and then the best is found out by cross validation which searches a grid of possible solutions by providing a bound on tunable parameters however this approach is yet to be taught in class and implementing it was also difficult in coding .Also Data preprocessing is important part of machine learning ,so data cleaning must be done before training happens ,Also you need to normalize the dimensions so that one does not end up affecting the result too much ,I also learnt Jupyter notebook usage ,I did not use kaggle since Internet connection was a problem .I am now aware of use of scikit-learn library ,also that pandas is used for data preprocessing .