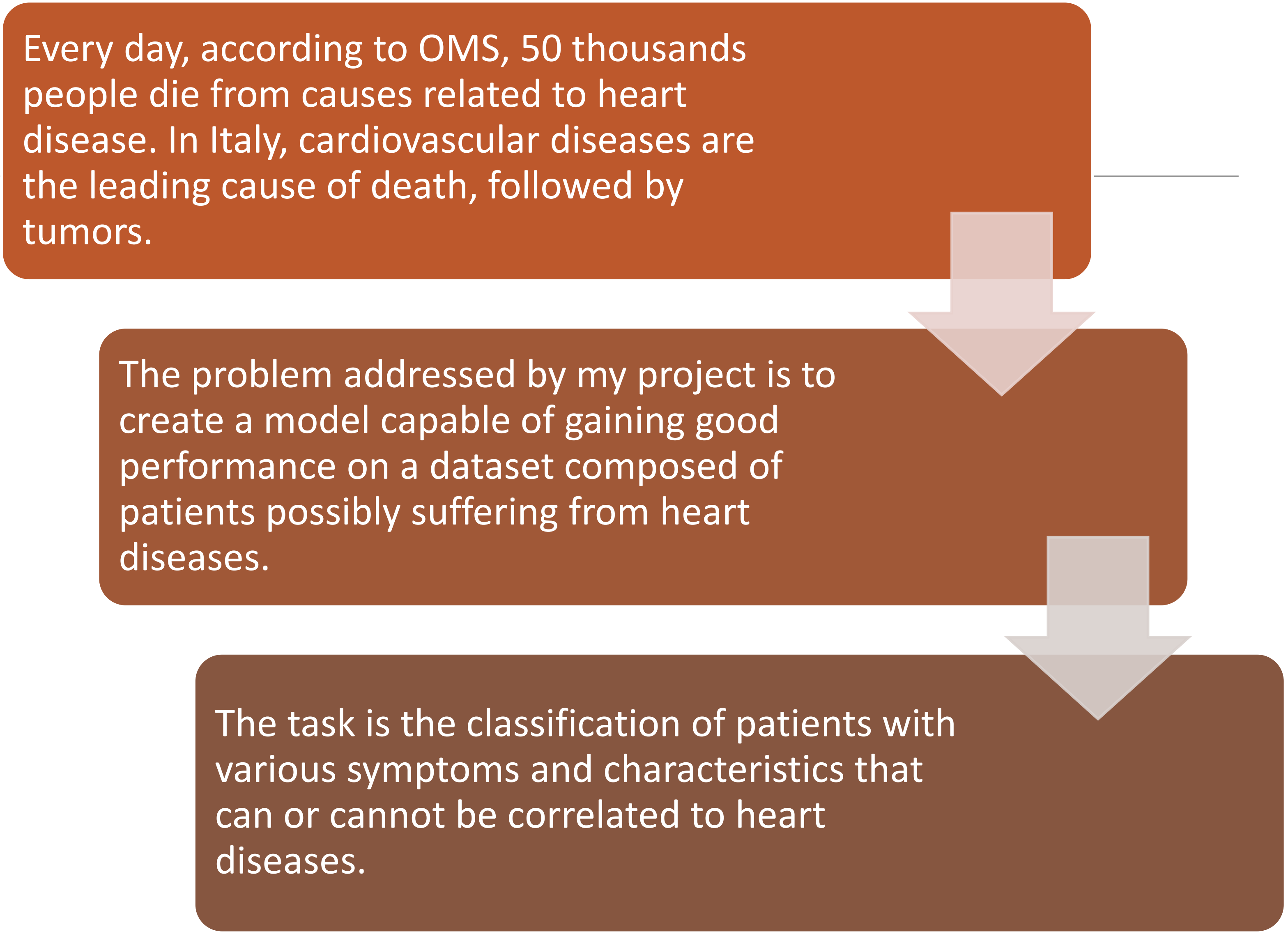
Comparison between ML models' performance on the classification task based on the UCI Parkinson telemonitoring Dataset

Motivations



Every day, according to OMS, 50 thousands people die from causes related to heart disease. In Italy, cardiovascular diseases are the leading cause of death, followed by tumors.

The problem addressed by my project is to create a model capable of gaining good performance on a dataset composed of patients possibly suffering from heart diseases.

The task is the classification of patients with various symptoms and characteristics that can or cannot be correlated to heart diseases.

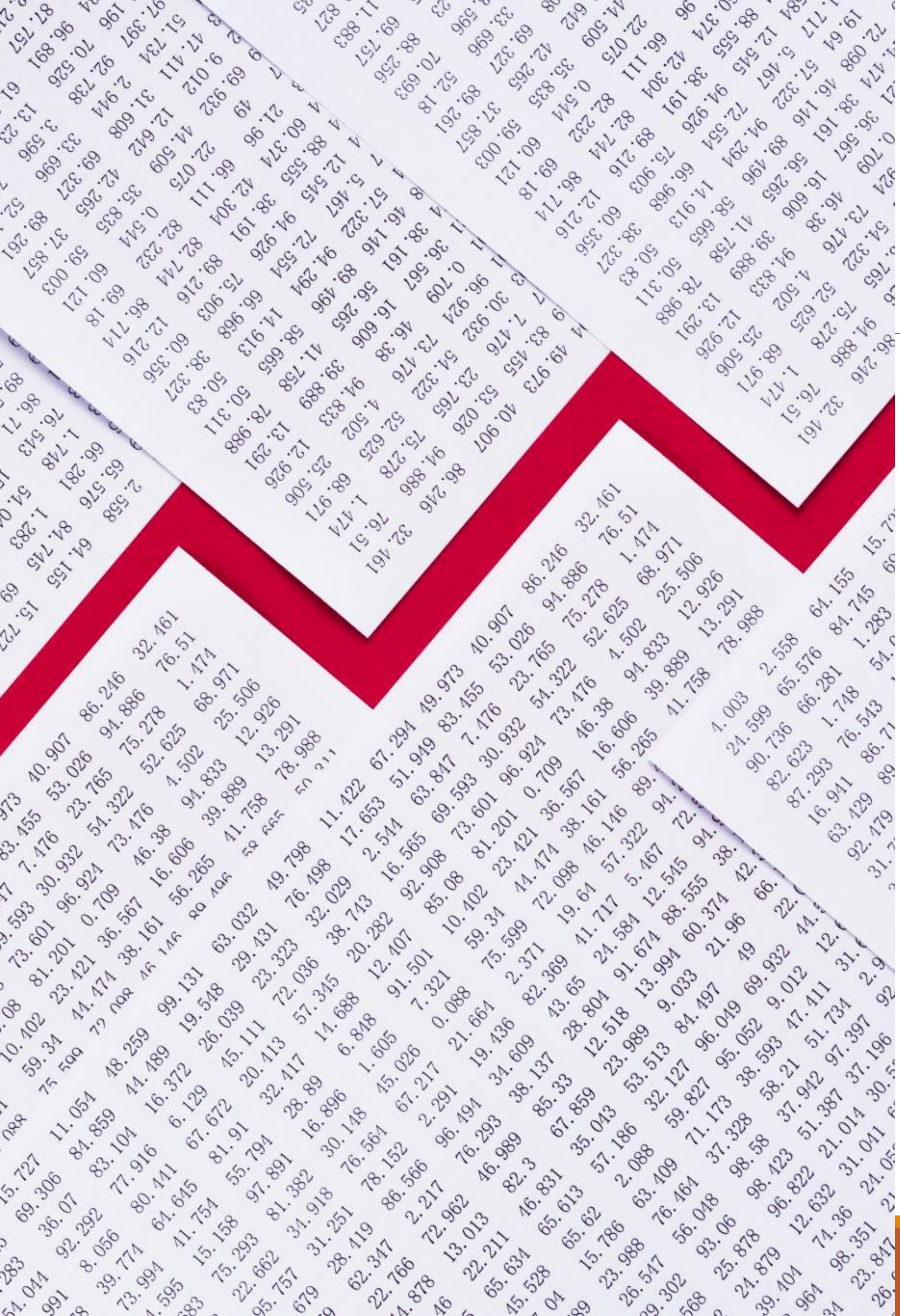
SOTA

State of the art with the UCI Heart Disease dataset can be reached with 3 different approaches:

Self-Attention Transformer: This model reached 96.51% accuracy on the Cleveland dataset. It is based on self-attention with a pre-trained Transformer fine-tuned on the dataset.

Extra Trees + Data Balancing: Combining oversampling and undersampling techniques with an Extra-Trees Classifier, this model reached 98.78% accuracy on this dataset. From the papers, it’s not clear if the results are independent or part of cross-validation.

Other results: Hybrid models like RF+Logistic Regression can reach 88.7% accuracy.



Preprocessing

The second phase is data preprocessing.

We deal with missing values by filling numerical features with the mean of the values and categorical features with the most frequent value.

Then, proceed with the one-hot encoding of the categorical features to process them as numerical values.

Finally, we check if there are still missing values; I chose to split the dataset in the code block of each model, because some models require validation while others don’t.

For each model 2 variants are implemented: **Feature selection** and **PCA.**

MODELS

I decided to make the project modular: each model has a function to run it named run_nameOfModel, making it easier to scale or update the work.

The model I used are 3: k-nn, Random forest and a SVM classifier.

For each model I tried some standard values for the hyperparameters, and chosen the ones that gave the best performance

K-NN: This model consists of a classifier that determines the class of a point based on the k nearest neighbors of that point. The distance measure is the simple Euclidean distance, possible using a Standard scaler.

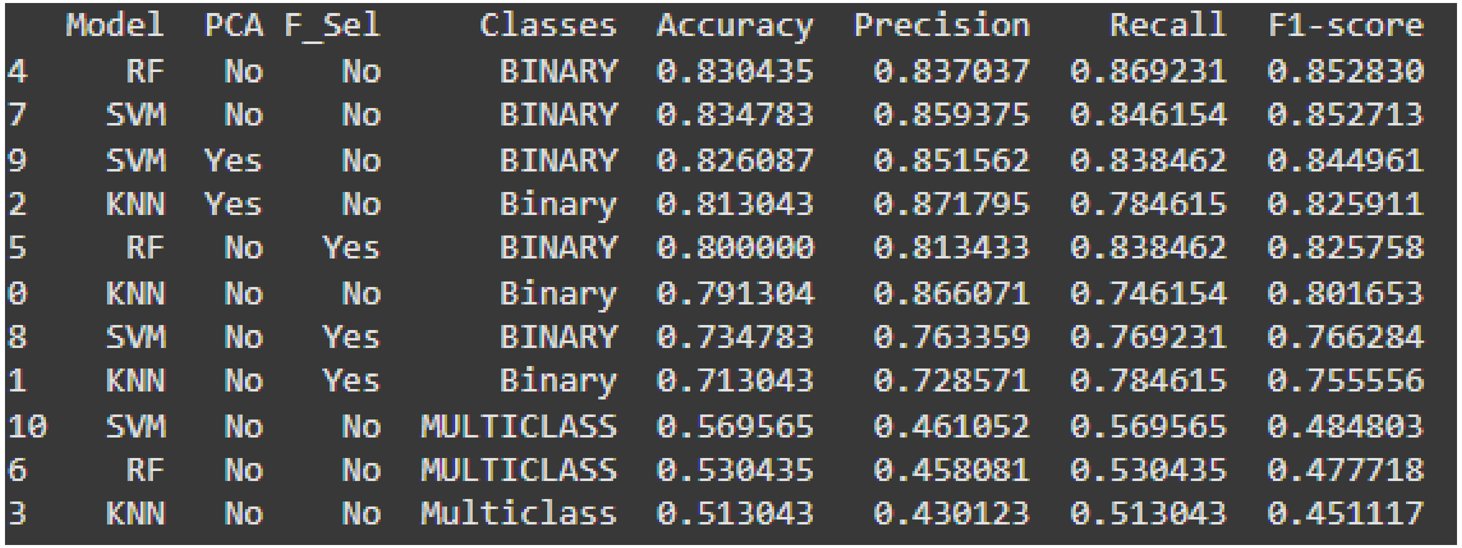
SVM: SVM classifier divides into 2 branches: linear and non-linear. In our case data is complex, and we need to use a non-linear kernel for the Support Vector Machine.

So first apply scaling on the data, then we consider an RBF kernel, which has 2 hyperparameters to optimize: **C** and **gamma**. The model implemented uses C=1.0 and gamma='scale'.

Random Forest: This model instead uses a random forest, which is an ensemble of decision trees.

Each tree is trained on a bootstrapped version of the training set (sampled with replacement), and the final prediction of the forest is obtained by averaging the prediction of each tree in the ensemble.

Results



```
   Model  PCA F_Sel      Classes  Accuracy  Precision    Recall  F1-score
4     RF   No    No       BINARY  0.830435   0.837037  0.869231  0.852830
7    SVM   No    No       BINARY  0.834783   0.859375  0.846154  0.852713
9    SVM  Yes    No       BINARY  0.826087   0.851562  0.838462  0.844961
2    KNN  Yes    No       Binary  0.813043   0.871795  0.784615  0.825911
5     RF   No   Yes       BINARY  0.800000   0.813433  0.838462  0.825758
0    KNN   No    No       Binary  0.791304   0.866071  0.746154  0.801653
8    SVM   No   Yes       BINARY  0.734783   0.763359  0.769231  0.766284
1    KNN   No   Yes       Binary  0.713043   0.728571  0.784615  0.755556
10   SVM   No    No   MULTICLASS  0.569565   0.461052  0.569565  0.484803
6     RF   No    No   MULTICLASS  0.530435   0.458081  0.530435  0.477718
3    KNN   No    No   Multiclass  0.513043   0.430123  0.513043  0.451117
```

We can see in the table that multiclass classification doesn’t suit very well this dataset, which can be used only for the binary task.

We can also see that with the binary task the models doesn’t perform well too, the results achieved are mediocre.

The cause can be the size of the dataset, and making it bigger could lead to better performance.

Conclusions

With this work I tried to compare performance of classic built-in classifiers on the UCI heart dataset, and see which one performs the best.

The models we used for a comparison are KNN,SVM and RF, all of the models had a similar performance on the binary task with or without PCA, and bad performance on multiclass task, this because the dataset is to unbalanced to predict all the classes in an acceptable way.

The use of Feature selection instead of PCA led to slight worst performance.

Possible future improvements should focus on changing to a bigger dataset or expanding the UCI one, because failures and mediocre performance are caused by the small size of it.