Comparison of Heart Disease Prediction Methods. Usage of Effective Machine Learning Techniques.

Bohdan Darmits

Abstract (Problem description and importance).

According to different sources heart disease is one of the greatest enemy of humanity. Different medical scientist say that it kills approximately one person every minute and it doesn't divide us on groups by race, sex or age, heart disease is a great thread to all of us. That's what induces scientists all around the world think how to deal with this problem and try to predict it instead of suffering the consequences. That's how the area of heart disease predation became one of the biggest area for predictions. A great variety of machine learning techniques have been used with different level of accuracy and success. But which of these techniques works better on todays most common datasets, and how the results will differ from the previous ones? Let's find out!

Data investigation and preprocessing.

Preprocessing:

To make every next step understandable, here is the explanation of all columns in dataset:

1. age – the age of patient, (int number);

2. sex – the sex of patient, (int number, 1 – male, 0 - female);

3. chest_pain_type – type of pain in chest, (int number, 1 – 4 means different types of pain);

4. blood_pressure – blood pressure, (int number);

5. cholestoral – cholestoral level, (int number);

6. blood_sugar – level of sugar in blood, (int number, 1 – not normal, 0 - normal);

7. electrocardiographic_results – results of electrocardiographic after a rest, (int number, 0 – 2 means different types);

8. maximum_heart_rate – maximum heart rate, (int number);

9. exercise_induced_angina – does exercises induce angina? (int number, 1- true, 0 - false);

10. st_depression - depression, by percentage of sport exercises, (int number, 1 – true, 0 – false);

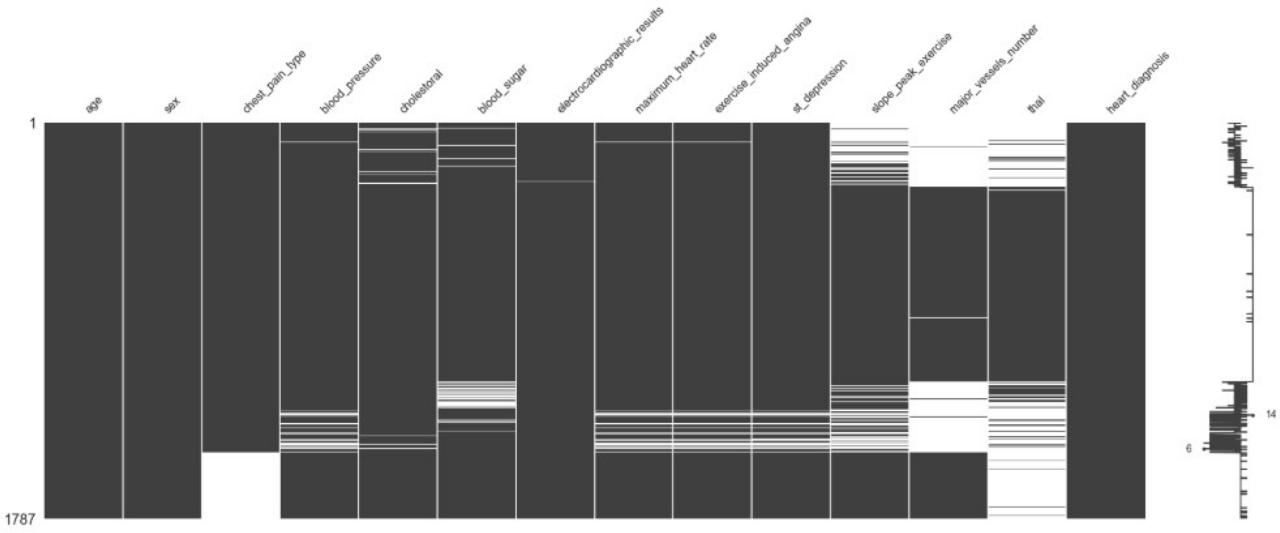
11. slope_peak_exercise – slope peak exercise segment, (float number, 1 – 5 means different types);

12. major_vessels_number – number of vessels, (int number, 0 – 3);

13. thal – defects, (int number, 3,6,7 means different types);

14. heart_diagnosis – does the patient have any heart diseases? (target variable, dependent), (int number, 1- true, 0 - false).

Firstly, Data Cleaning and Missing Data Imputation was done. Since I had few datasets with different number of rows, many data points were wrong and uncompleted at this stage. 11 datasets were combined in one. To do this I had to make a transformation of non-numeric columns into numeric ones, fit the representation of data in columns into one standard. Some columns in some sets were removed. After gluing the datasets into one base I faced the new problem. On exploratory data analysis stage (EDA), a missing values check showed far not the best result:



age : 0.0%; sex : 0.0%; chest_pain_type : 16.73195299384443 %; blood_pressure : 3.3%; cholestoral : 1.6%; blood_sugar : 4.9%; electrocardiographic_results : 0.1%; maximum_heart_rate : 3.07%; exercise_induced_angina : 3.07%; st_depression : 3.4%; slope_peak_exercise : 17.1%; major_vessels_number : 33.9%; thal : 42.8%;; heart_diagnosis : 0.0 %

The picture above shows the missing data with white gaps. Columns: 'blood_pressure', 'cholestoral', 'maximum_heart_rate', 'st_depression' had values that could vary from 0 to some max values. In these columns all blank cells were replaced by mean value. In all others, values were ranging for example from 1 to 4 or had three options like 3, 6, 7 with specific meaning for each digit. In these cases I decided to find the mode values in column for the values 1 and 0 in corresponding raws in target column 'heart_diagnosis'.

There is a list of other methods of imputing missing values, such as: Hot deck imputation – when user needs to find all the sample subjects who are similar variables, then randomly choose one to replace the missing one; Cold deck imputation – made by systematically chosen value from an individual who has similar values on other variables; Regression imputation – prediction of value based on other variables; Stochastic regression imputation – same prediction plus random residual value. All these methods were investigated and tried. But in this case due to the small amount of Nones in columns 'blood_pressure', 'cholestoral', 'maximum_heart_rate', 'st_depression', and specific data in other columns I decided that the method I chose, is the best one.

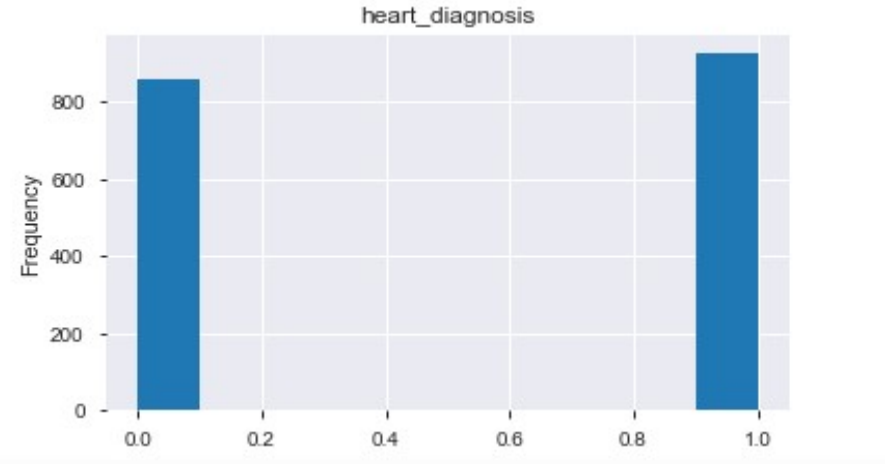
The final dataset was a result of combination of 11 datasets and an information collected from additional resources. This is Data Integration. At this part, all required work was done. Firstly, three main techniques types of data mapping were tried. Manual Data Mapping, Semi Automated Data Mapping and Automated Data Mapping. For this internet resources and SQL tables were used. Secondly task – schemas matching. At this step five approaches were discovered: Database Schema Matching Using Machine Learning with Feature Selection - this method improves on its predecessors by including one-to-one attribute matching rather than just matching one attribute with a set of possible attributes, it still has the same problem that it does not consider the possibility of one attribute matching to a set of attributes; Semantic Integration in Heterogenous Databases using Neural Network - this method implemented schema matching using Machine Learning

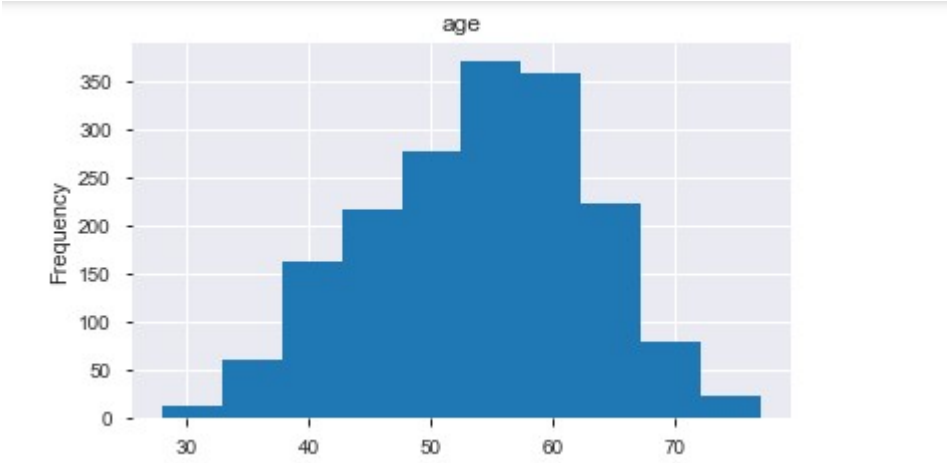
approach. Known as SemaInt. Provides with a similarity mapping of each attribute in one schema with a set of attributes in another, it does not consider the fact one might map to a set of others as well; Corpus-based Schema Matching - this method considers one to one matching of attributes and cannot make mappings like one to many or many to one; Generic Schema Matching with Cupid - a technique of matching which is schema based and not instance based. In the proposed method, heirarchical schemas are represented as trees and nonheirarchical schemas are generalied as graphs. This method maintains a thesaurus for finding linguistic similarity and also makes use of information like schema structure and relation of attributes with each when assigning scores; iMAP:Discovering Complex Semantic Matches between Database Schemas - iMAP is a new method of performing both one to one and one to many schema matching by converting the matching problem to a search problem in a relatively large search space of all possible schema mappings. For efficient searching custom searchers based on concatenation of text, arithmetic operations over numeric attributes etc. are used, and scoring each match to find the best possible matchings. Since the searchers are customized over type of data, they only search through a subset of search space and by this reducing system complexity. This method achieves one to many mapping. It requires a domain expert for creating custom searchers specific to a particular type of database. The method also makes use of only the data contained in the tables and not the schema names themselves. Within Data Integration stage, finding redundant attributes was done by correlation and outliers were found by using Q grams score IQR. There was no need of $\chi 2$ test because it is used when data in nominal. Here are the results of finding outliers:

```
Out[26]: age                          False
         blood_pressure               False
         blood_sugar                  False
         chest_pain_type              False
         cholestoral                  False
         electrocardiographic_results False
         exercise_induced_angina      False
         heart_diagnosis              False
         major_vessels_number         False
         maximum_heart_rate           False
         sex                          False
         slope_peak_exercise          False
         st_depression                False
         thal                         False
         dtype: bool
```

In addition to that, the PCA algorithm was implemented to find outliers. The principles will be explained a bit later. All tests gave negative result. Also some

frequency tests are shown below. As you can see, distribution of data on patients with positive and negative results regarding heart problems are almost the same.



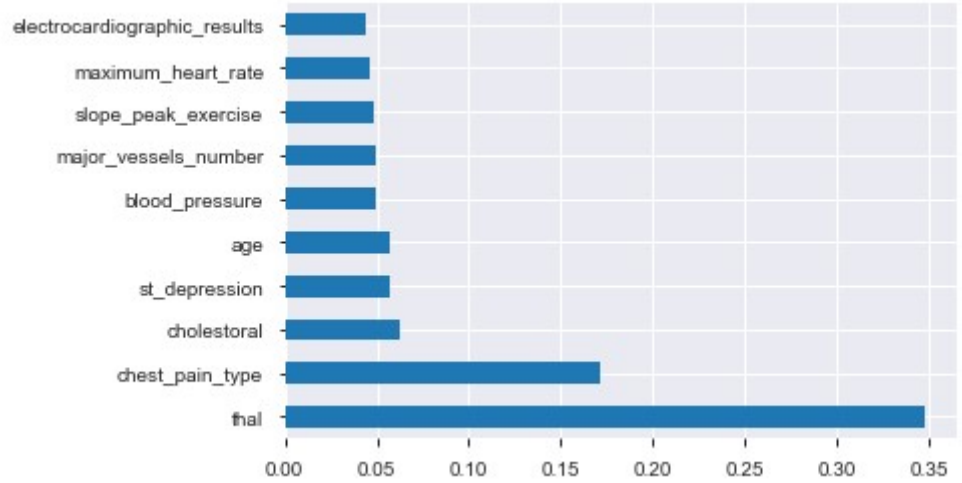


The correlation Analysis can be done by counting the coefficients. -1 strongly negative correlation, 1 – strongly positive correlation. One of the best methods for finding correlations in pearson method:

Out[35]:

	age	sex	heart_diagnosis
age	1.000000	-0.026351	0.175799
sex	-0.026351	1.000000	0.199629
heart_diagnosis	0.175799	0.199629	1.000000

All these features are parts of EDA that was mentioned earlier. The last, but very important step of EDA is Feature Importance Analysis. And this is the result:



The third step, to talk about is Data Transformation. The biggest attention was given to Rank Transformation and BoxCox transformation. Rank is good to obtain a variable that is will behave like a normally distributed one. BoxCox has similar intentions. It aims to transform a continuous variable into an almost normal distribution. It doesn't like negative values. In my investigation I decided to try rank transformation to see the influence of this decision on the results.

How do I unify and scale data? Data Normalization – the answer and the forth step. Insertion, Updation and Deletion problems – is the list of what might happen without proper normalization. In notebook a few ways of Data Normalization were applied. A MinMax function taken from .py library. Min-max normalization has only one significant con - it does not handle outliersIf set will have 99 variables in one range and one variable out of it – this one will have a great affect on transformation especially if it is much bigger than others. But we don't have any, so we shouldn't be scary of it.

Related works:

Accuracy obtained using different techniques:

2015	Priti Chandra et al.	Computational Intelligence Technique for early	Naïve Bayes	86.29%
2015	Cemil et al.	Propose application of knowledge discovering process on prediction of stroke patients	ANN	81.82% for training dataset 85.9% for test data set
			SVM	80.38% for train data set 84.26% for test data set
2016	Muhammad Saqlain et al.	Identification of Heart Failure by Using Unstructured	Logistic Regression	80.00%
			Neural Network	84.80%
			SVM	83.80%
			Random Forest	86.60%
			Decision Tree	86.60%
			Naïve Bayes	87.70%

2016	Marjia et al.	Heart disease prediction using WEKA tool and 10-Fold cross-validation	KStar	75%
			J48	86%
			SMO	89%
			Bayes Net	87%
			Multilayer Perceptron	86%
2016	Dr. S. Seema et al.	Predict chronic disease by mining the data containing in historical health records	Naïve Bayes Decision tree Support Vector Machine (SVM)	Highest accuracy in case of heart disease 95.556% is achieved by SVM.
2016	Tapas RanjanBaithar u	Analysis of Data Mining Techniques For Healthcare Decision Support System Using Liver Disorder Dataset	J48	68.97%
			ZeroR	57.97%
			Multilayer Perceptron	71.59%
			IBK	62.90%
			Naïve Bayes	55.36%
			VFI	60.29%
2016	VidyaK.Sudarshan et al.	Application of higher-order spectra for the characterization of coronary artery disease using electrocardiogram signals.	KNN	98.17%
			Decision Tree (DT)	98.99%
2016	Ashok Kumar Dwivedi	Evaluate the performance of different machine learning	Naïve Bayes	83%

Y. Alp Aslandogan, et. al. (2004), worked on classifiers called K-nearest Neighbour (KNN), Decision Tree, Naïve Bayesian and used Dempsters' rule This classification based on the combined idea showed very good accuracy (Y. Alp Aslandoganet. al.," Evidence Combination in Medical Data Mining", Proceedings of the international conference on Information Technology: Coding and Computing (ITCC'04) 0-7695-2108-8/04©2004 IEEE.).

Franck Le Duff (2004), worked on creating Decision tree with clinical data. He investigated and as a result gave data mining techniques which can help doctors to predict the survival of patients. (Franck Le Duff, CristianMunteanu, Marc Cuggiaa, Philippe Mabob, "Predicting Survival Causes After Out of Hospital Cardiac Arrest using Data Mining Method", Studies in health technology and informatics, Vol. 107, No. Pt 2, page no. 1256-1259, 2004 Boleslaw Szymanski, et. al. (2006), operated)

Niti Guru, et. al. (2007), was one of the first scientists who tried to predict

heart by using neural systems. Controlled network was used for analysis of heart diseases. Training was made by back-propagation technique. (Niti Guru, Anil Dahiya, NavinRajpal, "Decision Support System for Heart Disease Diagnosis Using Neural Network", Delhi Business Review, Vol. 8, No. 1, January - June 2007).

(http://www.statisticssolutions.com/multiple-linear-regression/) - The researchers developed tool to analyze the occurrence of chances of coronary disease. Two layered neuro-fuzzy approach is developed to predict occurrences heart disease. The implementation of this approach produced fault rate very low and a high work efficiency in performing analysis for heart disease prediction.

(Soni, J., Ansari, U., Sharma, D., & Soni, S. (2011) - Predictive data mining for medical diagnosis: An overview of heart disease prediction. International Journal of Computer Applications, 17(8), 43-48.) Six medical attributes are used to predict heart diseases and produced more accurate and efficient results. In this work three classifiers are used: Naive Bayes, Classification by clustering and Decision Tree. The achieved accuracy: 96.5%, 88.3% and 99.2% respectively.

(Qureshi, M. A. (2017). Comparative Study of Existing Techniques for Heart Diseases Prediction using Data Mining Approach. Asian Journal of Computer Science and Information Technology, 7, 50-56.) - Comparison of different data mining techniques. 13 attributes were used. Models developed and validated by using five algorithms: C5.0, Neural Network, Support Vector Machine (SVM), KNearest Neighborhood (KNN) and Logistic Regression. The accuracy is 93.02%, 80.23%, 86.05%, 88.37% respectively.

(DeepaliChandna “Diagnosis of Heart Disease Using Data Mining Algorithm”, IEEE Conf. on International Journal of Computer Science and Information Technologies, 2015, pp 1678-1680) - Algorithm that was explained uses search constraints to reduce the number of rules, searches for association rules on a training set and validates them on test set. Researchers applied association rules to predict heart diseases in more accurate manner. In medical terms, association rules relate heart perfusion measurements and risk factors to the degree of disease in four specific arteries. Search constraints and test set validation significantly reduced the number of association rules and produced a set of rules with high heart disease predictive accuracy. (2006).

Also many other web sites and books were found and investigated in order to achieve the best result.

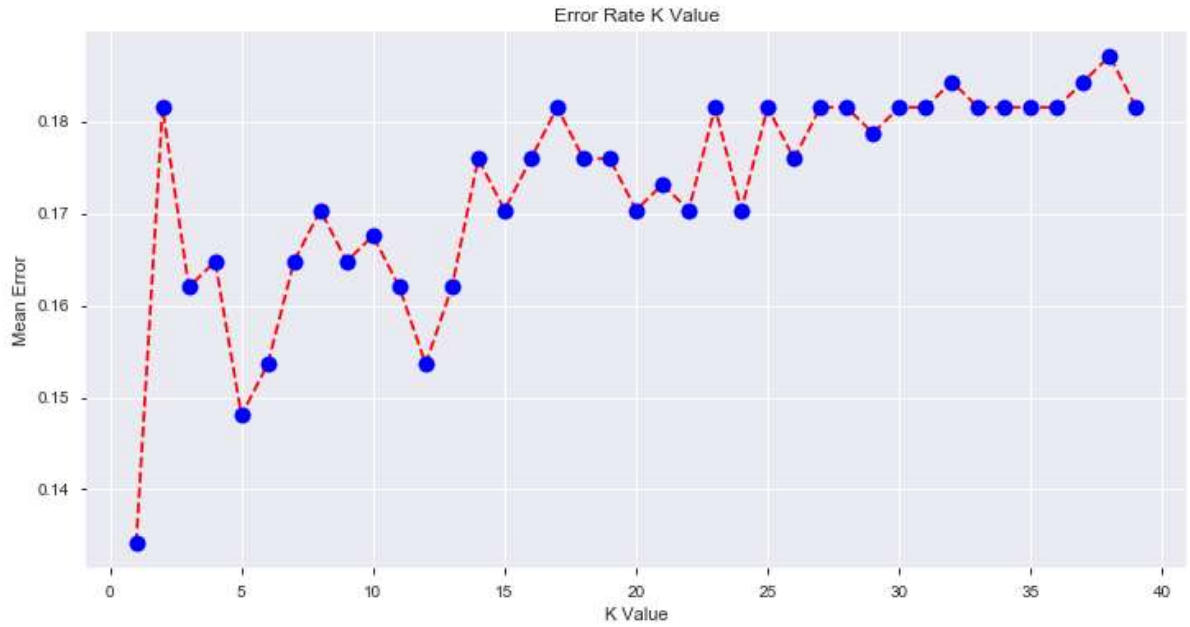
Solution.

Predictions were made using Linear regression, KNN, PCA with random forest algorithm, LDA with random forest algorithm, Descision tree, Gradient Boosting Classifier, XGBoost Classifier, ADA boost with Decision tree Base Estimator and with SVC Base Estimator, SVM, Naive Bayes, Neural Network MLPC

Linear regression – is used to show relations between one dependent variable and one or few independent. Our dependent variable was prostitution Status, independent – all others, except country and region. Linear regression is good when the relationship to between covariates and response variable is known to be linear. A clear disadvantage is that Linear Regression simplifies many problems. And that's why can't give a clear result with high accuracy. Very often covariates and response variables don't exhibit a linear relationship.

KNN - supervised machine learning algorithm. It calculates the distance of a new data point to all other training data points. The distance can be of any type like Euclidean or Manhattan. Then the algorithm selects the K-nearest data points, where K can be any integer. Finally, it assigns the data point to the class to which the majority of the K data points belong. Pros: Since the algorithm requires no training before making predictions, new data can be added during work; It is lazy learning algorithm and therefore requires no training prior to making real time predictions. That's why this algorithm is vey fast. Cons: The KNN algorithm doesn't work well with high dimensional data because with large number of dimensions, it becomes difficult for the algorithm to calculate distance in each dimension; The KNN algorithm has a high prediction cost for large datasets. This is because in large datasets the cost of calculating distance between new point and each existing point becomes higher; KNN algorithm doesn't work well with categorical features since it is difficult to find the distance between dimensions with categorical features.

Also the algorithm for finding the best k value vas written and applied to all trials. Points where the mean error is the lowest are the best K – values. It can be seen on the picture below.



PCA - Principal Component Analysis is a technique that is used to transform high dimensional data to low dimensional by selecting the most important features that capture maximum information about the dataset. The training time of the algorithms reduces significantly with smaller number of features. But it is mandatory to normalize features before applying PCA

LDA – Linear discriminant analysis is one more technique for reducing dimension level of the set. Unlike PCA, LDA tries to reduce dimensions of the feature set while retaining the information that discriminates output classes. LDA tries to find a decision boundary around each cluster of a class. Then LDA projects the data points to new dimensions in a way that the clusters are as separate from each other as possible and the individual elements within a cluster are as close to the centroid of the cluster as possible. The new dimensions are ranked on the basis of their ability to maximize the distance between the clusters and minimize the distance between the data points within a cluster and their centroids. These new dimensions form the linear discriminants of the feature set. In simple words, we just need to calculate the separability between different classes, calculate the variance in all classes, construct the lower-dimensional space that maximizes separability and minimizes variance…

PCA - unsupervised algorithm. It ignores class labels and wants to find the principal components that maximize variance in a given set of data. Linear Discriminant Analysis, on the other hand, is a supervised algorithm that finds the linear discriminants that will represent those axes which maximize separation between different classes.

After applying these two algorithms to the sets, I will try to make predictions by using random forest classification algorithm.

Random Forest Algorithm: Pick N random records from the dataset. Build a decision tree based on these N records. Choose the number of trees you want in your algorithm and repeat previous steps.

In case of a regression problem, for a new record, each tree in the forest predicts a value for Y (output). The final value can be calculated by taking the average of all the values predicted by all the trees in forest. Or, in case of a classification problem, each tree in the forest predicts the category to which the new record belongs. Finally, the new record is assigned to the category that wins the majority vote. Pros: The random forest algorithm is not biased. It works well when dataset has both categorical and numerical features. The random forest algorithm also works well when data has missing values or it has not been scaled well.

Descision tree - For each attribute in the dataset, the decision tree algorithm forms a node, where the most important attribute is placed. For evaluation we start at the root node and work our way down the tree by following the corresponding node that meets our condition or "decision". This process continues until a leaf node is reached, which contains the prediction or the outcome of the decision tree. Pros: it needs very small effort to train algorithm, very fast and efficient compared to KNN and other classification algorithms.

Gradient Boosting Classifier - Gradient boosting systems have two parts “learners”: a weak learner and an additive component. Decision trees are used as weak learners. A special procedure is a part of algorithm, special for minimizing the error between given parameters. This is done by taking the calculated loss and performing gradient descent to reduce that loss. In the end, parameters of the tree are modified to reduce the residual loss. The new tree's output is then appended to the output of the previous trees. This process will be repeated until a previously specified number of trees is reached, or the loss is reduced below a certain threshold.

XGBoost – is a bit modified and customized version of Gradient boosting.

ADA boost - Ada-boost or Adaptive Boosting is one of ensemble boosting classifier proposed by Yoav Freund and Robert Schapire in 1996. AdaBoost classifier builds a strong classifier by combining multiple poorly performing

classifiers so that you will get high accuracy classifier. Pros: allows to use many classifiers, to combine and improve their strength. Cons: Is sensitive to noise and outliers. Is quite slow.

Naive Bayes – This just approach based on Bayes theorem. Pros: Easy to implement and often outperforms more complicated algorithms, works very good with categorical data and numerical data, can be used to perform regression by using Gaussian Naive Bayes.

MLPC – Neural network. All artificial NN are inspired by human neural networks, and percentron. Has few phases: Feed-Forward, has 3 steps, the predicted output is not necessarily correct; it can be wrong, and we need to correct it. Back Propagation is the second step. The cycle of feed-forward and back propagation is called one "epoch". This process continues until a reasonable accuracy is achieved.

Results and evaluation.

A table comparing all results with the results of paper that were investigated

	Normalized	Rank	Avarage results of others
Linear Regression	73.9	76.4	86
KNN (k = 11)	91.3	89.3	80
PCA with random forest algorithm	72.9	80.1	-
LDA with random forest algorithm	81.8	85.1	-
Descision tree	86.5	90.5	87
Gradient Boosting Classifier	81.2	88.8	-
XGBoost Classifier	84.07	91.6	-
ADA boost with Decision tree Base Estimator	77.9	85.7	83
ADA boost with SVC Base Estimator(linear)	60.6	-	83
SVM (linear)	81.005	-	91
Naive Bayes	78.2	77.9	86
MLPC	88.5	81.2	82

The results are pretty good, it is seen that the “Rank results” are better in all cases except KNN, Bayes and MLPC because of the algorithms specifics, described in the algorithms description. Unexpectedly the worst result is shown by linear regression. The best in “Normalized” column is KNN, and the best in all – XGBBoost classifier, SVC and SVM for “Rank” are missed because of some specifics described higher. To compare with results of relevant works, the score is 4:4, not counting the additional algorithms, I didn’t find were implemented previously.

These algorithms were chosen because of their versatility, it's like a competition between different ways of prediction. And the comparing is the most interesting. This work gave a clear answer, what types of implementations are best – working for such datasets, and I am sure this could help in future work on predicting diseases.

Code: https://github.com/Bdarmits/Hear-Disease-Prediction-Ai-Coursework/

Datasets: https://drive.google.com/drive/folders/1BxWX-JFJ73ZEATz8FYtkwg6-aVsuTTfk?usp=sharing/