Advanced Data Science Capstone Project: Predicting cardiovascular events.

Architectural Decision Document (ADD)

Prepared by: Maxim Lukin
As of date: 2/10/2020

1.1 Data Source

1.1.1 Technology Choice

For current case study Heart Disease Data Set has been chosen. Here is a link to Kaggle: https://www.kaggle.com/ronitf/heart-disease-uci
Creators:
1. Hungarian Institute of Cardiology. Budapest: Andras Janosi, M.D.
2. University Hospital, Zurich, Switzerland: William Steinbrunn, M.D.
3. University Hospital, Basel, Switzerland: Matthias Pfisterer, M.D.
4. V.A. Medical Center, Long Beach and Cleveland Clinic Foundation: Robert Detrano, M.D., Ph.D.

1.1.2 Justification

This database contains 76 attributes, but all published experiments refer to using a subset of 14 of them. In particular, the Cleveland database is the only one that has been used by ML researchers to this date. The "goal" field refers to the presence of heart disease in the patient. It is integer valued from 0 (no presence) to 4.

1.2 Enterprise Data

1.2.1 Technology Choice

IBM Watson Studio has been used as cloud-based enterprise solution. Heart Disease Data Set has been extracted from the web source above and have been uploaded and stored in IBM Watson Studio.

1.2.2 Justification

IBM Watson Studio is an integrated environment designed to make it easy to develop, train, manage models, and deploy AI-powered applications and is a SaaS solution delivered on the IBM Cloud. It is evolving Data Science Experience on IBM Cloud with lot of new features to build AI applications.

1.3 Streaming analytics

1.3.1 Technology Choice

Streaming analytics (such as real-time data) do not require application.

1.3.2 Justification

Due to the nature of Heart Disease Data Set there is no need to apply any streaming analytics.

1.4 Data Integration

1.4.1 Technology Choice

Heart Disease Data Set from Kaggle has been initially available as a cleansed and transformed/adopted dataset.

1.4.2 Justification

During ETL process dataset has been checked once again for any possible missing values in dataframe and inappropriate attributes formats.

1.5 Data Repository

1.5.1 Technology Choice

IBM Watson Object Storage and GitHub have been used during the project as data repositories.

1.5.2 Justification

Due to the nature of integrated environment of IBM Watson, IBM Watson Object Storage has been used for data storing purposes. Additionally, personal GitHub repository has been integrated for secure purposes.

1.6 Discovery and Exploration

1.6.1 Technology Choice

Initially, Heart Disease Data Set is stored in .csv format on the kaggle.com.

1.6.2 Justification

Data set contains of the following attributes:
> 1. age
> 2. sex
> 3. chest pain type (4 values)
> 4. resting blood pressure
> 5. serum cholestoral in mg/dl
> 6. fasting blood sugar > 120 mg/dl
> 7. resting electrocardiographic results (values 0,1,2)
> 8. maximum heart rate achieved
> 9. exercise induced angina
> 10. oldpeak = ST depression induced by exercise relative to rest
> 11. the slope of the peak exercise ST segment
> 12. number of major vessels (0-3) colored by flourosopy

Reference: https://github.com/IBM/coursera/blob/master/coursera_capstone/AI_Methods_Overview_Romeo_Kienzler_v3.pdf
Article: The IBM Data and Analytics Reference Architecture, Page 13.

> 13. thal: 3 = normal; 6 = fixed defect; 7 = reversable defect
The names and social security numbers of the patients were recently removed from the database, replaced with dummy values. One file has been "processed", that one containing the Cleveland database. All four unprocessed files also exist in this directory.

1.7 Actionable Insights

1.7.1 Technology Choice

Based on the capstone project requirements and for educational purposes the following algorithms have been used:

- Logistic Regression,
- Support Vector Machines,
- Linear Support Vector Machines (SVC),
- k-Nearest Neighbors algorithm (KNN),
- Gaussian Naive Bayes,
- Perceptron,
- Stochastic Gradient Descent,
- Decision Tree Classifier,
- Random Forest,
- Ridge Classifier.

All these algorithms have been compared between each other with Train and Test Scores (accuracy).
After, Keras Sequential Neutral Network Model with one additional iteration has been deployed.

1.7.2 Justification: Selected algorithms

Logistic regression is a statistical model that in its basic form uses a logistic function to model a binary dependent variable, although many more complex extensions exist. It is used to model the probability of a certain class or event existing such as pass/fail, win/lose, alive/dead or healthy/sick.

Support-vector machines (SVMs) are supervised learning models with associated learning algorithms that analyze data used for classification and regression analysis. Given a set of training examples, each marked as belonging to one or the other of two categories, an SVM training algorithm builds a model that assigns new examples to one category or the other, making it a non-probabilistic binary linear classifier (although methods such as Platt scaling exist to use SVM in a probabilistic classification setting).

K-Nearest Nneighbors algorithm (k-NN) is a non-parametric method used for classification and regression. k-NN is a type of instance-based learning, or lazy learning, where the function is only approximated locally and all computation is deferred until classification.

Bayes classifiers are a family of simple "probabilistic classifiers" based on applying Bayes' theorem with strong (naïve) independence assumptions between the features. They are among the simplest Bayesian network models. Naive Bayes is a simple technique for constructing classifiers: models that assign class labels to problem instances, represented as vectors of feature values, where the class labels are drawn from some finite set. There is not a single algorithm for training such classifiers, but a family of algorithms based on a common principle: all naive Bayes classifiers assume that the value of a particular feature is independent of the value of any other feature, given the class variable.

Perceptron is an algorithm for supervised learning of binary classifiers. A binary classifier is a function which can decide whether or not an input, represented by a vector of numbers, belongs to some specific class. It is a type of linear classifier, i.e. a classification algorithm that makes its predictions based on a linear predictor function combining a set of weights with the feature vector.

Stochastic gradient descent (SGD) is an iterative method for optimizing an objective function with suitable smoothness properties (e.g. differentiable or subdifferentiable). It can be regarded as a stochastic approximation of gradient descent optimization, since it replaces the actual gradient (calculated from the entire data set) by an estimate thereof (calculated from a randomly selected subset of the data). Especially in big data applications this reduces the computational burden, achieving faster iterations in trade for a slightly lower convergence rate.

Decision tree learning is one of the predictive modeling approaches used in statistics, data mining and machine learning. It uses a decision tree (as a predictive model) to go from observations about an item (represented in the branches) to conclusions about the item's target value (represented in the leaves). Tree models where the target variable can take a discrete set of values are called classification trees; in these tree structures, leaves represent class labels and branches represent conjunctions of features that lead to those class labels. Decision trees where the target variable can take continuous values (typically real numbers) are called regression trees.

Random forest is a learning method for classification, regression and other tasks that operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees. Random decision forests correct for decision trees' habit of overfitting to their training set.

Tikhonov regularization, named for Andrey Tikhonov, is a method of regularization of ill-posed problems. Also known as **ridge regression** it is particularly useful to mitigate the problem of multicollinearity in linear regression, which commonly occurs in models with large numbers of parameters. In general, the method provides improved efficiency in parameter estimation problems in exchange for a tolerable amount of bias (see bias–variance tradeoff). This classifier first converts the target values into {-1, 1} and then treats the problem as a regression task (multi-output regression in the multiclass case).

1.7.3 Justification: Selected frameworks

NumPy is a library for the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays.

Pandas is a software library written for the Python programming language for data manipulation and analysis. In particular, it offers data structures and operations for manipulating numerical tables and time series.

Matplotlib is a Python 2D plotting library which produces publication quality figures in a variety of hardcopy formats and interactive environments across platforms. Matplotlib can be used in Python scripts, the Python and IPython shells, the Jupyter notebook, web application servers, and four graphical user interface toolkits.

Seaborn is a Python data visualization library based on matplotlib. It provides a high-level interface for drawing attractive and informative statistical graphics.

Scikit-learn is a free software machine learning library for the Python programming language. It features various classification, regression and clustering algorithms including support vector machines, random forests, gradient boosting, k-means and DBSCAN, and is designed to interoperate with the Python numerical and scientific libraries NumPy and SciPy.

Scikit-learn is used for the following matter:

- Splitting initial data set into train and test,
- performing cross validation technic,
- performing hyperparameter optimization though grid search,
- training and testing of the linear models as Logistic Regression, Perceptron, Ridge Classifier, SGD Classifier,
- training and testing of the SVC and LinearSVC
- training and testing of the Random Forest Classifier and DecisionTreeClassifier
- training and testing of the KNeighbors Classifier,
- training and testing of the Naïve Bayes Classifier.

Libraries and Versions:
Python: 3.6.9 |Anaconda, Inc.| (default, Jul 30 2019, 19:07:31)
[GCC 7.3.0]
Pandas: 0.25.3
Numpy: 1.15.4
Sklearn: 0.20.3
Matplotlib: 3.1.3

Keras is an open-source neural-network library written in Python. It is capable of running on top of TensorFlow, Microsoft Cognitive Toolkit, R, Theano, or PlaidML. Designed to enable fast experimentation with deep neural networks, it focuses on being user-friendly, modular, and extensible.

Keras library is used for the development of Sequential model is a linear stack of layers. During model development process, two Keras Sequential neutral networks has been trained and tested. During the first iteration of categorical model development, **relu** and **softmax** activation functions have been used with 200 epochs. Additionally, for model performance enhancement purpose, the binary model has been developed with **relu** and **sigmoid** activation functions with the same number of epochs.

Libraries and Versions:
Keras: 2.2.4

1.7.4 Justification: Selected model performance indicators

The **train** and **test scores** have been compared between the following models:

- Logistic Regression,
- Support Vector Machines,
- Linear Support Vector Machines (SVC),
- k-Nearest Neighbors algorithm (KNN),
- Gaussian Naive Bayes,
- Perceptron,
- Stochastic Gradient Descent,
- Decision Tree Classifier,
- Random Forest,
- Ridge Classifier.

More detailed performance analysis has been performed for Keras Sequential neutral networks:

- Model accuracy
- Model Loss
- Precision
- Recall
- F1-score

1.8 Applications / Data Products

1.8.1 Technology Choice

Data Product – Keras Sequential Neutral Network Model is deployed in IBM Watson Studio for collaboration and teamwork purposes.

1.9 Security, Information Governance and Systems Management

1.9.1 Technology Choice

For demonstrational purposes, Capstone project is available in the web within IBM Watson Studio. All external users could access the project with view only rights. Team members could access the project using their personal user credentials to cooperate and contribute to the project.