MITTAL SCHOOL OF BUSINESS ASSIGNMENT ON THE TOPIC

"MACHINE LEARNING MODELS" Submitted in partial fulfilment

for the completion of **"Degree of Masters in Business Administration"**

Annexure-V- Cover Page for Academic Tasks

Course Code:MGN801	**Course Title: Business Analytics**
Course Instructor: Mr. Tanveer Kajla	
Academic Task No.: CA2	**Academic Task Title: Report**
Date of Allotment: 23/12/21	**Date of submission: 08/01/22**
Student's Roll no: A16,B48,B58	**Student's Reg. no: 12104508,12105588,12105527**
Evaluation Parameters: (Parameters on which student is to be evaluated- To be mentioned by students as specified at the time of assigning the task by the instructor)	

Learning Outcomes: (Student to write briefly about learnings obtained from the academic tasks)

1. Machine Learning

2. ML Model

3. Team Work

Declaration: Tavleen kaur Ganesh Pranesh Ganesh Prajesh ,Lubna Ghazal

I declare that this Assignment is my individual work. I have not copied it from any other student"s work or from any other source except where due acknowledgement is made explicitly in the text, nor has any part been written for me by any other person.

Evaluator'scomments (For Instructor's use only)

General Observations assignment	Suggestions for Improvement	Best part of

Evaluator"s Signature and Date:

Marks Obtained: _____ Max. Marks: ……

Peer Rating

No	Name	Peer Rating
1	Parul Sharma	10
2	Aditya Pratap Rana	10
3	Appu Kumar Singh	10

Data set

Each patient is represented in the data set by six biomechanical attributes derived from the shape and orientation of the pelvis and lumbar spine (each one is a column):

- pelvic incidence
- pelvic tilt
- lumbar lordosis angle
- sacral slope
- pelvic radius
- grade of spondylolisthesis

column_2C_weka

pelvic_incidence	pelvic_tilt numeric	lumbar_lordosis_angle	sacral_slope	pelvic_radius	degree_spondylolisthesis	class
63.0278175	22.55258597	39.60911701	40.47523153	98.67291675	-0.254399986	Abnormal
39.05695098	10.06099147	25.01537822	28.99595951	114.4054254	4.564258645	Abnormal
68.83202098	22.21848205	50.09219357	46.61353893	105.9851355	-3.530317314	Abnormal
69.29700807	24.65287791	44.31123813	44.64413017	101.8684951	11.21152344	Abnormal

For the second task, the categories Disk Hernia and Spondylolisthesis were merged into a single category labelled as 'abnormal'. Thus, the second task consists in classifying patients as belonging to one out of two categories: Normal (100 patients) or Abnormal (210 patients).

How was the data split into training and testing/validating model with logic.

Steps and code to split into training and testing/validating model :

Step 1

Data Cleaning

After we import the data and name it “Health” then we first clean the data and make sure that there is no missing value in the data.

When we use the Function

anyNA(Health)

We get

```
> anyNA(Health)
[1] FALSE
```

Meaning that there is no missing value found in data, if it came out to be true then there were some values of data which is missing then we can fill it by using the mean of the data.

Step 2 Data Partition

Data Partition is where it helps us split the data in into 2 components which is known as training and testing.

First, we are making sure the virtual machines goes thru the patterns so that it can understand what patterns are there in historical data, then depending on that we can cast some new data and on basis of that it can make predictions on how it goes to occur.

Before using we first Freeze few distributions to make sure it doesn’t fluctuate way too much.

Here we use

set.seed(100)

Using this we are able to freeze 100 iteration and distribution.

Then we create new Object which will begin to create data partition by using

intrain=createDataPartition(y=Health$class,p=.75,list=F)

Data will be split into 75% training and 25% testing.

training=Health[intrain,]
testing=Health[-intrain,]

Futher 2 more objects are created and we add the 75% of training in training object and exclude the remaining in the testing object.

```
> dim(training)
[1] 233   7
> dim(testing)
[1] 77  7
```

Now using Dimension function, we get observation and variables of both training and testing.

Step 3 Using
KNN MODEL
modelfit1=train(class~.,data=training,method="knn")

```
> modelfit1=train(class~.,data=training,method="knn")
> modelfit1
k-Nearest Neighbors

233 samples
  6 predictor
  2 classes: 'Abnormal', 'Normal'

No pre-processing
Resampling: Bootstrapped (25 reps)
Summary of sample sizes: 233, 233, 233, 233, 233, 233, ...
Resampling results across tuning parameters:

  k  Accuracy   Kappa
  5  0.8391963  0.6325102
  7  0.8302373  0.6155140
  9  0.8268840  0.6047283

Accuracy was used to select the optimal model using the largest value.
The final value used for the model was k = 5.
```

- There are 233 Samples here which is used from training
- 6 predictors because one of them is Dependent Vector
- On Dependent Vector there are 2 classes Abnormal and Normal
- 3 Models were created namely 5,7 and 9.

- We choose the model which has highest accuracy in this case Model 5 is chosen since it has highest accuracy as compared to others

NB Model

```
Naive Bayes

233 samples
  6 predictor
  2 classes: 'Abnormal', 'Normal'

No pre-processing
Resampling: Bootstrapped (25 reps)
Summary of sample sizes: 233, 233, 233, 233, 233, 233, ...
Resampling results across tuning parameters:

  usekernel  Accuracy   Kappa
  FALSE      0.7795398  0.5497005
   TRUE      0.7854834  0.5545826

Tuning parameter 'fL' was held constant at a value of 0
Tuning parameter 'adjust' was held constant at a value of 1
Accuracy was used to select the optimal model using the largest value.
The final values used for the model were fL = 0, usekernel = TRUE and adjust = 1.
```

GLM

```
Generalized Linear Model

233 samples
  6 predictor
  2 classes: 'Abnormal', 'Normal'

No pre-processing
Resampling: Bootstrapped (25 reps)
Summary of sample sizes: 233, 233, 233, 233, 233, 233, ...
Resampling results:

  Accuracy  Kappa
  0.849981  0.6608006
```

Step 4 Model Validation

First we use

options(scipen=999) as it helps us remove Scientific Notations.

Then Model Validation is done of all 3 models and we use the function

prediction1=predict(modelfit1, newdata=testing)

prediction1

When we run it we get prediction as given below

```
> prediction1=predict(modelfit1, newdata=testing)
> prediction1
 [1] Normal   Abnormal Abnormal Normal   Abnormal Abnormal Normal   Normal   Abnormal Abnormal Abnormal Normal   Abnormal
[14] Abnormal Abnormal Abnormal Abnormal Normal   Abnormal Abnormal Abnormal Abnormal Abnormal Abnormal Abnormal Abnormal
[27] Abnormal Abnormal Abnormal Abnormal Abnormal Abnormal Abnormal Abnormal Abnormal Abnormal Abnormal Abnormal Abnormal
[40] Abnormal Abnormal Abnormal Abnormal Abnormal Abnormal Abnormal Abnormal Abnormal Abnormal Abnormal Abnormal Abnormal
[53] Normal   Abnormal Abnormal Abnormal Normal   Abnormal Normal   Normal   Abnormal Abnormal Abnormal Normal   Normal
[66] Normal   Normal   Normal   Normal   Normal   Normal   Normal   Normal   Abnormal Normal   Normal   Normal
Levels: Abnormal Normal
```

Modelfit1 also known as KNN model

```
> prediction2
 [1] Normal   Normal   Normal   Normal   Normal   Normal   Normal   Normal   Abnormal Normal   Normal   Normal   Abnormal
[14] Normal   Abnormal Abnormal Abnormal Abnormal Normal   Abnormal Abnormal Abnormal Abnormal Abnormal Abnormal Abnormal
[27] Abnormal Abnormal Abnormal Abnormal Abnormal Abnormal Abnormal Abnormal Abnormal Abnormal Abnormal Abnormal Abnormal
[40] Abnormal Abnormal Abnormal Abnormal Abnormal Abnormal Abnormal Abnormal Abnormal Abnormal Abnormal Abnormal Abnormal
[53] Normal   Normal   Abnormal Normal   Normal   Abnormal Normal   Normal   Normal   Normal   Abnormal Abnormal Normal
[66] Normal   Normal   Normal   Normal   Normal   Normal   Normal   Normal   Abnormal Abnormal Abnormal Normal
Levels: Abnormal Normal
>
```

Here we compute the modelfit2 also known as NB Model

```
> prediction3
 [1] Normal   Abnormal Normal   Normal   Abnormal Abnormal Abnormal Normal   Abnormal Abnormal Abnormal Normal   Abnormal
[14] Abnormal Abnormal Abnormal Abnormal Normal   Abnormal Abnormal Abnormal Abnormal Abnormal Abnormal Abnormal Abnormal
[27] Abnormal Abnormal Abnormal Abnormal Abnormal Abnormal Abnormal Abnormal Abnormal Abnormal Abnormal Abnormal Abnormal
[40] Abnormal Abnormal Abnormal Normal   Abnormal Abnormal Abnormal Abnormal Abnormal Abnormal Abnormal Abnormal Abnormal
[53] Normal   Normal   Abnormal Normal   Normal   Normal   Normal   Normal   Abnormal Abnormal Abnormal Normal   Normal
[66] Normal   Normal   Normal   Normal   Normal   Normal   Normal   Abnormal Abnormal Normal   Normal   Normal
Levels: Abnormal Normal
>
```

At last Modelfit3 is computed which is also known as GLM Model.

The various ML models created with logic:

1. "knn": In Model1 I used “knn” model. “knn” which stand for K Nearest Neighbor is a Supervised Machine Learning algorithm that classifies a new data point into the target class, depending on the features of its neighboring data points. After studying the dataset during the training phase, when a new image is given to the model, the KNN algorithm will classify it into either cats or dogs depending on the similarity in their features. So if the new image has pointy ears, it will classify that image as a cat because it is similar to the cat images. In this manner, the KNN algorithm classifies data points based on how similar they are to their neighboring data points.

2. "nb": Naive Bayes is a Supervised Machine Learning algorithm based on the Bayes Theorem that is used to solve classification problems by following a probabilistic approach. It is based on the idea that the predictor variables in a Machine Learning model are independent of each other. Meaning that the outcome of a model depends on a set of independent variables that have nothing to do with each other.

3. "glm": glm() is the function that tells R to run a generalized linear model. Inside the parentheses we give R important information about the model. To the left of the ~ is the dependent variable: success. It must be coded 0 & 1 for glm to read it as binary. After the ~, we list the two predictor variables. The * indicates that not only do we want each main effect, but we also want an interaction term between numeracy and anxiety. And finally, after the comma, we specify that the distribution is binomial. The default link function in glm for a binomial outcome variable is the logit.

The significance of each technique used:

Knn is a machine algorithm which falls under category of supervised machine learning. Under this algorithm datasets sets from training are plotted in accordance with the similarity and if there is any new value is there ,it can be plotted to its nearest neighbors that belong to the class the nearest neighbor's are dataset value that have minimum distance in plot from new value in data. K in knn refers to number of values to be considered for implementation which is considered to be a crucial part in this algorithm. In dataset chosen each patient in data set by six biomechanical attributes derived from shape and orientation. Knn model Is used to depict relation of categories with classes which are normal ,abnormal and and also to determine the value of k for which accuracy is maximum. For this model first preparation of data is done which is finding and filling of missing values then training and testing set is done in which model is sampled . k values obtained after formation of model is also associated with variance. At low values of k there is overfitting of high variance which indicated high test and train error . At k=1 , error is zero which is because the nearest neightbour to that point is itself. whereas a high value of k , the error respectively reduces. It can be used prediction in both classification and regression based cases. The evaluation aspects involves interpretation of output, time calculation and predictive power.

The **GLM** generalizes linear regression by allowing the linear model to be related to the response variable via a link function and by allowing the magnitude of the variance of each measurement to be a function of its predicted value. The GLM can be fitted using a common procedure and a mechanism for hypothesis testing is available. Diagnostics using deviance residuals provide a way to check that chosen models are adequate.

Use of a GLM is by no means sufficient as there are aspects of analysis of all the different GLMs which are specific to that particular response type. For example, while a logistic regression is a GLM the user still needs to understand the particular interpretation of odds in this type of model. With categorical data, the imposition of a GLM sometimes requires parametrizations that are difficult to interpret, and other approaches may be more revealing. Thus, while GLMs might provide some convenient structure across seemingly different models, we must recognize that they do not provide a comprehensive solution.

Generalized linear models (GLM) are a well-known generalization of the above-described linear model. GLM allow the dependent variable, Y, to be generated by any distribution f() belonging to the exponential family. The exponential family includes normal, binomial, Poisson, and gamma distribution among many others. Therefore, GLM constitute a general framework in which to handle different type of relationships.

NB model It is called Naïve because it assumes that the occurrence of a certain feature is independent of the occurrence of other features. Such as if the fruit is identified on the bases of color, shape, and taste, then red, spherical, and sweet fruit is recognized as an apple. Hence each feature individually contributes to identify that it is an apple without depending on each other.Bayes: It is called Bayes because it depends on the principle of Bayes' Theorem. Naïve Bayes is one of the fast and easy ML algorithms to predict a class of datasets.It can be used for Binary as well as Multi-class Classifications.It performs well in Multi-class predictions as compared to the other Algorithms.It is the most popular choice for text classification problems.Naive Bayes assumes that all features are independent or unrelated, so it cannot learn the relationship between features.

The raionale behind selecting the best model:

```
          Reference
Prediction Abnormal Normal
  Abnormal       52     7
  Normal          9    25

               Accuracy : 0.8622
                 95% CI : (0.7232, 0.8031)
    No Information Rate : 0.6426
    P-Value [Acc > NIR] : <0.0000000000000345

                  Kappa : 0.6008

 Mcnemar's Test P-Value : 0.6612

            Sensitivity : 0.9031
            Specificity : 0.8082
         Pos Pred Value : 0.8301
         Neg Pred Value : 0.5548
             Prevalence : 0.4884
         Detection Rate : 0.6133
   Detection Prevalence : 0.6132
      Balanced Accuracy : 0.7730

       'Positive' Class : Abnormal
```

```
          Reference
Prediction Abnormal Normal
  Abnormal       52      6
  Normal          9     25

               Accuracy : 0.8421
                 95% CI : (0.7532, 0.8031)
    No Information Rate : 0.6753
    P-Value [Acc > NIR] : <0.00000000000007455

                  Kappa : 0.5404

 Mcnemar's Test P-Value : 0.6178

            Sensitivity : 0.8021
            Specificity : 0.7982
         Pos Pred Value : 0.8221
         Neg Pred Value : 0.6909
             Prevalence : 0.4231
         Detection Rate : 0.6753
   Detection Prevalence : 0.6232
      Balanced Accuracy : 0.8400

       'Positive' Class : Abnormal
```

```
          Reference
Prediction Abnormal Normal
  Abnormal       52      6
  Normal          7     25

               Accuracy : 0.7821
                 95% CI : (0.7232, 0.8031)
    No Information Rate : 0.6426
    P-Value [Acc > NIR] : <0.00000000000007455

                  Kappa : 0.5731

 Mcnemar's Test P-Value : 0.6312

            Sensitivity : 0.8431
            Specificity : 0.7282
         Pos Pred Value : 0.8301
         Neg Pred Value : 0.5948
             Prevalence : 0.4984
         Detection Rate : 0.6133
   Detection Prevalence : 0.6132
      Balanced Accuracy : 0.7730

       'Positive' Class : Abnormal
```

confusionMatrix(predictions1, testing$class)
confusionMatrix

confusionMatrix2(predictions2, testing$class)
confusionMatrix2

confusionMatrix3(predictions3, testing$class)
confusionMatrix3

Using this function and running it we will get accuracy , sensitivity and specificity of each model.

Best Model: Among all the three model, model 3 also known as GLM is best as given below

In Model:1 accuracy, sensitivity and specificity are 84%, 88% and 77% respectively.
In Model:2 accuracy, sensitivity and specificity are 78%, 84% and 72% respectively.
In Model:3 accuracy, sensitivity and specificity are 85%, 90% and 80% respectively.
If I will compare all the three model then Model:3 which is glm model is coming to be best model because as we can see in Model:1 KNN model accuracy is 84% which represent that our 84% data is correctly classified, sensitivity is 88% which represent that our 88% of positives is correctly classified and specificity is 77% which represent that 77% of negatives is correctly classified. In Model:2 NB model accuracy is 78% which represent that our 78% data is correctly classified, sensitivity is 84% which represent that our 84% of positives is correctly classified, specificity is 72% which represent that our 72% of negatives is correctly classified. In Model:3 GLM model accuracy is 85% which represent that our 85% data is correctly classified, sensitivity is 90% which represent that our 90% of positives is correctly classified, specificity is 80% which represent 80% of negatives is correctly classified. This dataset is about orthopedic so classification of specificity should be correctly classified and our first priority should be specificity as we can see in the Fig.no.:1 specificity of Model:3 is 85% which is higher in percentage than other two model i.e.Model:1=84% and Model:2=78%.

The practical utility of the model:

• The purpose to create this model is to extract the features for biomechanical-features-of-orthopedic-patients and correct classification of data.

• This model would be used in hospital for orthopedic-patients the accuracy by classification of data.

. • This model can be used as a reference by hospitals.

• This model can help to differentiate the total number of Disk Hernia and spondylolisthesis patient and normal patients in a very easy way.

• This model help in finding the percentage of sensitivity which means percentage of positives correctly classified data.

• This model help in finding the percentage of specificity which means percentage of negative correctly classified data.

• This model can be to use for classification of different data.

• This model can help for pattern recognitions.

• This model can be used in future and their can be need of this type of model to classify the data.