Cardiovascular Disease Prediction Using Logistic Regression and K-Nearest Neighbours

ECS784P - Data Analytics

Submission Date:
Wednesday, April 7th, 2021

Student:
Erfan Divanizadeh

Table of Contents

1 Introduction

1.1 Project Definition

Data science has been a driving force in the advancement of medicine and medical practices; one such advancement includes the use of prediction algorithms to identify diseases or diagnose patients. This project will utilise such algorithms, particularly two classification algorithms, to determine whether a patient is diagnosed with a cardiovascular disease given a set of input parameters. Using K-Nearest Neighbours and Logistic Regression algorithms, this project will produce two sets of predictions that prove useful in real-life applications at industry- or research-level. While more research and optimisation techniques are required to further enhance prediction accuracy, we attempt to build a strong foundation for further works to use as reference. This report will describe the dataset and the algorithms used as well as analyse the findings of each algorithm to find the optimal classification for cardiovascular disease prediction.

Cardiovascular disease is a broad term used to describe various conditions affecting the heart or blood vessels [1] such as heart disease, heart attack, congenital heart disease, stroke, and so on. Patients with high risk conditions such as high blood pressure or high cholesterol are at high risk of having a cardiovascular disease [2] - these diseases are responsible for a quarter of premature deaths in the UK and are a significant cause of various disabilities [1].

1.2 Project Objectives

The aim of this project is to predict whether a patient is diagnosed with a cardiovascular disease or not and see how a model can be tuned in order to perform better in a critical scenario. To achieve this aim, we propose using two classification algorithms on a preprocessed dataset. The results of the two algorithms will then be compared and used in order to analyse how suitable each of these is and whether or not it should be used in the context of medical diagnosis.

2 Related Work

The first related work that will be discussed is a report titled 'Developing prediction models for clinical use using logistic regression' by Maren E. Shipe, Stephen E. Deppen, Farhood Farjah and Eric L. Grogan [4]. The aim of this work was to discuss assisting healthcare professionals and patients using prediction models using logistic regression. As seen in [4, p. 1], it is discussed that the authors will "describe a set of guidelines for clinicians to use to develop logistic regression-based prediction models for binary outcomes that are intended to augment clinical decision making". There are 6 different models designed using logistic regression. Each of the models is designed for a different outcome and for a specific population and there is a set of instructions for all models on how to develop these prediction models for clinical use. In addition to this, there is a set of instructions of how to evaluate the model performance and model validity. Finally, it is discussed in the report that it is important to conduct an impact

assessment on the healthcare professionals and/or individual patients with instructions on how to do so.

The second related work that will be discussed is a report titled 'Predicting heart diseases in logistic regression of machine learning algorithms by python jupytelab' by A. S. Thanuja Nishadi [5]. The aim of this study is to "identify the most significant predictors of heart diseases and predicting the overall risks by using logistic regression" [5, p. 1]. Using a relevant dataset from the Kaggle website, this author uses logistic regression to satisfy the aim of the study. As seen on [5, p. 2], "the dataset consists of 14 IVS and predicted value". The 14 IVS are used to evaluate the risk of coronary heart disease for the individuals in the dataset.

The third related work that will be discussed is an article titled 'Logistic regression was as good as machine learning for predicting major chronic diseases' [6] by S. Nusinovici et al. The aim of this article is to evaluate the performance of logistic regression against the performance of 5 different ML models. The five ML models that were considered are single-hidden-layer neural network, support vector machine, random forest, gradient boosting machine, and k-nearest neighbor [6]. The results show that logistic regression performs as well as ML models to predict the risk of major chronic diseases.

The fourth related work that will be discussed is an article titled 'RKNNMDA: Ranking-based KNN for MiRNA-Disease Association prediction' by Xing Chen, Qiao-Feng Wu and Gui-Ying Yan [7]. The aim of this study is to predict the micro-RNA (miRNA) relation to diseases using Ranking-based k -Nearest Neighbors (KNN). The study found that using KNN and leave-one-out cross validation yielded performance with area under ROC (AUC) = 0.8221 indicating reliable performance [7]. The study has also proven that KNN would be a great use for miRNA-disease association.

The fifth related work that will be discussed is an article titled 'Efficient heart disease prediction system using K-nearest neighbor classification technique' by Nida Khateeb and Muhammad Usman [8]. The aim of this study is to provide high accuracy for predicting heart disease. After different Heart Disease Prediction systems have been reviewed. The authors employ K-nearest neighbor classifiers to predict heart disease using 14 attributes. The results show that KNN classifier yielded a performance with 80% accuracy [8].

The final related work that will be discussed is an article titled 'classification of heart disease using K-nearest neighbor and genetic algorithm' by M. Akhil Jabbar, B. L. Deekshatulu, Priti Chandra [9]. The aim of this study is to enhance the accuracy in diagnosis of heart disease using a new algorithm which combines KNN with genetic algorithm (GA). As seen on [9, p. 7], "the results acquired by KNN and GA reveals that by integrating GA with KNN will improve the classification accuracy for many data sets".

3 Data Collection and Representation

3.1 Data Description

The dataset used in this project is provided by Kaggle [3]; this dataset has over 70,000 instances each having 12 attributes - the first 11 attributes can be used to predict the last attribute (target

attribute) which indicates the presence or absence of a cardiovascular disease. There are three types of input features:

- *Objective* - factual information
- *Examination* - results of medical examinations
- *Subjective* - information given by the patient

Moreover, this dataset comprises numerical, binary, and categorical values as shown in table 1 below. For a more detailed view of the dataset, please refer to the dataset sample table in the appendix.

Numerical	Binary	Categorical
Age (years)	Gender (1=Male, 2=Female)	Cholesterol (1=Normal, 2=Above Normal, 3=Well Above Normal)
Height (cm)	Smoke (0=No, 1=Yes) - refers to whether the patient smokes or not	Glucose (1=Normal, 2=Above Normal, 3=Well Above Normal)
Weight (kg)	Alcohol (0=No, 1=Yes) - refers to whether the patient drinks or not	
ap_hi (Systolic Blood Pressure)	Active (0=No, 1=Yes) - refers to whether the patient is active or not	
ap_lo (Diastolic Blood Pressure)	Cardio (0=No, 1=Yes) - refers to whether patient has cardiovascular disease	

Table 1 - Table detailing the types of values in the dataset

3.2 Data Exploration

We started by visualising the dataset and extracting information with the help of graphs . This is necessary in order to understand what feature might give more predictive power to the machine learning algorithms and which features are not as desirable, because the dataset might have focused too much on a specific group of people. For example, the first such finding was the divide between smoker and non smokers (Figure 1) which just by looking at is clear that the non smokers are under-represented and this is something that must be kept in mind throughout the project. Similar findings were for the alcohol consumption and activity both having similar distributions. Age and Weight had a good gaussian distribtuion.

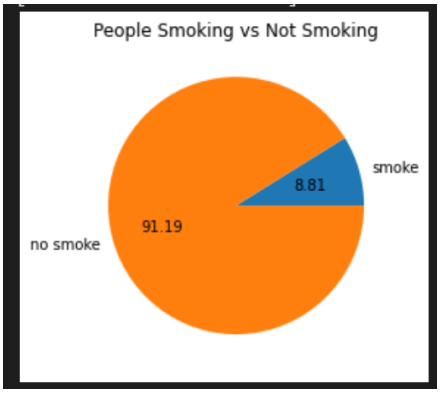


Figure 1 - Not Smoker vs Smoker Division

Looking at the overall split between the two classes of the dataset, it can be seen that the split is roughly equal which tells us that an algorithm such as KNN should not suffer too much because of bias.

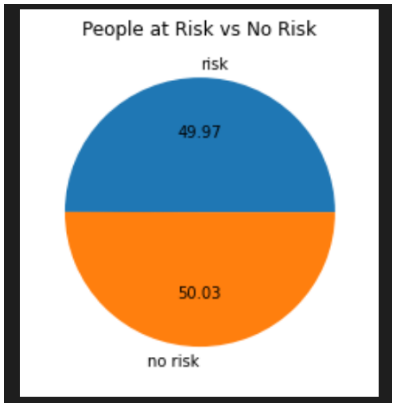
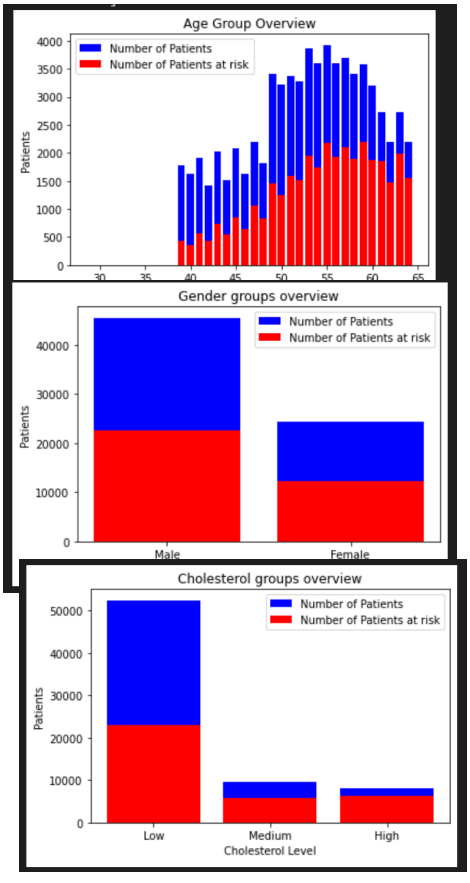


Figure 2 - No Risk vs Risk

After visualising how different features are distributed, we used Python (pandas + matplot) in order to generate graphs which helped extract information such as:



"What age group is at risk?"

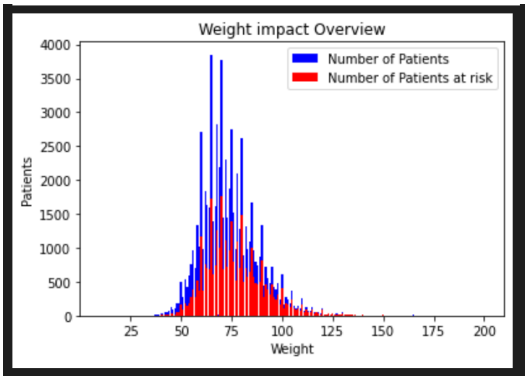
The graph shows how the older a person is the more likely that person will be at risk of having a cardiovascular disease, This results are in line with what we expected since with aging the body becomes weaker however it is also important to note that the higher risk percentage starts at around 50 yrs of age and onwards.

"Are men more likely to be at risk?"

By looking at the graph it is safe to say that the sex of a person does not have much of an impact on the risk factor. However it is also important to note that male group has more samples but the overall split for each seems equal at around 50%

"Does high cholesterol mean high risk?"

As we expected and found out, cholesterol is one of the main factors for a cardiovascular disease. People with a low level are mainly in the no risk category while those with a moderate level are mainly in the risk and those with high are almost all at risk



"Does weight increase the risk?"

As with cholesterol, weight is another known factor of cardiovascular disease so to no surprise the higher the weight, the higher the number of samples who are part of the risk group.

Even if some of the above were expected outcomes, data exploration was a crucial part of the project because not only did it reinforce what we already knew about the topic (in general) but also because by being what we expected it showed us that the data is not made up and it is of decent quality. It also made us aware of the high imbalance in some cases such as smoking groups.

3.3 Feature Selection

As mentioned in the previous section, we expected some features to give less learning power compared to other features which can be caused either by not enough representation within the dataset or by the distribution itself. If the chance of a male being at risk is the same as a female being at risk then for example gender tells us nothing in the context of a machine learning algorithm. This is why after the data exploration where we experimented with all the different features we had to remove some features that would not be valuable during the training phase and might result in poor performance during the testing phase caused by the algorithm focusing too much on learning outliers for example. In order to express this even further and see what effect this will have, we will generate models which use all the features present after pre-processing and models that have feature selection applied. The feature selection algorithm we used is CFS (Correlation Feature Selector) Subset Evaluation which is part of the Weka base package. It works by assessing the learning power which features get while in the same subset as other features, and it will select the subset with the best learning power overall. In our case, it reduced the number of features from 12 to just 5 and by looking at the 5, we can tell that these are in fact related such as blood pressure and cholesterol. Surprisingly weight was not selected but that is because even if it is correlated with cholesterol, might not give as much learning power in the context of the current subset with the other features taken into account. The best subset of features according to the algorithm is [ap_hi,ap_lo,cholesterol,active,age].

```
=== Attribute Selection on all input data ===

Search Method:
        Best first.
        Start set: no attributes
        Search direction: forward
        Stale search after 5 node expansions
        Total number of subsets evaluated: 67
        Merit of best subset found:    0.109

Attribute Subset Evaluator (supervised, Class (nominal): 11 cardio_binarized):
        CFS Subset Evaluator
        Including locally predictive attributes

Selected attributes: 4,5,6,10,12 : 5
                     ap_hi
                     ap_lo
                     cholesterol
                     active_binarized
                     age_yrs
```

Figure 3 - Results of CFS Subset Evaluation

3.4 External Libraries

Table 2 below provides a list of libraries used in this project for various purposes; it contains library names along with a corresponding concise description as well as specific library imports. Besides the mentioned libraries, there is a particular software called Weka that we used for the the feature selection, and the results presentation sections of our project. This software is essentially a collection of machine learning algorithms as it contains tools for preprocessing, classification, visualisation, and more. We used this software as a means to cross-validate our python work to ensure we are achieving our desired results.

Library	Description
csv	This library implements classes to read and write tabular data in CSV (comma separated values) format. CSV's reader and writer objects read and write sequences.
pandas	This library works with tabular data from CSV files. It is often used for data manipulation, analysis, and presentation. It offers operations for manipulating numerical tables.
numpy	This library provides a simple yet powerful data structure often known as the n-dimensional array. In other words, it provides data scientists with multi-dimensional arrays and matrix data structures to be able to perform statistical, algebraic, and trigonometric operations.
math	This library provides programmers with the ability to perform various mathematical functions such as square root, random integer, trigonometric functions, etc.
sk-learn	A very important library, scikit-learn (or sk-learn) is a machine learning library that is widely known and used in the data science field. It is used to implement various ML algorithms in a straightforward manner. Due to the specificity of our project in terms of algorithms of choice, we have imported a few packages necessary (such as metrics, roc_auc_score, and roc_curve) to split our dataset, implement our algorithms, and present results.
matplotlib	This library comprises a collection of data presentation tools. Essentially, matplotlib gives you full control over how your graph/table looks like - you can change labels, choose graph/chart type, colour the labels, choose axis labels, and so much more.
seaborn	This is a visualisation library that yields a high-level interface for drawing exquisite and informative graphs and charts. This is based on matplotlib.

Table 2 - Table detailing the external libraries imported on Python

4 Methodology

Predicting the presence or absence of a cardiovascular disease in a patient is a classification problem. As a result, we will be using the Logistic Regression and the K-Nearest Neighbour classification algorithms. While there are advantages and disadvantages between the two algorithms, we believe them to be an adequate fit for solving the problem at hand given our dataset.

4.1 Logistic Regression

Logistic Regression is an optimisation algorithm based on boundaries between two classes that are normally set between values 0 and 1 - the output of a logistic regression model can essentially be thought of as a probability value (of the instance belonging to one of the two classes). This value is obtained from the product of the weights and vectors and its distance from the boundary. Whether the value is less than or greater than the boundary (P=0.5) will determine whether the patient does have a cardiovascular disease. Therefore, in our case Logistic Regression will be utilised as a classifier.
The reasons why we chose this model are threefold. In the context of this project, this algorithm is ideal as our outcome is binary; the boundary introduced by Logistic Regression helps linearly separate the two classes. Moreover, this model can be regularised - the large coefficients are penalised in order to prevent overfitting. Lastly, we can use the probability value returned to measure the accuracy of the predictions.

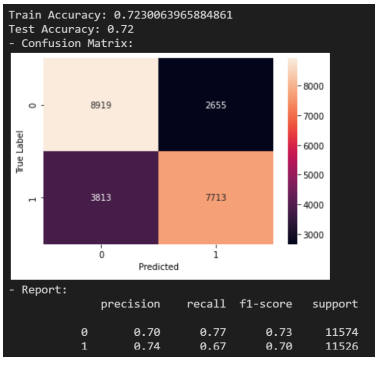
4.2 K-Nearest Neighbours

K-Nearest Neighbours is a simple classification algorithm that identifies what class an instance belongs to by looking at its K nearest neighbours; this is done based on a similarity measure or distance functions. The distance K is a constant (usually a Euclidean distance) defined by the user which serves as a radius that encompasses possibly multiple data points; these data points are then used to determine the predicted target attribute of the data instance at hand. For each of these data points, the Euclidean distance is calculated and appended to an array, which is then sorted in ascending order. After iterating through all data points in radius K, the predicted target attribute is assigned the mode of the labels in the array.
This algorithm is simple and efficient, and requires no pre-built model or training stage. Another advantage is that K-Nearest Neighbours is versatile meaning it can be utilised for problems such as regression and search as well as classification. The only disadvantage is that this model's decrease in performance speed is proportional to the increase in number of instances and predictor variables.

5 Testing and Results

5.1 Dataset Preprocessing

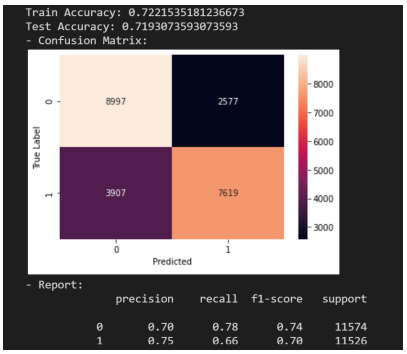
1. Removed ID from dataset
2. Convert age from days to years
3. Attempted standardization but it did not offer any improvements



5.2 Logistic Regression Results

Logistic Regression was the first model that we trained. We trained a model in python using sklearn and a subset of all

features and the solver was liblinear with a C=1 so the penalty was not too harsh.
As can be seen from the results, the model performed relatively well with a test accuracy of 72%. Accuracy is a good metric for this specific dataset because the dataset has two classes which are balanced. Looking at the confusion matrix we see that it has a bias towards the 0 (Not at risk) class which is not ideal in the context of detecting cardiovascular diseases as it will tell people they are healthy when in fact they are not (False Positive). If such a model is to be used in the context of medical diagnosis, it must have a higher recall value for class 1. It is important to note however that precision for class 1 is in fact better which is another target for the system as it has to catch the patients who are actually at risk.

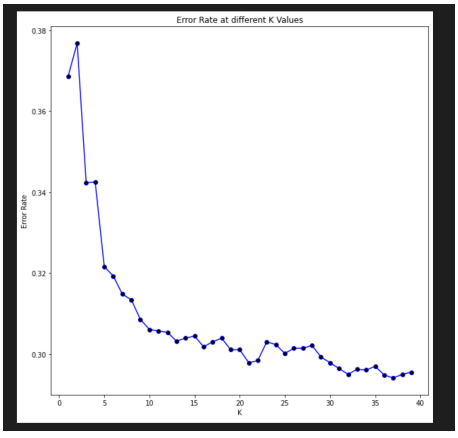


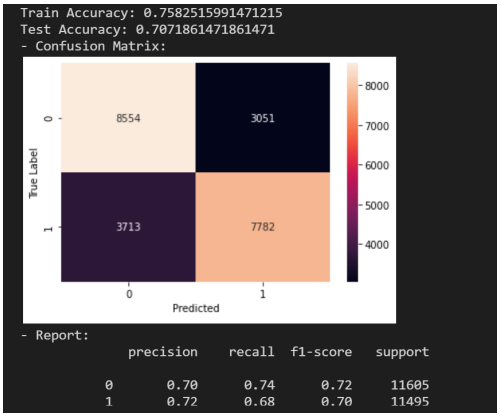
As can be seen, training a model using the best feature subset did not improve performance and in fact it slightly decreased it. The FPR is high which as mentioned previously is not what we want, For this to work we need the bias / variance tradeoff to be tuned so that the FPR is lower

5.3 K-Nearest Neighbours Results

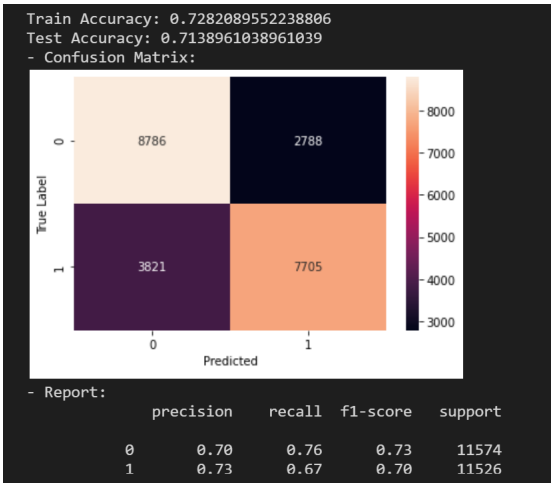
KNN was the second model we trained. Similarly to Logistic Regression, we first made a model into Python using sklearn and a subset made of all the features and then one model using the best subset of features. What is different however is a cross validation stage that we added in our python notebook so that we can find the best hyperparameter k.

This graph is showing the error rate at different k values. Based upon the “Elbow method” we decided that 9 would be the best parameter for our model since it is a binary classification problem so the k must be odd and it is the point at which you get the best increase in performance per computational power (the elbow of the graph).





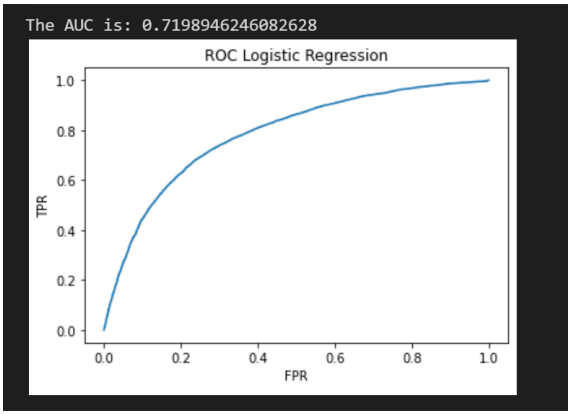
The results were pretty good as the report shows, the model scored well when it comes to accuracy at around 71% however notice the drop from 75% during training to 75%during testing. This means that unfortunately the model is trying to overfit one of the classes. It focuses too

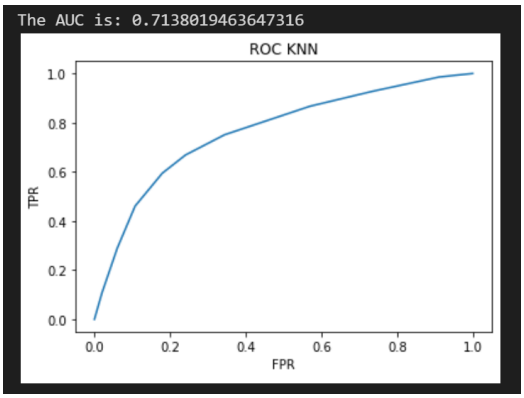


much on learning specific values for every single feature which shows KNN's weakness when it comes to high dimensional datasets. Still nonetheless it performed better than expected on the subset with all the features. As the confusion matrix shows, it managed pretty well but unfortunately as it stands by default the False Positives are too high for it to be ideal in a clinical scenario.

As expected, we got better results with the lower dimension feature set because of how KNN does its decision boundary. We can also see that now there is less difference in performance between training and testing which further proves the point that KNN suffers from overfitting when dealing with high dimensions. Even if these are better results The FPR is still too high for this model to be suitable for diagnosis. Meaning that it still requires some tuning of the threshold

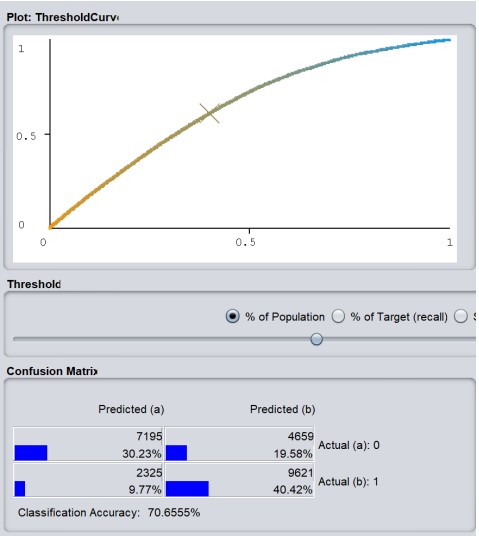
5.4 Results Comparison





In order to compare the two models used, we will use the AOC metric as it is a good representation of how well a model can perform and it can also show us how well it could perform when tuned. In order for the comparison to be fair, we will look at the ROC curve for each model at its best which means we are comparing the LR model with the high dimension feature set against the KNN model with the smaller feature subset (best features).
As can be seen, at their best both algorithms performed similarly and gave quite satisfactory results. The logistic Regression model scores slightly better when compared to the KNN one (0.72 against 0.71) which tells us that it is the superior model. Both models as we saw previously do suffer by default from a high FPR, meaning that as they are these should not be used in a diagnosis system because falsely telling a patient that they are healthy will have severe consequences. When it comes to implementation and computing power needed to run, KNN is

definitely more demanding as it will need cross validation for its hyperparameter as well as computing the distance to a high number of neighbours for each sample, making it also slower. We consider Logistic Regression to be the better model based on the results but both models have the potential to be useful with some tuning.



This is an example generated using Weka of how manipulating the threshold can change the rates. As can be seen now we got less FPs which makes this model more suitable for clinical diagnosis. It will have the negative effect that more people will be wrongly diagnosed at risk but it is not as bad compared to the alternative, meaning that such a clinical system must have a bias towards the negative outcome.

6 Conclusion

As a conclusion it is safe to say that when dealing with clinical data and diagnosis it is crucial to minimise the amount of patients wrongly classified as a whole but even more importantly it is to minimise the amount of sick patients categorised as being healthy. Usually, clinical data about patients tends to have a high number of dimensions so unless a subset is being used, we cannot see KNN as being the go to algorithm. On the other hand Logistic Regression does not suffer from high dimensions making it a more natural choice. The fact that these two algorithms performed well but by using the default threshold, the rates are not ideal is also showing how a supervised learning model such as these two also makes more sense to be used as it can be tuned by the scientists until it will eventually find the sweet spot at which the rates are acceptable. Finally, we consider these two models post threshold tuning to be a good choice for our dataset and the results are satisfactory since from the beginning accuracy was very high that alone being enough for us to call the project a successful experiment and we are sure that with a different dataset that has a more balanced distribution across all its features these models can achieve around 80%-82% accuracy.

7 Bibliography

[1] N. England, "NHS England » Cardiovascular disease (CVD)", England.nhs.uk, 2021. [Online]. Available: https://www.england.nhs.uk/ourwork/clinical-policy/cvd/#:~:text=Cardiovascular%20disease%20(CVD)%20is%20a,hypertension%2C%20stroke%20and%20vascular%20dementia. [Accessed: 16- Mar- 2021].

[2] "Cardiovascular disease", nhs.uk, 2021. [Online]. Available: https://www.nhs.uk/conditions/cardiovascular-disease/. [Accessed: 16- Mar- 2021].

[3] "Cardiovascular Disease dataset", Kaggle.com, 2021. [Online]. Available: https://www.kaggle.com/sulianova/cardiovascular-disease-dataset. [Accessed: 05- Mar- 2021].

[4] M. Shipe, S. Deppen, F. Farjah and E. Grogan, Developing prediction models for clinical use using logistic regression: an overview. 2019.

[5] A. Nishadi, Predicting Heart Diseases In Logistic Regression Of Machine Learning Algorithms By Python Jupyterlab. Colombo, 2019, pp. 1-2.

[6] S. Nusinovici et al., Logistic regression was as good as machine learning for predicting major chronic diseases. 2017.

[7] X. Chen, Q. Wu and G. Yan, RKNNMDA: Ranking-based KNN for MiRNA-Disease Association prediction. 2016, pp. 1-2.

[8] N. Khateeb and M. Usman, Efficient Heart Disease Prediction System using K-Nearest Neighbor Classification Technique. 2017.

[9] M. Jabbar, B. Deekshatulu and P. Chandra, Classification of Heart Disease Using K-Nearest Neighbor and Genetic Algorithm. 2013.

8 Appendix

8.1 Dataset Sample

age	gender	height	weight	ap_hi	ap_lo	cholesterol	gluc	smoke	alco	active	cardio
50	2	168	62	110	80	1	1	0	0	1	0
55	1	156	85	140	90	3	1	0	0	1	1
51	1	165	64	130	70	3	1	0	0	0	1
48	2	169	82	150	100	1	1	0	0	1	1
47	1	156	56	100	60	1	1	0	0	0	0
60	1	151	67	120	80	2	2	0	0	0	0
60	1	157	93	130	80	3	1	0	0	1	0
61	2	178	95	130	90	3	3	0	0	1	1
48	1	158	71	110	70	1	1	0	0	1	0
54	1	164	68	110	60	1	1	0	0	0	0
61	1	169	80	120	80	1	1	0	0	1	0
51	2	173	60	120	80	1	1	0	0	1	0
40	2	165	60	120	80	1	1	0	0	0	0
54	1	158	78	110	70	1	1	0	0	1	0
39	2	181	95	130	90	1	1	1	1	1	0
45	2	172	112	120	80	1	1	0	0	0	1
58	1	170	75	130	70	1	1	0	0	0	0
45	1	158	52	110	70	1	3	0	0	1	0
47	1	154	68	100	70	1	1	0	0	0	0
59	2	162	56	120	70	1	1	1	0	1	0

8.2 KNN Code

```
]
#### Please allow it to run for a few minutes! ####

rawData = [] #Hold all dataset raw in memory
processedData = []
#Load the dataset into the program and store it into a list
def loadData(path):
    with open(path) as f:
        reader = csv.reader(f)
        for line in reader:
            rawData.append(line)

loadData("cardio_train.csv")
#print(rawData[0][0])
#print(rawData[1])#

headers = preProcess(rawData)

#print(headers)
#print(processedData[1])
#Feature vector
features = [entry[:11]for entry in processedData]
#print(features[1])
#Labels for the entries
labels = [entry[11]for entry in processedData]
#print(labels[1])
#Feature + Label is the x-y pair needed for the model in order to start training
#The label is the risk factor 0 (No risk) or 1 (At risk)

#For the purpose of validation, ensuring that the model works well, the dataset will be split into training and testing (held out)
features_train, features_test, labels_train, labels_test = train_test_split(features, labels, test_size=0.33, random_state=10)
#features_train, features_val, labels_train, labels_val = train_test_split(features_train, labels_train, test_size=0.1, random_state=0)

# Uncomment this if you want to use standardized features with a deviation = 1 From my testing this performs similar or worse
#scaler = StandardScaler()
#features_train = scaler.fit_transform(features_train)
#features_val = scaler.fit_transform(features_val)
#features_test = scaler.fit_transform(features_test)

KNNClassifier = KNeighborsClassifier(n_neighbors=9, metric='euclidean')
KNNClassifier.fit(features_train, labels_train)

labels_pred = KNNClassifier.predict(features_test)
print('Train Accuracy: '+ str(KNNClassifier.score(features_train, labels_train)))
print('Test Accuracy: ' + str(KNNClassifier.score(features_test, labels_test)))
print('- Confusion Matrix:')
matrix = confusion_matrix(labels_test, labels_pred)
sns.heatmap(matrix, annot = True, fmt = 'g')
plt.xlabel("Predicted")
plt.ylabel("True Label")
plt.show()
print('- Report:')
print(classification_report(labels_test, labels_pred))
```

8.3 Logistic Regression

```
#Logistic Regression - Supervised

rawData = [] #Hold all dataset raw in memory
processedData = []
#Load the dataset into the program and store it into a list
def loadData(path):
    with open(path) as f:
        reader = csv.reader(f)
        for line in reader:
            rawData.append(line)

loadData("cardio_train.csv")
#print(rawData[0][0])
#print(rawData[1])

headers = preProcess(rawData)

#print(headers)
#print(processedData[1])
#Feature vector
features = [entry[:11]for entry in processedData]
#print(features[1])
#Labels for the entries
labels = [entry[11]for entry in processedData]
#print(labels[1])
#Feature + Label is the x-y pair needed for the model in order to start training
#The label is the risk factor 0 (No risk) or 1 (At risk)

#For the purpose of validation, ensuring that the model works well, the dataset will be split into training and testing (held out)
features_train, features_test, labels_train, labels_test = train_test_split(features, labels, test_size=0.33, random_state=0)

# Uncomment this if you want to use standardized features with a deviation = 1 From my testing this performs similar or worse
#scaler = StandardScaler()
#features_train = scaler.fit_transform(features_train)
#features_test = scaler.fit_transform(features_test)

LRClassifier = LogisticRegression(solver='liblinear', penalty = 'l2',C=1, random_state=0)
LRClassifier.fit(features_train, labels_train)

labels_pred = LRClassifier.predict(features_test)
print('Train Accuracy: '+ str(LRClassifier.score(features_train, labels_train)))
print('Test Accuracy: ' + str(LRClassifier.score(features_test, labels_test)))
print('- Confusion Matrix:')
matrix = confusion_matrix(labels_test, labels_pred)
sns.heatmap(matrix, annot = True, fmt = 'g')
plt.xlabel("Predicted")
plt.ylabel("True Label")
plt.show()
print('- Report:')
```