Programming for Data Analytics 1

POE PART 2

Classificatiion and Model Improvement

Mukeba Mbunda Eliezer

ST10486340

PDDA0801 VCWCCR Term1 GRO1

Varsity College Cape Town Newlands

DUE: May 29, 2025

LECTURER NAME: Dr Rudolf Holzhausen

List of Figures

Contents

1 Introduction

The medical scheme chose to begin with cancer cases to accelerate their ability to apply dreaded disease benefits to customers in need. We chose the **Breast Cancer Dataset** due to its quality and suitability.

This dataset, downloaded from Kaggle, contains the characteristics of patients diagnosed with cancer and includes the following information:

- Id : represents a unique ID of each patient
- Diagnosis: indicates the type of cancer (M = Malignant, B = Benign)
- Clinical features: radius_mean, texture_mean, perimeter_mean, area_mean, smoothness_mean, compactness_mean, concavity_mean, concave_points_mean (represents the mean values of the cancer's visual characteristics)

Link to the dataset : https://www.kaggle.com/datasets/erdemtaha/cancer-data/data

First, in terms of quality, the dataset is clean (without missing values, duplicates, or inconsistencies), well-organized in rows and columns, ready for analysis. Each record contains the patient's unique ID, the cancer diagnosis, and the average values of the cancer's visual characteristics. In terms of suitability, the dataset contains labeled data, including both input features (radius_mean, texture_mean, perimeter_mean, area_mean, smoothness_mean, compactness_mean, concavity_mean, concave points_mean, etc) and the correct output (M = Malignant :dangerous and requires urgent attention , B = Benign: a type of medical condition or growth that is not cancerous or dangerous) making it suitable for the medical scheme's goal which is to classify customers based on their cancer diagnosis.

Logistic regression model is chosen because it provides a simple and efficient model for binary classification problems like this one, where the task is to predict whether a patient has a benign or malignant tumor. Because the dataset is small, well-structured, and has numerical values as features , logistic regression is a good choice. Although other models like SVM perform well for classification tasks, they often require complex kernel and parameter tuning to avoid overfitting (Singh, 2024). KNN relies heavily on large, clean datasets and can struggle with high-dimensional data (Rishabh. 2024). Random forest is powerful but can be computationally expensive and less interpretable (Samy. 2024). In contrast, logistic regression is simple, efficient, computationally inexpensive and works well with smaller datasets, and provides easily interpretable results (Abhishek. 2024), making it particularly suitable for this breast cancer classification task.

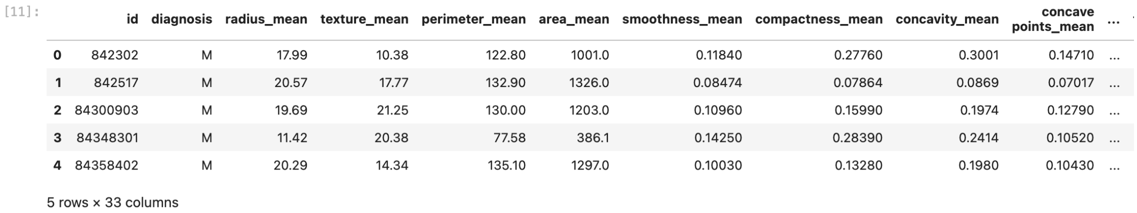
This report provides detailed findings on how we conducted a classification analysis to support the medical scheme. We start by explaining the reasons for choosing the dataset and the algorithm. Next, we perform exploratory data analysis (EDA) to uncover patterns within the data. After that, we train the model and finally test its performance on generated data.

2 Data exploration

In this section, we begin by importing the necessary libraries and exploring the dataset using Pandas. `Pandas` is a powerful Python library commonly used for data manipulation and analysis. It provides data structures and functions that make it easy to clean, analyze, and visualize structured data (Wschool 2019)

To load the data and display the first five rows, we use:

```
df = pd.read_csv('data.csv')
df.head(5)
```



[11]:

	id	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean	compactness_mean	concavity_mean	concave points_mean	...
0	842302	M	17.99	10.38	122.80	1001.0	0.11840	0.27760	0.3001	0.14710	...
1	842517	M	20.57	17.77	132.90	1326.0	0.08474	0.07864	0.0869	0.07017	...
2	84300903	M	19.69	21.25	130.00	1203.0	0.10960	0.15990	0.1974	0.12790	...
3	84348301	M	11.42	20.38	77.58	386.1	0.14250	0.28390	0.2414	0.10520	...
4	84358402	M	20.29	14.34	135.10	1297.0	0.10030	0.13280	0.1980	0.10430	...

5 rows × 33 columns

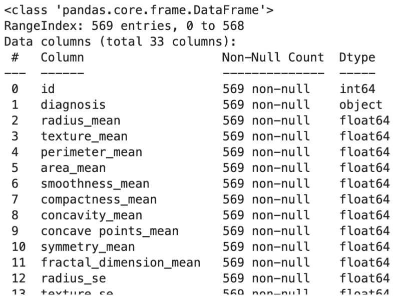
Figure 1: Dataset overview

To get the dimension of the dataset, we use :

```
df.shape
(569, 33)
```

To show all the information about the dataset, we use :

```
df.info()
(refer to the notebook for the full picture)
```



```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 569 entries, 0 to 568
Data columns (total 33 columns):
 #   Column                  Non-Null Count  Dtype  
---  ------                  --------------  -----  
 0   id                      569 non-null    int64  
 1   diagnosis               569 non-null    object 
 2   radius_mean             569 non-null    float64
 3   texture_mean            569 non-null    float64
 4   perimeter_mean          569 non-null    float64
 5   area_mean               569 non-null    float64
 6   smoothness_mean         569 non-null    float64
 7   compactness_mean        569 non-null    float64
 8   concavity_mean          569 non-null    float64
 9   concave points_mean     569 non-null    float64
 10  symmetry_mean           569 non-null    float64
 11  fractal_dimension_mean  569 non-null    float64
 12  radius_se               569 non-null    float64
 13  texture_se              569 non-null    float64
```

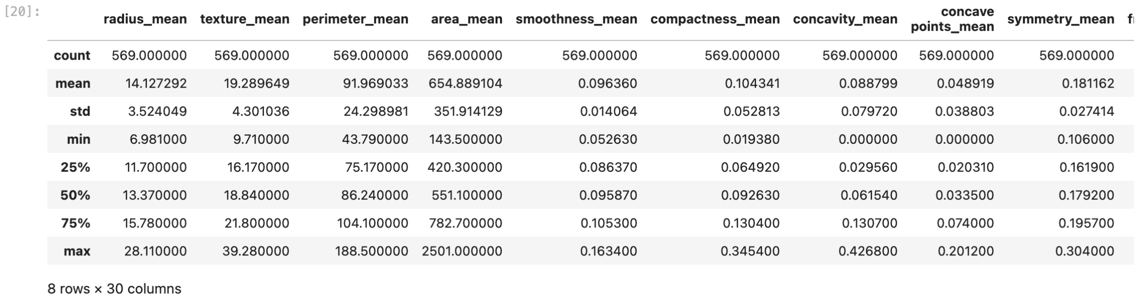
Figure 2: Dataset information

To remove unnecessary columns , we use :

```
df = df.drop(['id','Unnamed: 32'], axis=1)
```

To get a statistical summary of the the dataset , we use :

```
df.describe()
```



[20]:

	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean	compactness_mean	concavity_mean	concave points_mean	symmetry_mean	f
count	569.000000	569.000000	569.000000	569.000000	569.000000	569.000000	569.000000	569.000000	569.000000	
mean	14.127292	19.289649	91.969033	654.889104	0.096360	0.104341	0.088799	0.048919	0.181162	
std	3.524049	4.301036	24.298981	351.914129	0.014064	0.052813	0.079720	0.038803	0.027414	
min	6.981000	9.710000	43.790000	143.500000	0.052630	0.019380	0.000000	0.000000	0.106000	
25%	11.700000	16.170000	75.170000	420.300000	0.086370	0.064920	0.029560	0.020310	0.161900	
50%	13.370000	18.840000	86.240000	551.100000	0.095870	0.092630	0.061540	0.033500	0.179200	
75%	15.780000	21.800000	104.100000	782.700000	0.105300	0.130400	0.130700	0.074000	0.195700	
max	28.110000	39.280000	188.500000	2501.000000	0.163400	0.345400	0.426800	0.201200	0.304000	

8 rows × 30 columns

Figure 3: Statistical summary

To check for missing and duplicate values, we use :

```
df.isnull().sum()
df.duplicated().sum()
0  # No missing or duplicate values
```

3 Exploratory Data Analysis

According to Geeksforgeeks (2025) , Exploratory Data Analysis (EDA) is an important first step in data science projects. It involves looking at and visualizing data to understand its main features, find patterns, and discover how different parts of the data are connected. In this section, we analyzed the dataset by exploring three types of relationships: univariate, bivariate, and multivariate. We used the `matplotlib`, `seaborn`, and `plotly` libraries for data visualization.

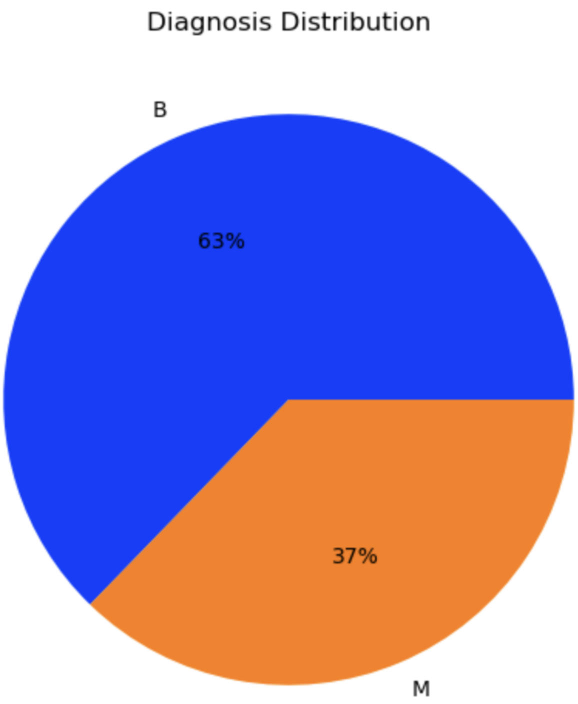


Figure 4: Pie chart distribution

From the figure above, we can observe that approximately 63% of the patients are diagnosed with benign tumors, while 37% have malignant tumors.

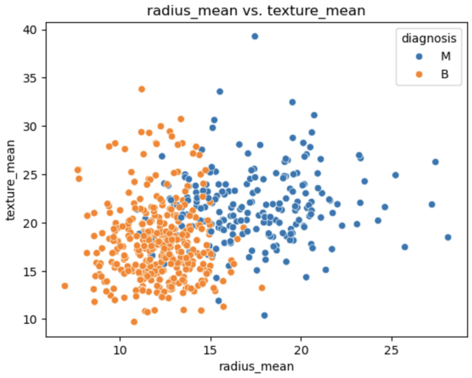


Figure 5: radium_mean vs. textur_mean

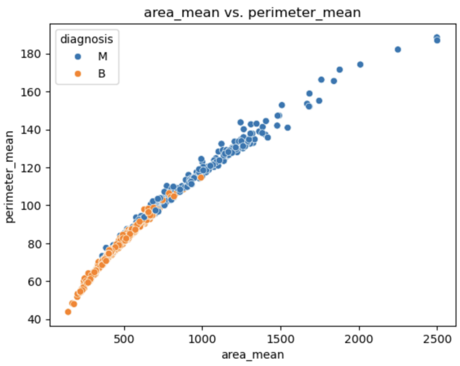


Figure 6: area_mean, perimeter_mean

From the figures above, The scatter plots show a clear separation between the two classes, indicating that the most features are effective for distinguishing between benign and malignant class.

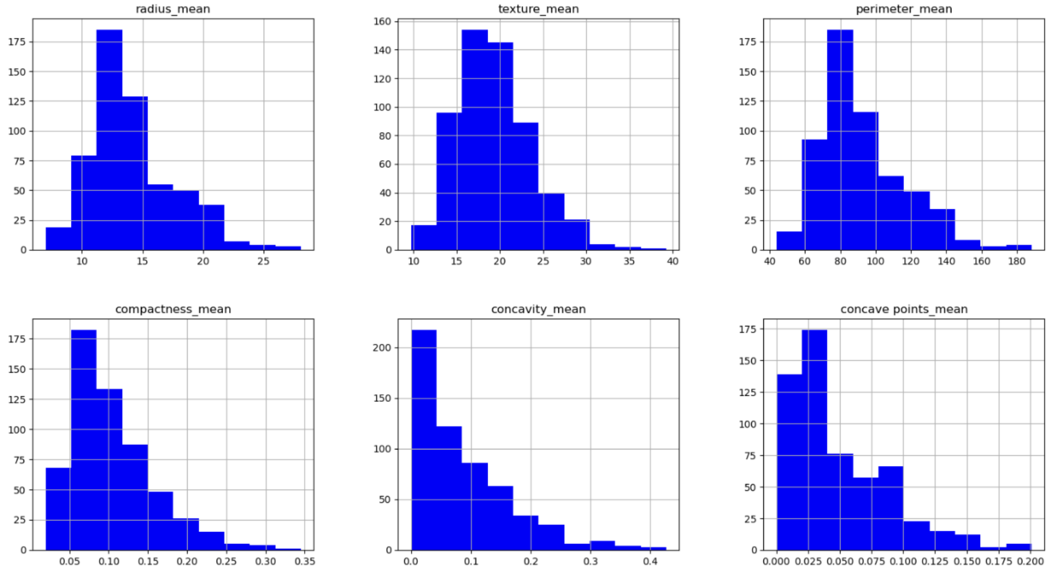


Figure 7: Histogram distribution

From the figures above, show that most features are right-skewed. This suggests the need for scaling or transformation before model training.

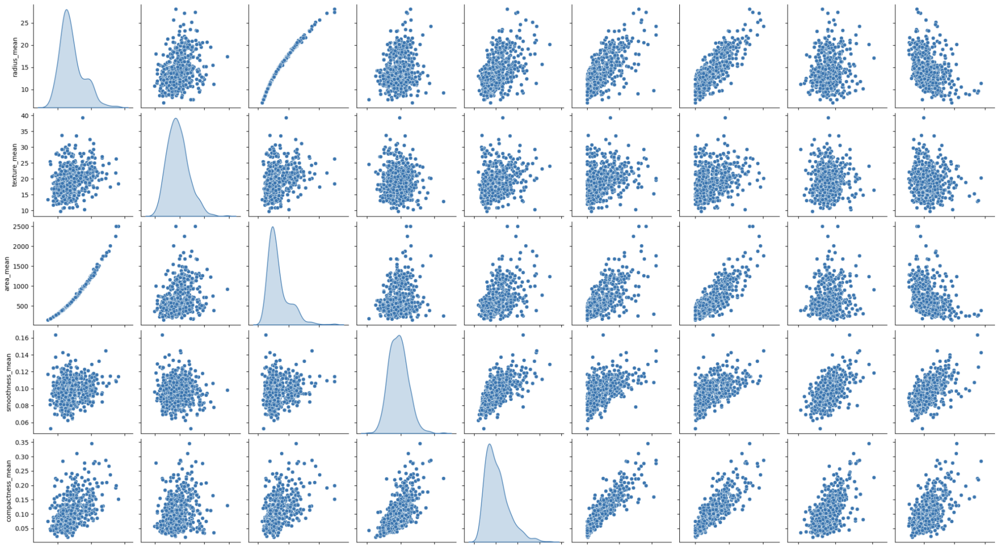


Figure 8: Pair Plot Distribution

From the figures above, The diagonal plots show the kde-plots distribution for each mean features, while the upper and lower triangles display the pairwise relationships between features

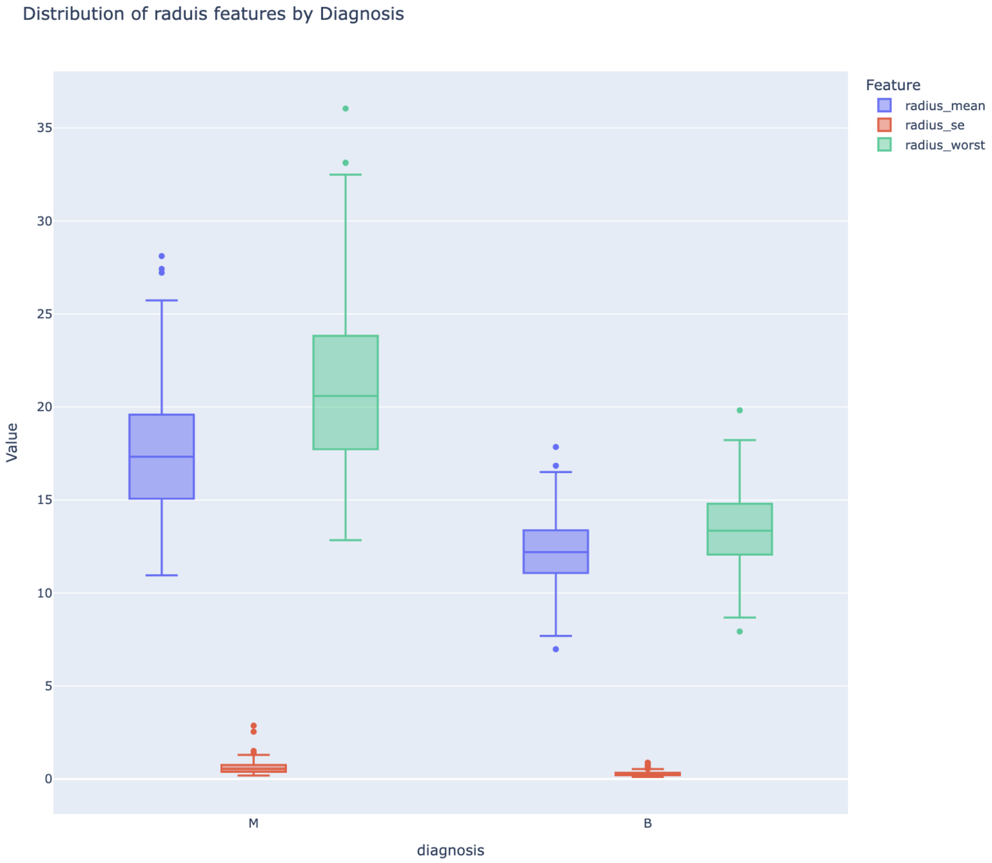


Figure 9: Box Plot Distribution

From the figure above, Each of the features have outliers as shown in the boxplot and therefore needs to be addressed during data preprocessing.

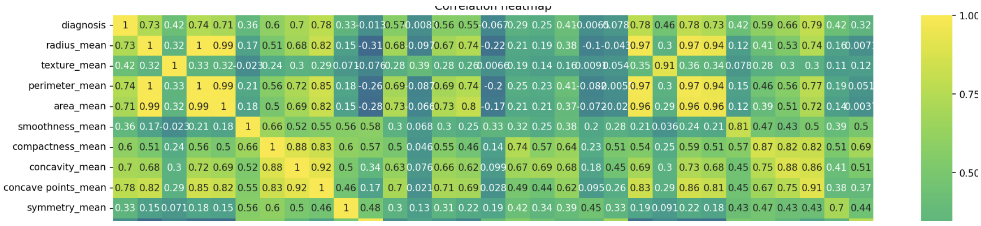


Figure 10: Correlation Heatmap

We can observe from the figure above that most of variables are highly correlated with others. For instance, radius, area, and perimeter show a strong correlation (corr_value = 0.90). This indicates the presence of multicollinearity in the data (*refer to the notebook for the full picture*).

4 Train model

We trained the Logistic regression model using a pipeline composed of four main steps:

1. **Min-Max Scaling:** We scaled the features to the range [0, 1] using `MinMaxScaler()`. This step standardizes the input values and improves the model convergence.

2. **GridSearchCV:** We used GridSearchCV to search for the best combination of hyperparameters that optimize model performance using `GridSearchCV()`.

3. **Dimensionality Reduction:** We applied Principal Component Analysis (PCA) to reduce the feature set to 5 principal components, thereby decreasing dimensionality and computational complexity.

4. **Classification:** Finally, we used `LogisticRegression` as the classifier.

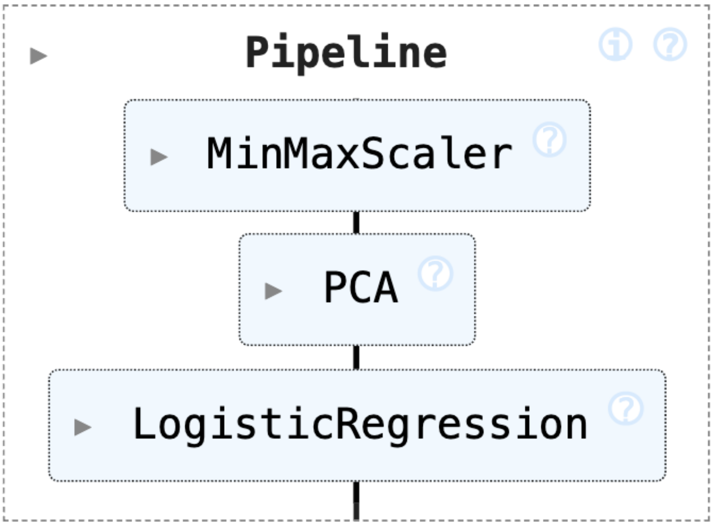
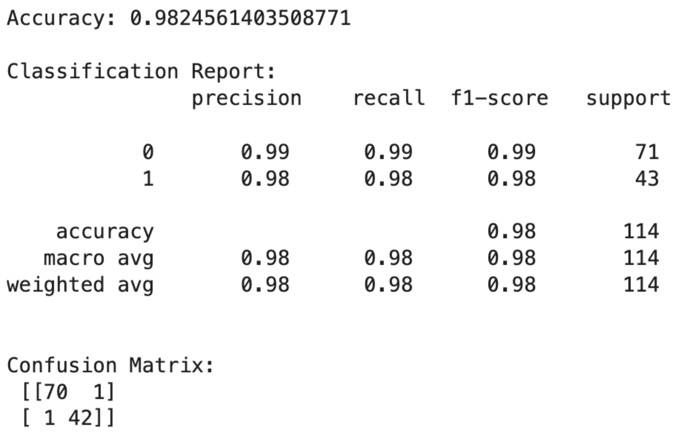


Figure 11: Train model

5 Model Evaluation and Interpretation



```
Accuracy: 0.9824561403508771

Classification Report:
              precision    recall  f1-score   support

           0       0.99      0.99      0.99        71
           1       0.98      0.98      0.98        43

    accuracy                           0.98       114
   macro avg       0.98      0.98      0.98       114
weighted avg       0.98      0.98      0.98       114


Confusion Matrix:
 [[70  1]
 [ 1 42]]
```

Figure 12: Model evaluation

The logistic regression model performs well, with an accuracy of about 98.24%. It correctly predicts the diagnosis of cancer, whether it is class B or class M . The precision for benign tumors is 99% and 0.98 for malignant tumors. The recall values are high for benign (0.99) and malignant (0.98) tumors, and the f1 score is also higher for both, demonstrating the reliability of the model.

The model achieved an average of 98% in terms of precision, recall, and F1 score by treating both classes equally, regardless of the number of samples in each class. This means that the model performs very well for both classes. When we also consider the number of samples in each class (called the weighted average), the performance remains high, showing that the model is still accurate across the entire dataset.

The confusion matrix, also known as the error matrix, allows visualization of the performance of the algorithm :

- true positive (TP = 42) : Malignant tumour correctly identified as malignant
- true negative (TN = 70) : Benign tumour correctly identified as benign
- false positive (FP = 1) : Benign tumour incorrectly identified as malignant
- false negative (FN = 1) : Malignant tumour incorrectly identified as benign

6 Model Improvement : Ensemble leaners

We used an ensemble model to improve prediction accuracy by combining the strengths of multiple classifiers, including Random Forest, KNN, SVM, and logistic regression, resulting in a more robust and reliable final model.

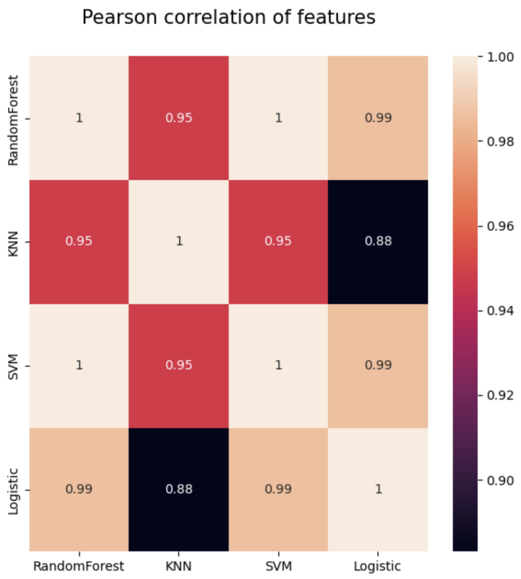
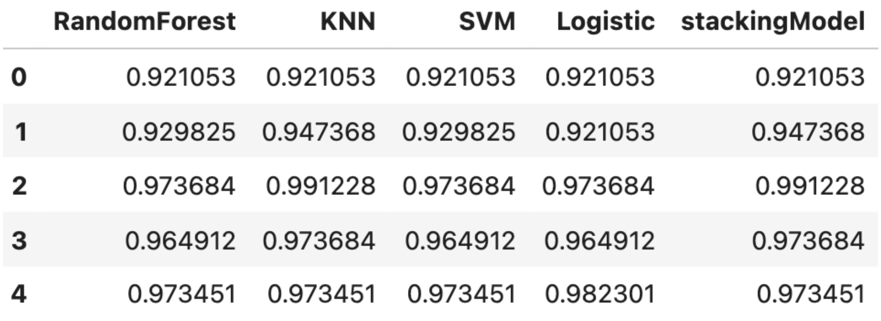


Figure 13: Correlation Heatmap of models

The figure above shows the correlation heatmap of model performance across different levels. Highly correlated models tend to behave similarly. For example, KNN and Random Forest show a high correlation, indicating similar behavior. In contrast, logistic regression and KNN show a weaker correlation, suggesting that their predictions differ across levels.



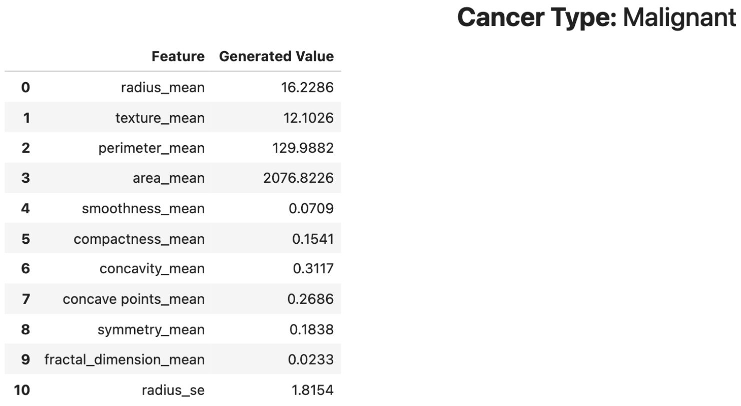
	RandomForest	KNN	SVM	Logistic	stackingModel
0	0.921053	0.921053	0.921053	0.921053	0.921053
1	0.929825	0.947368	0.929825	0.921053	0.947368
2	0.973684	0.991228	0.973684	0.973684	0.991228
3	0.964912	0.973684	0.964912	0.964912	0.973684
4	0.973451	0.973451	0.973451	0.982301	0.973451

Figure 14: Result of models

We observed that even after applying ensemble learning methods such as stacking, the model accuracy decreased slightly indicating that ensemble learning did not improve performance in this case. logistic model trained alone was already good and performed well.

7 Predicting Cancer Type

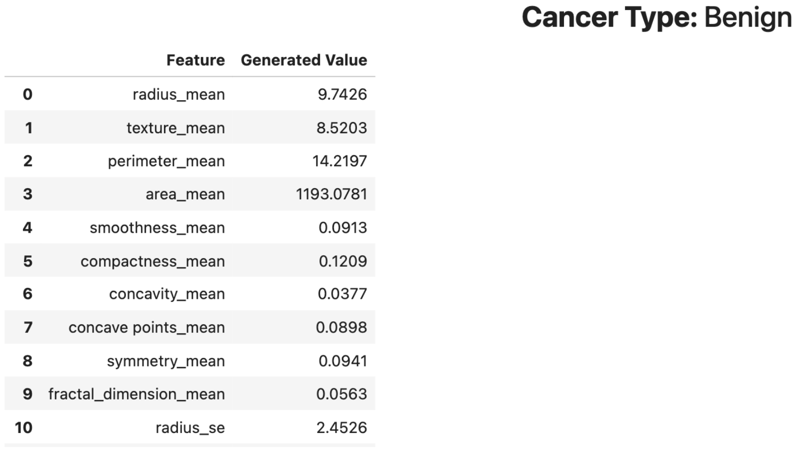
We created a random function that generates features randomly and used the trained model to predict whether the patient is likely to have a malignant or benign cancer. (*Not all 30 features are displayed; please refer to the notebook for the dataframe.)*



Cancer Type: Malignant

	Feature	Generated Value
0	radius_mean	16.2286
1	texture_mean	12.1026
2	perimeter_mean	129.9882
3	area_mean	2076.8226
4	smoothness_mean	0.0709
5	compactness_mean	0.1541
6	concavity_mean	0.3117
7	concave points_mean	0.2686
8	symmetry_mean	0.1838
9	fractal_dimension_mean	0.0233
10	radius_se	1.8154

Figure 15: Malignant Cancer Type



Cancer Type: Benign

	Feature	Generated Value
0	radius_mean	9.7426
1	texture_mean	8.5203
2	perimeter_mean	14.2197
3	area_mean	1193.0781
4	smoothness_mean	0.0913
5	compactness_mean	0.1209
6	concavity_mean	0.0377
7	concave points_mean	0.0898
8	symmetry_mean	0.0941
9	fractal_dimension_mean	0.0563
10	radius_se	2.4526

Figure 16: Benign Cancer Type

References

Abhishek., S (2024). *Logistic Regression and Its Role in Classification Problems*. https://medium.com/@abhishekshaw020/logistic-regression-and-its-role-in-classification-problems-504ff348bb41. Accessed: 28 May 2025.

Geeksforgeeks (2025). *What is Exploratory Data Analysis?* Accessed: 22 April 2025. URL: https://www.geeksforgeeks.org/what-is-exploratory-data-analysis/.

Rishabh., S (2024). *KNN (K-Nearest Neighbour)*. https://medium.com/@RobuRishabh/support-vector-machines-svm-27cd45b74fbb. Accessed: 28 May 2025.

Samy., B (2024). *Random Forest, Explained: A Visual Guide with Code Examples*. https://medium.com/data-science/random-forest-explained-a-visual-guide-with-code-examples-9f736a6e1b3c. Accessed: 28 May 2025.

Wschool (2019). *Pandas Introduction*. Accessed: 22 April 2025. URL: https://www.w3schools.com/python/pandas/pandas_intro.asp.