Christopher E. Liedel, March 2022

Architectural Decisions Document

Evaluating multiple models for detecting early-stage heart disease

According to the World Health Organization, heart disease, or cardiovascular diseases (CVDs) take an estimated 17.9 million lives each year, around 32% of all deaths globally. CVDs are a group of disorders of the heart and blood vessels such as cerebrovascular disease, coronary heart disease, and rheumatic heart disease. Four out of 5 CVD deaths are due to heart attacks and strokes; and one-third of these deaths occur in people under the age of 70.

People with cardiovascular diseases or those at high risk for developing them due to the presence of one or more risk factors would benefit greatly from early detection and management. A fast, accurate, and simple to use machine learning based diagnostic tool can assist health care practitioners in reducing the morbidity and mortality of these diseases.

This project evaluates the following models:

- Dummy Classifier (Baseline)
- Logistic Regression
- Decision Tree
- Random Forest
- XGBoost
- Support Vector Machine
- K-Nearest Neighbors
- Stacked Classifiers
- Fully Connected Neural Network

Data Source

Datasets used in this project:

1. UCI Machine Learning Heart Disease dataset

https://archive-beta.ics.uci.edu/ml/datasets/heart+disease (303 instances)

The dataset is a Public Health Dataset dating from 1988, consisting of four databases: **Cleveland, Hungary, Switzerland, Stalog (Heart), and Long Beach V**. It contains 76 attributes; however, all published experiments refer to using a subset of 14 attributes.

2. Heart Disease Dataset (Comprehensive)

https://www.kaggle.com/sid321axn/heart-statlog-cleveland-hungary-final (1190 instances)

This dataset is a combination of datasets that are available independently. They have been combined over 11 common features and 1190 instances making it the largest heart disease dataset available for research purposes.

Datasets used for curation of the Comprehensive dataset:

Datasets and number of instances

1. Cleveland: 303
2. Hungarian: 294
3. Switzerland: 123
4. Long Beach VA: 200
5. Stalog (Heart) Data Set: 270

Total 1190

Technology Choice

Datasets are contained in CSV file format

Justification

End user ease of use. File can be expanded as more data becomes available.

Enterprise Data

Technology Choice

Not applicable

Justification

Project does not use enterprise data

Streaming analytics

Technology Choice

Not applicable

Justification

Project does not use streaming data

Data Integration

Technology Choice

Data is cleansed and transformed using Python, pandas, and scikit-learn libraries.

Justification

- Python is an easily readable high-level, object-oriented, programming language.
- The pandas library, built on top of the Python programming language allows for ease of data manipulation and analysis.
- Scikit-learn, largely written in Python is used in the data stage for examining features using tree based importances, low variance filtering, and univariate feature selection.

Data Repository

Technology Choice

Cloud object storage for data.

Justification

- Object storage is cost-effective.
- Any data type is supported.
- Scales to the petabyte range.
- Can access specific storage locations through folder and file names and file offsets.

Discovery and Exploration

Technology Choice

Pandas, Seaborn, scikit-learn, NumPy and Matplotlib are used for visualizations, correlations, and statistical analysis of data set features.

Justification

Pandas, Seaborn, scikit-learn, NumPy and Matplotlib are widely used, open-source libraries with extensive documentation and community support.

Actionable Insights

Heart disease prediction is commonly explored as a binary classification problem predicting either "**0**" suggesting heart disease is not present or not likely to occur according to the input," or "**1**" suggesting heart disease is present or is likely to occur according to the input. This project uses a multiclass approach to view the probability of each class.

Model evaluation is made using **Accuracy, F1 score, Matthews Correlation Coefficient, Jaccard Similarity Coefficient,** and **ROC_AUC score. True model skill** is estimated using a **95% confidence interval** for model error and to determine the upper and lower bounds of model accuracy. Performance metrics are calculated from a confusion matrix and include **Sensitivity, Specificity, Precision, Negative Predictive Value, False Positive Rate, False Negative Rate**, and **False Discovery Rate.**

Algorithms

- Dummy Classifier (Baseline)
- Logistic Regression (For multiclass classification)
- Decision Tree
- Random Forest
- XGBoost Classifier
- Stacked Classifiers (With MLP as the final estimator)
- Neural Network (Keras Sequential)

Best performing models:

- SVM classifier is the best performer on the small dataset. **(92%)**
- XGBoost classifier is the best performer on the comprehensive dataset. **(97%)**

SVM Performance:

- Accuracy: 0.918
- MCC: 0.836
- AUC Score: 0.919
- Jaccard Score: 0.857 (Positive label)

Evaluation using an industry standard **95% confidence interval** shows that true classification error of the SVM model on unseen data is likely to be between 1% and 15%; thus, the true classification accuracy of the model on unseen data is likely between 85% and 99%**.**

SVM Parameters:

```
{'C': 1.0,
 'break_ties': False,
 'cache_size': 200,
 'class_weight': None,
 'coef0': 0.025,
 'decision_function_shape': 'ovr',
 'degree': 3,
 'gamma': 'scale',
 'kernel': 'poly',
 'max_iter': -1,
 'probability': True,
 'random_state': 42,
 'shrinking': True,
 'tol': 0.001,
 'verbose': False}
```

XGBoost Performance:

- Accuracy: 0.966
- MCC: 0.931
- AUC Score: 0.986
- Jaccard Score: 0.945 (Positive label)

Evaluation using an industry standard **95% confidence interval** shows that true classification error of the XGBoost model on unseen data is likely to be between 0% and 6%; thus, the true classification accuracy of the model on unseen data is likely between 93% and 99.9%

XGBoost Parameters:

```
{'objective': 'binary:logistic',
 'use_label_encoder': True,
 'base_score': 0.5,
 'booster': 'gbtree',
 'colsample_bylevel': 1,
 'colsample_bynode': 1,
 'colsample_bytree': 1,
 'enable_categorical': False,
 'gamma': 0.6,
 'gpu_id': -1,
 'importance_type': None,
 'interaction_constraints': '',
 'learning_rate': 0.300000012,
 'max_delta_step': 0,
 'max_depth': 10,
 'min_child_weight': 1,
 'missing': nan,
 'monotone_constraints': '()',
 'n_estimators': 100,
 'n_jobs': -1,
 'num_parallel_tree': 1,
 'predictor': 'auto',
 'random_state': 0,
 'reg_alpha': 0,
 'reg_lambda': 1,
 'scale_pos_weight': 1,
 'subsample': 1,
 'tree_method': 'exact',
 'validate_parameters': 1,
 'verbosity': None,
 'eval_metric': ['logloss', 'error']}
```

Technology Choice

Jupyter notebooks with each step named according to convention. Python, pandas, scikit-learn along with TensorFlow and Keras frameworks are used for model development and evaluation.

Justification

Development:

- The Jupyter Notebook is a popular application for creating and sharing computational documents. It offers a simple, streamlined, document-centric experience.
- Python is an easily readable high-level, object-oriented, programming language.
- The pandas library, built on top of the Python programming language allows for ease of data manipulation and analysis.
- Scikit-learn groups all necessary machine learning algorithms together.
- TensorFlow is one of the most widely used deep learning frameworks. At its core, it is a linear algebra library supporting automatic differentiation. Keras provides an abstraction layer on top of TensorFlow.

Evaluation:

- **Accuracy -** shows a model’s ability to differentiate correctly.

- **Performance metrics from a confusion matrix -** A confusion matrix is a summary of prediction results. It provides insight not only into the errors being made by the model, but more importantly the types of errors that are being made.

- **Matthews correlation coefficient (MCC) -** The (MCC) is a statistical rate which produces a high score only if the prediction obtained good results in all of the four confusion matrix categories (true positives, false negatives, true negatives, and false positives), proportionally to both the size of positive elements and the size of negative elements in the dataset.

- **AUC Score -** is the measure of the ability of a classifier to distinguish between classes and is used as a summary of the ROC curve.

- **Jaccard similarity coefficient -** defined as the size of the intersection divided by the size of the union of label sets. It is used to compare the similarity of the set of predicted labels for a sample to the corresponding set of true labels.

Applications / Data Products

Technology Choice

Jupyter notebook and trained model with explanation of the entire process.
Trained model and weights ready for integrating into a mobile device application.
Final deployment will depend upon the intended use scenario and resources available.

Justification

A Jupyter notebook with a trained model and explanation is simple to use for testing in a local environment.

Exported model can be pushed to mobile devises for use outside a clinical setting by the general public, even where internet service may be sparce or non-existent. This application can provide access to a low-cost diagnostic aid almost anywhere.

Security, Information Governance and Systems Management

Technology Choice

Project Jupyter notebooks and saved model stored on the IBM Cloud Platform and GitHub repository. Data files in cloud object storage.

Justification

Security for these low or no cost services is managed by the vendors.