Comparing Machine Learning Algorithms for Alzheimer's Disease Classification

Group 3

INTRODUCTION: 10 marks

Aims and objectives of the project:

Detecting Alzheimer's disease in its early stages is vital to ensure support and intervention for patients before the situation worsens. However some of the existing techniques pose difficulties, in terms of comfort, affordability or accuracy (e.g., MRI). The use of machine learning offers an avenue for creating diagnostic approaches that are more easily accessible, cost efficient and accurate. This research investigates the impact of different machine learning algorithms on the accuracy of classifying Alzheimer's disease using features identified in exploratory data analysis as having significant association with Alzheimer's diagnosis.

Roadmap of the report:

- Background
    - We explain what we are trying to answer about Alzheimer's classification and how our findings should be used by our target audience of healthcare professionals and researchers.
- Step Specification
    - We describe how we approached each part of the analysis (data sourcing, data cleaning, data exploration and visualisation, and machine learning implementation), and the high-level steps involved in each part.
- Implementation and execution
    - We describe our blended waterfall-agile development approach and how work was delegated in pairs based on strengths and weaknesses.
    - We describe the specific python libraries used in each part of the analysis pipeline and specific functions for specific jobs.
    - We describe our biggest achievements, challenges, and any decisions to change analysis steps.
- Results
    - We describe the key findings of our exploratory analyses (i.e., which features show significant associations with diagnosis) and machine learning implementation and evaluation.
- Conclusion
    - We summarise our key findings (i.e., which is the best model) and outline future directions.

BACKGROUND: 5 marks

What we are trying to answer

Our project employs a model-centric approach to enhance Alzheimer's disease classification. We want to identify the best machine learning algorithm for this classification task. To do this, we have trained and evaluated six different algorithms on all features identified as showing a significant association with Alzheimer's diagnosis in the exploratory analysis phase. We will focus on accuracy, recall and F1, as well as ROC/AUC and confusion matrices.

Our target audience

Healthcare professionals and researchers are the primary audience for this project. They will be able to use our findings to identify which machine learning algorithm leads to the optimal detection of Alzheimer's. By understanding the most effective algorithms, researchers can build on this work to develop more accurate and reliable methods for identifying Alzheimer's at its earliest stages, which is critical for timely intervention and treatment.

STEPS SPECIFICATIONS: 15 marks

Data sourcing

We used the Kaggle API to import the Alzheimer's dataset. This dataset includes demographic variables, lifestyle factors, medical history, clinical measurements, cognitive and functional assessments, symptoms and diagnosis for 2149 individuals.

Data preprocessing

To prepare the dataset for exploratory analysis and visualisations, we planned to conduct a thorough checking and cleaning of the data set, including handling missing values and outliers, as well as transforming variables to appropriate data types. However, inspection of the data revealed that there were no missing values or any outliers, therefore, this step mainly consisted of dropping unnecessary variables (e.g., patient ID) and changing data types (e.g., one-hot encoding of the Ethnicity variable).

Exploratory data analysis (EDA) and visualisation

The first stage of visualisation involved looking at the distributions of each feature, overall and split by Alzheimer's diagnosis. For this, we used various methods, including violin plots, box plots, and histograms. We then used QQ-plots and Shapiro Wilk tests to assess the normality of our continuous variables, to guide statistical analysis selection. As this step indicated non-normality of all variables, we used Spearman Rank correlation to look at the (continuous) inter-feature associations, and Mann Whitney U to examine which continuous variables significantly varied between diagnosis groups. Both of these tests are non-parametric, and therefore do not require the data to be normally distributed. We visualised the Spearman Rank results using a heatmap of correlation values, and the Mann Whitney using a bar chart of p-values for each feature. To examine categorical inter-feature associations and how each categorical feature relates to Alzheimer's diagnosis, we used the Chi-square test of association. We visualised the inter-feature associations with a heatmap of p-values and visualised the associations with diagnosis with a bar chart.

The results of the Mann Whitney and Chi-square tests were used to identify features showing a significant association with Alzheimer's diagnosis, which were then used in the next step of machine learning evaluation.

Machine learning evaluation and visualisation

The machine learning part of the project involved the following four steps:

1. Selecting the dataset containing predictors identified in univariate EDA
2. Encoded and standardised data for the train, validation, and test data sets.
3. Ran and evaluated six models using a classification report. The models were: K-Neighbours Classifier, Support Vector Classifier, Logistic Regression, Decision Tree Classifier, Random Forest Classifier, and Linear Support Vector Classifier.
4. Visualised the model evaluation using ROC curves and Confusion Matrices.

From this output, we were able to identify the best performing algorithm.

IMPLEMENTATION AND EXECUTION: 15 marks

Team delegation

We used SWAT analysis to identify the team's strengths and weaknesses. Once defined, we split the team into pairs and assigned each pair to the section of the project in which they will be the strongest contributors. The team was divided as follows:

- **Alazne and Eva:** First pass at initial data fetching, exploration and cleaning (including some data visualisations).
- **Wendy and Yasmeen:** Deeper data exploration using statistical analysis and visualisations, to establish which features to use in our Machine Learning models.
- **Bronte and Saoirse:** Machine learning model implementation, evaluation, and visualisation.

Report writing was divided across the team, and we used a Google doc to allow simultaneous collaboration. Bronte and Saoirse were additionally responsible for finalising the jupyter notebook for submission, including code reviews, and editing the report for submission.

Development approach

Given the context of the project and the team members' schedules, we used a blend of waterfall and agile methodology.

Waterfall aspects: We divided the project into different sections at the start, and agreed on a fixed timeline for when each section would be completed, and the next section could begin. This accommodates the limited availability of team members and the fast approaching deadline by allowing each pair to plan, develop, review and finalise their section, making it ready for the next team to pick up and continue from seamlessly (by utilising Github to review and share our code).

Agile aspects: In line with Agile principles, each sub-group pair communicated with other sub-teams (using Slack) to ensure alignment between sections. Once each pair developed their codebase, they refactored the code to make it in line with best practices and added comments to help other team members understand what they had done. Bronte and Saoirse completed multiple code reviews as the project progressed to refine code, streamline where anything became redundant, and catch any errors in data analysis.

Diving deeper into each part of the analysis (inc. tools and libraries used)

Data sourcing

To retrieve the dataset via the Kaggle API, we followed the instructions here to create a Kaggle account and download a json file containing our username and API key. We copied this json file into the directory containing the jupyter notebook. There is an empty version of this file in the GitHub repo as an example. This dataset is stored in data/source as a csv file. We used the pandas library to read this data in. This library has functions for analysing, cleaning, exploring, and manipulating data.

Data preprocessing

We used pandas functions to initially sanity check the data, including *head, shape, columns*. We then use *isnull* to check for missing values, *duplicated* to check for duplicated data, and *info* to check data types. We used *astype* to change numerical variables to categorical and *describe* to look at (1) counts, mean, std, min, quartiles, and max of continuous variables, and (2) counts, unique values, and frequency of top values for categorical variables. We then made a custom function to check for outliers in the continuous data using the *quantile* function. This step indicated there were no outliers. The final step was to identify if any categorical variables should be one-hot encoded (i.e., non-ordinal variables). Only Ethnicity was one-hot encoded as the levels do have an inherent ordering. For this, we used *get_dummies*.

Exploratory data analysis and visualisation

We used Seaborn, a data visualisation library built on top of Matplotlib to generate insightful graphics. We used the *violinplot, boxplot, histplot, countplot* and *heatmap* functions. We also used the re package to make the axis labels and titles suitable for plotting. Additionally, we employed the NumPy package to create arrays to mask the upper triangle of our heat maps.

In addition to these exploratory visualisations, we researched and applied various statistical tests, ensuring that our analysis was appropriate for the type of data we were dealing with. For our statistical analysis, we used the SciPy.stats sub-package of SciPy. We used this to conduct Shapiro tests of normality and create QQ plots for continuous data, Mann Whitney U tests, and Chi-square tests of association. We used the pandas *corr* function to calculate Spearman's Rank correlation.

Machine learning evaluation and visualisation

The first step was to use pandas to select the input features. Sklearn was used to transform data and run the machine learning pipeline. To scale our continuous variables and encode our categorical data as numeric data, we used the StandardScaler, OrdinalEncoder (for x data), and the LabelEncoder (for y data). Encoding is necessary for machine learning models because they require numerical input to compute relationships between different variable types. We used *train_test_split* to split our data into training, validation, and test data. We fit and applied the transformers on the training data, and then applied the transforms to the validation and test sets to prevent data leakage.

Next, we initialised the model objects in a dictionary with various parameters. We set the random state argument to ‘50’ for all models to ensure reproducibility and for all models except K-neighbours and Linear SVC, we balanced the class weights to deal with the imbalance in diagnosis classification in the dataset. We then ran the following steps for each model:

- Fit model to training data
- Used model to make predictions on validation and test set
- Used *classification_report* function to generate reports for our validation and test sets, which include precision, recall, F1, and accuracy (macro average and weighted average).

Using these outputs for each model, we generated ROC curve plots and Confusion matrices for each model (total 6).

Achievements

Our exploratory data analysis and visualisations enabled us to identify key features that show significant associations with Alzheimer's diagnosis (based on the data we have). This was critical for determining the features most likely to influence diagnosis predictions when applying machine learning models. We implemented an advanced, and flexible, Machine Learning pipeline, using functions to improve modularity and abstraction. This pipeline efficiently and successfully implemented six model types, and printed out appropriate evaluations for validation and test sets. We also created appropriate visualisations for model evaluation, showcasing an in-depth understanding of model metrics.

Challenges

In general, coordinating between team members with busy schedules was challenging, as team members were rarely available at the same time and some were less able to communicate via slack regularly.

Fetching the data from the Kaggle API was initially quite challenging, particularly for team members who were using corporate managed devices. One of our initial implementation challenges was determining which variables to retain or discard from the dataset. Variables such as Patient ID and DoctorInCharge were excluded because they did not provide useful information for the analysis.

During data preprocessing, there was some confusion regarding outlier detection and data types. Initially, we applied the IQR function to all variables, regardless of whether they were categorical or continuous. However, this function is only appropriate for continuous variables. There was also confusion over when to use one-hot encoding. Originally, this was applied to a subset of categorical variables (some binary, some ordinal, some non-ordinal), however, we identified that this was only necessary to apply to non-ordinal data (i.e., Ethnicity).

Data exploration: we found it very challenging to determine which tests to use to first assess the normality of the data and then explore the association between different types of variables: continuous variables and a categorical variable (Diagnosis) and between categorical variables. Another difficulty faced was selecting the most insightful and visually appealing representation of our analysis.

Machine learning implementation: After implementing cross-validation on the training set, we observed significant overfitting, likely due to a data leak caused by standardising the entire dataset prior to cross-validation. Given the time constraints and this issue falling outside the project scope, we decided to remove cross-validation entirely. Instead we refined our existing code to adhere to conventional standardisation practices - fitting out scalers exclusively to the training data, before transforming on train-val-test splits. The final model's performance was validated using validation and test data; however, future work may investigate integrating some type of cross-validation for a more robust architecture to improve generalisation capacity. It was also challenging to understand how Linear SVC worked and differed from the other five models, and the implications of these differences on how to create appropriate visualisations (i.e., ROC curve not possible).

Decisions to change analysis steps

Initially, BMI was encoded into a categorical variable ("normal," "overweight," and "obese"), however, we decided not to pursue this categorisation as it didn't serve a particular function. Consequently, we chose to keep BMI as a continuous variable.

Regarding exploratory data analysis and visualisation, Spearman Rank correlation was initially used to create a heatmap to visualise the categorical inter-feature associations, however, we identified this was inappropriate for categorical variables, and so created a heatmap based on the p-values from a series of chi-squared tests instead.

Regarding machine learning implementation, we had originally planned to use the following two sets of input features: (1) 23 variables comprising demographics, lifestyle factors, family history, and clinical measurements, excluding functional assessments and symptoms, and (2) 9 variables identified as being important in previous research. However, we decided it would be more in line with our goals if we focused only on predictors identified as being important in exploratory data analysis.

RESULT REPORTING: 10 marks

Exploratory data analysis

The following variables showed significant associations with Alzheimer's diagnosis: 'Sleep Quality', 'ADL', 'Cholesterol HDL', 'Functional Assessment', 'Mini Mental State Exam', 'Behavioral Problems', and 'Memory Complaints'. We used these as input features to our model evaluation. We used these seven variables as input features for model evaluation and selection.

Machine Learning Evaluation

To determine the 'best' overall machine learning algorithm for classifying Alzheimer's in our dataset, we decided to primarily focus on *Accuracy*, *Recall* and the *F1 score*. By focusing on Average Recall and the Average F1 score, we ensure that our model not only captures as many true Alzheimer's cases as possible but also maintains a reasonable level of precision, reducing the risk of over-diagnosis. This approach helps in selecting a machine learning algorithm that performs well in the context of our specific goal—accurate and reliable identification of Alzheimer's cases in our dataset.

The Random Forest Classifier and Decision Tree Classifier were tied along all three metrics: 93% overall accuracy, 91% average recall, and 92% average F1 score.

To further distinguish between the two, we also looked at the AUC metric and ROC curve (Figure 1). The higher the AUC, the better the model is at distinguishing between the positive and negative classes. The Random Forest and Support Vector Machine Classifiers had the same AUC metric (0.95). In terms of the shape of the curve, the Random Forest Classifier has a steeper incline to the top-left corner, indicating that the model has a high true positive rate (y-axis) while maintaining a low false positive rate (x-axis).

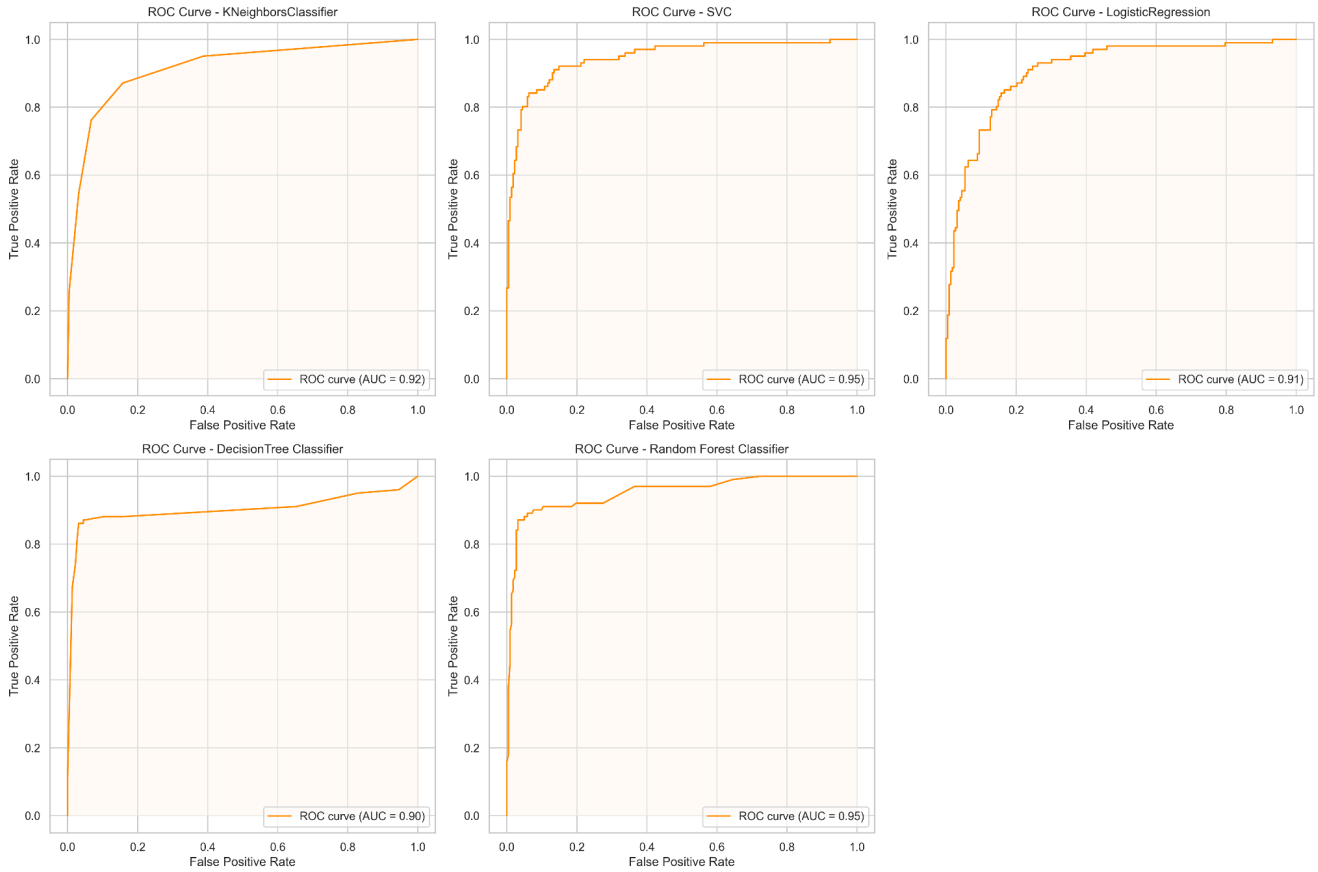

Figure 1. ROC curves of each of the six Machine Learning Classifiers evaluated on the Alzheimer's data set. Y-axis shows the true positive rate, while x-axis shows the false positive rate. The AUC metric is plotted in the bottom right corner. The steeper the incline and the higher the AUC metric, the better the model.

Finally, we looked at the Confusion Matrices (Figure 2). Confusion Matrices provide insights into the distribution of errors by showing true positives, false positives, true negatives, and false negatives, which is useful for understanding error types and class imbalance handling. Overall, the Confusion Matrices indicated that the Random Forest and Decision Tree Classifiers perform similarly and the differences are minor:

- True positive (TP): Decision tree had slightly higher (88 vs 86)
- True negative (TN): Random forest had slightly higher (215 vs 212)
- False positive (FP): Random forest had slightly lower (7 vs 10)
- False negative (FN): Decision tree had slightly lower (13 vs 15)

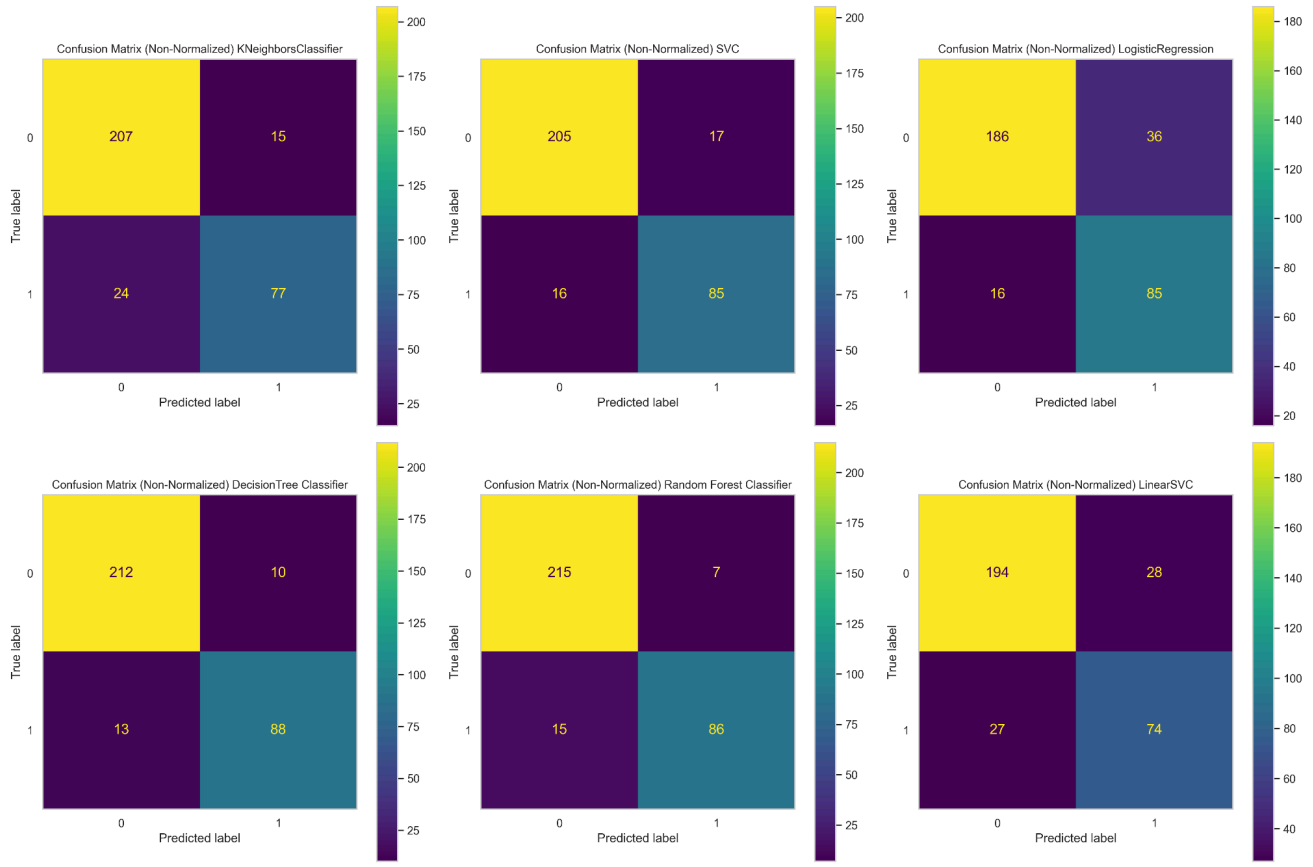

FIgure 2. Confusion Matrices of each of the six Machine Learning Classifiers evaluated on the Alzheimer's data set. X-axis plots the predicted label, while y-axis plots the true label. 0 = no Alzheimer's diagnosis, 1 = Alzheimer's diagnosis. Higher values in top left (true negative) and bottom right (true positive), and lower values in top right (false positive) and bottom left (false negative) are desirable.

CONCLUSION: 5 marks

The Random Forest Classifier and Decision Tree Classifier performed identically on accuracy, recall, and F1-score. The Random Forest Classifier had a higher AUC metric and steeper ROC curve, and a similar TP/TN/FP/FN balance. Between the two, they perform almost identically, so either could be chosen as the best depending on other considerations like model complexity and interpretability. If simplicity and interpretability are critical, the Decision Tree Classifier might be preferred; otherwise, Random Forest Classifier could offer slightly better generalisation due to its ensemble nature and in this dataset, the AUC metric was higher. Future work could tune hyperparameters of each model to maximise model performance and consider using neural networks for a deep-learning approach.