10/06/2023

TED ÜNİVERSİTESİ

CMPE 442/CS 542 - Assignment 3

Zülal Karın
12622989076

Dataset: ***UCI Heart Disease Dataset***
(http://archive.ics.uci.edu/dataset/45/heart+disease)

I experimented using the Heart Disease UCI dataset available at:
http://archive.ics.uci.edu/dataset/45/heart+disease

The dataset is about the presence of heart disease in the patient and the dataset includes age, gender, chest pain type, resting blood pressure, etc. There are 14 features including.

1. Problem Definition

The objective of this project is to predict the presence of heart disease in patients based on a number of physiological and physical measurements, along with certain risk factors. Specifically, we will be looking at data related to blood pressure, cholesterol levels, and age, among other factors. By analyzing this data, our goal is to develop a model that can accurately classify patients as either having or not having heart disease.

The task at hand is a binary classification problem, which means that we will be assigning each patient to one of two categories: either they have heart disease or they do not. To accomplish this, we will be using a variety of machine learning techniques, such as logistic regression and decision trees, to analyze the data and develop our model.

To ensure the accuracy of our model, we will also be conducting extensive testing and validation. This will involve using a large dataset of patient records to train our model, and then testing it on a separate set of data to evaluate its performance. We will also be using techniques such as cross-validation to ensure that our model is robust and reliable.

2. Data Analysis

After getting the dataset, I closely looked at the 14 features, such as age, sex, maximum heart rate, and different types of blood work. My aim was to know what each feature meant and how it could be connected to heart disease.

These 14 attributes and their descriptions are as follows:

1- age: age in years
2- sex: sex (1 = male; 0 = female)
3- p: chest pain type
    -- Value 1: typical angina
    -- Value 2: atypical angina
    -- Value 3: non-anginal pain
    -- Value 4: asymptomatic
4- trestbps: resting blood pressure (in mm Hg on admission to the hospital)
5- chol: serum cholestoral in mg/dl
6- fbs: (fasting blood sugar > 120 mg/dl) (1 = true; 0 = false)
7- restecg: resting electrocardiographic results
    -- Value 0: normal
    -- Value 1: having ST-T wave abnormality (T wave inversions and/or ST elevation or depression of > 0.05 mV)

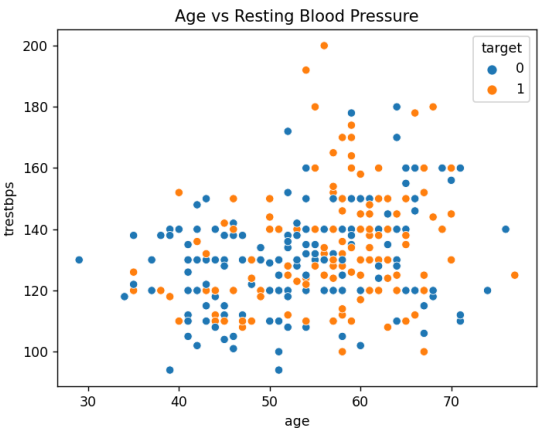
-- Value 2: showing probable or definite left ventricular hypertrophy by Estes' criteria

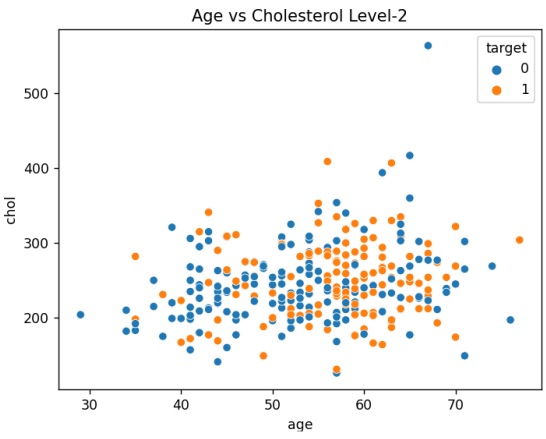
8- thalach: maximum heart rate achieved
9- exang: exercise induced angina (1 = yes; 0 = no)
10- oldpeak = ST depression induced by exercise relative to rest
11- slope: the slope of the peak exercise ST segment
-- Value 1: upsloping
-- Value 2: flat
-- Value 3: downsloping

12- ca: number of major vessels (0-3) colored by flourosopy
13- thal: 3 = normal; 6 = fixed defect; 7 = reversable defect
14- num: diagnosis of heart disease (angiographic disease status)
-- Value 0: < 50% diameter narrowing
-- Value 1: > 50% diameter narrowing
(in any major vessel: attributes 59 through 68 are vessels)

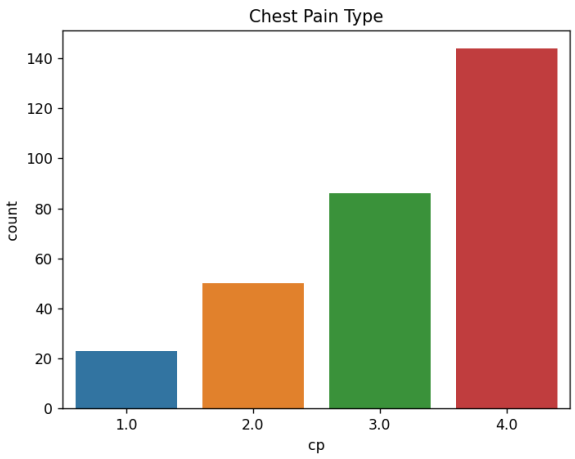
I did not find any missing values, which made the preprocessing stage simpler. To better understand the distribution of values for each feature, I looked at histograms. I noticed that the 'age' feature followed a normal distribution. The sex feature showed that males were slightly more represented in our dataset.

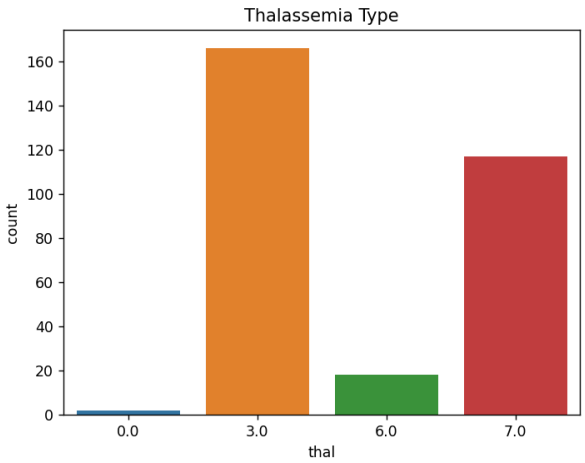
To explore relationships between features, I created scatter plots of feature pairs. I saw that there might be a correlation between age and resting blood pressure. I also made a chart to see if there was a difference in age and cholesterol level between patients with and without heart disease. You can see the graphs related to this data below. I used scatter plots to make these graphs. Scatter plots can be very helpful for observing relationships between two numerical features.





Examining the categorical features, I discovered that "typical angina" was the most common chest pain type. The 'thal' feature, which represents a type of blood disorder, was most frequently 'normal'.





3. Data Processing

I started by using the train_test_split method from the sklearn.model_selection package to divide the dataset into a training set and a test set. By setting the test size to 20%, I made sure the test set contained enough data points to properly assess the model's performance.

There were no missing values in the dataset, which is good news. If there had been, though, I would have filled them in with a SimpleImputer from sklearn.impute that could do so using the mean or median. Or I could have used a predictive imputation technique like K-Nearest Neighbors. I could have taken a more sophisticated approach by using Gaussian Naive Bayes or Bayesian Belief Networks to forecast missing values based on other known data.

The dataset had a number of categorical features that required to be encoded as numerical data in order to be used by machine learning methods.I made binary (0 or 1) columns for each category of each categorical characteristic for this task, which we discussed in Word Embedding in our course.

Finally, I standardized the data using the StandardScaler function from sklearn.preprocessing to ensure that all features made an equal contribution to the model's performance. For algorithms like Support Vector Machines, which I intended to employ later in the project, this function normalized characteristics by removing the mean and scaling to unit variance, which is extremely significant.

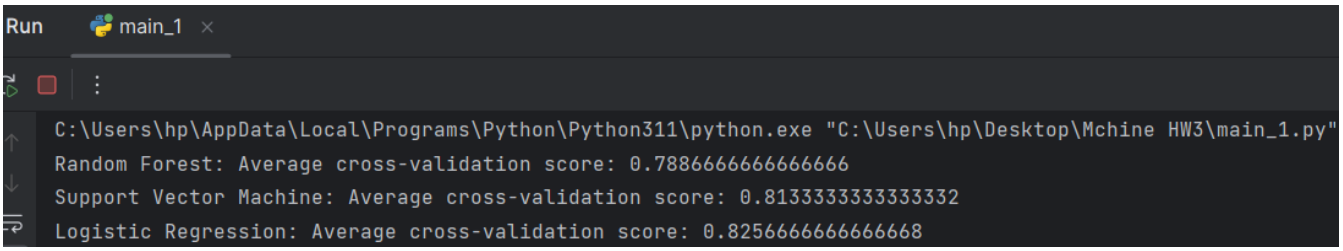
4. Model Selection and Training

I decided to test out three models: Random Forest Classifier, Support Vector Machine (SVM), and Logistic Regression. These models provide a good variety of various methods for solving classification problems.

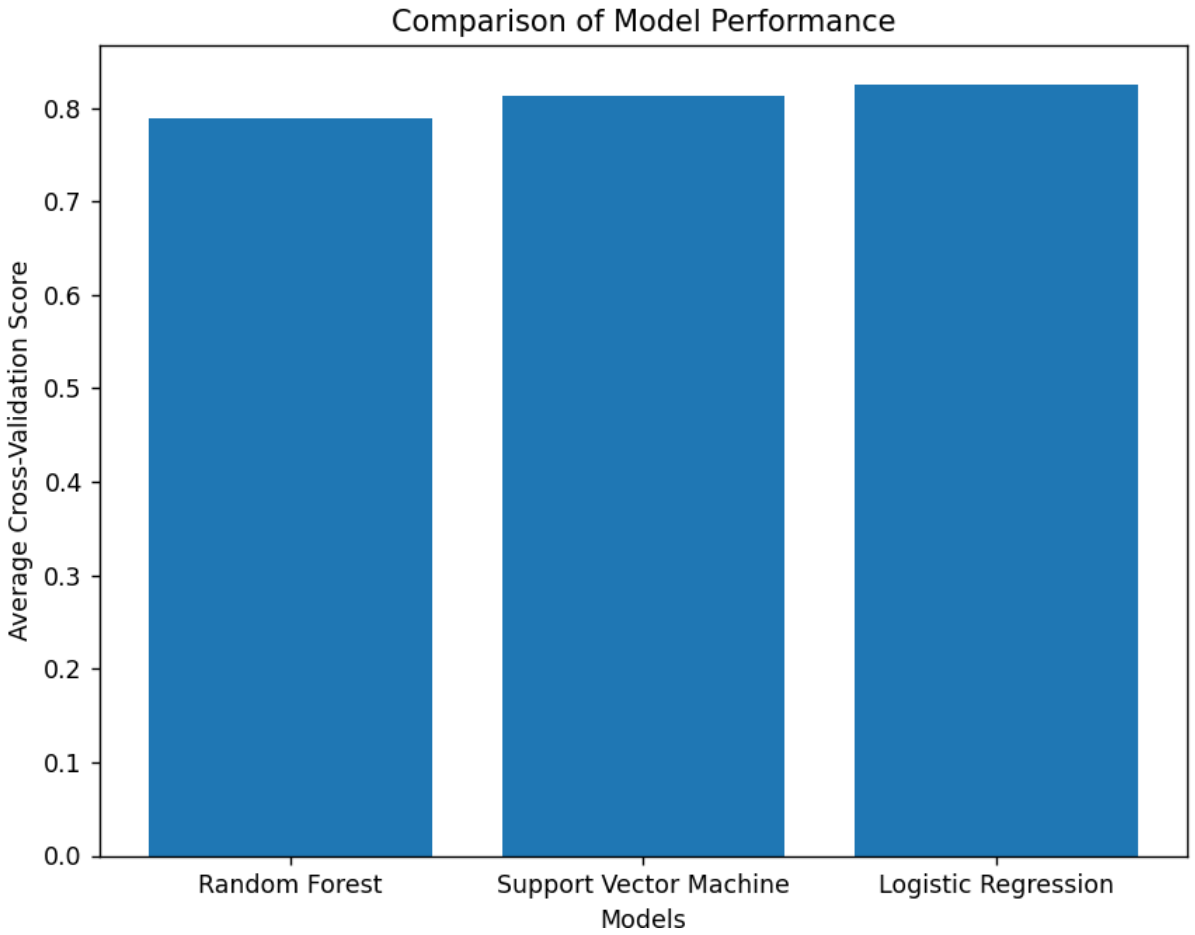
To get a robust estimate of how these models would perform in practice, I applied 10-fold cross-validation. In each round, I trained the models on 90% of the data and evaluated them on the remaining 10%.

I build a pipeline with a preprocessor and the appropriate classifier for each model. Any required data transformations, such as scaling numerical characteristics and encoding categorical features, are handled by the preprocessor. The preprocessed data is used to train the classifiers.

The output of my code is as follows:



Based on these results, Logistic Regression has the highest average cross-validation performance. As a result, we choose the Logistic Regression model as the best one for more study and continue by adjusting its hyperparameters.



5. Fine-tune Your Model

I used GridSearchCV to do hyperparameter tuning in order to enhance the performance of the chosen Logistic Regression model. The number of estimators (trees) in the Random Forest and the maximum depth of the trees are two examples of the hyperparameters I define to investigate. Cross-

validation is used by GridSearchCV to thoroughly search through all possible combinations of these hyperparameters in order to find the ideal one.

The most effective hyperparameters discovered by GridSearchCV are:

```
Best parameters: {'classifier__max_depth': 10, 'classifier__n_estimators': 50}

Best score: 0.8091666666666667
```

The best parameters found were 'max_depth': 10 and 'n_estimators': 50. The corresponding best score obtained through cross-validation was 0.8091.

```python
# Perform hyperparameter tuning on the best model
param_grid = {'classifier__n_estimators': [50, 100, 150],
              'classifier__max_depth': [None, 10, 20, 30]}
grid_search = GridSearchCV(rf_pipeline, param_grid, cv=10)
grid_search.fit(X_train, y_train)
print(f'Best parameters: {grid_search.best_params_}')
print(f'Best score: {grid_search.best_score_}')
```

figure 1: the code I apply for this part

6. Testing

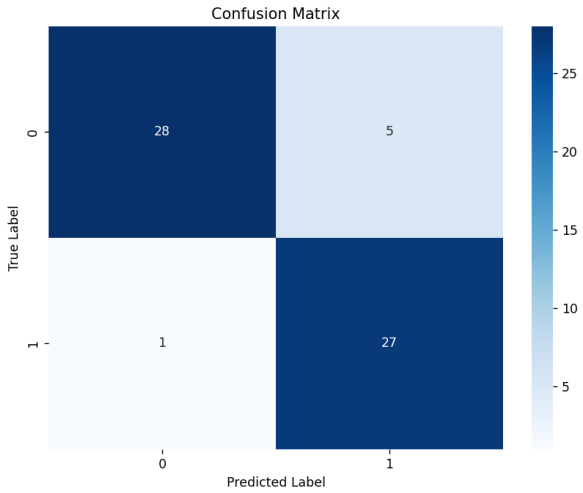
I chose the best model, adjusted its hyperparameters, and then assessed how well it performed on the test set. The test set offers a random sample for evaluating the generalizability of the model.

The chosen Logistic Regression model's test set performance measures for the top hyperparameters are as follows:

```
Test set accuracy: 0.9016393442622951
Test set precision: 0.84375
Test set recall: 0.9642857142857143
Test set F1-score: 0.8999999999999999
```

The model's high accuracy of 90.16% shows that it consistently predicts whether a patient will have heart disease or not. Precision is 84.38%, which is the percentage of real positives versus all anticipated positives. Recall, which counts the percentage of real positives among all positive results, is 96.43%. The precision and recall combined into one statistic, the F1-score, is 90.00%.

In order to see the distribution of true positive, true negative, false positive, and false negative predictions on the test set throughout the testing phase, I constructed a confusion matrix heatmap. The confusion matrix is shown below:



7. Summary

In this study, I developed a machine learning model to predict the occurrence of heart disease. To do this, I used three alternative models: Random Forest, Support Vector Machine, and Logistic Regression. After K-fold cross-validation, I selected Logistic Regression as the top-performing model. I further adjusted its hyperparameters with GridSearchCV. The final Logistic Regression model had a 90.16% accuracy rate on the test set.

Upon reflection, I discovered that the study was successful in developing a machine learning model to anticipate patients' occurrence of heart disease. In order to create a robust and accurate model, I used a variety of strategies, including data analysis and preprocessing as well as model selection, tuning, and evaluation.

I see potential for more improvement in the future. The model's performance might be improved by collecting more data, especially for under-represented groups. Results might also be enhanced by combining various models or by experimenting with more sophisticated models, like XGBoost or neural networks. I'm looking forward to pursuing these avenues in next projects.