**Appliance Energy Prediction**

**Rohit Kumar sharma, Data science trainee,**

**Alma Better, Bangalore**

**Abstract:**

Home energy monitoring by appliance-level information can provide consumers awareness on energy saving. The system can be implemented through a smart meter which requires an efficient data analysis algorithm for providing an accurate energy consumption profile, the purpose for proper home energy management.

***Keywords: machine learning, heart risk prediction, classified labels***

**1.Problem Statement**

Heart disease is the major cause of morbidity and mortality globally: it accounts for more deaths annually than any other cause. For example, an estimated 17.9 million people died from heart diseases in 2016, representing 31% of all global deaths. Over three quarters of these deaths took place in low- and middle-income countries. It is, however, difficult to identify high risk patients because of the multi-factorial nature of several contributory risk factors such as diabetes, high blood pressure, high cholesterol etc. Due to such constraints, scientists have turned towards modern approaches like Data Mining and Machine Learning for predicting the disease.

The main objective is to build a predictive model, which could help them in predicting the heart risk proactively.

There are lots of attributes taken to identify the risk:

* **Sex**: male or female
* **Age**: Age of the patient
* **Current Smoker**: whether or not the patient is a current smoker.
* **Cigs Per Day**: the number of cigarettes that the person smoked on average in one day.
* **BP Meds**: whether or not the patient was on blood pressure medication
* **Prevalent Stroke**: whether or not the patient had previously had a stroke
* **Prevalent Hyp**: whether or not the patient was hypertensive
* **Diabetes**: whether or not the patient had diabetes
* **Tot Chol**: total cholesterol level
* **Sys BP**: systolic blood pressure
* **Día BP**: diastolic blood pressure
* **BMI**: Body Mass Index
* **Glucose**: glucose level
* **CHD:** 10-year risk of coronary heart disease (CHD) - (binary: “1”, means “Yes”, “0” means “No”)

**2. Introduction**

### Most heart diseases are highly preventable and simple lifestyle modifications (such as reducing tobacco use, eating healthily, obesity and exercising) coupled with early treatment greatly improve their prognoses. It is, however, difficult to identify high risk patients because of the multi-factorial nature of several contributory risk factors such as diabetes, high blood pressure, high cholesterol etc. Due to such constraints, scientists have turned towards modern approaches like Data Mining and Machine Learning for predicting the disease.

Machine learning (ML), due to its superiority in pattern detection and classification, proves to be effective in assisting decision making and risk assessment from the large quantity of data produced by the healthcare industry on heart disease.

### Our goal here is to build a predictive model, which could help doctors and hospitals in predicting Cardiovascular heart risk proactively.

## **3. Reasons for Heart Risk**

The reasons for surge heart diseases are:

* Diabetes
* Excessive use of alcohol or caffeine
* High blood pressure
* Smoking
* Stress

**3. Steps involved:**

* **Exploratory Data Analysis**

After loading the dataset, we performed this method by comparing our target variable that is CHD with other independent variables. This process helped us figuring out various aspects and relationships among the target and the independent variables. It gave us a better idea of which feature behaves in which manner compared to the target variable.

* **Null values Treatment**

Our dataset contains a large number of null values which might tend to disturb our accuracy hence we handle them at the beginning of our project in order to get a better result.

* **Encoding of categorical columns**

We used dummies to produce binary integers of 0 and 1 to encode our categorical features because categorical features that are in string format cannot be understood by the machine and needs to be converted to numerical format.

* **Feature Selection**

In these steps we used algorithms like Boruta to check the results of each feature i.e., which feature is more important compared to our model and which is of less importance.

* **Standardization of features**

Our main motive through this step was to scale our data into a uniform format that would allow us to utilize the data in a better way while performing fitting and applying different algorithms to it.

The basic goal was to enforce a level of consistency or uniformity to certain practices or operations within the selected environment.

* **Fitting different models**

For modelling we tried various classification algorithms like:

1. **Logistic Regression**
2. **SVM Classifier**
3. **K nearest Neighbour**
4. **Decision Tree**

* **Tuning the hyperparameters for better accuracy**

Tuning the hyperparameters of respective algorithms is necessary for getting better accuracy and to avoid overfitting in Decision tree and k nearest neighbor.

**7.1. Algorithms:**

1. **Logistic Regression:**

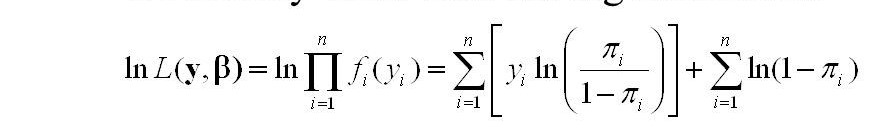
Logistic Regression is actually a classification algorithm that was given the name regression due to the fact that the mathematical formulation is very similar to linear regression.

The function used in Logistic Regression is sigmoid function or the logistic function given by:

f(x)= 1/1+e ^(-x)



The optimization algorithm used is: Maximum Log Likelihood. We mostly take log likelihood in Logistic:

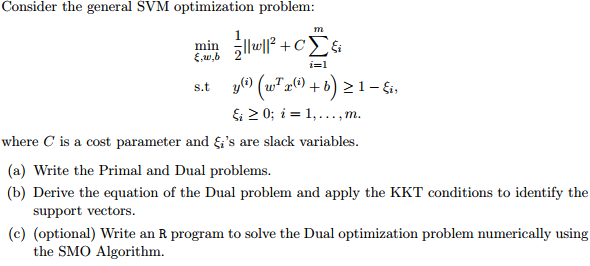


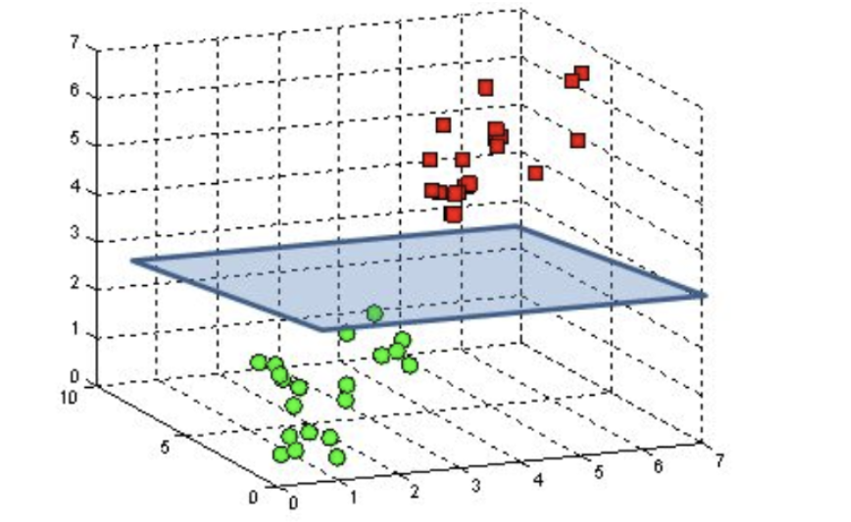
.

1. **Support Vector Machine Classifier:**

SVM is used mostly when the data cannot be linearly separated by logistic regression and the data has noise. This can be done by separating the data with a hyperplane at a higher order dimension.

In SVM we use the optimization algorithm as:





We use hinge loss to deal with the noise when the data isn’t linearly separable.

Kernel functions can be used to map data to higher dimensions when there is inherent non linearity.

1. **K Nearest Neighbour:**

The k-nearest-neighbours is a data classification algorithm that attempts to determine what group a data point is in by looking at the data points around it. An algorithm, looking at one point on a grid, trying to determine if a point is in group A or B, looks at the states of the points that are near it. The range is arbitrarily determined, but the point is to take a sample of the data. If the majority of the points are in group A, then it is likely that the data point in question will be A rather than B, and vice versa.



1. **Decision Tree-**

A decision tree is a tree-like graph with nodes representing the place where we pick an attribute and ask a question; edges represent the answers the to the question; and the leaves represent the actual output or class label.

Decision trees classify the examples by sorting them down the tree from the root to some leaf node, with the leaf node providing the classification to the exam

**7.2. Model performance:**

Model can be evaluated by various metrics such as:

1. **Confusion Matrix**-

The confusion matrix is a table that summarizes how successful the classification model is at predicting examples belonging to various classes. One axis of the confusion matrix is the label that the model predicted, and the other axis is the actual label.

1. **Precision/Recall**-

Precision is the ratio of correct positive predictions to the overall number of positive predictions: TP/TP+FP

Recall is the ratio of correct positive predictions to the overall number of positive examples in the set: TP/FN+TP

1. **Accuracy**-

Accuracy is given by the number of correctly classified examples divided by the total number

of classified examples. In terms of the confusion matrix, it is given by: TP+TN/TP+TN+FP+FN

1. **Area under ROC Curve (AUC)**-

ROC curves use a combination of the true positive rate (the proportion of positive examples predicted correctly, defined exactly as recall) and false positive rate (the proportion of negative examples predicted incorrectly) to build up a summary picture of the classification performance.

**7.3. Hyper parameter tuning:**

Hyperparameters are sets of information that are used to control the way of learning an algorithm. Their definitions impact parameters of the models, seen as a way of learning, change from the new hyperparameters. This set of values affects performance, stability and interpretation of a model. Each algorithm requires a specific hyperparameters grid that can be adjusted according to the business problem. Hyperparameters alter the way a model learns to trigger this training algorithm after parameters to generate outputs.

We used Grid Search CV for hyperparameter tuning. This also results in cross validation and in our case, we divided the dataset into different folds.

**Grid Search CV-**Grid Search combines a selection of hyperparameters established by the scientist and runs through all of them to evaluate the model’s performance. Its advantage is that it is a simple technique that will go through all the programmed combinations. The biggest disadvantage is that it traverses a specific region of the parameter space and cannot understand which movement or which region of the space is important to optimize the model.

**8. Conclusion:**

That's it! We reached the end of our exercise.

Starting with loading the data so far, we have done EDA, null values treatment, encoding of categorical columns, feature selection and then model building.

In all of these models our accuracy revolves in the range of 70 to 85 %. And there is no such improvement in accuracy score even after hyperparameter tuning. So, the accuracy of our best model is 85%.

* The Support vector machine with the radial kernel was the best performing model in terms of accuracy and the F1 score. Its high AUC shows that it has a high true positive rate.
* The most important features in predicting the ten-year risk of developing CHD were age and systolic blood pressure.
* Balancing the dataset by using the SMOTE technique helped in improving the models' sensitivity, this is when compared to the performance metrics of other models on different notebooks on the same dataset
* With more data (especially that of the minority class) better models can be built

**References-**

1. MachineLearningMastery
2. GeeksforGeeks
3. Analytics Vidhya