Jonathan Asanjarani 07/11/2024

Final Paper DATA71200 Advanced Data Analysis Summer 2024

Supervised and Unsupervised Learning with the Framingham Heart Study

The following machine learning project applies a supervised and unsupervised learning algorithm to a clinical dataset to predict whether a patient has a risk of developing coronary heart disease over a ten-year period, using various data preprocessing and transformation techniques to enhance model performance. Supervised learning is a type of machine learning where the algorithm is trained on labeled data to make predictions or decisions. It learns to map input data to the correct output. This machine learning technique is optimal for classification tasks. The results of different machine learning models, such as Naive Bayes, Random Forest, and XGB Classifier, were compared, with cross-validation used to ensure the reliability of the performance estimates. Machine learning models were evaluated by calculating the F1-score, recall score, accuracy score, and precision score. The precision score measures how many predicted positive instances were correct. Accuracy measures the overall correctness of the model. The recall score focuses on how well a model evaluated negative prediction. For this dataset, a model is considered to have a negative prediction if a patient is not diagnosed with a ten-year risk of coronary heart disease. The F1-score combines precision and recall into a single number, balancing the trade-offs. It is the average of precision and recall, with a greater emphasis on the smaller value. The score goes from 0 to 100%, with higher values indicating better performance. It is particularly useful for imbalanced datasets because it considers both false positives and false negatives. Unsupervised learning techniques were also employed to explore how well the data could be separated without predefined labels. Unsupervised learning is a type of machine learning that works with unlabeled data, meaning there are no pre-existing labels or categories. Its goal is to find patterns and relationships in the data on its own. Three types of clustering algorithms were tested: K-means clustering, Density-Based Spatial Clustering of Applications with Noise (DBSCAN), and Agglomerative Hierarchical Clustering. Unsupervised learning models were evaluated using the ARI score and the silhouette coefficient. The ARI (Adjusted Rand Index) measures the similarity between clusters by comparing the clustering results with ground truth labels, adjusting for chance. The silhouette coefficient assesses how similar an object is to its own cluster compared to other clusters, indicating the quality of the clustering. These metrics provide insights into the effectiveness and accuracy of the clustering models.

Project 1

The dataset used for this project is derived from the Framingham Heart Study and includes various medical and demographic variables used to assess the risk of developing coronary heart disease over a ten-year period. This data can be found on GitHub using the following link (https://github.com/GauravPadawe/Framingham-Heart-Study/blob/master/framingham.csv). The Framingham Heart Study is a long-term, ongoing cardiovascular cohort study on residents of the city of Framingham, Massachusetts. The study began in 1948 with 5,209 adult subjects and has added second and third generation subjects since. It is a major source of data on factors affecting heart disease, and its findings have contributed to the understanding of heart health that has led to the development of guidelines for blood pressure, cholesterol, and other risk factors in maintaining cardiovascular health. The columns for this dataset include the age of the participant, education of the participant, whether or not they are a smoker, how many cigarettes do they smoke per day, whether the participant is on blood pressure medication, whether the participant had a stroke previously or not, whether the participant has prevalent hypertension, whether the participant has diabetes, the total cholesterol of the participant, the systolic blood pressure of the participant, the diastolic blood pressure of the participant, the heart rate of the participant, and whether or not the participant was labeled with a risk of developing coronary heart disease over a ten-year period.

The following project was produced using Google collab and Python 3.0. Exploratory data analysis was conducted using the pandas, NumPy, matplotlib, SciPy, SKlearn, and seaborn libraries. The data set was uploaded to GitHub and imported using the pandas library. The first five rows of the dataset were examined to get an understanding of the different types of values available in each column. Additionally, it is necessary to examine the datatype to ensure that the data is compatible with machine learning algorithms.

This particularly dataset has already been preprocessed to some degree. Columns with binary values, such as the column indicating whether the participant had a stroke previously or not, have already been converted to numerical values. A value of “0” indicates that the participant does not have a particular condition and value “1” indicates they do have said condition. All the columns have either integer or float data types.

The dataset was then split into a training and testing set using SKlearn. Our data was separated into features (X) and labels (y). The term feature refers to the input label. In other words, a feature is one column of data that is used to predict the target variable. The target variable can also be referred to as the (y) variable. For this project, the target variable is the final column in the dataset, indicating whether or not a particular patient has a risk of developing heart disease over a ten-year period. The features(X) consist of all the remaining columns in the dataset(gender, age, education, currentSmoker(are you a current smoker?), CigPerDay(how many cigarettes do you smoke per day?), BPMeds(Do you take BPMeds?), prevalent Stroke(Are you having a prevalent Stroke?), Prevalent Hyp(Do you have prevalent hypertension?), diabetes(do you have diabetes?), totChol(What is your total cholesterol?), sysBP(What is your systolic blood pressure?), diaBP(What is you diastolic blood pressure?), BMI(What is your BMI?), and heartrate(what is your average heart rate?). The data frame was split up into four sections: y_train, y_test, X_train, and X_test. The model was trained and fitted using the X_train and y_train sets. To determine whether the model is correctly predicting the outputs and labels, the X_test and y_test sets are utilized. Maintaining our training sets larger than the test sets is advised. Therefore, for this project, twenty percent of the dataset was used as the test set, and eighty percent of the data set was used as the training set.

Exploratory data analysis was then conducted on the training set. The “. describe” method was used to understand the mean, standard deviation, minimum value, maximum value, and the quartile range. Most participants were middle-aged adults between the ages of forty-one to fifty-five, and only had some high school education. There were an equal number of patients with a healthy BMI score and an overweight BMI score. Most participants had an average heart rate. There also seemed to be an equal number of smokers vs. non-smokers. In total, there are three thousand three hundred and ninety-two rows indicating the number of patients and fourteen columns indicating the number of features/columns used to describe said patients.

To train our dataset with supervised learning, it is necessary to account for missing values. To visualize the number of missing values by column I imported the “missingno” library. A bar graph was created to visualize the number of values by column. Of all the columns, glucose seemed to have the lowest number of values, and the highest number of missing values. A heatmap was also created to visualize the nullity (missingness) correlation in the data. Based on the results, there was low, if any, correlation between missingness in values. Mean imputation was than used to fill missing values for variables that had over ten percent of total values missing. This included the following variables: “glucose”, “education”,” BPMed”, and “totChol”. Variables/columns that had less than ten percent of total values missing, had their missing value dropped. Lastly, the dataset was checked for duplicates. There were no duplicate values found.

The target variable, or the variable we are trying to predict for our supervised learning project, is a binary value indicating whether a particular patient has a risk of developing coronary heart disease over a ten-year period, called “TenYearCHD”. A value of “0” indicates that the participant does not have a risk of developing coronary heart disease over a ten-year period and value “1” indicates they do have a risk of developing coronary heart disease over a ten-year period. It is imperative to understand how these values are distributed. The majority value is “0”, indicating that most patients do not have a risk of developing coronary heart disease over a ten-year period. Five hundred and fifteen patients were labeled with said risk and two thousand eight hundred and seventy-seven patients were not. This appears to be a class imbalance. There also seems to be a relatively low number of patients assigned a ten-year risk.

To further understand the dataset, an individual histogram was made for each column/variable to understand how the values are distributed. As stated earlier, most participants were middle-aged adults with only some high school education. These participants also had a relatively healthy BMI, and there were an equal number of smokers and non-smokers. Most smokers smoked less than ten cigarettes per day. Most patients were not on blood pressure medication, did not have a prevalent stroke, did not have prevalent hypertension, and did not have diabetes. There was a relatively normal distribution of values for the variables heart rate and diastolic blood pressure. Systolic blood pressure, glucose, and total cholesterol have a right-skewed distribution of values.

Furthermore, a series of scatter matrices were produced to understand how different variables related with one another. More specifically, a scatter matrix can help visually identify any possible correlations between variables. Based on the results, there seemed to be a strong positive correlation between systolic and diastolic blood pressure. There is a weak positive correlation between age and total cholesterol, systolic blood pressure, diastolic blood pressure, and BMI. There also seemed to be a weak positive correlation between systolic blood pressure and diastolic blood pressure with BMI.

Following this exploratory data analysis, additional data transformation was implemented on appropriate variables. Logarithmic, squared, cubed, and exponential transformation was applied to diastolic blood pressure and systolic blood pressure to experiment whether any of these transformations would normalize the distribution of values and account for the right-skew. Many machine learning algorithms assume that the data features are normally distributed, this is why handling skewed distribution is essential in the data transformation process. Logarithmic transformation seemed to be most effective in normalizing the distribution of values for systolic and diastolic blood pressure. Logarithmic transformation is a technique that is useful for normalizing right skewed or positively skewed data. This transformation applies natural log values to all data points. To visualize the differences, a histogram of the raw data distribution for both systolic and diastolic blood pressure was compared to a histogram depicting the distribution of values for these variables after logarithmic, squared, cubed, and exponential transformation.

Project 2

The library "sklearn" was used to standardize and normalize the dataset for our machine learning project. A custom transformer class labeled "Log Transformer" was used to apply a logarithmic transformation to the specified columns systolic and diastolic blood pressure. This transformation was then integrated into a preprocessing pipeline using the attribute “Column Transformer”. The “Log Transformer” class takes column names as input, fits without changes, and applies the np.log1p transformation (log(1 + x)) to the specified columns during transformation. The preprocessing pipeline then combines this custom transformation with a standard scaling step using “Standard Scaler” for specific columns. “Standard Scaler” is a preprocessing technique which transforms the distribution of each feature to have a mean of zero. This prevents any single feature from dominating the learning process due to a larger magnitude. The pipeline is configured to drop any columns not specified in the transformations. For this model test, only columns transformed through logarithmic transformation were utilized as features. Finally, the pipeline is fitted and applied to the training data (X_train) and used to transform the test data (X_test).

There were multiple attempts to transform the data for this model. Originally, missing values were filled by inputting the mean using ‘Simple Imputer’ to replace missing values. This imputation was then applied to the training and testing data. Additionally, under sampling was applied to the training data to address the class imbalance between patients with a ten-year risk and patients without one. Under sampling involves randomly deleting rows from the majority class (patients without a ten-year risk). This technique helps yield a balanced dataset and ensures that both classes are equally comparable.

This method of data transformation yielded decent results when applied to a Naïve Bayes classifier. The Naïve Bayes classifier is a common supervised learning algorithm that is used for classification problems. The model predicts the probability an instance belongs to a certain class given the feature values. The Naïve Bayes classifier calculates probability using Bayes Theorem, which finds the probability of an event occurring given the probability of another event that has already occurred. Each feature contributes to the predictions with no relation between each other. When trained on the resampled data, the Bayes classifier showed an accuracy of 0.72, an F1 score of 0.71, a precision score of 0.69, and a recall score of 0.72. However, after running cross validation, these results were significantly worse. Cross-validation is a method used in machine learning to better evaluate a model's performance. It involves dividing the data into multiple subsets or folds. In each iteration, one-fold is used for validation while the rest are used for training the model. This process repeats several times, with each fold getting a turn as the validation set. Finally, the results from all the iterations are combined to provide a more accurate estimate of the model's performance. Following cross-validation, the precision score dropped to 0.60, the recall score dropped to 0.59, and the F1 score dropped to 0.58.

Furthermore, this method of data transformation yielded even worse results when the training data was applied to a Random Forrest classifier. A Random Forest classifier builds many decision trees using random subsets of the data and features. Each tree makes its own prediction about the class of the input data and the majority vote is the final prediction. This approach improves accuracy and reduces overfitting compared to a single decision tree. The training data yielded an accuracy score of 0.58, an F1 score of 0.64, a precision score of 0.58, and a recall score of 0.59. After applying cross-validation, these scored remained relatively consistent. However, it is worth noting that the F1 score dropped to 0.58. A confusion matrix was plotted to display the number of accurate and inaccurate instances based on the model’s predictions. Out of the one hundred and twenty-nine patients labeled with a ten-year risk in the test-data, the model correctly predicted the label for seventy-two patients and incorrectly predicted the label for fifty seventy patients. Out of the seven hundred and twenty-two patients who were not labeled with a ten-year risk, the model correctly predicted the label for four hundred and twenty-six patients, and incorrectly predicted the label for two-hundred and ninety-three patients.

Following the poor results of this model's performance, a new strategy for data transformation was utilized to produce better results.

The dataset was reuploaded to the Google Collab notebook. For the purposes of this project, the data frame was sampled before being split into a training and testing set. This strategy was implemented to address the class imbalance and relatively low number of patients assigned a ten-year risk. Using 'SKlearn', the data was split into a testing and training set. Similarly to the first attempt, twenty percent of the data was used as the testing set, and eighty percent of the data was used for training. The split was also stratified. Stratified sampling is used in machine learning and statistics to make sure that samples are distributed across classes or categories in a way that is representative of the population.

Various types of imputation were used to fill missing values. Mean imputation was used when the distribution of data was normal and median imputation was used for skewed data. The variable "Heart Rate" had missing values filled using mean imputation. Mean imputation was used because variable "Heart Rate" has a relatively normal distribution of values. The following variables were filled using median imputation since their values had a more skewed distribution: education, Cigarettes per day, Blood Pressure Medication, Total Cholesterol, BMI, and glucose. Furthermore, variables with continuous values had missing values imputed using the IQR (InterQuartile Range) method. Data transformation was applied to the training and testing test. Logarithmic transformation was also applied to the systolic blood pressure column and diastolic blood pressure column. Finally, a preprocessing pipeline was created to scale numeric columns, and one-hot encode categorical columns, dropping all other columns. The pipeline was then fitted on the training set and applied to both training and test datasets.

The accuracy, F1 score, and recall score decreased significantly when the newly transformed data was trained on the Naïve Bayes classifier. The accuracy score decreased to 0.59, the recall score decreased to 0.59 and the F1 Score decreased to 0.63. However, the precision score increased to 0.74. Of the seven hundred and nineteen patients that were labeled with a ten-year risk, the model correctly predicted two hundred and thirty-four patients and incorrectly predicted four hundred and eighty-five patients. Of the seven hundred and twenty patients that were not labeled with a ten-year risk, the model correctly predicted the label for six hundred and twenty-one patients and incorrectly predicted the label for ninety-nine patients. Based on these results, we can conclude that the model was more accurate in classifying patients without a ten-year risk than classifying patients with a ten-year risk. Based on these results, we can conclude that this strategy of data transformation was not effective for this model.

The random Forrest classification model yielded much more promising results. The accuracy, F1-score, recall score, and precision of the model increased to 0.88. Of the seven hundred and nineteen patients that were labeled with a ten-year risk, the model correctly predicted the label for six hundred and sixty-three patients and incorrectly predicted the label for 56 patients. Of the seven hundred and twenty patients that were not labeled with a ten-year risk, the model correctly predicted the label for five hundred and ninety-eight patients and incorrectly predicted the label for one hundred and twenty-two patients. Based on these results, we can conclude that the model more accurately predicted participants labeled with a ten-year risk for cardiovascular disease. After cross validation, these results decreased significantly. The average accuracy score was reduced to 0.74, the average precision score was reduced to 0.72, the average recall score was reduced to 0.78, and the average f1-score was reduced to 0.75.

As an additional step, the training data was assessed on an XGB Classifier model. An XGB (XGBoost) Classifier creates a series of decision trees, each one improving on the mistakes of the last. It combines their predictions to make a final decision. It's fast and good at handling complex data. This model seemed to also yield promising results. For patients labeled without a ten-year risk, the model classified patients with an accuracy of 0.96, a recall score of 0.86, and an f1-score of 0.91. For patients that were labeled with a ten-year risk. The model classified patients with an accuracy of 0.87, a recall score of 0.97, and an f-1 score of 0.92. Similarly, the performance of the model was reduced significantly after cross-validation. The average accuracy score was reduced to 0.75, the average precision score was reduced to 0.73, the average recall score was reduced to 0.81, and the average f-1 score was reduced to 0.77.

Hyperparameter tuning was an additional strategy that was used to improve the performance of the random forest classifier and the XGBBoost Classifier. Finding the ideal values for a machine learning model's hyperparameters—parameters that regulate the model's learning process, such as the learning rate, the number of neurons in a neural network, or the kernel size in a support vector machine—is known as hyperparameter tuning. “Grid Search CV” was used to identify the best and worst parameters for both models.

The XGB Classifier model was rerun with modified parameters. The parameters set the maximum depth of each model to 7 layers and adjusted the subsample to use 70% of the data for training each model. Results were evaluated using cross-validation. There was little to no difference in results after hyperparameter tuning.

The random forest classifier was rerun with adjusted parameters for better clarity as well. We set the number of decision trees to 50 and used a seed value of 42 to ensure consistent results. Each tree's maximum depth was limited to 30 levels. For splitting nodes, the model used a logarithmic function (log2) to select features. Additionally, at least 2 samples were required to split any node. Despite these changes, cross-validation results showed little to no improvement in performance.

Project 3

The more features or dimensions in a dataset, the larger the amount of data required to obtain a statistically significant result. This can lead to issues such as overfitting and reduced accuracy. As the number of dimensions increases, the number of feature combinations grows exponentially, making it hard to get a representative sample of the data. Feature engineering techniques such as dimensionality reduction can help reduce the number of features while retaining as much of the original information as possible.

Principle component analysis is a dimensionality reduction technique that was implemented on the training data to reduce the number of features in the dataset. Principal Component Analysis (PCA) is an unsupervised technique used to study the relationships between variables. It reduces the number of variables in a dataset by finding a new, smaller set of variables that keeps most of the original information, making it useful for regression and classification.

Principle component analysis was used to assess if it could improve the performance of the random forest classification model. PCA was used to assess how many features are needed to retain 95% of the variance. This means PCA was applied to determine the minimum number of features needed to keep 95% of the data's original variability or information. All other features were dropped from the training data. When applied to the random forest classification model, we saw little to no difference in performance.

Three types of clustering algorithms were applied to the dataset: K-means clustering, Density Based Scanning Clustering, and Agglomerative/Hierarchical Clustering. These clustering algorithms were also evaluated with and without principal component analysis.

The K-means clustering algorithm was first applied without principal component analysis. First KMeans clustering was used to find the best number of clusters for a dataset by looking at how inertia changes with different cluster counts. It tries cluster numbers from 1 to 9, calculates the inertia (a measure of how well the data points fit within the clusters) for each cluster count, and stores these values in a dictionary. Finally, the number of clusters against their inertia values were plotted to help identify the optimal number of clusters by finding the "elbow point," where adding more clusters doesn't significantly improve the fit. The K-means clustering algorithm was implemented using two clusters based on the results of the plot. This clustering model yielded poor results, having an Adjusted Rand Index of 0.05 and a Silhouette coefficient of 0.17. The clustering model yielded the same results when principal component analysis was applied to the dataset.

The density-based scanning model and agglomerative clustering model did not seem to be compatible with the raw data or the data transformed through principal component analysis. The density-based clustering algorithm had an Adjusted Rand Index of 0.33 and a Silhouette coefficient of less than one percent. The agglomerative clustering algorithm had a silhouette coefficient of less than one percent as well as an ARI score of 0.02. There was little to no difference in results when principal component analysis was applied to the dataset.

As a final attempt at yielding better results, the training data was transformed a third time using a dimensionality reduction technique known as UMAP (Unifold Manifold Approximation and Projection). UMAP is a dimensionality reduction technique that can reduce the dimensionality of the data while maintaining its topological structure. UMAP is particularly useful for high-dimensional datasets.

Following this data transformation, the clustering algorithms yielded much better results. The K-means clustering algorithm had an increased ARI score of 0.26 and an increased silhouette coefficient score of 0.48 when the number of clusters was increased to ten. It is important to note that the K-means cluster map suggested a lower number of clusters, however, ten clusters yielded the best results. The density-based clustering algorithm had an increased ARI score of 0.48 and a similar Silhouette Coefficient of 0.31. Finally, the results of the agglomerative clustering model had an increased ARI score of 0.32 and an increased Silhouette Coefficient of 0.42.

Discussion

The Naïve Bayes classifier seemed to yield better results when there were less features added to the model. However, the random forest classification model and the XGB Classification model yielded the best result when all the data was used and missing values were imputed using mean, median, and interquartile range imputation rather than just mean imputation.

Based on the results, we can conclude that decision tree models had the best performance. A decision tree model may perform better than a Naïve Bayes classifier because it can capture complex interactions between features without assuming they are independent. Decision trees handle non-linear relationships, can manage missing values, and are robust to outliers, making them more effective for many real-world datasets.

The clustering models did not seem to lend themselves to the raw data or the data transformed through principal component analysis. Mild success was found when the dimensionality reduction technique UMAP or Unifold Manifold Approximation and Projection was applied to the dataset. However, even this result did not seem too promising. It is important to note that up sampling the dataset altered how the dataset would be normally distributed in the real world. Most patients in general, do not have a risk of developing coronary heart disease. Since the data has been up sampled, it contains an equal number of patients with and without a risk of developing coronary heart disease over a ten-year period. This may have negatively impacted the result of the supervised and unsupervised learning algorithms.