Breast Cancer Modelling with Neural Networks

Contents

Dukgeun Choi

Written on 28th July 2024

The project aims to use neural networks to train a machine learning model to predict if a breast cancer diagnosis is malignant or benign using features computed from a digitised image of a fine needle aspirate (FNA) of a breast mass, describing the characteristics of the cell nuclei. This is a **binary classification** problem, as the model aims to predict whether a diagnosis is malignant (M) or benign (B).

The input data for the model consists of 30 numeric features from the FNA images, such as cell nucleus radius, texture, perimeter, area, smoothness, compactness, concavity, symmetry, and fractal dimension. More information on these characteristics is covered in the **Dataset Collection** section. The model outputs a binary classification, either benign or malignant, for each diagnosis.

To assess the model's performance, the following evaluation metrics will be used.

- Accuracy: Overall correctness of predictions
- Precision: Proportion of true positive predictions among all positive predictions
- Recall: Proportion of true positive predictions among all actual positive cases
- F1-Score: Harmonic mean of precision and recall (Buhl, 2023)

2. Dataset Collection

2.1 Dataset Source

The dataset that will be used for the project is Diagnostic Wisconsin Breast Cancer Database that is extracted from UC Irvine Machine Learning Repository. The data dictionary can be found in the appendix, **8.1a)**.

2.2 Data Cleaning and Preprocessing

The data was cleaned checking if there were any missing values. This was done by running '*print("\nMissing values:\n", df.isnull().sum())*'. This output showed that there were no missing values in the dataset, therefore it was not required to remove any rows.

Afterwards, the id variable was removed as it provides little to no importance to the output of the model. Then, the 'diagnosis' column was set as the target variable and LabelEncoder was used to convert categorical values ('M' and 'B') to numerical values (1 and 0).

Furthermore, the StandardScaler was applied to normalise all numeric features, ensuring that they are on the same scale. This is an important step as it improves the model's performance and prevents inputs with larger magnitudes from dominating those with smaller magnitude.

Then, the data was randomly shuffled to ensure that the order of the samples doesn't affect the model training. This is important in reducing the bias and ensuring that the training, validation and test sets are representative of the overall distribution of the data. The shuffled data was then split into training, validation and test set, 60%, 20% and 20%.

These data cleaning and preprocessing steps ensure that the data is clean and consistent, and optimally prepared for model training.

3. Exploratory Data Analysis

In this section, exploratory data analysis of the dataset will be discussed to help better understand the data and discover some key findings that will help in training the model.

As shown in **8.1a**, the ten most important features include those related to the nuclei's area, perimeter, radius and concave points. Cancerous cells are characterised by a larger nucleus compared to benign or normal cells (Eldridge, 2023), which is indicative of the importance of nuclei's area, perimeter, radius and concave points. Furthermore, malignant tumours have irregular shapes with many indentations or protrusions (Eldridge, 2023), indicative of the importance of the concave points. This is further illustrated in the correlation graph **8.1.b**, where these features have the highest correlation with the target variable.

By plotting box plots for each feature by diagnosis, most of the variables had a clear distinction between malignant and benign cases, like the few examples shown in **8.1.c**. Yet, there were few features where the median was the same for both malignant and benign cases, such as texture_se, smoothness_se and symmetry_se, as seen in **8.1.d**. The standard error is the measure of variability in these characteristics, representing how much these measurements deviate from the mean values. This indicates that the variability in the texture, smooth and symmetry of a cell nucleus have less discriminative power in distinguishing between malignant and benign tumour.

4. Benchmark Model

For the benchmark model for the project, a **logistic regression** was fitted to the data. Logistic regression was chosen as it is a straightforward non-deep learning model that provides probabilities of class membership, with the output being either 0 or 1, making it well suited for binary classification problems.

To fit the logistic regression to the data, the data cleaning and preprocessing mention in Section 2 was initially processed. After splitting the data into training and test sets, a simple logistic regression model was fitted using LogisticRegression from sklearn.linear.model, with default hyperparameters.

As mentioned in Section 1, the model's accuracy, precision, recall and f1-score will be used to evaluate the model's performance for this project. The performance metrics for the benchmark logistic regression is as follows.

	Accuracy	Precision	Recall	F1-Score
Validation	0.96	0.98	0.94	0.96
Test	0.96	1.00	0.91	0.95

5. Deep Learning Models

5.1 Dense Sequential Model

5.1.1 Model Outline

The first deep learning model performed was a Dense Sequential Model. This model is a neural network architecture that consist of a linear stack of layers, where the output of each layer is fed as the input for the next layer (Tracyrenee, 2023). It aims to mimic the functionality of a human brain, where each neuron in a layer is connected to the neurons in the next layer. The model does have its limitations, as it does not allow branching of and is restricted to a single output layer (Swain, 2021).

This model was chosen mainly due to its simplicity, as (mentioned in the **3. Exploratory Data Analysis**) most features in the dataset had a clear influence on the target variable. Furthermore, the single output layer restriction is well suited for the supervised learning problem for the project.

5.1.2 Hyperparameter Tuning

To optimise the model's performance, numerous manually hyperparameter tuning was performed. The hyperparameters that were tuned were the number of layers, number of neurons in each layer, batch size, activation functions, learning rate and dropout rate. The number of epochs used in the model was set to 1000 for all models, with the application of early stopping with a patience of 15. This was to ensure the model does not overfit the training data, and automatically stops the training process if the performance is not improving. The patience represents the number of epochs to wait for an improvement before the stoppage, allowing for some fluctuations. After the stoppage, the best performing model is restored.

As seen on the table of the hyperparameter tuning process, the number of layers and the number of neurons in each layer were first tuned. Although the results for the for the first three models were great overall, it can be observed that the recall were significantly less than the other metrics. Recall is a metric that measures how often a machine learning model correctly identifies positive instances (true positives) from all the actual positive samples in the dataset (Evidently AI, 2024). This implied that there it was failing to identify few actual positive cases. Upon revisiting the dataset, it could be found that there was uneven distribution of the target variables, with 357 benign cases and 212 malignant cases. As such, Synthetic Minority Over-sampling Technique (SMOTE) was applied to balance the data. After this implication, the results across the performance were more consistent, as seen from **Model 3** onwards.

Other hyperparameters were further tuned, where for the activation functions for the outer layer, only sigmoid and tanh functions were used. This is because they restrict the output ranges between 0 and 1, which is good for the binary classification problem of this project.

Model No.	No. of layers	No. of neurons	Batch size	Activation function	Learning rate	Dropout rate	Accuracy	Precision	Recall	F1-Score
1	5	128, 64, 32, 16, 1	32	Relu Sigmoid (output)	0.001	0.03	0.95	0.96	0.91	0.94
2	4	64, 32, 16, 1	32	Relu Sigmoid (output)	0.001	0.03	0.95	0.96	0.91	0.94
3	3	32, 16, 1	32	Relu Sigmoid (output)	0.001	0.03	0.97	0.97	0.91	0.95
4 (after SMOTE)	5	128, 64, 32, 16, 1	32	Relu Sigmoid (output)	0.001	0.03	0.96	0.99	0.93	0.98
5	4	64, 32, 16, 1	32	Relu Sigmoid (output)	0.001	0.03	0.96	0.99	0.93	0.96

6	3	32, 16, 1	32	Relu Sigmoid (output)	0.001	0.03	0.97	0.99	0.95	0.97
7	2	16, 1	32	Relu Sigmoid (output)	0.001	0.03	0.97	0.97	0.96	0.97
8	**2**	**10, 1**	**32**	**Relu Sigmoid (output)**	**0.001**	**0.03**	**0.98**	**0.99**	**0.96**	**0.97**
9	2	10, 1	32	Relu tanh (output)	0.001	0.03	0.95	0.96	0.95	0.95
10	2	10, 1	32	Relu Sigmoid (output)	0.001	0.05	0.96	0.99	0.93	0.96
11	2	10, 1	64	Relu Sigmoid (output)	0.001	0.03	0.96	0.98	0.93	0.96
12	2	10, 1	32	Relu Sigmoid (output)	0.01	0.03	0.96	0.98	0.93	0.96
13	2	10, 1	32	Relu Sigmoid (output)	0.0001	0.03	0.97	0.98	0.94	0.96

5.2 Wide and Deep Model

5.2.1 Model Outline

The second neural network architecture used on the dataset was a Wide and Deep Model. This model jointly trains wide linear models and deep neural networks to combine the benefits of memorization and generalization for recommender systems (arXiv, 2016). The Wide and Deep model was chosen to capture both linear and complex non-linear relationships between the features and the target variable. The wide path directly inputs numerical features to capture linear relationships, while the deep path captures more complex interactions.

5.2.2 Hyperparameter Tuning

For this model, the hyperparameters that were manually tuned was the number of hidden layers for the deep part, number of neurons in each layer, activation functions, the learning rate and the dropout rate. Similarly, early stopping was applied to reduce overfitting and SMOTE was used to balance the data.

Having found that **Model 8** was the most optimal model in the previous section, the same hyperparameters were used for the deep part with the aim to build upon it. As the number of hidden layers and the number of neurons were increased, the recall decreased, like the first model. As seen in the table below, the hyperparameters were changed yet the first one performed the best.

Model No.	No. of hidden layers	No. of neurons	Activation function	Learning rate	Dropout rate	Accuracy	Precision	Recall	F1-Score
1	**1**	**10**	**Relu Sigmoid (output)**	**0.001**	**None**	**0.97**	**0.99**	**0.96**	**0.97**
2	2	16, 8	Relu Sigmoid (output)	0.001	None	0.96	0.96	0.96	0.96
3	2	32, 16	Relu Sigmoid (output)	0.001	None	0.96	0.99	0.93	0.96
4	3	64, 32, 16	Relu Sigmoid (output)	0.001	None	0.96	0.99	0.93	0.96
5	1	10	Relu Sigmoid (output)	0.0001	None	0.97	0.99	0.96	0.97
6	1	10	Relu Sigmoid (output)	0.01	None	0.96	0.98	0.94	0.96
7	1	10	Relu Sigmoid (output)	0.01	0.03	0.95	0.95	0.94	0.95
8	1	10	Relu Sigmoid (output)	0.001	0.03	0.96	0.99	0.95	0.96
9	1	10	Relu tanh (output)	0.001	0.03	0.94	0.94	0.93	0.94

6. Discussion

The best performing models from each architecture was then evaluated on the test set to simulate how it would perform on unseen data. The Deep Sequential Model and Wide and Deep Model performed equally well, with test accuracy of 0.98, precision of 0.99, recall of 0.97 and F1-score of 0.98. In comparison to the benchmark model, both deep learning models performed better overall, especially in the recall. This was an issue faced and improved with the hyperparameter tunings.

7. Potential Ethical Concerns

One potential ethical concern for using artificial intelligence for cancer treatment is data privacy and security. Usage of the dataset of the previous breast cancer diagnoses may be highly sensitive to the patients, requiring them to give consent to their data to be used. Furthermore, there may be issues with accountability and liability, as it is unclear who would be held responsible, the developers, the healthcare providers, or the institution implementing the system.

8. Appendix

8.1. Data Dictionary

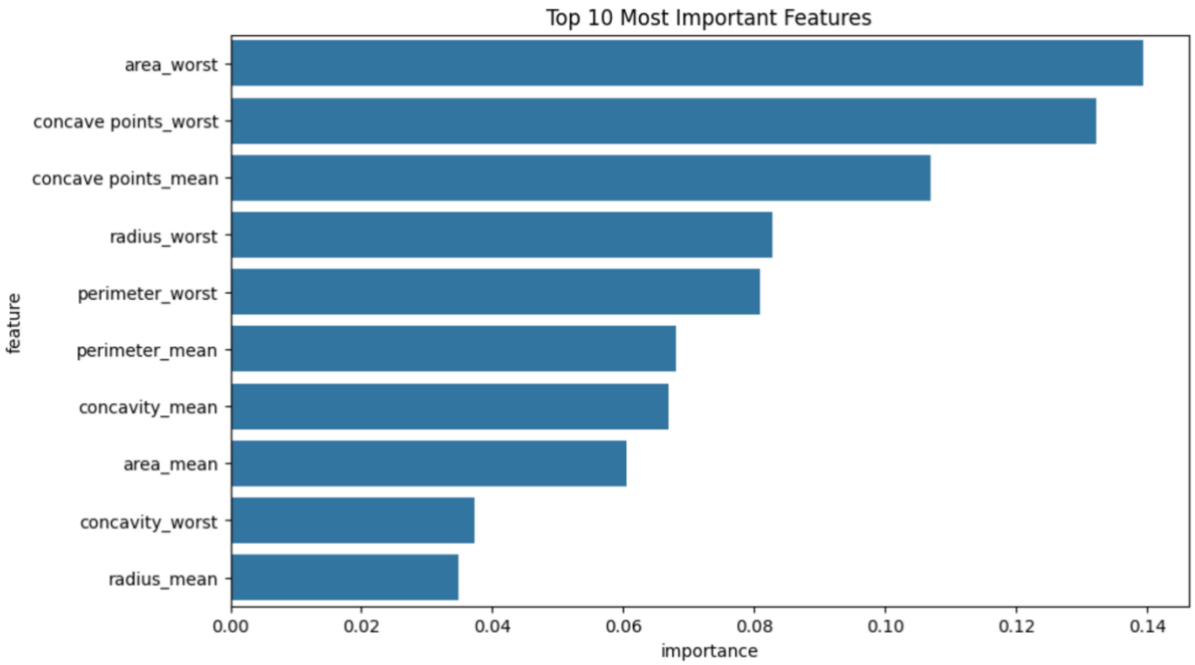
Variable Name	Description	Datatype
id	Unique identifier for each sample	Numeric
radius_mean	Mean of distances from centre to points on the perimeter of the cell nucleus	Numeric
texture_mean	Standard deviation of grayscale values in the cell nucleus	Numeric
perimeter_mean	Mean size of the core tumour	Numeric
area_mean	Mean area of the cell nucleus	Numeric
smoothness_mean	Mean of local variation in radius lengths	Numeric
compactness_mean	Mean of (perimeter^2 / area - 1.0)	Numeric
concavity_mean	Mean of severity of concave portions of the contour	Numeric
concave points_mean	Mean for number of concave portions of the contour	Numeric
symmetry_mean	Mean of symmetry of the cell nucleus	Numeric
fractal_dimension_mean	Mean for "coastline approximation" - 1	Numeric
radius_se	Standard error of the mean of distances from centre to points on the perimeter	Numeric
texture_se	Standard error of grayscale values in the cell nucleus	Numeric
perimeter_se	Standard error of the mean size of the core tumour	Numeric
area_se	Standard error of the cell nucleus area	Numeric
smoothness_se	Standard error of local variation in radius lengths	Numeric
compactness_se	Standard error of (perimeter^2 / area - 1.0)	Numeric
concavity_se	Standard error of severity of concave portions of the contour	Numeric
concave points_se	Standard error for number of concave portions of the contour	Numeric
symmetry_se	Standard error of symmetry of the cell nucleus	Numeric
fractal_dimension_se	Standard error for "coastline approximation" - 1	Numeric
radius_worst	"Worst" or largest mean value for distance from centre to points on the perimeter	Numeric
texture_worst	"Worst" or largest mean value for standard deviation of grayscale values	Numeric
perimeter_worst	"Worst" or largest mean value for core tumour size	Numeric
area_worst	"Worst" or largest mean value for cell nucleus area	Numeric

smoothness_worst	"Worst" or largest mean value for local variation in radius lengths	Numeric
compactness_worst	"Worst" or largest mean value for (perimeter^2 / area - 1.0)	Numeric
concavity_worst	"Worst" or largest mean value for severity of concave portions of the contour	Numeric
concave points_worst	"Worst" or largest mean value for number of concave portions of the contour	Numeric
symmetry_worst	"Worst" or largest mean value for symmetry of the cell nucleus	Numeric
fractal_dimension_worst	"Worst" or largest mean value for "coastline approximation" - 1	Numeric
diagnosis (TARGET)	**The diagnosis of breast tissues (M = malignant, B = benign)**	**Nominal Categorical**

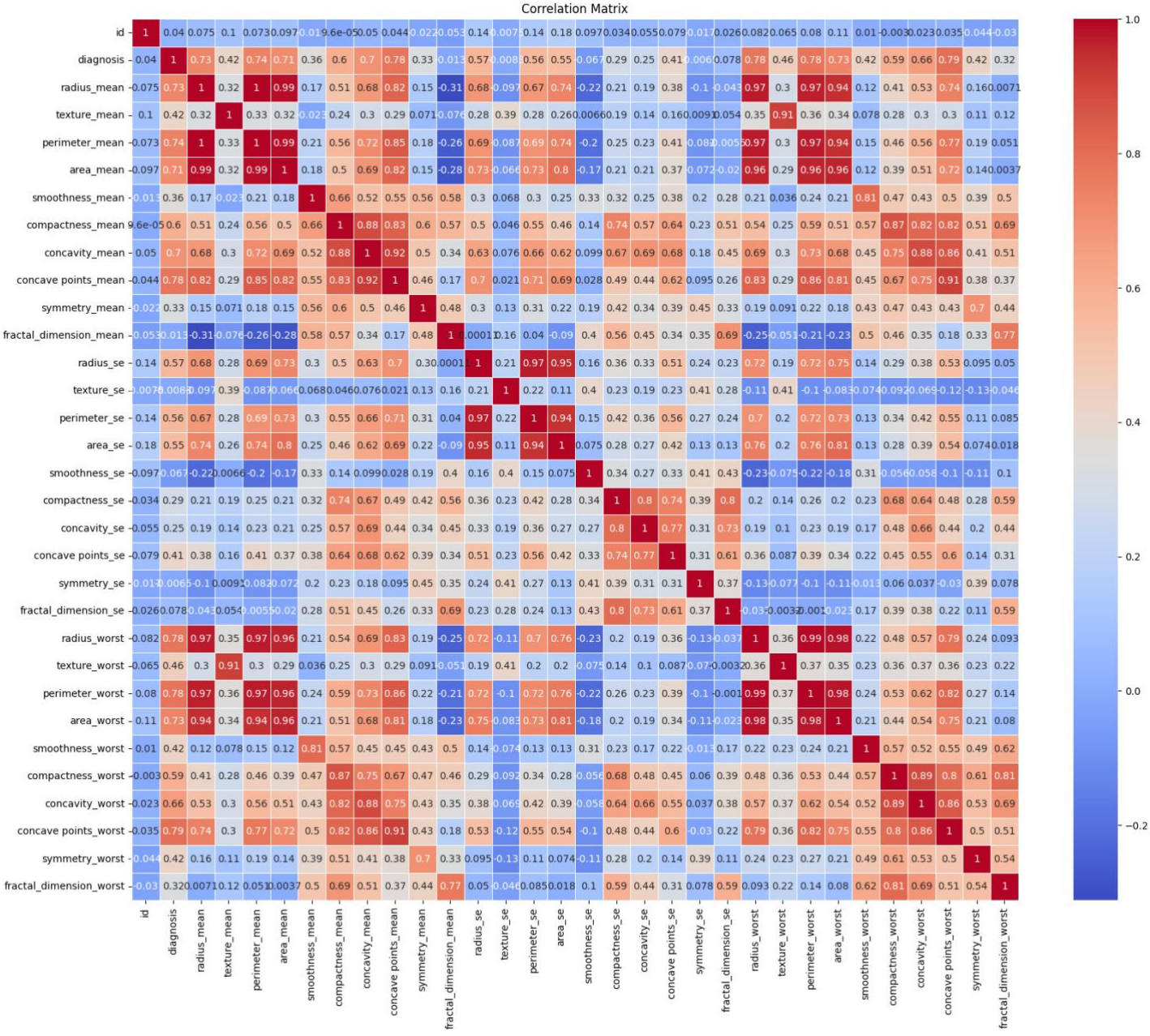
The ‘diagnosis’ column is the **target variable**, which has a datatype of nominal categorical. It is a binary classification problem, with ‘M’ representing malignant cases and ‘B’ representing benign cases.

8.2. Exploratory Data Analysis Plots

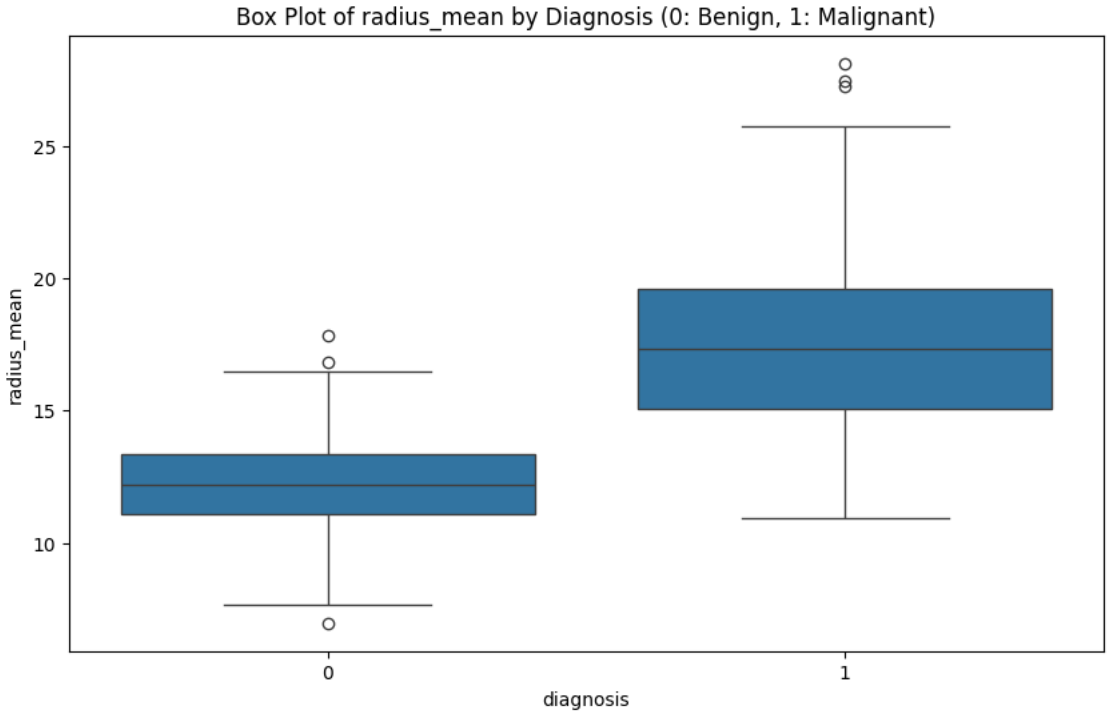
8.2 a)

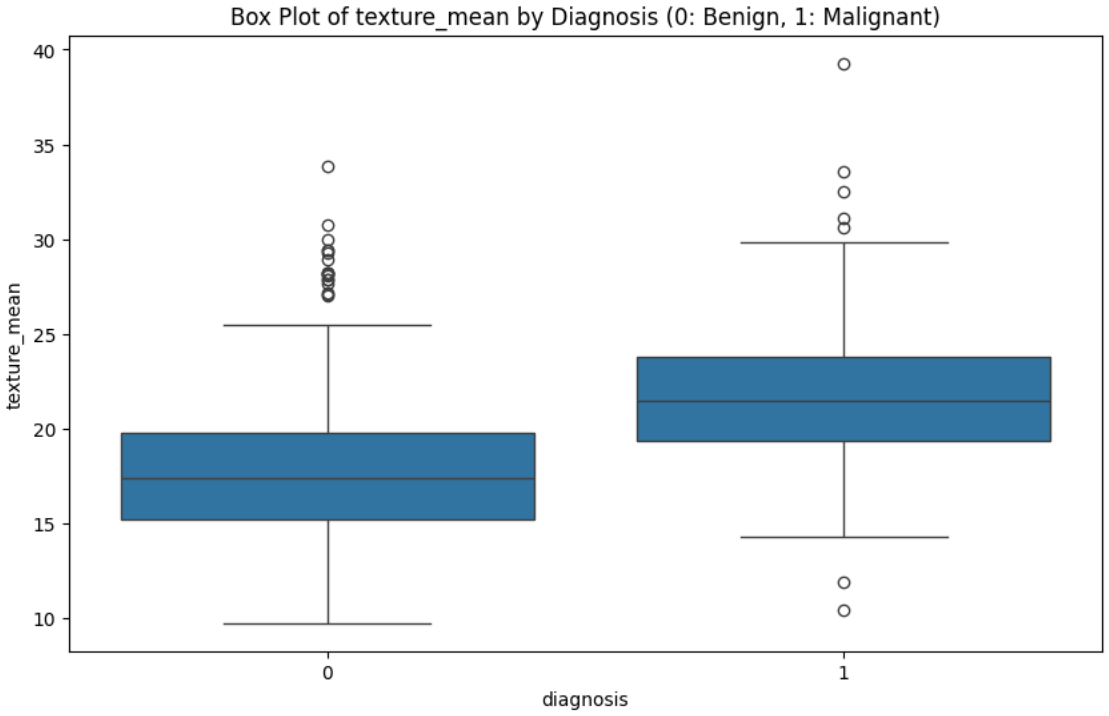


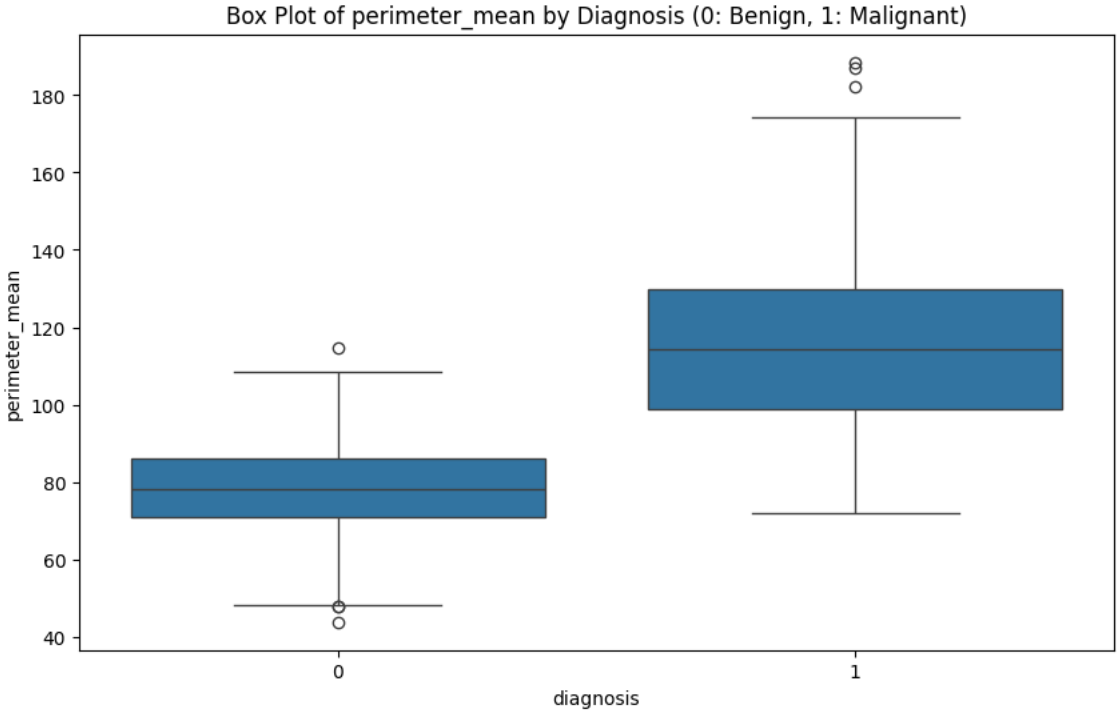
8.2 b)


Correlation Matrix
id
diagnosis
radius_mean
texture_mean
perimeter_mean
area_mean
smoothness_mean
compactness_mean
concavity_mean
concave points_mean
symmetry_mean
fractal_dimension_mean
radius_se
texture_se
perimeter_se
area_se
smoothness_se
compactness_se
concavity_se
concave points_se
symmetry_se
fractal_dimension_se
radius_worst
texture_worst
perimeter_worst
area_worst
smoothness_worst
compactness_worst
concavity_worst
concave points_worst
symmetry_worst
fractal_dimension_worst
1.0
0.8
0.6
0.4
0.2
0.0
−0.2

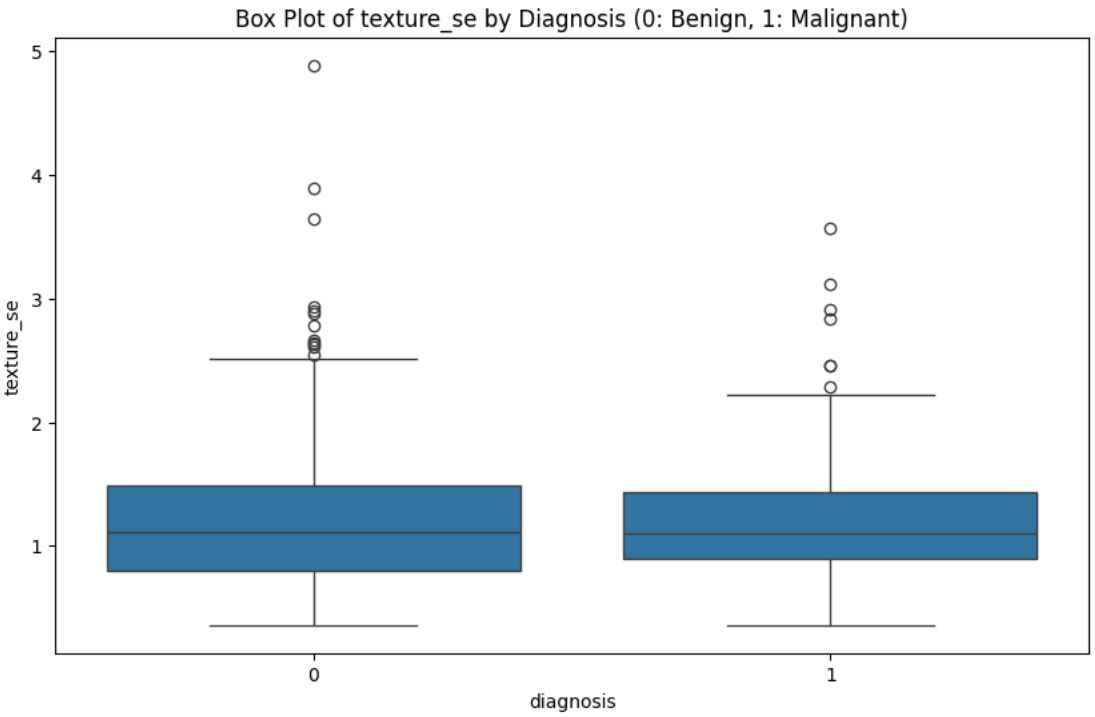
8.2 c)

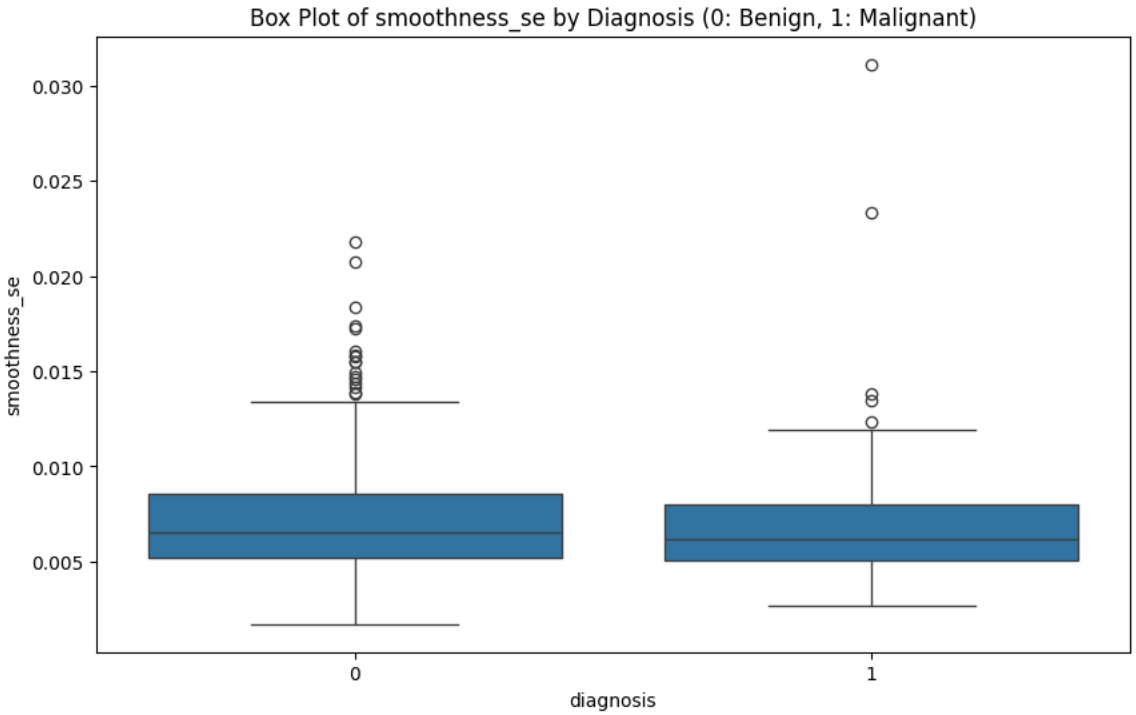

Box Plot of radius_mean by Diagnosis (0: Benign, 1: Malignant)
radius_mean
25
20
15
10
0
1
diagnosis

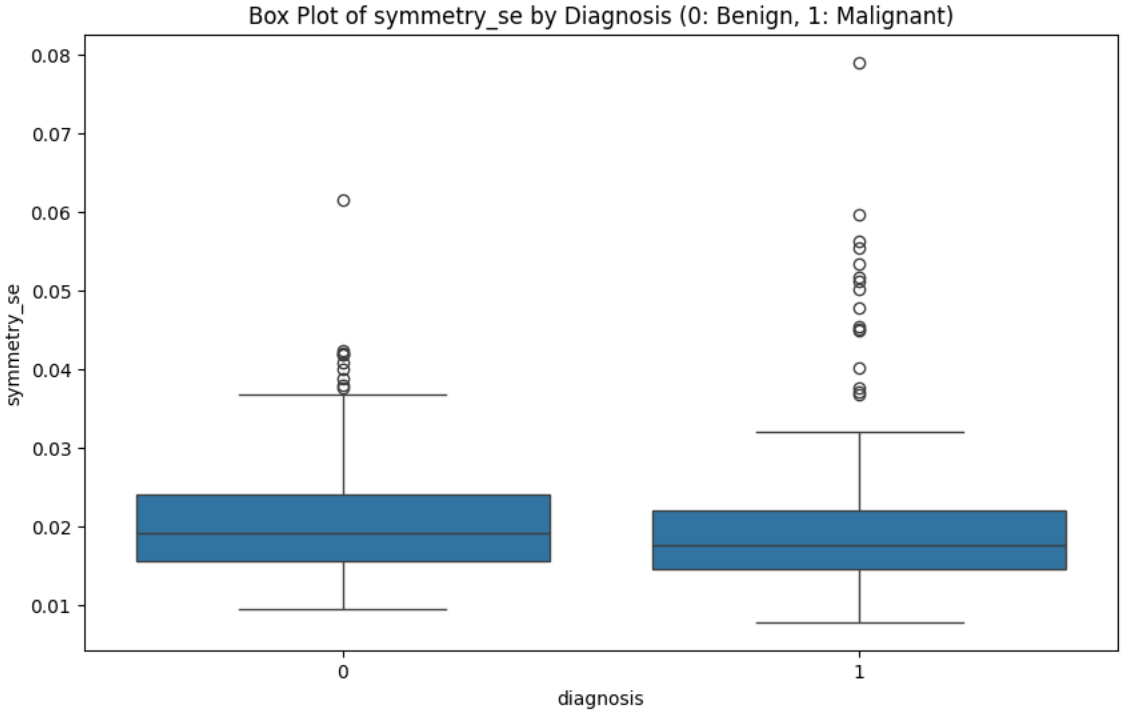

Box Plot of texture_mean by Diagnosis (0: Benign, 1: Malignant)
texture_mean
40
35
30
25
20
15
10
0
1
diagnosis


Box Plot of perimeter_mean by Diagnosis (0: Benign, 1: Malignant)
perimeter_mean
180
160
140
120
100
80
60
40
0
1
diagnosis

8.2 d)


Box Plot of texture_se by Diagnosis (0: Benign, 1: Malignant)
texture_se
5
4
3
2
1
0
1
diagnosis





8.3. Generative AI Usage

"I am looking to choose a supervised learning problem to investigate using a deep-learning model, give me some ideas"

This prompt was used to brainstorm ideas for the project.

"Check for any grammatical errors here: [text]"

This prompt was used to double check any grammatical errors with the assignment.

"My model is performing extremely well in every metric except recall. Give me some possible things to check to fix this."

> This prompt was used to help me investigate why the recall was always lower, eventually leading me to balance the data using SMOTE, improving this metric.

9 Reference

Wolberg,William, Mangasarian,Olvi, Street,Nick, and Street,W.. (1995). Breast Cancer Wisconsin (Diagnostic). UCI Machine Learning Repository. https://doi.org/10.24432/C5DW2B. (Accessed: 20 June 2024).

Buhl, N. (2023) *F1 score in Machine Learning*, *Encord*. Available at: https://encord.com/blog/f1-score-in-machine-learning/ (Accessed: 20 June 2024).

Learning, U.M. (2016) *Breast cancer wisconsin (diagnostic) data set*, *Kaggle*. Available at: https://www.kaggle.com/datasets/uciml/breast-cancer-wisconsin-data (Accessed: 20 June 2024).

Narkhede, S. (2021) *Understanding AUC - roc curve*, *Medium*. Available at: https://towardsdatascience.com/understanding-auc-roc-curve-68b2303cc9c5 (Accessed: 22 June 2024).

Ak, M.F. (2020) *A comparative analysis of breast cancer detection and diagnosis using data visualization and machine learning applications*, *Healthcare (Basel, Switzerland)*. Available at: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7349542/ (Accessed: 22 June 2024).

Lynne Eldridge, M. (2023) *Cancer cells vs. normal cells: How are they different?*, *Verywell Health*. Available at: https://www.verywellhealth.com/cancer-cells-vs-normal-cells-2248794 (Accessed: 22 June 2024).

Tracyrenee (2023) *What is the difference between a sequential and functional model in Keras tensorflow?*, Available at: https://tracyrenee61.medium.com/what-is-the-difference-between-a-sequential-and-functional-model-in-keras-tensorflow-ba2bc3ec700d#:~:text=Sequential%20model%3A%20The%20Sequential%20model,layer%2C%20in%20a%20sequential%20manner. (Accessed: 26 July 2024).

Swain, S. (2021) *Understanding sequential vs functional API in Keras*, *Analytics Vidhya*. Available at: https://www.analyticsvidhya.com/blog/2021/07/understanding-sequential-vs-functional-api-in-keras/ (Accessed: 27 July 2024).

EvidentlyAI (2024) *Accuracy vs. precision vs. recall in machine learning: What's the difference?*, *Evidently AI - Open-Source ML Monitoring and Observability*. Available at: https://www.evidentlyai.com/classification-metrics/accuracy-precision-recall#:~:text=Recall%20is%20a%20metric%20that,the%20number%20of%20positive%20instances. (Accessed: 27 July 2024).

Cheng, H.-T. *et al.* (2016) *Wide & Deep Learning for Recommender Systems*, *arXiv.org*. Available at: https://arxiv.org/abs/1606.07792 (Accessed: 27 July 2024).