# UKS31744.docx

*by*

**Submission date:** 29-Apr-2023 03:05AM (UTC-0500)
**Submission ID:** 2079069418
**File name:** UKS31744.docx (1.36M)
**Word count:** 3558
**Character count:** 20040

# DATA MINING THE HEALTHCARE DATASET

## Abstract

In this paper, a health-related dataset of 1584 observations and 26 variables, including patients' medical conditions and risk kinds, has become analyzed. For identifying patterns and relationships among characteristics, descriptive analysis and a correlation heatmap visualization are employed. The dataset has been preprocessed and then divided into test and training sets, which are utilized to train a KNN model. The study demonstrates the potential benefits of machine learning in forecasting people's health statuses by identifying those who are at risk of getting particular health issues or illnesses.

# Table of Contents

1.0 Introduction ..... 4

2.0 Summary of features ..... 4

3.0 Data Pre-processing ..... 7

4.0 Supervised Model Training, Evaluation, and submitting the predicted labels ..... 8

5.0 Unsupervised Clustering ..... 14

6.0 Summary of findings ..... 15

7.0 Conclusion ..... 16

References ..... 17

## 1.0 Introduction

In the case of the healthcare profession, mining of data has become an increasingly important tool for finding useful methods in large and useful datasets. In this context, the healthcare dataset that is used for the analysis has 1584 observations and 26 variables that represent a diverse range of clinical and patient characteristics. Additionally, predictions based on the dataset have been made using three distinct machine learning models, KNN, Random Forest, and SVM. From straightforward decision trees to more intricate algorithms that can uncover hidden relationships between variables, these models offer a variety of predictive capabilities. Besides, the jupyter notebook platform has been considered for this task.

## 2.0 Summary of features

**Load health related datasets**

```
In [2]: health_train_db= lds.read_csv('D:/Dataset/health_train.csv')
        health_test_db = lds.read_csv('D:/Dataset/health_test.csv')

In [4]: health_train_db.head(10)
```

Out[4]:

| | id | x1 | x2 | x3 | x4 | x5 | x6 | x7 | x8 | x9 | ... | x16 | x17 | x18 | x19 | x20 | x21 | x22 | x23 | x24 | target |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | PA1001 | 1406 | 145.0 | F | 0.005 | 0.000 | 0.002 | 0.000 | 0.0 | 0.0 | ... | 104.0 | 171.0 | 4.0 | 0.0 | 155.0 | 153.0 | 154.0 | 4.0 | 1.0 | Low risk |
| 1 | PA1002 | 258 | 127.0 | M | 0.012 | 0.000 | 0.008 | 0.004 | 0.0 | 0.0 | ... | 53.0 | 191.0 | 12.0 | 1.0 | 133.0 | 126.0 | 131.0 | 41.0 | 0.0 | Low risk |
| 2 | PA1003 | 479 | 145.0 | F | 0.000 | 0.000 | 0.000 | 0.002 | 0.0 | 0.0 | ... | 111.0 | 157.0 | 1.0 | 1.0 | 150.0 | 146.0 | 149.0 | 6.0 | 1.0 | Low risk |
| 3 | PA1004 | 906 | 146.0 | F | 0.004 | 0.000 | 0.005 | 0.003 | 0.0 | 0.0 | ... | 107.0 | 169.0 | 2.0 | 2.0 | 150.0 | 147.0 | 149.0 | 7.0 | 0.0 | Low risk |
| 4 | PA1005 | 1921 | 140.0 | F | 0.002 | 0.003 | 0.006 | 0.006 | 0.0 | 0.0 | ... | 75.0 | 228.0 | 9.0 | 0.0 | 142.0 | 118.0 | 142.0 | 20.0 | 0.0 | Low risk |
| 5 | PA1006 | 70 | 144.0 | F | 0.001 | 0.000 | 0.005 | 0.000 | 0.0 | 0.0 | ... | 138.0 | 168.0 | 3.0 | 0.0 | 162.0 | 157.0 | 160.0 | 5.0 | 1.0 | Low risk |
| 6 | PA1007 | 1702 | 137.0 | F | 0.006 | 0.002 | 0.005 | 0.002 | 0.0 | 0.0 | ... | 69.0 | 178.0 | 3.0 | 1.0 | 148.0 | 143.0 | 149.0 | 43.0 | 1.0 | Low risk |
| 7 | PA1008 | 712 | 126.0 | M | 0.005 | 0.003 | 0.005 | 0.000 | 0.0 | 0.0 | ... | 50.0 | 150.0 | 4.0 | 0.0 | 133.0 | 130.0 | 132.0 | 10.0 | 1.0 | Low risk |
| 8 | PA1009 | 1734 | 134.0 | F | 0.008 | 0.001 | 0.007 | 0.004 | 0.0 | 0.0 | ... | 80.0 | 189.0 | 6.0 | 0.0 | 150.0 | 146.0 | 150.0 | 33.0 | 0.0 | Low risk |
| 9 | PA1010 | 1147 | 122.0 | M | 0.000 | 0.000 | 0.006 | 0.009 | 0.0 | 0.0 | ... | 91.0 | 135.0 | 4.0 | 0.0 | 126.0 | 117.0 | 121.0 | 19.0 | 1.0 | Low risk |

10 rows × 26 columns

**Figure 1: import and view health_train dataset**

(Source: Self-Created)

The above dataset is about the health condition of a patient, which also included risk type and other medical conditions of patients. The ID is unique for each patient and is a benefit to identify each patient with the particular ID. In this dataset total 1584 observations with patients takes place. There are a total 26 variables included in this dataset for this 1584 observation, that is included patients id, patients risk type like (Low risk, high risk, Moderate risk). In this case some personal information of the patients are also involved, that is supposed to identify the particular patients.

```
In [5]: col_names = list(health_train_db)
        print(col_names)

['id', 'x1', 'x2', 'x3', 'x4', 'x5', 'x6', 'x7', 'x8', 'x9', 'x10', 'x11', 'x12', 'x13', 'x14', 'x15', 'x16', 'x17', 'x18', 'x1
9', 'x20', 'x21', 'x22', 'x23', 'x24', 'target']

In [6]: col_names = list(health_test_db)
        print(col_names)

['id', 'x1', 'x2', 'x3', 'x4', 'x5', 'x6', 'x7', 'x8', 'x9', 'x10', 'x11', 'x12', 'x13', 'x14', 'x15', 'x16', 'x17', 'x18', 'x1
9', 'x20', 'x21', 'x22', 'x23', 'x24']

In [7]: # information about the dataframe
        health_train_db.info(verbose=True, null_counts=True)

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1584 entries, 0 to 1583
Data columns (total 26 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   id      1584 non-null   object
 1   x1      1584 non-null   int64
 2   x2      1584 non-null   float64
 3   x3      1584 non-null   object
 4   x4      1584 non-null   float64
 5   x5      1540 non-null   float64
 6   x6      1584 non-null   float64
 7   x7      1584 non-null   float64
 8   x8      1567 non-null   float64
 9   x9      1584 non-null   float64
 10  x10     1557 non-null   float64
 11  x11     1584 non-null   float64
 12  x12     1584 non-null   float64
 13  x13     1584 non-null   float64
 14  x14     1584 non-null   object
```

**Figure 2: View health_train dataset details**

(Source: Self-Created)

This information is about the reading of the particular dataset, in this case load of the dataset is also performed with the help of pandas Library. In this dataset total 26 variables are placed with the medical information of patients. It also visualized the variables in the above image, it is clear that in this dataset null values are not placed. The info () method helps to provide the dataset details, such as total columns, data types, range index values and others for which users can better understand the data and evaluate the task more accurately (Obaid, Dheyab and Sabry, 2019).

```
In [8]: health_train_db.describe()
```

Out[8]:

| | x1 | x2 | x4 | x5 | x6 | x7 | x8 | x9 | x10 | x11 | ... | x15 | 15 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| count | 1584.000000 | 1584.000000 | 1584.000000 | 1540.000000 | 1584.000000 | 1584.000000 | 1567.000000 | 1584.000000 | 1557.000000 | 1584.000000 | ... | 1584.000000 | 15 |
| mean | 1053.188131 | 133.297980 | 0.003169 | 0.009906 | 0.004347 | 0.001854 | 0.000003 | 0.000157 | 47.094412 | 1.337816 | ... | 70.409091 | |
| std | 615.996716 | 10.002632 | 0.003821 | 0.048627 | 0.002948 | 0.002940 | 0.000050 | 0.000593 | 17.269621 | 0.899092 | ... | 38.993892 | |
| min | 0.000000 | 106.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 13.000000 | 0.200000 | ... | 3.000000 | |
| 25% | 523.750000 | 126.000000 | 0.000000 | 0.000000 | 0.002000 | 0.000000 | 0.000000 | 0.000000 | 32.000000 | 0.700000 | ... | 36.000000 | |
| 50% | 1049.500000 | 133.000000 | 0.002000 | 0.000000 | 0.004000 | 0.000000 | 0.000000 | 0.000000 | 49.000000 | 1.200000 | ... | 68.000000 | |
| 75% | 1583.250000 | 141.000000 | 0.006000 | 0.003000 | 0.007000 | 0.003000 | 0.000000 | 0.000000 | 61.000000 | 1.700000 | ... | 100.000000 | 1 |
| max | 2125.000000 | 160.000000 | 0.019000 | 0.477000 | 0.014000 | 0.015000 | 0.001000 | 0.005000 | 86.000000 | 7.000000 | ... | 176.000000 | 1 |

8 rows × 22 columns

**Figure 3: Summary of the dataset**

(Source: Self-Created)

In the Summary table of the dataset, it is about the descriptive analysis of the dataset, where data cunt, Mean, standard deviation, min, max values are identified. In this case the mean value of the first variable is 1053.188, Std is 615.99 min value is 0 and max value is 2125 respectively. When used to an overall dataset, statistical analysis, which uses measures like average, standard deviation, as well as and maximum and lowest values, can offer helpful insights into the general wellness state of the group of people under investigation. The distribution of various health-related dataset variables, such as age, BMI, blood pressure, cholesterol levels, and it can be identified with descriptive analysis (Burdack *et al*. 2020). The prevalence of various health conditions and risk factors in the population can be better understood with the help of this data. Outliers and extreme values in the dataset, which may be signs of data errors or unusual health conditions that need further investigation, can be identified with the assistance of descriptive analysis.

```
[9]: plt_grp.figure(figsize=(16,6))
     sns_grp.heatmap(health_train_db.corr() ,cmap='gist_heat',annot=True)
```

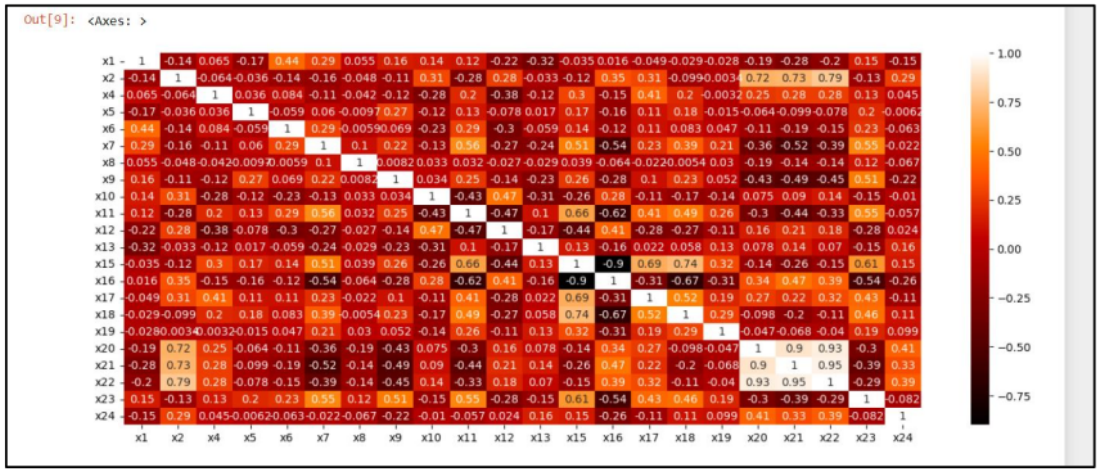



**Figure 4: Generate Heatmap based on the train dataset**

(Source: Self-Created)

The given code generates a heatmap plot of a correlation matrix for a health-related data utilizing Python's seaborn as well as matplotlib modules. The heatmap plot provides an illustration of the health-related dataset's correlation matrix. Every heatmap square represents the correlation among two variables within the dataset. Every cell's color demonstrates the degree of relationship between the variables, with red getting a strong positive connection, blue representing a strong negative correlation, as well as white representing no association.

This visualization can help researchers uncover patterns and correlations among parameters in the data set. It could prove useful in detecting which factors are strongly related to one another while others have a weak or no relationship. This data is able to inform additional data analysis or modeling (Guo *et al*. 2021). Furthermore, the plot can assist in identifying any potential problems with plurality that might impact the precision and stability of the regression models. In the end, the heatmap plot may be an effective analysis of exploratory data as well as a feature selection tool in health-related information.

## 3.0 Data Pre-processing

**Data Cleaning and Data Pre-processing**

```
In [10]: health_train_db.isnull().sum().sum()
Out[10]: 88

In [11]: health_train2 = health_train_db.dropna()
         health_train2.isnull().sum().sum()
Out[11]: 0

In [12]: # Drop unnecessary column of the dataset
         health_train2 = health_train2.drop('x14', axis=1)

In [13]: # Convert Male and Female to binary Number
         health_train2['x3'] = health_train2['x3'].map({'F': 1, 'M': 0})
         health_train2["target"] = health_train2["target"].map({"Low risk": 3, "Moderate risk": 4, "High risk": 5})

In [14]: health_train2.head()
```

Out[14]:

| | id | x1 | x2 | x3 | x4 | x5 | x6 | x7 | x8 | x9 | ... | x16 | x17 | x18 | x19 | x20 | x21 | x22 | x23 | x24 | target |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | PA1001 | 1406 | 145.0 | 1 | 0.005 | 0.000 | 0.002 | 0.000 | 0.0 | 0.0 | ... | 104.0 | 171.0 | 4.0 | 0.0 | 155.0 | 153.0 | 154.0 | 4.0 | 1.0 | 3 |
| 1 | PA1002 | 258 | 127.0 | 0 | 0.012 | 0.000 | 0.008 | 0.004 | 0.0 | 0.0 | ... | 53.0 | 191.0 | 12.0 | 1.0 | 133.0 | 126.0 | 131.0 | 41.0 | 0.0 | 3 |
| 2 | PA1003 | 479 | 145.0 | 1 | 0.000 | 0.000 | 0.000 | 0.002 | 0.0 | 0.0 | ... | 111.0 | 157.0 | 1.0 | 1.0 | 150.0 | 146.0 | 149.0 | 6.0 | 1.0 | 3 |
| 3 | PA1004 | 906 | 146.0 | 1 | 0.004 | 0.000 | 0.005 | 0.003 | 0.0 | 0.0 | ... | 107.0 | 169.0 | 2.0 | 2.0 | 150.0 | 147.0 | 149.0 | 7.0 | 0.0 | 3 |
| 4 | PA1005 | 1921 | 140.0 | 1 | 0.002 | 0.003 | 0.006 | 0.006 | 0.0 | 0.0 | ... | 75.0 | 228.0 | 9.0 | 0.0 | 142.0 | 118.0 | 142.0 | 20.0 | 0.0 | 3 |

**Figure 5: Cleaning the dataset and pre-process the data before make predictions**

(Source: Self-Created)

The process of data cleaning and data preprocessing is an important part of a data analysis project. The prediction will be successful due to the process of data cleaning in this case.

**Dataset Splitting into train and test format**

```
In [15]: # Split the dataset into features and target
         X_value = health_train2.drop(["id", "target"], axis=1)
         y_value = health_train2["target"]

In [16]: # Splitting dataset on train/test
         X_train, X_test, y_train, y_test = split_process(X_value, y_value, test_size=0.3, random_state=0)
```

**Figure 6: Dataset splitting process**

(Source: Self-Created)

The above codes are utilized to divide a dataset into two sections, one for the features (X_value) along with one for the target variable (y_value). The features are the variables that are entered which are employed to forecast the target variable, that in the present scenario is an individual's health state.

This can be done by employing the split_process function with a test size of 0.3 along with a random state of 0. It implies that 70% of the information will be employed for training the algorithm and 30% would be utilized for evaluating its performance. The model may be trained on the training data before being tested on the test data to determine how well it predicts the health status of new people by dividing the data into test and training sets (Nischitha *et al*. 2020). This assists the healthcare industry in forecasting people's health state by helping them to construct reliable prediction models according to important attributes, which can then be utilized to identify persons at risk of acquiring particular health disorders or diseases.

## 4.0 Supervised Model Training, Evaluation, and submitting the predicted labels

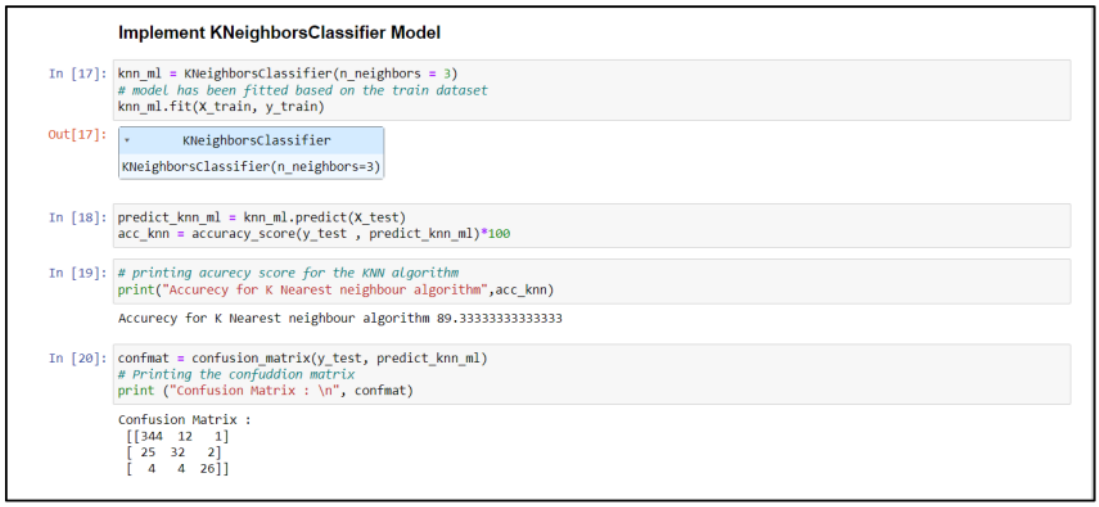


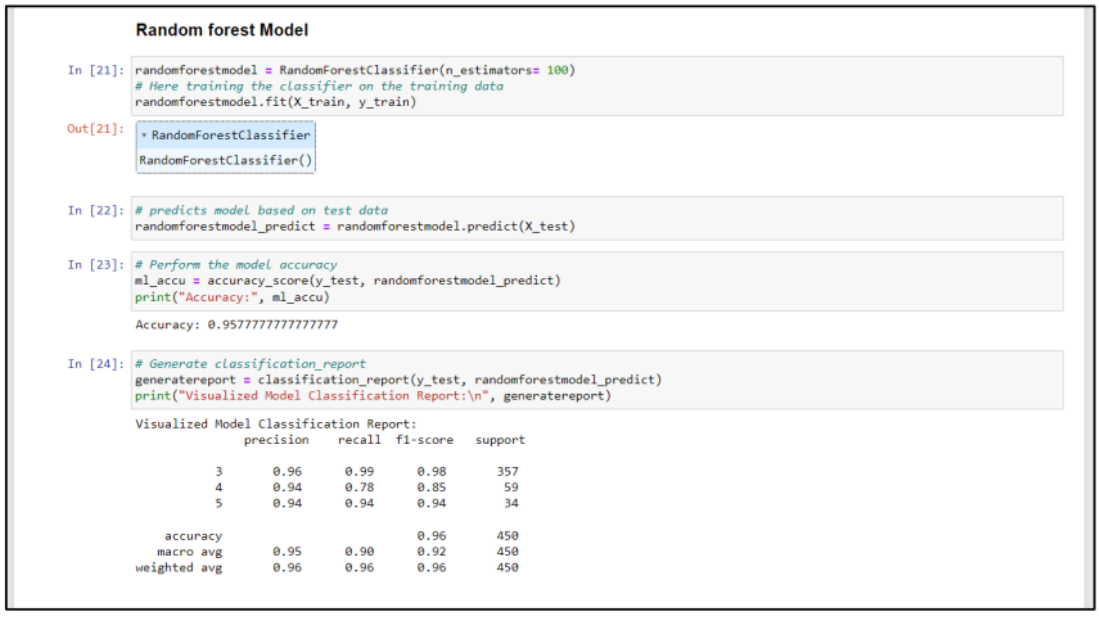

**Figure 7: Execute KNN model**

(Source: Self-Created)

The following codes are utilized here in order to execute the KNN model based on the health sector's train dataset. The "KNeighborsClassifier ()" method helps to build the model successfully and the "fit ()" method provides the accessibility in order to fit the model corresponding to the task (Kohli, Godwin, and Urolagin, 2021). After performing the task, the accuracy score has been

generated as 89% as well as confusion matrix is also shown here in the above image. The confusion matrix comprises data regarding the total number of actual positives, erroneous positives, real negatives, as well as erroneous negatives.

In the medical field, this KNN classifier model might be applied to predict patients' health condition on the basis of their medical histories, lab findings, or other health-related information. The model can categorize patients' health state into distinct risk groups, which healthcare providers can utilize to monitor patients' health, prioritize therapies, and distribute resources appropriately (Nair and Kashyap, 2019). In some cases, healthcare practitioners utilize KNN classifier models in order to estimate a patient's risk of heart disease determined by their medical history, lifestyle habits, and other health-related information. Patients can be classified in different formats, such as "low," "moderate," or "high" risk through the KNN model. This data may be utilized to direct treatment decisions, track the health of the individual, as well as allocate supplies.



**Random forest Model**

In [21]:
```
randomforestmodel = RandomForestClassifier(n_estimators= 100)
# Here training the classifier on the training data
randomforestmodel.fit(X_train, y_train)
```

Out[21]:
```
RandomForestClassifier
RandomForestClassifier()
```

In [22]:
```
# predicts model based on test data
randomforestmodel_predict = randomforestmodel.predict(X_test)
```

In [23]:
```
# Perform the model accuracy
ml_accu = accuracy_score(y_test, randomforestmodel_predict)
print("Accuracy:", ml_accu)
```

```
Accuracy: 0.9577777777777777
```

In [24]:
```
# Generate classification_report
generatereport = classification_report(y_test, randomforestmodel_predict)
print("Visualized Model Classification Report:\n", generatereport)
```

```
Visualized Model Classification Report:
              precision    recall  f1-score   support

           3       0.96      0.99      0.98       357
           4       0.94      0.78      0.85        59
           5       0.94      0.94      0.94        34

    accuracy                           0.96       450
   macro avg       0.95      0.90      0.92       450
weighted avg       0.96      0.96      0.96       450
```

**Figure 8: Random Forest model**

(Source: Self-Created)

The offered code constructs a Random Forest Classifier ML model to forecast the target variable's value according to the input characteristics. Here, the model accuracy is shown as 95% after performing the model successfully. This “accuracy_score ()” function determines the precision of the random forest classification model through a comparison of the label predictions

"(randomforestmodel_predict)" along with the actual labels (y_test). Besides, this "classification_reports ()" function provides the classification report regarding the model's predictions on the tested dataset. The classification analysis includes accuracy, recall, and F1 scores for each class.

In the health industry, the Random Forest Classifier model is extensively used to forecast health outcomes, detect hazards to health, as well as generate treatment regimens for patients. For example, the model may be trained on a dataset comprising data on individuals' health state, medical history, along with other relevant parameters (Liu, Gu and Wang, 2021). Utilizing this data, the model may forecast the likelihood of acquiring particular health issues or the efficacy of certain therapies. Therefore, the offered code uses a Random Forest Classifier model to forecast the target variable according to input data. The model may be trained on health-related data to predict health outcomes, detect risk factors, and build patient treatment strategies. The model's categorization report gives essential information for healthcare providers to make educated decisions about patient treatment.

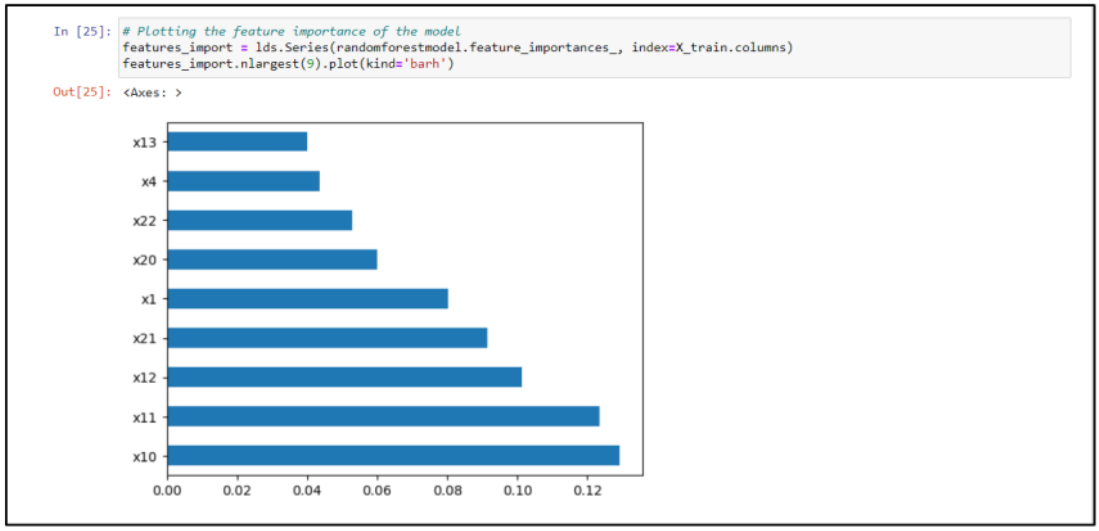



**Figure 9: Plotting the features importance of the model**

(Source: Self-Created)

The code offered plots the feature significance of a random forest model. A predictive machine learning model's feature significance reveals the relative relevance of every characteristic in the algorithm's forecasts. Employing the plot () function, this plot(kind='barh') code makes a horizontal bar chart representing the top nine most essential characteristics. The filter option is set

to 'barh' to generate a horizontal bar chart, and the output is a graphical depiction of the most significant nine most essential feature significance ratings.

In summary, this code is helpful in identifying the most relevant features in a random forest model, that is able to be utilized to get understandings of which characteristics are most crucial in producing the model's predictions (Keerthan Kumar, Shubha and Sushma, 2019). This data may be utilized to impact judgements regarding the components to include or remove from the model, thereby improving its accuracy and efficacy. Furthermore, because the visual representation of the feature's significance scores is simple to comprehend and interpret, this code may prove useful in clarifying the model's forecasts to participants who lack technical expertise.

**Implement Decision Tree Model**

In [26]:
```
# Generate the decision tree classifier
Decision_tree_clasifier_ml = DecisionTreeClassifier()
# train the classifier based on training dataset
Decision_tree_clasifier_ml.fit(X_train, y_train)
```

Out[26]:
```
DecisionTreeClassifier
DecisionTreeClassifier()
```

In [27]:
```
# Predictting model labels for test dataset
Decision_tree_clasifier_ml_predict = Decision_tree_clasifier_ml.predict(X_test)
# generate model accuracy
Decision_tree_clasifier_ml_accuracy = accuracy_score(y_test, Decision_tree_clasifier_ml_predict)
print(f"Model Accuracy is:---- {Decision_tree_clasifier_ml_accuracy}")
```

```
Model Accuracy is:---- 0.92
```

In [28]:
```
# Create a classification report
gen_classification_report = classification_report(y_test, Decision_tree_clasifier_ml_predict)
print(gen_classification_report)
```

```
              precision    recall  f1-score   support

           3       0.95      0.96      0.95       357
           4       0.76      0.71      0.74        59
           5       0.86      0.91      0.89        34

    accuracy                           0.92       450
   macro avg       0.86      0.86      0.86       450
weighted avg       0.92      0.92      0.92       450
```

**Figure 10: Code to execute the Decision tree model**

(Source: Self-Created)

This code generates a Decision Tree Classifier employing a machine learning algorithm. The classifier has been taught on a training set as well as employed to forecast the modeling categories for the dataset being tested. It subsequently evaluates the model's accuracy or generates a categorization report. This assists the health sector in determining an individual's health state by analyzing the attributes specified in the dataset and forecasting the health status as low, moderate, or high. The model accuracy is shown as 92% after performing the train dataset. The classification report is also shown where precision values, recall values and other values are shown.

This model may also be trained using a dataset comprising data pertaining to individuals' medical histories, lab findings, factors affecting lifestyles, along with other health-related factors. The algorithm may then utilize this data to build a decision tree that categorizes patients into various risk groups based on their medical state (Mishra *et al*. 2020). After training, the model may be employed to predict the health condition of new patients to allocate it to the right risk group.

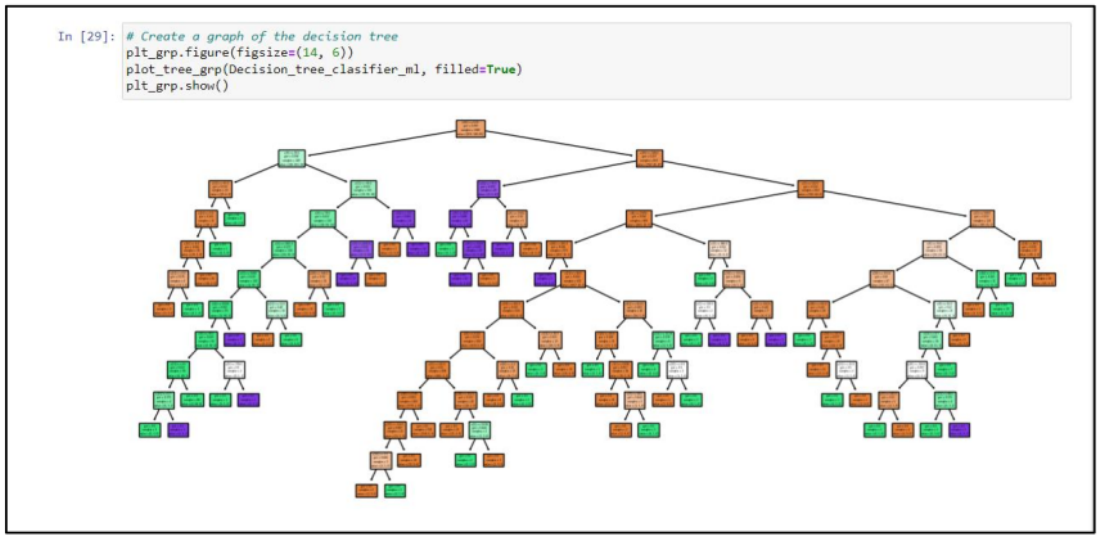



**Figure 11: Decision tree graphs based on the model**

(Source: Self-Created)

The given code is employed to visualize the classifier's decision tree. A decision tree is a visual representation of the standards that the model acquired throughout training. plt_grp.show (): The following line uses the matplotlib library's display() method for showing the graph on the monitor. The decision tree is depicted in the figure, with each node indicating a test of some property or feature of the data as well as the branches reflecting the different results of that testing (Charbuty and Abdulazeez, 2021). The node color represents the majority of the class in that node, with deeper colors reflecting a greater percentage of that class.

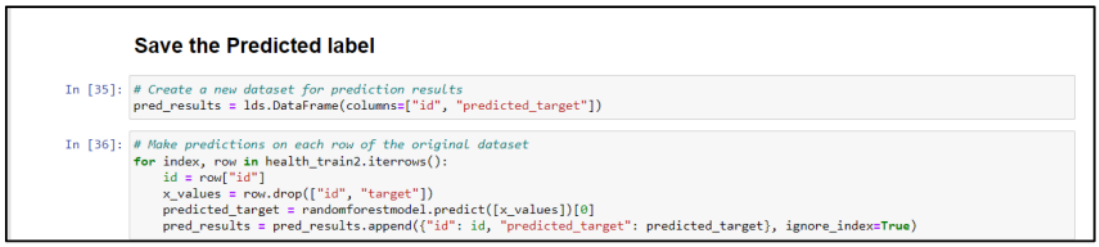
In the health industry, this visualization may be used to evaluate the model's decision-making process and determine which factors or attributes are most relevant in forecasting patients' health state. The decision tree may demonstrate, for example, that a patient's age, gender, and BMI are the most relevant criteria in determining their risk of getting particular health disorders. Healthcare providers can utilize this data to generate tailored interventions and therapies based on patients' particular risk factors.


**Figure 12: Code to perform the SVM model**

(Source: Self-Created)

This code generates a support vector machine (SVM) model and trains it with the training data. The algorithm is subsequently utilized to generate predictions on the test data set, as well as the accuracy of the predictions is calculated. Additionally, it generates a confusion matrix that shows the prediction reliability. In regards to healthcare, such codes can be utilized for predicting a patient's health state according to medical data.

While the data being input could contain information such as the patient's blood pressure, cholesterol levels, and age, SVM model might be trained to classify the patient's health state as low, moderate, or high (Islam *et al*. 2021). Moreover, these details should be utilized through healthcare practitioners in order to monitor their patients' wellness as well as take necessary steps to enhance their well-being. In addition, the confusion matrix could offer data on the accuracy of the SVM model, allowing health care professionals to identify areas for improvement as well as boost the testing strategies.



```
In [37]: # Save the prediction results to a new CSV file
         pred_results.to_csv("D:/task/predictedTarget.csv", index=False)
```

**Figure 13: Code to save the best model to the predicted target dataset**

(Source: Self-Created)

Here, the above codes are utilized here in order to save the predicted target file with respect to the id and target result after performing the random forest model and it shows the high accuracy values with respect to the other machine learning models that are performed based on the dataset.

## 5.0 Unsupervised Clustering

**Unsupervised Clustering**

```
In [38]: from sklearn.cluster import KMeans
         from sklearn.preprocessing import LabelEncoder

In [40]: # Fit the k-means model
         kmeans_clustering_analysis = KMeans(n_clusters=3, random_state=0)
         kmeans_clustering_analysis.fit(X_value)
```

```
Out[40]: KMeans(n_clusters=3, random_state=0)
```

```
In [41]: # Make predictions
         kmeans_clustering_analysis_pred = kmeans_clustering_analysis.predict(X_value)

In [42]: # Scatter plot in order to display the clusters based on the Low risk, Moderate risk, High risk
         plt_grp.scatter(X_value.iloc[:, 0], X_value.iloc[:, 1], c=kmeans_clustering_analysis_pred)
         plt_grp.title("K-Means Clustering for Low risk, Moderate risk, High risk")
         plt_grp.xlabel("Feature 1")
         plt_grp.ylabel("Feature 2")
         plt_grp.show()
```

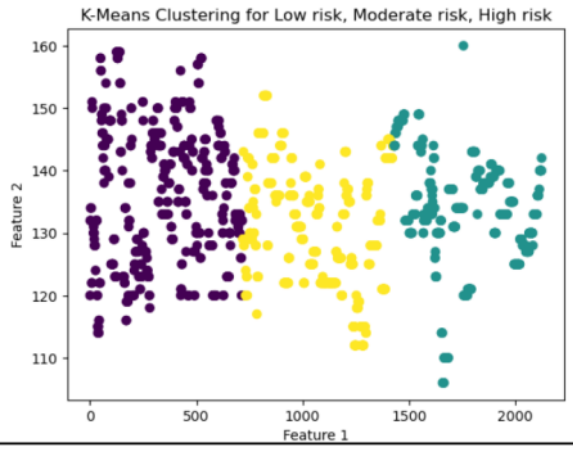



**Figure 14: Python codes to perform the unsupervised clustering analysis**

(Source: Self-Created)

The k-means technique has to be employed by the code above to do a clustering analysis. The initial step is to generate a KMeans object with a specific number of clusters, which is 3 in this case, as well as its random state has been established for repeatability. After that, the fitting method on the KMeans object begins with the data set X_value as input. The cluster centers are determined by fitting the KMeans model to the data. The predict approach has been employed to assign clusters to every data point based on its closeness to the cluster centers. This information is saved in kmeans_clustering_analysis_pred (Raj and Masood, 2020). At last, to visualize the clusters, a scatter plot is constructed with matplotlib. The plot's x- and y-axes correspond to Feature 1 as well as Feature 2, correspondingly. Every point of data on the display is coloured to show the cluster assignment obtained by the KMeans model. The plot's title shows that the clusters represent Low, Moderate, as well as High risk.

The health industry might use k-means clustering to classify patients depending on their health state. As an example, if the two plotted characteristics are linked to certain health measures, the resultant clusters might possibly represent collections of patients with varying risks for illness (e.g., low, moderate, and high risk). This data could subsequently be utilized to improve patient care as well as allocate resources.

## 6.0 Summary of findings

The study utilizing the health_train datasets aim to comprehend patients' health conditions while making predictions about their current state of health. The dataset had 1584 observations as well as 26 variables, covering the patient's ID, risk category, and various medical problems. While being utilized to create predictions, the data received a cleansing and pre-processed (Dhillon and Singh, 2019). The study employed numerous supervised machines learning models, including KNN, Random Forest, Decision Tree, and SVM, to predict patients' health state. The findings indicated that the Random Forest model had the maximum accuracy of 95%. Furthermore, the Random Forest model's feature importance has been displayed in order to comprehend the most significant characteristics in its forecasts.

The k-means approach has additionally been employed for unsupervised clustering in the study. The data served for training the k-means model, which is subsequently utilized for allocating clusters to each data point according to its proximity to the cluster centers. The clusters reflected Low, Moderate, as well as High risk, according to the data. This data might be utilized to enhance

patient care as well as better manage resources. The Random Forest model, considered the best, has been restored to the anticipated target dataset (Qureshi *et al*. 2020). The Random Forest Classifier model got the greatest accuracy score of 95%, then came the Decision Tree model with 92% accuracy as well as the SVM model with a lower accuracy score. The Random Forest model's feature importance has been plotted, and the decision tree graph appears to better comprehend the model's decision-making process. Each information point is assigned towards a cluster according to its proximity to the cluster centers, which indicates low, moderate, and high risk.

Finally, the study demonstrated that the implementation of machine learning models may offer useful information into patients' health conditions and assist healthcare providers in making informed decisions about patient care (Ngiam and Khor, 2019). The study's findings emphasize the significance of data preparation and the application of appropriate models in producing accurate predictions. Unsupervised clustering algorithms may also be employed to categorize patients depending on their health state.

## 7.0 Conclusion

At the end, it can be concluded that the health_train dataset has been analyzed via several machine learning methods, both supervised and unsupervised, providing insights into patients' health status and risk levels. The research highlighted the importance of data pre-processing or cleansing in producing accurate forecasts. While compared to other models, the Random Forest model achieved the greatest accuracy score of 95%, and its feature significance displayed to comprehend the most relevant aspects in its forecasts. The Decision Tree model received a 92% accuracy score, whereas the SVM model had a less favorable accuracy score. The k-means clustering technique has also been used to divide patients into low, moderate, and high-risk groups depending on their health problems. The findings emphasize the need of using AI models in healthcare in order to enhance patient care and efficiently manage resources. The study also sheds light on the process of how healthcare sectors can utilize the data-driven approaches for making more informed decisions as well as enhance patient outcomes.

## References


Burdack, J., Horst, F., Giesselbach, S., Hassan, I., Daffner, S. and Schöllhorn, W.I., 2020. Systematic comparison of the influence of different data preprocessing methods on the performance of gait classifications using machine learning. *Frontiers in bioengineering and biotechnology*, *8*, p.260.

Charbuty, B. and Abdulazeez, A., 2021. Classification based on decision tree algorithm for machine learning. *Journal of Applied Science and Technology Trends*, *2*(01), pp.20-28.

Dhillon, A. and Singh, A., 2019. Machine learning in healthcare data analysis: a survey. *Journal of Biology and Today's World*, *8*(6), pp.1-10.

Guo, W., Che, L., Shahidehpour, M. and Wan, X., 2021. Machine-Learning based methods in short-term load forecasting. *The Electricity Journal*, *34*(1), p.106884.

Islam, N., Rashid, M.M., Wibowo, S., Xu, C.Y., Morshed, A., Wasimi, S.A., Moore, S. and Rahman, S.M., 2021. Early weed detection using image processing and machine learning techniques in an Australian chilli farm. *Agriculture*, *11*(5), p.387.

Keerthan Kumar, T.G., Shubha, C.A. and Sushma, S.A., 2019. Random forest algorithm for soil fertility prediction and grading using machine learning. *Int J Innov Technol Explor Eng*, *9*(1), pp.1301-1304.

Kohli, S., Godwin, G.T. and Urolagin, S., 2021. Sales prediction using linear and KNN regression. In *Advances in Machine Learning and Computational Intelligence: Proceedings of ICMLCI 2019* (pp. 321-329). Springer Singapore.

Liu, C., Gu, Z. and Wang, J., 2021. A hybrid intrusion detection system based on scalable K-Means+ random forest and deep learning. *Ieee Access*, *9*, pp.75729-75740.

Mishra, S., Mallick, P.K., Tripathy, H.K., Bhoi, A.K. and González-Briones, A., 2020. Performance evaluation of a proposed machine learning model for chronic disease datasets using an integrated attribute evaluator and an improved decision tree classifier. *Applied Sciences*, *10*(22), p.8137.

Nair, P. and Kashyap, I., 2019, February. Hybrid pre-processing technique for handling imbalanced data and detecting outliers for KNN classifier. In *2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon)* (pp. 460-464). IEEE.

Ngiam, K.Y. and Khor, W., 2019. Big data and machine learning algorithms for health-care delivery. *The Lancet Oncology*, *20*(5), pp.e262-e273.

Nischitha, K., Vishwakarma, D., Ashwini, M.N. and Manjuraju, M.R., 2020. Crop prediction using machine learning approaches. *International Journal of Engineering Research & Technology (IJERT)*, *9*(08), pp.23-26.

Obaid, H.S., Dheyab, S.A. and Sabry, S.S., 2019, March. The impact of data pre-processing techniques and dimensionality reduction on the accuracy of machine learning. In *2019 9th annual information technology, electromechanical engineering and microelectronics conference (iemecon)* (pp. 279-283). IEEE.

Qureshi, K.N., Din, S., Jeon, G. and Piccialli, F., 2020. An accurate and dynamic predictive model for a smart M-Health system using machine learning. *Information Sciences*, *538*, pp.486-502.

Raj, S. and Masood, S., 2020. Analysis and detection of autism spectrum disorder using machine learning techniques. *Procedia Computer Science*, *167*, pp.994-1004.

# UKS31744.docx

ORIGINALITY REPORT

| 3% | 3% | % | 0% |
|---|---|---|---|
| SIMILARITY INDEX | INTERNET SOURCES | PUBLICATIONS | STUDENT PAPERS |

PRIMARY SOURCES

| | | |
|---|---|---|
| 1 | doctorpenguin.com<br>Internet Source | <1% |
| 2 | www.researchgate.net<br>Internet Source | <1% |
| 3 | digital.library.unt.edu<br>Internet Source | <1% |
| 4 | Submitted to Swinburne University of Technology<br>Student Paper | <1% |
| 5 | bspace.buid.ac.ae<br>Internet Source | <1% |
| 6 | hdl.handle.net<br>Internet Source | <1% |
| 7 | link.springer.com<br>Internet Source | <1% |
| 8 | www.mdpi.com<br>Internet Source | <1% |

| | | | |
|---|---|---|---|
| Exclude quotes | On | Exclude matches | Off |
| Exclude bibliography | On | | |