THE UNIVERSITY
OF ARIZONA

Biometric Predictors of Emotional States: A Deep Learning Approach

INFO 521: Introduction to Machine Learning
Taught by: Prof. Clayton Morrison
School of Information

A Project Report

Riyanshi Bohra

M.S. in Data Science

riyanshibohra@arizona.edu

I. INTRODUCTION

A. Background

The use of AI and machine learning techniques for measuring how physiological and behavioral factors are related to each other, and how relevant they are to the healthcare industry, has become an important area of interest recently among scholars. This has recently gained more importance due to its ability in understanding mental health related factors, using physical data metrics. The use of digital technology, such as wearables, capture an individual's physical variables constantly, which can provide information on the different measures of mood and other mental disorders, if used efficiently. **[1]** In this context, machine learning enables a deeper understanding of mental health patterns with respect to an individual's physical health.

B. Objective

The project's objective is divided into two parts. The first part is using a synthetic dataset, which consists of an individual's mood states and physical variables that are available from the use of digital technology such as wearables, and generating mood descriptions by employing a pre-trained GPT-2 model on the relevant columns in the data. This is further used to classify the generated sentences into predefined categories of mood states using deep learning and transformer models that deal best with language sequential data.

The second part focuses on using real-world data consisting of several health metrics, such as sleep cycles, physical activity and biological metrics, to classify an individual's stress levels into three distinct mood categories.

C. Method Overview

This project employs various advanced methods in AI and machine learning, including deep learning, transformer and multi-class classification models. Firstly, it uses an Ensemble model, a Multi-class model, and a BERT (Bidirectional Encoder Representations from Transformers) model for mood classification based on synthesized mood descriptions. Further, for the second part of the project, different classification algorithms are applied to real-world health metrics data and compared to classify stress levels into mood states.

D. Problem Statement

This project aims to explore the use of various machine learning techniques in finding mood patterns using health-related datasets. It uses both synthetic and real-world data to derive actionable insights.

II. DATA

A. Synthetic Dataset

The synthetic data used in this project is sourced from Kaggle **[2]**. This data was created originally only for experimental purposes and consists of several health metrics derived from wearables sensors technology. Key features include:

1. Sleep/Activity data: Information about Sleep Duration, deep sleep, REM sleep
2. User Data and Physical metrics: User ID, Age, Gender, Weight,etc.
3. Lifestyle: Information based on an individual's smoking habits, alcohol consumption and exercise details.
4. Biological factors: Information about heart rate, blood oxygen level and skin temperature.
5. Mood: The target feature for this project is the 'Mood' column, which consists of mood states such as Happy, Sad, Anxious, Neutral.
6. Mood descriptions: The generated column using a pre-trained GPT-2 transformer model describing how an individual might feel based on a particular mood state.

B. Real World Dataset

The real-world dataset is relatively smaller and focuses on factors that are directly related to sleep and daily lifestyle habits **[3]**. Key features include:

1. Sleep data: Detailed information about sleep duration, quality and sleep disorder
2. User Data and Physical Metrics: Physical activity levels, stress levels and BMI Index of an individual.
3. Cardiovascular Health: Information about blood pressure and heart rate measurements.

III. ALGORITHMS

A. Deep Learning Algorithms

The deep learning models employed in this project include an Ensemble model and a BERT (Bidirectional Encoder Representations from Transformer) model. **[5]**

1. Ensemble model:

The Ensemble model employed in this project combines predictions from various individual models to enhance overall performance.The architecture combines outputs from multiple neural networks LSTM (Long Short-Term Memory) models, each focusing on a unique class, to improve prediction accuracy. This is based on the ensemble learning principles as described by Polikar (2012) **[4]**.

If P_i is the prediction of the i^{th} LSTM model and w_i is it's corresponding weight, the final prediction P is calculated as:

$$P = \sum_{i=1}^{n} w_i \cdot P_i$$

2. BERT(Bidirectional Encoder Representations from Transformers) Model

BERT is a powerful transformer model designed to pre-train bidirectional representations from unlabeled texts by using attention layers and transformer encoders.This pre-trained model can be fine tuned further with one additional output layer to perform several tasks such as classifying mood descriptions and text classification in the project's scenario. The design and application of BERT is described in the paper by Devlin et al. (2018)**[6]**.

The Attention mechanism used in BERT is expressed as:

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

Where Q, K, and V represent query, key and value metrics respectively and d_k is the dimension of the key

B. Classification Algorithms

1. Decision Tree

Decision tree is one of the most popular approaches for binary and multi-class classification purposes. It consists of nodes to form a directed rooted tree with a root node and multiple decision nodes. For the multi-class classification problem, the goal is to predict the value of the target variable with three or more possibilities.It is one of the simplest models which works well on smaller datasets. This methodology is further described in Rokach and Maimon's work on decision trees (2005) **[7][8].**

2. Random Forest

Random Forest is an ensemble of decision trees, utilized for its high accuracy. It works by creating multiple decision trees during training and after a large number of trees is generated, they vote for the most popular class. The output is the class with the most votes as the prediction. Breiman's research on random forests provides a foundational understanding of this model (Breiman, 2001)**[9]**. It is represented as:

$$Y = mode\left\{Y_1, Y_2, Y_3, ..., Y_N\right\}$$

Where Y_i is the prediction of the i^{th} tree

3. Multinomial Logistic Regression

Multinomial logistic regression is used for multi-class classification purposes and is an extension of logistic regression. It models the target using a multinomial probability distribution function. The standard logistic regression algorithm is used for binary classification and utilizes the log loss function to train the model. For multinomial logistic regression, the loss function used is cross-entropy loss, which outputs one probability for each class label **[10]**. The probability of class K can be represented as:

$$P(X) = \frac{e^{\beta_k^T X}}{\sum_{j=1}^{K} e^{\beta_k^T X}}$$

Where β_k are coefficients for class k and X is the feature vector.

Two variants of Logistic Regression were applied, each using a unique regularization technique (L1 and L2) to address overfitting. L1 regularization promotes sparsity in the model, while L2 regularization prevents overfitting by adding a penalty term.

Regularization – L1(Lasso):

$$Cost\ Function\ =\ Loss\left(y,\ y^{\bullet}\right) +\ \lambda \sum |w|$$

Regularization – L2(Ridge):

$$Cost\ Function\ =\ Loss\left(y,\ y^{\bullet}\right) +\ \lambda \sum w^2$$

Here λ is the regularization parameter, and w represents the weights of the model

4. Gradient Boosting Classifier

Gradient Boosting is an algorithm based on Boosting, an ensemble learning approach in Machine Learning. It learns from each of the weak learners to build a stronger model and works well in both regression and classification purposes. It is essentially an iterative functional gradient algorithm which minimizes a loss function iteratively by choosing a function that points towards a negative gradient **[11]**.

$$F_m(x) = F_{m-1}(x) + v \sum_{i=1}^{n} \gamma_i h_i(x)$$

Where $F_{m-1}(x)$ is the model at iteration m-1, v is the learning rate, are the coefficients and $h_i(x)$ are the weak learners.

IV. PROCEDURE

The procedure is divided into two sections to describe separate analyses conducted with synthetic and real-world datasets. The first part involves the application of deep learning models on the synthetic dataset, and the second part focuses on the use of classification models for the real-world dataset. These two sections represent unique and specific purposes, with different objectives.

PART 1: Analysis with Synthetic Dataset and Deep Learning Models

A. Text Generation

1. Data Loading and Text Generation Pipeline Initialization:
   - This step involves loading all of the synthetic datasets (activity,health,interaction) into separate dataframes using the load_data function
   - Next, using the text_generation function, a pipeline is generated for text generation using GPT-2 transformer model
2. Data Preprocessing and Text Generation:
   - For the Preprocessing part in the 'main', as a first step, all the datasets are merged together using common columns('User_ID' and 'Timestamp')
   - Next, all the unnecessary columns are dropped such as 'User_ID', 'Timestamp', and 'Notifications_Received'.
   - Sentence starters and mood specific words were defined to create unique prompts. The various mood states were 'Happy', 'Sad', 'Neutral', and 'Anxious'.
   - Then, I created a function to add synonyms for these mood words for diverse prompts.
   - Then, for each row, mood descriptions are generated in batches using the batch_generate function which creates prompts.

Running the Script:

- Run the main script to load and preprocess the data(load_data), Initialize the text generation pipeline(text_generation), and save the updated dataset.
- This script can be run by executing the main() function.

B. Implementation of Deep Learning Models

1. Data Loading and Transformation:
   - The first step is loading the updated dataset with mood descriptions using the load_and_transform_data function. This dataset is the output of Part 1.
   - Next, one-hot encoding was applied on the target variable 'Mood' to feed into the deep learning neural network model.
2. Data Preprocessing using NLP:
   - Using the preprocess_data function, next step is to convert the textual mood descriptions into lower case for standardization.
   - Next, the text data is tokenized and converted to sequences. It returns padded sequences, labels, vocabulary size, and maximum sequence length.
3. Model Training and Evaluation:
   - Using the train_ensemble_model function, an ensemble of four LSTM models(one for each class) is generated. Each model is a Sequential model with multiple layers including embedding, LSTM, and Dense layers.

- Using the evaluate_ensemble_model, the ensemble model is evaluated on the test data.
- Using the train_bert_model function, a BERT model is created which uses pre-trained BERT embeddings in combination with an additional output Sequential layer to fine tune the model.
- Using the evaluate_bert_model, the BERT model is evaluated on the test data.

4. **Result Visualization:**
   The values for precision, recall, f1 scores for each model are compared and visualized using the create_and_save_plots function.

Running the Script:

- Run the script to load the data, preprocess, train, evaluate models, and generate visualizations.
- Run the script by executing the main() function.

PART 2: Analysis with Real-World Dataset and Classification Models

1. Data Loading and Preprocessing:
   - The first step is to load the dataset through the load_data function, which reads the rate from the CSV file.
   - Next, using the preprocess_data function, the columns in the dataset are preprocessed through the following tasks:
     1. Dealing with BMI categories and re-defining them.
     2. Converting blood pressure values into two separate columns for an easier and accurate analysis.
     3. Categorizing stress levels into moods (Calm, Neutral, Overwhelmed)
     4. Using Label Encoding to encode all the categorical columns to use it for model development.

2. Model Training and Evaluation:
   - As the first step, the dataset is split into training and testing sets.
   - The classification models used are Decision Tree, Random Forest (with hyperparameter tuning), Logistic Regression (with L1 and L2 regularization), and Gradient Boosting.
   - Next, using separate functions for each, the classification models are trained on the train data. The respective functions are train_decision_tree, train_random_forest, train_logistic_regression, train_gradient_boosting.
   - Using the evaluate_model function, the models are evaluated.

3. Model Comparison and Visualization:
   - Using the compare_and_visualize_models function, the precision, recall and F1 score of each model is evaluated.
   - The function creates a bar plot for comparison of all the metrics for each model.

Running the Script:

1. Run the script to do data processing, model training, evaluation, and visualization steps.
2. Run the script by executing the main() function.

V. EVALUATION

In evaluating the performance of the machine learning models used in this project, the primary focus was on three metrics– Precision, Recall, and F1 Score. A confusion matrix is also beneficial while finding the summary of the predictions. These metrics are widely recognized in the field of machine learning. They provide insights into the models' performance, specifically in case of classification problems. **[12]**

1. Precision

Precision measures the accuracy of positive predictions. It is the ratio of true positives to the total predicted positives and is particularly useful when it is important to know the number of false positives. The equation is given as:

$$Precision = \frac{True\ Positive}{True\ Positives + False\ Positives}$$

2. Recall

Recall, or sensitivity, indicates the ability of the model to find all the relevant instances in a dataset. It is crucial in scenarios where missing a positive instance (false negative) is costlier than falsely labeling negative instances as positive. The equation is given as:

$$Recall = \frac{True\ Positive}{True\ Positives + False\ Negatives}$$

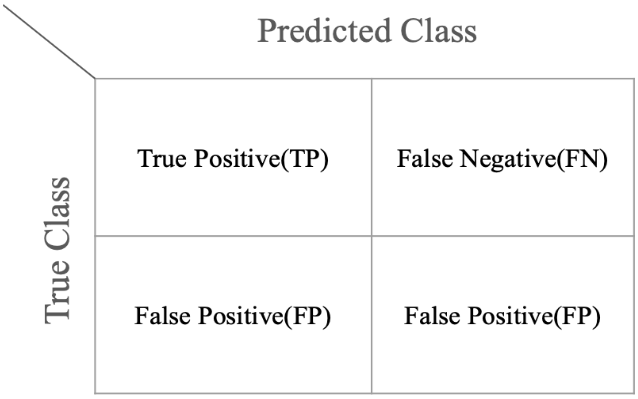
3. F1 score

F1 Score is the weighted average of both precision and recall metrics. It measures the accuracy of the model by combining the scores of both of these metrics. The equation is given as:

$$F1\ Score = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

4. Confusion matrix

Confusion matrix is a very popular measure used while doing classification problems. It can be applied to binary classification as well as for multiclass classification problems. It essentially shows the prediction summary in a matrix form.**[13]**



	Predicted Class	
True Class	True Positive(TP)	False Negative(FN)
	False Positive(FP)	False Positive(FP)

Table 1: Confusion matrix

VI. RESULTS

A. Synthetic Data Results

The evaluation of models using the synthetic dataset revealed distinct performances across different mood labels. The Ensemble model consistently performed better in comparison to the BERT model across all mood categories—Neutral, Anxious, Happy, and Sad—when considering the Precision, Recall, and F1 Score.

Precision: The Ensemble model showcased higher precision in most categories, which indicates a lower rate of false positives. For 'Neutral' and 'Happy' states, the precision was significantly higher compared to the BERT model, indicating that it had more number of correct positive predictions.

Recall: In terms of recall, there was a significant difference between the values given by the Ensemble model as compared to the BERT model which indicates better performance from the Ensembler in correctly identifying all moods, particularly the 'Neutral' and 'Anxious' states.

F1 Score: The F1 Score, which balances precision and recall, also gave better results for the Ensemble model.

It is worth taking into consideration that this analysis was performed on a synthetic dataset, which might be why the model is performing extremely well. Reproducing the same code for a real-world dataset with more patterns and outliers will help in analyzing the actual performance of the two models in this scenario.

Model	Mood	Precision	Recall	F1 - Score
Ensemble Model	Neutral	0.982	0.989	0.986
	Anxious	0.994	0.993	0.993
	Happy	0.994	0.986	0.990
	Sad	0.991	0.994	0.993
BERT	Neutral	0.946	0.955	0.950
	Anxious	0.981	0.866	0.920
	Happy	0.979	0.953	0.966
	Sad	0.872	0.987	0.926

Table 2: Comparative Analysis of Precision, Recall, and F1 Scores Across Mood Labels for Ensemble and BERT Models

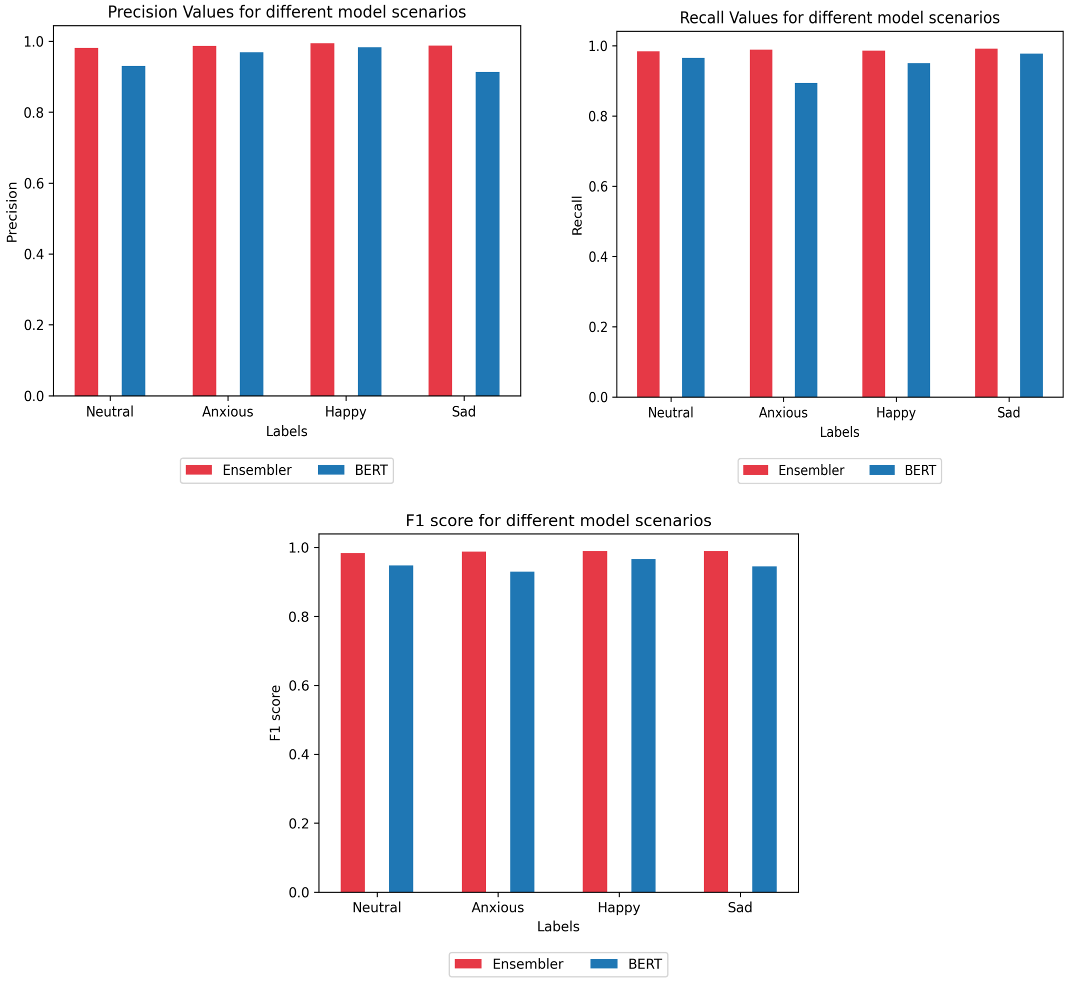


Figure 1: Performance Evaluation of Ensemble and BERT Models:
This figure presents three bar plots comparing the effectiveness of Ensemble and BERT models in classifying four mood categories: Neutral, Anxious, Happy, and Sad. The evaluation metrics include precision, recall, and F1 score. Overall, the Ensemble model shows a higher level of accuracy in all these metrics.

B. Real World Data Results

The evaluation of models using the real world dataset revealed almost similar performances across different performance metrics Precision, Recall and F1 Score. Each visualization plot gives insights on how well each model performed to classify real-world observations into the predefined stress level categorie–: 'Calm', 'Overwhelmed', and 'Neutral'.

Precision: In terms of precision, the Random Forest Model proves to be the one with the highest accuracy after dealing with overfitting through Hyperparameter Tuning. Decision Tree is a weaker, simple model and hence even though the values are similar for the tree based models, it cannot be considered as the best model. High precision indicates that the model has a low rate of false positives.

Recall: Similar to Precision, the value for Recall across all models are significantly similar but Random forest dealt with Overfitting the best and gave the best result out of all.

F1 Score: Consistent with other metrics, F1 score is the highest for Random Forest which indicates its superior performance for multi-class classification of the three stress level categories.

The overall superior performance of the Random Forest across all three metrics indicates that the tree-based Ensemble method performs the best for Multi-class classification purposes and is suited for dealing with the patterns present in real-world health data. Logistic Regression with L1 and L2 regularization gives lower performance in comparison which demonstrates the importance of using appropriate penalty terms in managing overfitting. Decision Tree, while simple and interpretable, does not deal with overfitting well.

Model	Precision	Recall	F1 - Score
Decision Tree	0.982	0.982	0.981
Random Forest	0.982	0.982	0.981
Logistic Regression (L1 Regularization)	0.717	0.690	0.690
Logistic Regression (L2 Regularization)	0.957	0.955	0.956
Gradient Boost	0.972	0.973	0.972

Table 3: Comparative Analysis of Precision, Recall, and F1 Scores across five multi-class classification models

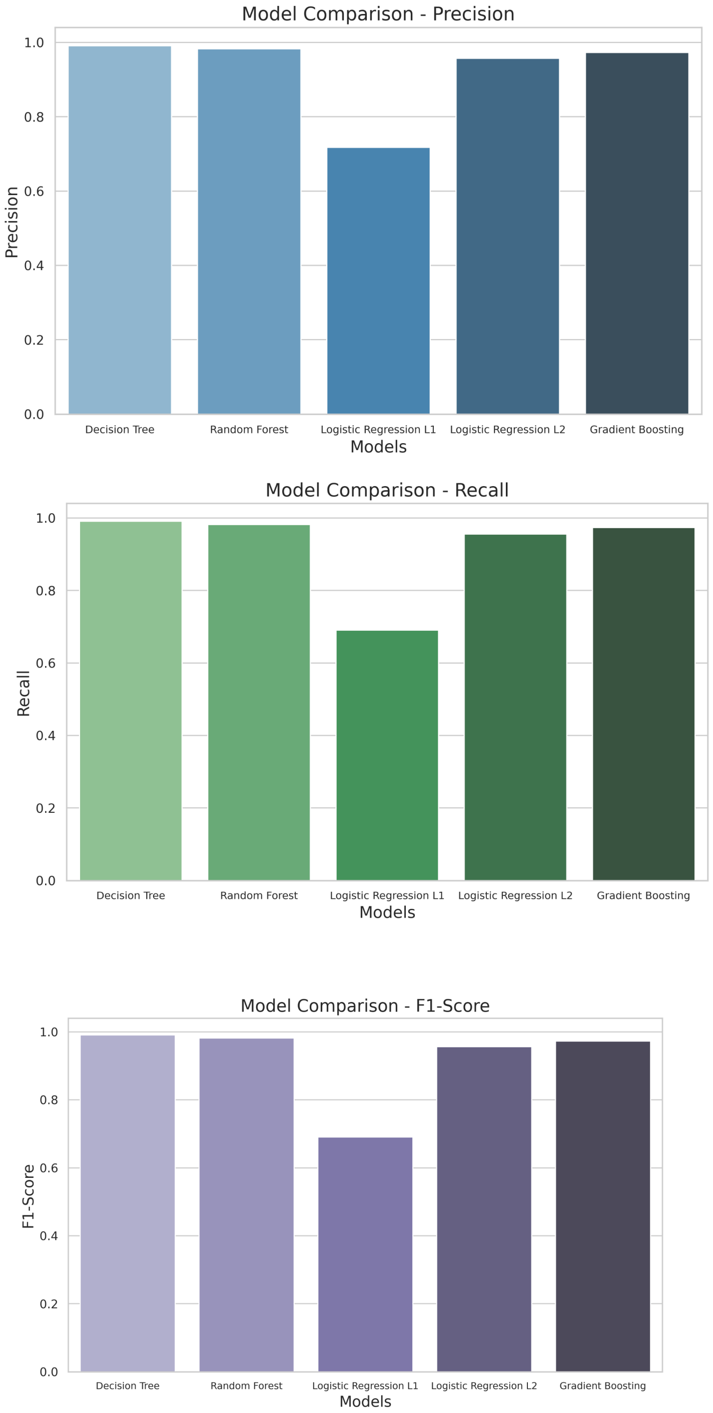


Figure 2: A comparison of classification models on Real World Dataset: Set of bar charts displaying the comparison based on Precision, Recall and F1 Score for five different models—Decision Tree, Random Forest, Logistic Regression with L1 and L2 Regularization, and Gradient Boosting Classifier. This was applied to real-world data, highlighting the better performance of Random Forest on all three metrics.

VII. REFERENCES

[1]. De Angel, V., Lewis, S., White, K., Oetzmann, C., Leightley, D., Oprea, E., Lavelle, G., Matcham, F., Pace, A., Mohr, D. C., Dobson, R., & Hotopf, M. (2021). Digital health tools for the passive monitoring of depression: a systematic review of methods(2022)
https://www.nature.com/articles/s41746-021-00548-8#:~:text=,using%20machine%20learning%20methods

[2]. Manideep, M.(2023), Synthetic dataset. Wearables dataset.
https://www.kaggle.com/datasets/manideepreddy966/wearables-dataset/data

[3]. Laksika,T.(2023),Real-life Sleep Health and Lifestyle Dataset.
https://www.kaggle.com/datasets/uom190346a/sleep-health-and-life style-dataset/data

[4]. Polikar, R. (2012). Ensemble learning.
https://users.rowan.edu/~polikar/ensemble.html#:~:text=URL%3A%20https%3A%2F%2Fusers.rowan.edu%2F

[5]. LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. Nature.
https://www.nature.com/articles/nature14539#:~:text=URL%3A%20https%3A%2F%2Fwww

[6]. Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2018). BERT: Pre-training of deep bidirectional transformers for language understanding.
https://ar5iv.labs.arxiv.org/html/1810.04805

[7] Rokach, L., & Maimon, O. (2005). Decision Trees.
https://www.researchgate.net/publication/225237661_Decision_Trees

[8] Visual Studio Magazine(2023). Multi-Class Classification Using a scikit Decision Tree.
https://visualstudiomagazine.com/articles/2023/03/17/scikit-classification.aspx

[9]. Leo Breiman(2001). Random Forests.
https://www.stat.berkeley.edu/~breiman/randomforest2001.pdf

[10]. Jason Brownlee(2020). Multinomial Logistic Regression with Python.
https://machinelearningmastery.com/multinomial-logistic-regression-with-python/

[11]. Paperspace. Gradient Boosting In Classification: Not a Black Box Anymore!
https://blog.paperspace.com/gradient-boosting-for-classification/

[12]. H Dalianis (2018). Springer Link. Evaluation Metrics and Evaluation.
https://link.springer.com/content/pdf/10.1007/978-3-319-78503-5_6.pdf

[13] Science Direct(2020). Confusion Matrix.
https://www.sciencedirect.com/topics/engineering/confusion-matrix#:~:text=A%20confusion%20matrix%20represents%20the,by%20model%20as%20other%20class.