ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ
ΤΜΗΜΑ ΨΗΦΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ
ΠΜΣ "Πληροφοριακά Συστήματα & Υπηρεσίες"

EXPERTISE: Big Data & Analytics

COURSE: Planning and Infrastructure of Majors Data: Python and Cloud Computing

Professor: Nikolaos Sgouros

POSTGRADUATE INTERNSHIP FEB. 2022-23

Student: Magirias Georgios

A.M. me2220

Academic year: 2022-23

28/02/2023

Contents

1. Overview of the problem

Cardiovascular diseases are disorders of the heart and blood vessels, including coronary heart disease, strokes, heart failure and other types of pathology [1]. Overall, cardiovascular disease kills around 17 million people worldwide each year, with death tolls rising for the first time in 50 years in the UK [2]. Specifically, heart failure occurs when the heart is unable to pump enough blood to the body, and is usually associated with diabetes, high blood pressure, or other dysfunction of the patient's heart.

In this paper, a data set containing medical records of patients with heart failure is analyzed and an attempt is made to develop models that predict whether patients will survive or not. More specifically, the work is divided into the following parts. First, a description and analysis of the data set is made (Section: Data Description and Analysis), in order to check the correctness of the statistical descriptions given by Chicco, D., & Jurman, G [3] for the numerical and categorical characteristics . In the following (Section: Methodology), reference is made to the Machine Learning methods used, as well as to other techniques applied, to reduce the characteristics of the data set. Following are the results obtained from what was applied (Section: Results) and finally, (Section: Conclusions), reference is made to the conclusions drawn.

2. Data Description and Analysis

The dataset used contains the medical records of 299 heart failure patients collected at the Faisalabad Cardiac Institute and Allied Hospital in Faisalabad (Punjab, Pakistan) from April to December 2015. Of the patients, 105 are female and the 194 men, while their ages range between 40 and 95 years (Table 1). All 299 patients had left ventricular systolic dysfunction and had prior heart failure.

The dataset contains 13 features, which report clinical, body and lifestyle information of the patients. Some characteristics are binary: anemia, high blood pressure, diabetes, sex, and smoking. Table 1 shows the statistical quantitative description of these categorical characteristics, while at the same time confirming the correctness of tables 2, 3 of the publication [3]. The rest of the characteristics are numerical and their statistical quantitative description is presented in Table 2.

		Category Total #	Category percentage	Survived patients #	Survived patients %	Dead patients #	Dead patients %
anaemia	0	170	56.86	120	70.59	50	29.41
	1	129	43.14	83	64.34	46	35.66
high_blood_pressure	0	194	64.88	137	70.62	57	29.38
	1	105	35.12	66	62.86	39	37.14
diabetes	0	174	58.19	118	67.82	56	32.18
	1	125	41.81	85	68.00	40	32.00
sex	0	105	35.12	71	67.62	34	32.38
	1	194	64.88	132	68.04	62	31.96
smoking	0	203	67.89	137	67.49	66	32.51
	1	96	32.11	66	68.75	30	31.25

Table 1: Statistical description of categorical characteristics.

		mean	std	min	25%	50%	75%	max
age	Full sample	60.83	11.89	40.0	51.00	60.0	70.00	95.0
	Dead patients	65.22	13.21	42.0	55.00	65.0	75.00	95.0
	Survived patients	58.76	10.64	40.0	50.00	60.0	65.00	90.0
creatinine_phosphokinase	Full sample	581.84	970.29	23.0	116.50	250.0	582.00	7861.0
	Dead patients	670.20	1316.58	23.0	128.75	259.0	582.00	7861.0
	Survived patients	540.05	753.80	30.0	109.00	245.0	582.00	5209.0
ejection_fraction	Full sample	38.08	11.83	14.0	30.00	38.0	45.00	80.0
	Dead patients	33.47	12.53	14.0	25.00	30.0	38.00	70.0
	Survived patients	40.27	10.86	17.0	35.00	38.0	45.00	80.0
platelets	Full sample	263358.03	97804.24	25100.0	212500.00	262000.0	303500.00	850000.0
	Dead patients	256381.04	98525.68	47000.0	197500.00	258500.0	311000.00	621000.0
	Survived patients	266657.49	97531.20	25100.0	219500.00	263000.0	302000.00	850000.0
serum_creatinine	Full sample	1.39	1.03	0.5	0.90	1.1	1.40	9.4
	Dead patients	1.84	1.47	0.6	1.08	1.3	1.90	9.4
	Survived patients	1.18	0.65	0.5	0.90	1.0	1.20	6.1
serum_sodium	Full sample	136.63	4.41	113.0	134.00	137.0	140.00	148.0
	Dead patients	135.38	5.00	116.0	133.00	135.5	138.25	146.0
	Survived patients	137.22	3.98	113.0	135.50	137.0	140.00	148.0
time	Full sample	130.26	77.61	4.0	73.00	115.0	203.00	285.0
	Dead patients	70.89	62.38	4.0	25.50	44.5	102.25	241.0
	Survived patients	158.34	67.74	12.0	95.00	172.0	213.00	285.0

Table 2: Statistical description of categorical characteristics.

A sample from the data set is then given (Table 3), where all the characteristics are shown, but also the dependent variable (DEATH_EVENT), which indicates whether the patient survived (0), or did not survive (1).

	age	anaemia	creatinine_phosphokinase	diabetes	ejection_fraction	high_blood_pressure	platelets	serum_creatinine	serum_sodium	sex	smoking	time	DEATH_EVENT
0	75.0	0	582	0	20	1	265000.00	1.9	130	1	0	4	1
1	55.0	0	7861	0	38	0	263358.03	1.1	136	1	0	6	1
2	65.0	0	146	0	20	0	162000.00	1.3	129	1	1	7	1
3	50.0	1	111	0	20	0	210000.00	1.9	137	1	0	7	1
4	65.0	1	160	1	20	0	327000.00	2.7	116	0	0	8	1

Table 3: Sample dataset.

In terms of data imbalance, the surviving patients (DEATH_EVENT = 0) are 203, while the dead patients (DEATH_EVENT = 1) are 96. In statistical terms, there are 32.11% positive and 67.89% negative.

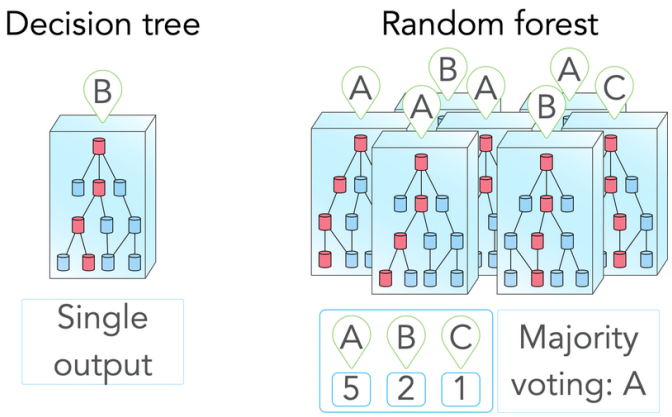
3. Methodology

In this section, initially, reference is made to the machine learning methods that were used to predict the survival of patients, using all characteristics, excluding "time". Then, there is a brief description of the PCA (Principal Component Analysis) technique, which was applied to reduce the features and finally the machine learning methods used to predict the survival of the patients are mentioned, but with the use of the new features produced by the PCA. What will be mentioned was implemented using the Python programming language.

3.1 Survival prediction classifiers

For the binary classification problem, using all features except the time of hospitalization (feature: “time”), two methods based on trees, Random Forests and Decision Trees, were applied. The random forests and the

decision trees are both popular machine learning algorithms used for classification and regression tasks. Decision trees are single trees that recursively partition the data based on the most important feature at each step, while random forests combine multiple decision trees and use random subsets of features to reduce overfitting and improve generalization performance.



Picture 1: Decision Tree, Random Forest

To evaluate the performance of the models, the Matthews correlation coefficient (MCC), F1-score, accuracy of the models, percentage of correctly classified positive samples (TPR), percentage of correctly classified negative samples (TNP) were calculated, as and the area under the AUC curve (ROC AUC). The specific outcome evaluation metrics were used so that the comparison with the research results of Chicco, D., & Jurman, G [3] could be made.

```python
# prepare data
X = df.drop(columns=['DEATH_EVENT', 'time', 'dummy'])
y = df['DEATH_EVENT']
scaler = MinMaxScaler()
X = scaler.fit_transform(X)

# initialize results list
results = []

# repeat process 100 times
for i in range(100):
    X_train, X_test, y_train, y_test = train_test_split(X, y, stratify=y, test_size=0.2)
    model = RandomForestClassifier()
    model.fit(X_train, y_train)
    y_pred = model.predict(X_test)

    # calculate metrics
    mcc = matthews_corrcoef(y_test, y_pred)
    f1 = f1_score(y_test, y_pred)
    acc = accuracy_score(y_test, y_pred)
    precision, recall, _ = precision_recall_curve(y_test, y_pred)
    pr_auc = auc(recall, precision)
    roc_auc = roc_auc_score(y_test, y_pred)
    tp_rate = np.sum((y_pred == 1) & (y_test == 1)) / np.sum(y_test == 1)
    tn_rate = np.sum((y_pred == 0) & (y_test == 0)) / np.sum(y_test == 0)

    # append results to list
    results.append([mcc, f1, acc, tp_rate, tn_rate, pr_auc, roc_auc])

# calculate mean of each metric and create dataframe
results_df = pd.DataFrame(results, columns=['MCC', 'F1 Score', 'Accuracy', 'TPR', 'TNR', 'PR AUC', 'ROC AUC'])
mean_results = results_df.mean()
final_results1 = pd.DataFrame(round(mean_results, 3)).T
final_results1.index = ['Random Forest (stratified_split, MinMaxScaler, iter=100)']

final_results1
```

Figure 2: Code section, applicationRF

Using the above code, the Random Forest method is applied and the averages of the metrics are calculated over the 100 runs. The same is done for the Decision Tree method.

3.2 PCA technique

PCA is a transformation of data (whose variables may be uncorrelated) whereby the new data acquire maximum variance and new variables, which are uncorrelated. This is how it is achieved:

1. Maximize Variance: The range of data values increases on one of the new axes (and decreases on the others).

2. Correlation minimization: The data variables after the transformation are uncorrelated, and therefore likely to be independent of each other.

3. Dimensionality reduction: After applying PCA, it is possible to exploit the data in fewer dimensions (those corresponding to higher variance), resulting in better understanding and visualization, as well as easier processing.

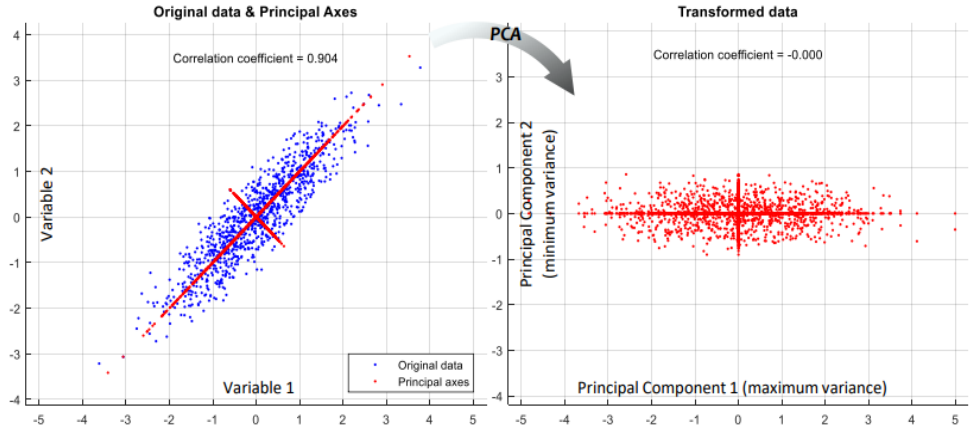


Figure 3: Two-dimensional data before and after applying the techniquePCA

The PCA technique was implemented using the scikit-learn library implementation.

3.3 Predicting survival using Principal Components

After the PCA technique was applied, three classifiers were then applied, one of which was the Random Forest and two new ones, the Gradient Boosting Classifier and the SVM with a radial kernel (SVM radial). Gradient Boosting Classifier is an ensemble learning method that successively combines weak classifiers to create a strong classifier, while Radial Kernel SVM classifier is a powerful algorithm that separates the data by mapping it into a high-dimensional space and finding the optimal threshold that maximizes the margin between different classes.

To evaluate the performance of these models, the same metrics mentioned in section 3.1 were again used. The code in the image below shows its implementation

SVM radial, using the 2 principal components resulting from PCA.

```
# prepare data
X = df.drop(columns=['DEATH_EVENT', 'time', 'dummy'])
y = df['DEATH_EVENT']
scaler = MinMaxScaler()
X = scaler.fit_transform(X)

# Initialize PCA with 2 components
pca = PCA(n_components=2)

# Fit PCA to the data and transform X
X_pca = pca.fit_transform(X)

# initialize results list
results = []

# repeat process 100 times
for i in range(1000):
    X_train, X_test, y_train, y_test = train_test_split(X_pca, y, stratify=y, test_size=0.2)
    model = SVC(kernel='rbf')
    model.fit(X_train, y_train)
    y_pred = model.predict(X_test)

    # calculate metrics
    mcc = matthews_corrcoef(y_test, y_pred)
    f1 = f1_score(y_test, y_pred)
    acc = accuracy_score(y_test, y_pred)
    precision, recall, _ = precision_recall_curve(y_test, y_pred)
    pr_auc = auc(recall, precision)
    roc_auc = roc_auc_score(y_test, y_pred)
    tp_rate = np.sum((y_pred == 1) & (y_test == 1)) / np.sum(y_test == 1)
    tn_rate = np.sum((y_pred == 0) & (y_test == 0)) / np.sum(y_test == 0)

    # append results to list
    results.append([mcc, f1, acc, tp_rate, tn_rate, pr_auc, roc_auc])

# calculate mean of each metric and create dataframe
results_df = pd.DataFrame(results, columns=['MCC', 'F1 Score', 'Accuracy', 'TPR', 'TNR', 'PR AUC', 'ROC AUC'])
mean_results = results_df.mean()
final_results3 = pd.DataFrame(round(mean_results, 3)).T
final_results3.index = ['SVC(radial)-PCA=2 (stratified_split, MinMaxScaler, iter=1000)']

final_results3
```

Figure 4: Code section, applicationSVM radial combined with PCA.

4. Results

In this section, we first describe the results obtained for survival prediction in the full data set (section “Survival prediction with all clinical features”), the results obtained by applying PCA (section “PCA technique results”) and the results for survival prediction when only the first two PCA principal components are used (section “Survival Prediction with Principal Components.

4.1 Prediction of survival by all clinical characteristics

As mentioned in the previous section, Random Forest and Decision Tree methods were applied. We checked the performance of the algorithms for different combinations of iterations and different preprocessing tools, as well as using or not using a trained partition of the data so that the two classes are correctly separated. The two tables below show the results obtained, where a multitude of metrics have been calculated to evaluate performance.

	MCC	F1 Score	Accuracy	TPR	TNR	PR AUC	ROC AUC
Random Forest (StandardScaler, iter=100)	0.378	0.540	0.743	0.493	0.860	0.637	0.677
Random Forest (StandardScaler, iter=1000)	0.361	0.524	0.735	0.467	0.865	0.632	0.666
Random Forest (MinMaxScaler, iter=100)	0.390	0.549	0.745	0.493	0.868	0.651	0.680
Random Forest (MinMaxScaler, iter=1000)	0.370	0.532	0.738	0.476	0.865	0.637	0.670
Random Forest (stratified_split, StandardScaler, iter=100)	0.372	0.534	0.742	0.477	0.864	0.634	0.671
Random Forest (stratified_split, MinMaxScaler, iter=100)	0.363	0.527	0.738	0.466	0.865	0.628	0.665

Table 4: Mean values of the valuation measures (random forest)

	MCC	F1 Score	Accuracy	TPR	TNR	PR AUC	ROC AUC
Decision Tree (StandardScaler, iter=100)	0.281	0.505	0.689	0.508	0.773	0.588	0.640
Decision Tree (StandardScaler, iter=1000)	0.259	0.490	0.675	0.497	0.761	0.579	0.629
Decision Tree (MinMaxScaler, iter=100)	0.266	0.491	0.682	0.498	0.769	0.579	0.633
Decision Tree (MinMaxScaler, iter=1000)	0.262	0.492	0.676	0.497	0.764	0.582	0.630
Decision Tree (stratified_split, StandardScaler, iter=100)	0.250	0.478	0.677	0.476	0.770	0.568	0.623
Decision Tree (stratified_split, MinMaxScaler, iter=100)	0.266	0.501	0.677	0.511	0.754	0.583	0.632

Table 5: Mean values of the valuation measures ((Decision Tree)

In both cases, the data was initially split, with 80% of the set used for training and 20% for testing the models. Judging from the results, the Random Forest method achieves the highest Matthews coefficient value (MCC= +0.390), the highest accuracy (Accuracy= 0.745), as well as the largest area under the ROC curve (ROC AUC= 0.680). These results are obtained when the MinMaxScaler transformer is used to scale the features, while the average values are obtained after 100 iterations. The performance of the Decision Tree model is quite worse in all cases, as can be seen from Table 5.

4.2 Results of PCA technique

By applying PCA, the contribution of the newly created features, i.e. the principal components, was tested. This was achieved by calculating the explanatory variance contributed by each characteristic. In the table below (Table 6), the percentage of variation offered by each characteristic is shown, while the cumulative variation diagram was then created (Figure 5), so that the elbow that is created can be seen, i.e. the point at which the data variance. Knowing this point, it is easy to find the optimal number of features (Principal Components) to perform dimension reduction.

	Feature	Explained Variance Ratio
0	PC1	0.276792
1	PC2	0.194606
2	PC3	0.174012
3	PC4	0.157112
4	PC5	0.095127
5	PC6	0.035307
6	PC7	0.024571
7	PC8	0.012917
8	PC9	0.010968
9	PC10	0.010119
10	PC11	0.008470

Table 6: Explanatory ratio of trait variancePCA.

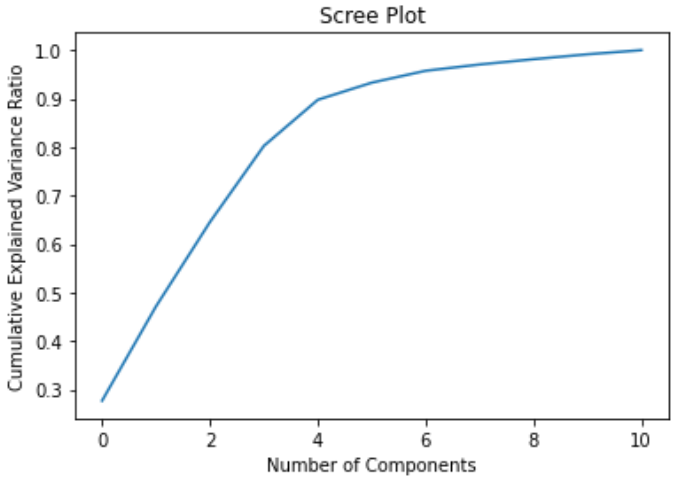


Figure 5: Plot of cumulative variance as a function of its number of featuresPCA

Looking at Table 6, it can be seen that the first characteristic (PC1) provides about 27% of the variance of the original data. In Figure 5, the cumulative variance diagram shows the point where the elbow is created, which corresponds to the four most important features of the PCA. In the next section, subsets of the PCA features are used to train the models mentioned in section 3.3.

4.3 Survival Prediction with Principal Components

Random Forest, GradientBoostingClassifier and SVM radial classification techniques were applied, using subsets of PCA principal components for training. Specifically, initially the performance of the methods was checked with the first 2 components, then with 4 and finally with 5 components. The obtained results are presented in Table 7.

	MCC	F1 Score	Accuracy	TPR	TNR	PR AUC	ROC AUC
Random Forest-PCA=2 (stratified_split, MinMaxScaler, iter=1000)	0.092	0.325	0.635	0.283	0.799	0.454	0.541
GradientBoostingClassifier-PCA=2 (stratified_split, MinMaxScaler, iter=1000)	0.127	0.331	0.656	0.274	0.833	0.471	0.554
SVC(radial)-PCA=2 (stratified_split, MinMaxScaler, iter=1000)	-0.002	0.000	0.683	0.000	0.999	0.651	0.499
Random Forest-PCA=4 (stratified_split, MinMaxScaler, iter=1000)	0.104	0.333	0.640	0.289	0.803	0.462	0.546
GradientBoostingClassifier-PCA=4 (stratified_split, MinMaxScaler, iter=1000)	0.067	0.303	0.628	0.261	0.798	0.436	0.529
SVM(radial)-PCA=4 (stratified_split, MinMaxScaler, iter=1000)	-0.027	0.006	0.673	0.004	0.983	0.554	0.494
RandomForestClassifier-PCA=5 (stratified_split, MinMaxScaler, iter=1000)	0.121	0.342	0.649	0.294	0.814	0.471	0.554
GradientBoostingClassifier-PCA=5 (stratified_split, MinMaxScaler, iter=1000)	0.080	0.315	0.632	0.272	0.798	0.446	0.535
SVC(radial)-PCA=5 (stratified_split, MinMaxScaler, iter=1000)	-0.019	0.060	0.663	0.038	0.952	0.421	0.495

Table 7: Mean values of the valuation measures using thePCA components

Based on the above table (Table 7), the best results were obtained using the GradientBoostingClassifier using the two main features of the PCA (MCC= + 0.127). Using more features does not seem to improve the performance of the models, not even features that coincide with where the elbow is created.

5. Conclusions

In this particular section, the conclusions drawn based on the results obtained from the above experiments and their comparison with the research results of Chicco, D., & Jurman, G [3] are mentioned. The conclusions were as follows:

➢ With the statistical descriptions of the categorical and numerical characteristics of the set data, the correctness of the corresponding research results was confirmed.

➢ By using all the characteristics of the data set, discrepancies were found in the performance of the models compared to those of the survey. These discrepancies are most likely due to different pre-processing of the data.

➢ The techniquePCA showed that the optimal number of components is four, however, the best performance was presented by a model trained with the two principal components.

➢ The results obtained with the use of its componentsPCA were much worse than those of the publication, in the case that the variables "serum creatinine" and "ejection fraction" were used. So the choice between using PCA or feature ranking depends on the specific problem and the nature of the data.

Bibliography

1. World Health Organization, World Heart Day. https://www.who.int/cardiovascular_diseases/world-heart-day/en/ . Accessed 7 May 2019.

2. The Guardian. UK heart disease fatalities on the rise for the first time in 50 years. https://www.theguardian.com/society/2019/may/13/heartcirculatory-disease-fatalities-on-risein-uk. Accessed 25 Oct 2019 .

3. Chicco, D., & Jurman, G. (2020). Machine learning can predict survival of patients with heart failure from serum creatinine and ejection fraction alone. BMC Medical Informatics and Decision Making, 20(1), 16. https://doi.org/10.1186/s12911-020-1023-5.

4. Breiman, L. (2001). Random forests. Machine learning, 45(1), 5-32.

5. Cortes, C., & Vapnik, V. (1995). Support-vector networks. Machine Learning, 20(3), 273-297.