**Feedback**

**Comparing different predictive models applied to a heart disease dataset**

**Summary**:

The project comprised a comparative analysis of binary classifiers and ANN models targeting the prediction of heart disease. Twelve models were tested and assessed against traditional performance metrics (accuracy, precision, and recall). Some hyperparameter tuning approaches are discussed and justified. Results are compared and discussed, with a candidate model being selected.

- Topic/scenario: [12/15]

This is an interesting and well-known problem, and AI can serve as a supporting tool for decision making and diagnosis. Overall, good choice of topic with a consistent implementation and substantial work on assessing different models. The major points to be reconsidered or further explained comprise: i) data exploratory analysis targeting data quality and the suitability of all variables to the proposed models; ii) literature review to identify similar datasets and existing gold standards – which variables are more suitable as predictors for heart disease; and iii) the need for changing your neural network architecture to include more layers, considering the size of your dataset, and whether such modifications have effectively contributed to the results.

- Data: [6/10]

A specific dataset was used, comprising 11 candidate features and 918 instances. Some missing data is reported, especially for `Cholesterol`. It would be nice to present and discuss the quality of the chosen dataset, which would better inform your decision about dropping or keeping attributes with missing attributes; or even investing in some data imputation method. If Cholesterol is proved to be a good predictor for heart disease, then the missing records would be misclassified. A comprehensive data exploration step is crucial for gaining an understanding of the data and which preparation tasks are needed – for instance, dealing with categorical and continuous variables.

No explicit mention of class labels is provided, although the dataset brings a column `HeartDisease` stating the outcome/diagnosis. Some comments on the class distribution would be fine, so we can assess whether the models are able to correctly train over a similar distribution of samples.

Some data preparation steps are described, especially in terms of one-hot encoding of some features and normalisation.

- Modelling: [8/10]

It is mentioned that no reference results are available, except for some Kaggle data. It would be nice to look at the literature for standard predictors for heart disease and then discuss the coverage of the chosen dataset – i.e. to what extent the chosen dataset has the minimum set of predictors for a satisfactory prediction task.

One important modelling task, associated with exploratory data analysis, is to clearly identify those variables with more predictive power and then check the distribution of your data. If Age can be considered a good predictor, what is the distribution of your samples in age groups, for instance? The same applies to Gender/Sex. This is a necessary exercise for calibrating your input features. You went through some grid search, which is technically more focused on finding the correct calibration for the model but does not always offer good support for picking the more suitable variables.

The rest is fine. You went through the main pipeline steps and have explored reasonable approaches for tuning your models. Perhaps one point to revise is related to the proposed ANN models and whether each new/modified architecture has effectively contributed to improving the results, as the accuracy of each version seems to fall into a standard deviation interval.

- Analytical models: [13/15]

A set of six binary classifiers was tested with default configurations and then with improved hyperparameters through grid search. Regarding ANN, six different models were proposed, with varying configurations in terms of the number of hidden layers and associated nodes, as well as optimizers. Overall, they represent standard approaches for prediction and were correctly used. No bespoke model is discussed or proposed – this could perhaps be a good tentative contribution.

- Results and discussion: [24/30]

Among the binary classifiers with default configuration, SVC is reported as the best and decision trees as the worst models, respectively. When applying hyperparameter tuning and cross-validation, the random forest has proved the best model and decision trees have presented an impressive improvement. There is a good summary of the best parameters for each model. There is also a good guess why SVM and logistic regression did not present significant improvements, and whether this could have been caused by the random choice of hyperparameters during tuning. Overall, the analysis based on these six classifiers is fine and has targeted the main (and potentially possible) improvements.

The ANN approach is based on a variation of the first model, progressively adding dense layers, exploring dropout and alternative optimizers. The main point is to justify whether the initial model can be considered good/efficient, and thus the need for improvement. Comparing the results for all the six architectures, we see some stability within a potential standard deviation range. Perhaps the most notable difference is for Recall, but still within a delimited range. This means, if we apply these models as an auxiliary diagnostic tool, all of them would be capable of presenting similar results.

One point to further explore is the use of dropout and whether it really contributes to your model - you started by adding new Dense layers then exploring dropout, which means you are increasing the number of trainable parameters across the network and then dropping part of them. You mentioned potential overfitting, but this was really observed in your data? My advice for justifying any changes in the proposed architecture will be starting with a new research question and then defining the approach to answer it. For instance, based on the results from a previous model and supposing you noticed overfitting or any other undesirable behaviour, propose the necessary modifications and then assess the model again.

As the dataset is described as small and you have only 11 features, I would consider a smaller model (up to 2 hidden layers) and invest a bit more on the optimization side. Perhaps increasing the number of epochs as well. Overall, the approach for testing NN is based on trial-and-error, and you went through it coherently, but make sure you clearly justify any changes/new guesses.

You see that all six models present similar results, and you have chosen the sixth model due to its apparent superior performance. That’s fine and is supported by the results. Perhaps testing with other datasets would improve your contribution and help in assessing the generalizability of your solution.

A final comparison is made between the best model from each approach, highlighting the advantages and disadvantages of each one; or whether each model can be used based on accuracy and recall. Random Forest is presented as the final choice.

- Technology: [7/10]

The implementation is based on Pandas, Scikit-learn and Tensorflow, with the correct use of the main functions. There is a clear demonstration of technical skills regarding code design, data preparation and analytical steps. No bespoke class/functions were reported. The report and the code lack data visualisation, which would improve the results and help in justifying your choices.

- Documentation: [6/10]

The documentation is overall fine. The report is well structured and brings a consistent discussion of the project topic, all the steps taken and a summary of results. The code brings minimal comments, which are fine for an experienced reader but not for a layman/beginner. Some visualisation would improve both.

- Topic/scenario: 12 [/15]

- Data: 6 [/10]

- Modelling: 8 [/10]

- Analytical models: 13 [/15]

- Results and discussion: 24 [/30]

- Technology: 7 [/10]

- Documentation: 6 [/10]

=> Total: 76 [/100]