|  |
| --- |
| 1. Find the best performing features using feature extraction in scikit Learn. (10) 2. Standardize your features to Gaussian distribution. (5)   patients\_tobeScaled = patients.drop(['Diagnosis'],axis=1) scaler = preprocessing.StandardScaler() |
| C. Split the dataset into 60/40 training and testing. (5)  y = patients.Diagnosis x\_train,x\_test,y\_train,y\_test = train\_test\_split(patients\_scaled,y,test\_ size=0.4,random\_state=42, stratify=y)   1. Create a logistic regression model (call it LRM1) using your best features. Describe   your model. (10)  # First drop the least important attribute you identified in your feature selection. Replace the “FeatureName” with the attribute you want to exclude x\_train = x\_train.drop(['FeatureName'],axis=1) x\_test = x\_test.drop(['FeatureName'],axis=1)  #Create the first logistic regression model LRM1 = LogisticRegression() LRM1.fit(x\_train,y\_train)   1. Create classification report of your model. (5)   predictions\_LRM1 = LRM1.predict(x\_test)  print(classification\_report(y\_test, predictions\_LRM1))   1. Describe your classification report (precision, recall, F1 score, and support). (10) 2. Create the accuracy score of your model. Describe the accuracy score. (10) 3. Create another logistic regression model (call it LRM2). Use all the independent   features this time (instead of your best performing features). (10)   1. Compare the two models (LRM1 and LRM2) based on the classification report and   accuracy score. Which one is a better model? Why? (10)   1. Create a Naïve Bayes model (call it NBM) using 60/40 split. (10) 2. Create classification report of your NBM model. (5) 3. Describe your classification report of NBM (precision, recall, F1 score, and support).   (5)   1. Create the accuracy score of your NBM model. Describe the accuracy score. (10) 2. Compare the logistic regression (LRM1 or LRM2) with the Naïve Bayes model   (NBM). Which one is better? Why? (10)   1. What would be your suggestions for further improving the accuracy of your chosen   model? (10) |

page1image7890880

from sklearn.ensemble import ExtraTreesClassifier X = patients.drop(['Diagnosis'],axis=1)  
Y = patients.Diagnosis  
model = ExtraTreesClassifier()

model.fit(X, Y)  
# Print the relative importance of each feature print(model.feature\_importances\_) print(patients.columns)