Est. 1841 YORK ST JOHN UNIVERSITY

1 Module – LDS7003M Artificial Intelligence and Machine Learning

Student Number (as shown on student ID card):	240184339
Word Count / Pages / Duration / Other Limits:	4290
Attempt Number:	1
Date of Submission:	28.01.2025

I have read and understood the Academic Misconduct statement.	Tick to confirm ☒
I have read and understood the Generative Artificial Intelligence use statement.	Tick to confirm ☒
I am satisfied that I have met the Learning Outcomes of this assignment (please check the Assignment Brief if you are unsure)	Met ☒

Self-Assessment – If there are particular aspects of your assignment on which you would like feedback, please indicate below.

Optional for students

Suggested prompt questions-

How have you developed or progressed your learning in this work?

What do you feel is the strongest part of this submission?

What feedback would you give yourself?

What part(s) of this assignment are you still unsure about?

Table of Contents

List of Figures

2 Part 1 – Theatrical Analysis

The essence of ontology in structuring knowledge in Artificial Intelligence systems

2.1 What is ontology

In AI terminology an ontology specifies a domain's discussion through an explicit formal way. A structured framework exists to categorize and arrange all information. The backbone structure of intelligent systems exists because ontologies help these systems grasp and handle complicated data through defined hierarchical connections between concepts (Andre, 2023). Knowledge bases consist of ontology structures that combine domains with proper examples of their classes. The boundary between an ontology and a knowledge base remains delicate.

2.2 The Importance of the Ontology on Knowledge Organization

Ontology in AI is a formalized specification of what is presumed to be known within some domain, as well as what can be considered an object, its properties, its relations, and the constraints that govern it. It offers a worldwide language that guarantees coherence and integration with other systems that engage in Artificial Intelligence. Ontologies also have a crucial role in structuring knowledge because when knowledge is organized ontologies assist the specific AI systems to be able to easily categorize information and data. They complement the AI applications by applying semantic understanding on data and polarizing them to mean something and something'. Ontologies improve the reasoning ability since the information is well arranged. For instance, an ontology in a system used to diagnose medical conditions would show connections between symptoms, diseases and their cures. Acquired in this manner such structured knowledge facilitates the proper assessment of possible diagnoses from observed symptoms thus enhancing the decision-making process within the system.

2.3 Impact on reasoning and Decision making

The organizational pattern of Ontology directly shapes both reasoning operations and decision-making activities. Granularity measurements control how detailed ontological information becomes which affects both computational performance and diagnostic precision. The combination of Coherence and Scalability generates systems that both resist illogical decisions and scale efficiently with new information. Applications using domain-specific ontologies provide specialized insights but extreme specificity within the ontology limits its applicability. A robust ontology infrastructure leads to intelligent flexible AI systems as evidenced through chatbots which use ontology to organize intents alongside responses while promoting contextual interactions

2.4 Future impact in evolving domains

Ontologies represent fundamental constructs within emerging research fields by organizing their knowledge base. The diagnostic and research capabilities of AI applications find their foundation in healthcare systems SNOMED CT and ICD. By using ontologies in finance experts capture transaction data and regulatory information to build systems that detect fraud while ensuring compliance protocols. Using ontological frameworks autonomous vehicles describe environments and objects which leads to superior decision-making capabilities under dynamic conditions. Traditionally used as foundational ultimate enabling components for artificial intelligence applications throughout emerging domains.

2.5 Healthcare field case study

The technological system Gene Ontology (GO) demonstrates crucial implementation of ontology within bioinformatics applications. The Gene Ontology framework gives complete coverage for representing fundamental gene and gene product attributes throughout all species (Anon., 2025). Through tagging genes and proteins GO enables systematic biological process analysis which drives genomics and personalized medicine advancements. Upgrading the Gene Ontology system through machine learning for predictions and interactive graphical displays and instantaneous updates will significantly increase its reach and effects in bioinformatics research

2.6 Summary

Ontologies are prerequisite to organizing the knowledge in the AI systems as these presidencies and decisions are reflectively shaped and moulded by their architecture. The use of these concepts in newly developing professions is evidence of their ability to propose actual changes. However, the tactics in building and sustaining ontologies for scalability, usability, and tight integration with new forms of Artificial Intelligence applications are crucial.

3 Part 2: Practical Implementation

Utilizing Classification and Clustering for predictive analysis and identifying patterns in healthcare sector

3.1 Identify Problem

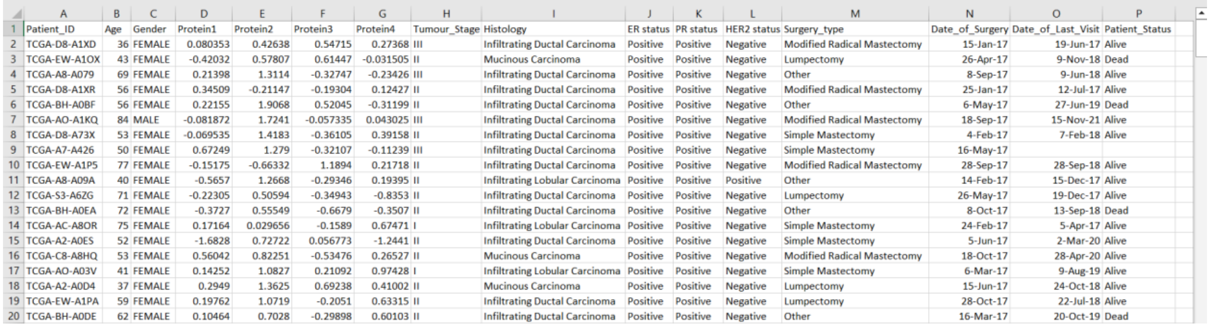
In healthcare, large data sets allow predicting its future behaviours with the help of classification and clustering. Classification helps to predict the outcome given by a set of characteristics and could be used to help in treatment of patients depending on whether the disease will progress or not. It allows offering targeted interventions and reveals patterns within data about patients' characteristics not revealed using conventional statistical methods. All these methods benefiting decision making, patient care, and precision medicine. With these techniques, it is therefore possible to turn data into useful information that can fuel innovation and enhance efficiency in the ever-increasing data laden health care provision realm.

The present research is concerned with the application of big data analysis for the improvement of breast cancer patient treatment. Applying the approach of classification, it allows to forecast the state of recovering and can be useful for further individual approach to therapy. Furthermore, mechanisms of clustering will also be used to group the patients based on their clinical and demographic characteristics for subsequent targeted intervention as well as further patient characterization. Both of these approaches are intended to enhance treatment and benefits as well as further the development of precision medicine for breast cancer treatment.

One example utilization in breast cancer care relates to classification with clustering and other kinds of tactics. One of the possibilities of a classification is to state, whether a patient is recovering, requires further treatment, or has a relapse to ensure timely action. On the other hand, what is referred to as clustering looks for subsets of patients who have similarity in factors like stage of cancer, type of surgery, and biomarkers. These clusters help healthcare facilities to devise specific therapies and disparage the nature of diseases more effectively thus helping the patients and boosting the formation of personalized medicine.

3.2 Dataset selection

The research uses real breast_cancer_data.csv (downloaded file name is BRCA.csv, renamed it for easiness for readers.) dataset entailing details such as demography and clinical data of breast cancer patients. Demographic access comprises gender and age, providing information on the patients. Clinical characteristics include disease extent or stage, histological tumour type, receptor status including; estrogen receptor (ER), progesterone receptor (PR) and HER2 (human epidermal growth factor receptor. Among others, and details of surgeries performed. The outcome variable revolves around patient recovery, that is, a patient's recovery status; either has or has not recovered, is on treatment or has even relapsed (amandam1, 2021). This type of data offers a resistant framework for pattern recognition and prediction necessary for creating better models of breast cancer treatment.



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1	Patient_ID	Age	Gender	Protein1	Protein2	Protein3	Protein4	Tumour_Stage	Histology	ER status	PR status	HER2 status	Surgery_type	Date_of_Surgery	Date_of_Last_Visit	Patient_Status
2	TCGA-D8-A1XD	36	FEMALE	0.080353	0.42638	0.54715	0.27368	III	Infiltrating Ductal Carcinoma	Positive	Positive	Negative	Modified Radical Mastectomy	15-Jan-17	19-Jun-17	Alive
3	TCGA-EW-A1OX	43	FEMALE	-0.42032	0.57807	0.61447	-0.031505	II	Mucinous Carcinoma	Positive	Positive	Negative	Lumpectomy	26-Apr-17	9-Nov-18	Dead
4	TCGA-A8-A079	69	FEMALE	0.21398	1.3114	-0.32747	-0.23426	III	Infiltrating Ductal Carcinoma	Positive	Positive	Negative	Other	8-Sep-17	9-Jun-18	Alive
5	TCGA-D8-A1XR	56	FEMALE	0.34509	-0.21147	-0.19304	0.12427	II	Infiltrating Ductal Carcinoma	Positive	Positive	Negative	Modified Radical Mastectomy	25-Jan-17	12-Jul-17	Alive
6	TCGA-BH-A0BF	56	FEMALE	0.22155	1.9068	0.52045	-0.31199	II	Infiltrating Ductal Carcinoma	Positive	Positive	Negative	Other	6-May-17	27-Jun-19	Dead
7	TCGA-AO-A1KQ	84	MALE	-0.081872	1.7241	-0.057335	0.043025	III	Infiltrating Ductal Carcinoma	Positive	Positive	Negative	Modified Radical Mastectomy	18-Sep-17	15-Nov-21	Alive
8	TCGA-D8-A73X	53	FEMALE	-0.069535	1.4183	-0.36105	0.39158	II	Infiltrating Ductal Carcinoma	Positive	Positive	Negative	Simple Mastectomy	4-Feb-17	7-Feb-18	Alive
9	TCGA-A7-A426	50	FEMALE	0.67249	1.279	-0.32107	-0.11239	III	Infiltrating Ductal Carcinoma	Positive	Positive	Negative	Simple Mastectomy	16-May-17		
10	TCGA-EW-A1P5	77	FEMALE	-0.15175	-0.66332	1.1894	0.21718	II	Infiltrating Ductal Carcinoma	Positive	Positive	Negative	Modified Radical Mastectomy	28-Sep-17	28-Sep-18	Alive
11	TCGA-A8-A09A	40	FEMALE	-0.5657	1.2668	-0.29346	0.19395	II	Infiltrating Lobular Carcinoma	Positive	Positive	Positive	Other	14-Feb-17	15-Dec-17	Alive
12	TCGA-S3-A6ZG	71	FEMALE	-0.22305	0.50594	-0.34943	-0.8353	II	Infiltrating Ductal Carcinoma	Positive	Positive	Negative	Lumpectomy	26-May-17	19-Dec-17	Alive
13	TCGA-BH-A0EA	72	FEMALE	-0.3727	0.55549	-0.6679	-0.3507	II	Infiltrating Ductal Carcinoma	Positive	Positive	Negative	Other	8-Oct-17	13-Sep-18	Dead
14	TCGA-AC-A8OR	75	FEMALE	0.17164	0.029656	-0.1589	0.67471	I	Infiltrating Lobular Carcinoma	Positive	Positive	Negative	Simple Mastectomy	24-Feb-17	5-Apr-17	Alive
15	TCGA-A2-A0ES	52	FEMALE	-1.6828	0.72722	0.056773	-1.2441	II	Infiltrating Ductal Carcinoma	Positive	Positive	Negative	Simple Mastectomy	5-Jun-17	2-Mar-20	Alive
16	TCGA-C8-A8HQ	53	FEMALE	0.56042	0.82251	-0.53476	0.26527	II	Mucinous Carcinoma	Positive	Positive	Negative	Modified Radical Mastectomy	18-Oct-17	28-Apr-20	Alive
17	TCGA-AO-A03V	41	FEMALE	0.14252	1.0827	0.21092	0.97428	I	Infiltrating Lobular Carcinoma	Positive	Positive	Negative	Simple Mastectomy	6-Mar-17	9-Aug-19	Alive
18	TCGA-A2-A0D4	37	FEMALE	0.2949	1.3625	0.69238	0.41002	II	Mucinous Carcinoma	Positive	Positive	Negative	Lumpectomy	15-Jun-17	24-Oct-18	Alive
19	TCGA-EW-A1PA	59	FEMALE	0.19762	1.0719	-0.2051	0.63315	II	Infiltrating Ductal Carcinoma	Positive	Positive	Negative	Lumpectomy	28-Oct-17	22-Jul-18	Alive
20	TCGA-BH-A0DE	62	FEMALE	0.10464	0.7028	-0.29898	0.60103	II	Infiltrating Ductal Carcinoma	Positive	Positive	Negative	Other	16-Mar-17	20-Oct-19	Dead

Figure 1: breast_cancer_dataset (BRCA) raw data

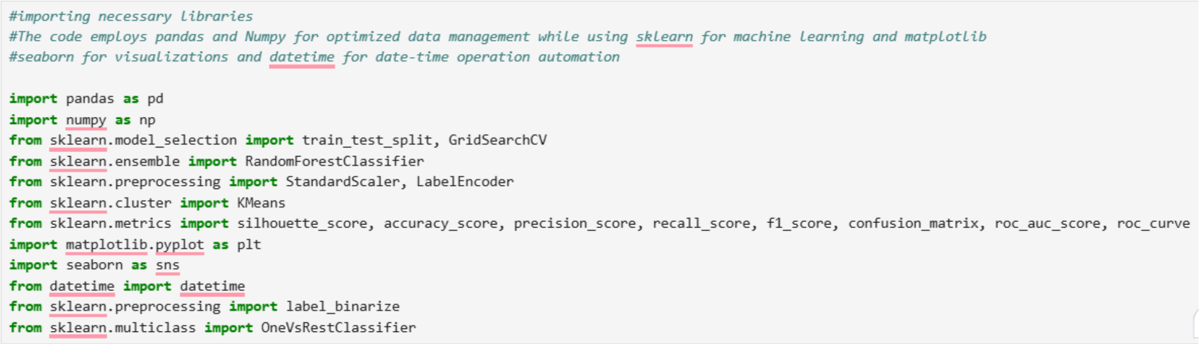
This set of features includes both categorical and numerical data making it suitable for classification and clustering. The dataset contains a number of predictors required for the outcomes assessment and subgrouping that adds to the more complex analysis in the field of breast cancer study.

3.3 Model Development

This work involves the process of modelling because it turns mere data into knowledge by which one can make accurate predictions and discoveries. This way research is able to foresee patient prospection, pick out patterns and classify comparable occurrences in order to treat them. It also guarantees efficient utilization of data in execution of decisions, enhancement of healthcare and lifelong learning for personalized healthcare. Model development also affirms the accuracy of conclusions and establishes a model of applicability for breast cancer intervention and treatment improvement.

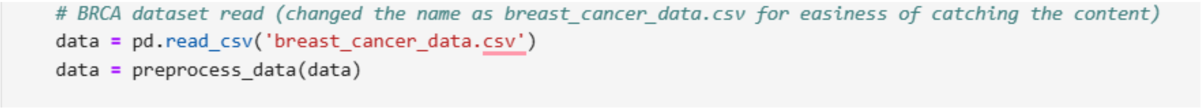
3.3.1 Data Pre-processing approach

Data preprocessing helps to make the collected datasets accurate and structured, free from any inconsistencies so that upon analysis there will be improved model performance.



```python
#importing necessary libraries
#The code employs pandas and Numpy for optimized data management while using sklearn for machine learning and matplotlib
#seaborn for visualizations and datetime for date-time operation automation

import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split, GridSearchCV
from sklearn.ensemble import RandomForestClassifier
from sklearn.preprocessing import StandardScaler, LabelEncoder
from sklearn.cluster import KMeans
from sklearn.metrics import silhouette_score, accuracy_score, precision_score, recall_score, f1_score, confusion_matrix, roc_auc_score, roc_curve
import matplotlib.pyplot as plt
import seaborn as sns
from datetime import datetime
from sklearn.preprocessing import label_binarize
from sklearn.multiclass import OneVsRestClassifier
```

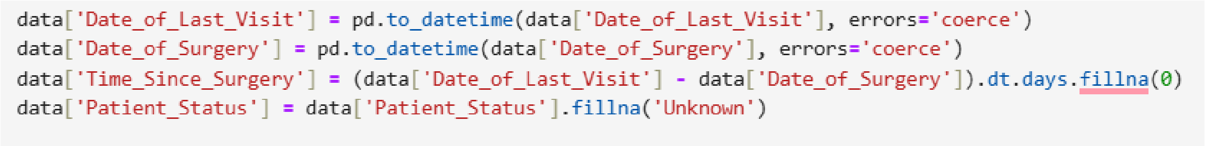
Figure 2: importing essential libraries



```
# BRCA dataset read (changed the name as breast_cancer_data.csv for easiness of catching the content)
data = pd.read_csv('breast_cancer_data.csv')
data = preprocess_data(data)
```

Figure 3: loading the raw data to process

➢ Handling missing value



```
data['Date_of_Last_Visit'] = pd.to_datetime(data['Date_of_Last_Visit'], errors='coerce')
data['Date_of_Surgery'] = pd.to_datetime(data['Date_of_Surgery'], errors='coerce')
data['Time_Since_Surgery'] = (data['Date_of_Last_Visit'] - data['Date_of_Surgery']).dt.days.fillna(0)
data['Patient_Status'] = data['Patient_Status'].fillna('Unknown')
```

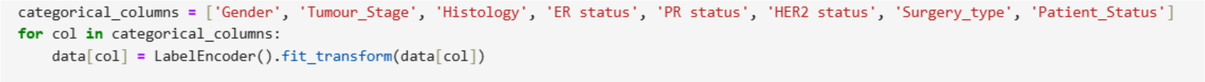
Figure 4: handle missing values on dataset

This code manages the missing values on a data set. All entries in Date_of_Last_Visit and Date_of_Surgery columns are converted to datetime format and entering an invalid date will result to NaT. The difference between these dates is expressed in days and called Time_Since_Surgery, while missing values are benchmarked to zero. For the categorical data, given values in Patient_Status are imputed by fill_value = ‘Unknown’. This ensures that for the dataset one is handling that the data is clean and has been standardized to reduce chances of making wrong analysis due to missing and invalid data.

➢ Feature engineering

This code carries out feature engineering, where it creates a new feature, which is the Time_Since_Surgery. It defines the simple difference between Date_of_Last_Visit and Date_of_Surgery to give essence into treatment plan of specific patient. The dt.days extracts the difference in days, and .fillna(0) fills any NA’s in the feature with zero so the feature is useable for analysis. This feature is enabling for the identification of surgery time and effects on a patient’s outcome.

➢ Encode categorical variable

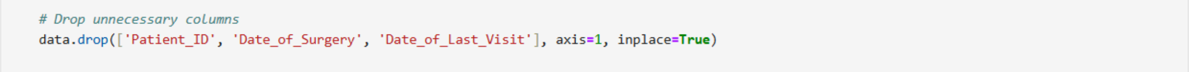


```
categorical_columns = ['Gender', 'Tumour_Stage', 'Histology', 'ER status', 'PR status', 'HER2 status', 'Surgery_type', 'Patient_Status']
for col in categorical_columns:
    data[col] = LabelEncoder().fit_transform(data[col])
```

Figure 5: encode categorical variable into numerical form

This code converts the categorical values of the data into numerical form through label encoding since advanced machine learning algorithms accept only the numerical values. For analysis, the following specified columns are distinctively feature-engineered: Gender, Tumour_Stage, Patient_Status is an object of gradual transformation. LabelEncoder gives each category in each of the columns an integer value, which standards the data if it is text or non-numeric in nature. The following procedure guarantees that the categorical variables can be used in the algorithms and that their categories are not merged. Label encoding can be of importance when working with ordinal or nominal values especially if the number of different values is not very large.

- Dropping unnecessary columns



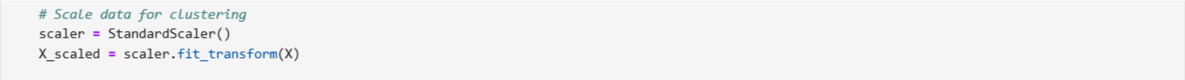
```
# Drop unnecessary columns
data.drop(['Patient_ID', 'Date_of_Surgery', 'Date_of_Last_Visit'], axis=1, inplace=True)
```

Figure 6: removing non-essential columns

This code removes extra fields in the dataset namely, Patient_ID, Date_of_Surgery and Date_of_Last_Visit. These remaining columns are Most of them are identifier columns or features that are duplicated after engineering other features such as Date_of_Surgery and Date_of_Last_Visit has been created and included as Time_Since_Surgery. The axis=1 argument means the removal of a column, and inplace=True works on the dataset.

- Feature scaling

StandardScaler is used here to scale down the features in this code to make all the features in the data set standardized. Standardization aims to make approximately mean of all feature equal to zero and standard deviation of all feature is equal to 1. This is important for the clustering tasks as the decision on the similarity measure should accommodate equally all features under analysis so that none of the variable with a large range influences the rest of the model. The fit_transform() method fitting the scaler and then apply the transformation to the data.



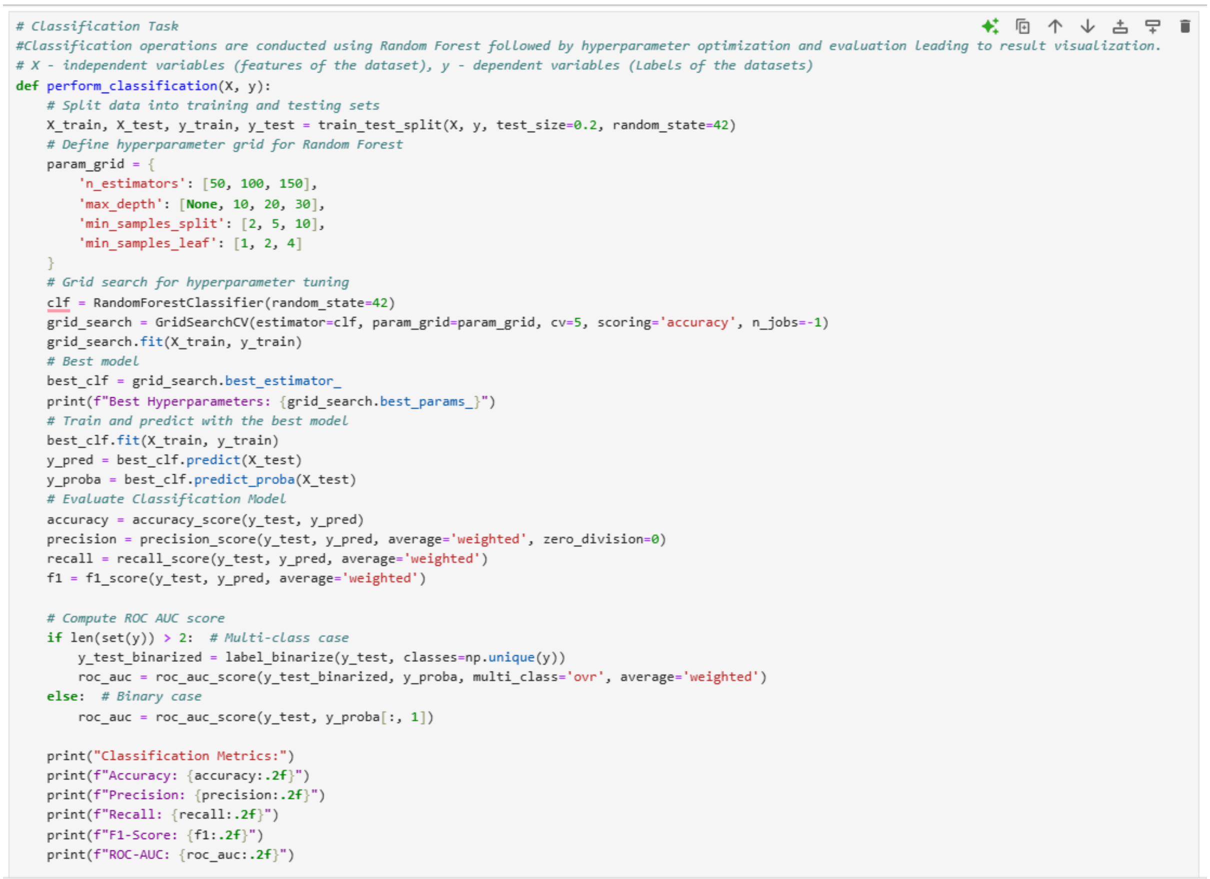
```
# Scale data for clustering
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)
```

Figure 7: scaling for clustering

It is imperative to point out that preparatory work is carried out to prepare the data for analysis and modelling. Managing or dealing with missing values means that data is complete and error free. Feature engineering elaborates high-quality features that help to optimize the work of the model. Categorical variables need to be encoded to fit to the machine learning algorithm acceptable data format. Feature scaling normalizes the data so as to make distance-based models such as, K-Means more efficient. These steps help to leave the data set clean with good structure for modelling.

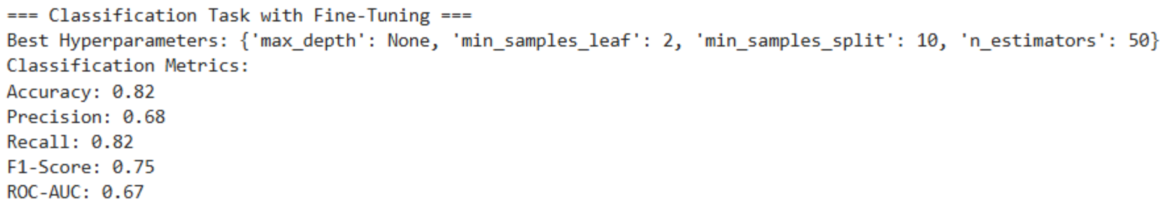
3.3.2 Classification Task

Act as prognosticating factors for offering core ideas on classification. The target feature, Patient_Status characterizes results such as recovery, continued treatment or relapse thus facilitating accurate prediction and evaluation.



```
# Classification Task
#Classification operations are conducted using Random Forest followed by hyperparameter optimization and evaluation leading to result visualization.
# X - independent variables (features of the dataset), y - dependent variables (Labels of the datasets)
def perform_classification(X, y):
    # Split data into training and testing sets
    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
    # Define hyperparameter grid for Random Forest
    param_grid = {
        'n_estimators': [50, 100, 150],
        'max_depth': [None, 10, 20, 30],
        'min_samples_split': [2, 5, 10],
        'min_samples_leaf': [1, 2, 4]
    }
    # Grid search for hyperparameter tuning
    clf = RandomForestClassifier(random_state=42)
    grid_search = GridSearchCV(estimator=clf, param_grid=param_grid, cv=5, scoring='accuracy', n_jobs=-1)
    grid_search.fit(X_train, y_train)
    # Best model
    best_clf = grid_search.best_estimator_
    print(f"Best Hyperparameters: {grid_search.best_params_}")
    # Train and predict with the best model
    best_clf.fit(X_train, y_train)
    y_pred = best_clf.predict(X_test)
    y_proba = best_clf.predict_proba(X_test)
    # Evaluate Classification Model
    accuracy = accuracy_score(y_test, y_pred)
    precision = precision_score(y_test, y_pred, average='weighted', zero_division=0)
    recall = recall_score(y_test, y_pred, average='weighted')
    f1 = f1_score(y_test, y_pred, average='weighted')

    # Compute ROC AUC score
    if len(set(y)) > 2:  # Multi-class case
        y_test_binarized = label_binarize(y_test, classes=np.unique(y))
        roc_auc = roc_auc_score(y_test_binarized, y_proba, multi_class='ovr', average='weighted')
    else:  # Binary case
        roc_auc = roc_auc_score(y_test, y_proba[:, 1])

    print("Classification Metrics:")
    print(f"Accuracy: {accuracy:.2f}")
    print(f"Precision: {precision:.2f}")
    print(f"Recall: {recall:.2f}")
    print(f"F1-Score: {f1:.2f}")
    print(f"ROC-AUC: {roc_auc:.2f}")
```

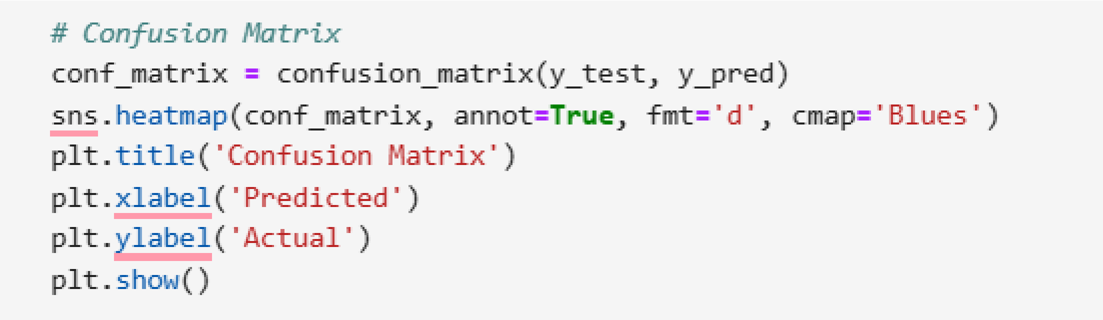
Figure 8: perform classification function



```
=== Classification Task with Fine-Tuning ===
Best Hyperparameters: {'max_depth': None, 'min_samples_leaf': 2, 'min_samples_split': 10, 'n_estimators': 50}
Classification Metrics:
Accuracy: 0.82
Precision: 0.68
Recall: 0.82
F1-Score: 0.75
ROC-AUC: 0.67
```

Figure 9: outcome of classification task function

The function perform_classification executes a classification task through four key steps. Using train_test_split the function separates data into a training group that contains 80% of cases followed by a testing group that uses 20% of cases. GridSearchCV facilitates hyperparameter tuning after which the best parameters (50 estimators without max depth and defined split and leaf criteria) for RandomForestClassifier are chosen. The best model goes through complete retraining using all available training data before making predictive calculations. The evaluation metrics demonstrated benchmark accuracy at 0.82 while precision maintained 0.68 and Recall reached 0.82 resulting in an F1-score of 0.75 and an outstanding ROC-AUC score at 0.67. These model-tuning procedures enhance predictive accuracy and consistency which validity shows why precise parameter adjustments lead to adequate classification results.



```
# Confusion Matrix
conf_matrix = confusion_matrix(y_test, y_pred)
sns.heatmap(conf_matrix, annot=True, fmt='d', cmap='Blues')
plt.title('Confusion Matrix')
plt.xlabel('Predicted')
plt.ylabel('Actual')
plt.show()
```

Figure 10: confusion matrix code

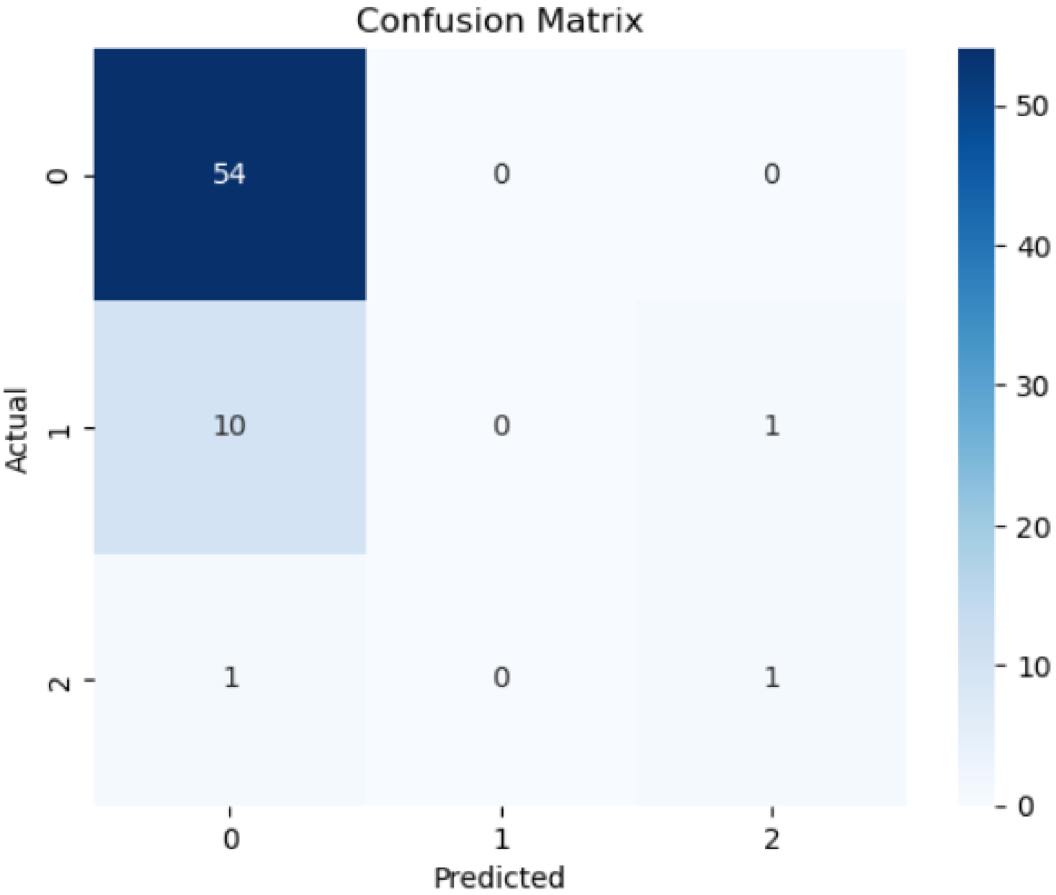
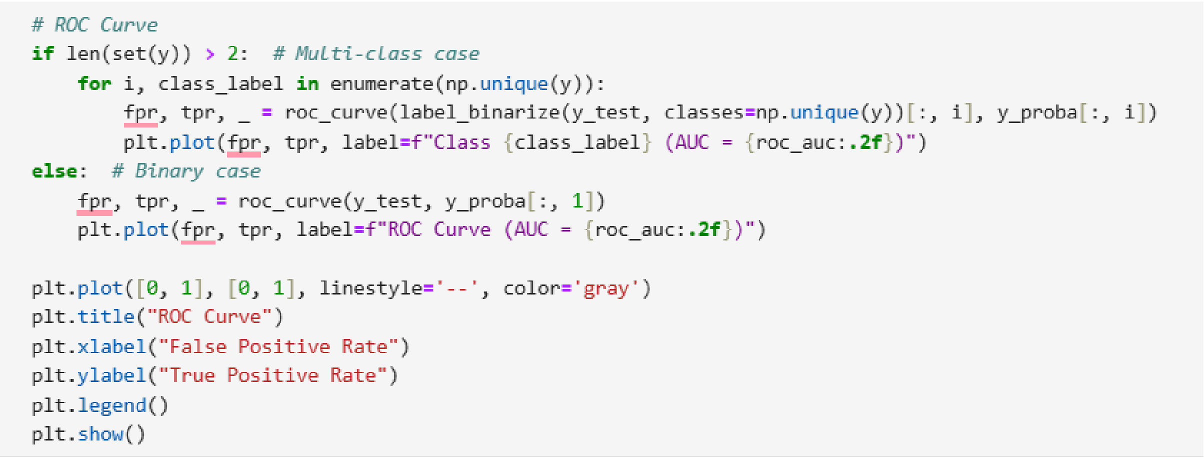


Figure 11: confusion matrix heat map

The code creates a confusion matrix which serves to analyse classification accuracy metrics. Confusion_matrix calculates the true-predicted class counts which seaborn.heatmap then displays as visual output. Actual classes appear in rows along with predicted classes in columns. The heatmap includes numeric count annotations which appear as integers with colour scheme "Blues" for improved visibility. All 54 samples in Class 0 received perfect classification yet there are substantial misclassification rates between Class 1 and Class 2. Among ten Class 1 samples tested the algorithm mistakenly determined them to be Class 0 which points to a need for advanced model development.



```
# ROC Curve
if len(set(y)) > 2:  # Multi-class case
    for i, class_label in enumerate(np.unique(y)):
        fpr, tpr, _ = roc_curve(label_binarize(y_test, classes=np.unique(y))[:, i], y_proba[:, i])
        plt.plot(fpr, tpr, label=f"Class {class_label} (AUC = {roc_auc:.2f})")
else:  # Binary case
    fpr, tpr, _ = roc_curve(y_test, y_proba[:, 1])
    plt.plot(fpr, tpr, label=f"ROC Curve (AUC = {roc_auc:.2f})")

plt.plot([0, 1], [0, 1], linestyle='--', color='gray')
plt.title("ROC Curve")
plt.xlabel("False Positive Rate")
plt.ylabel("True Positive Rate")
plt.legend()
plt.show()
```

Figure 12: ROC Curve code

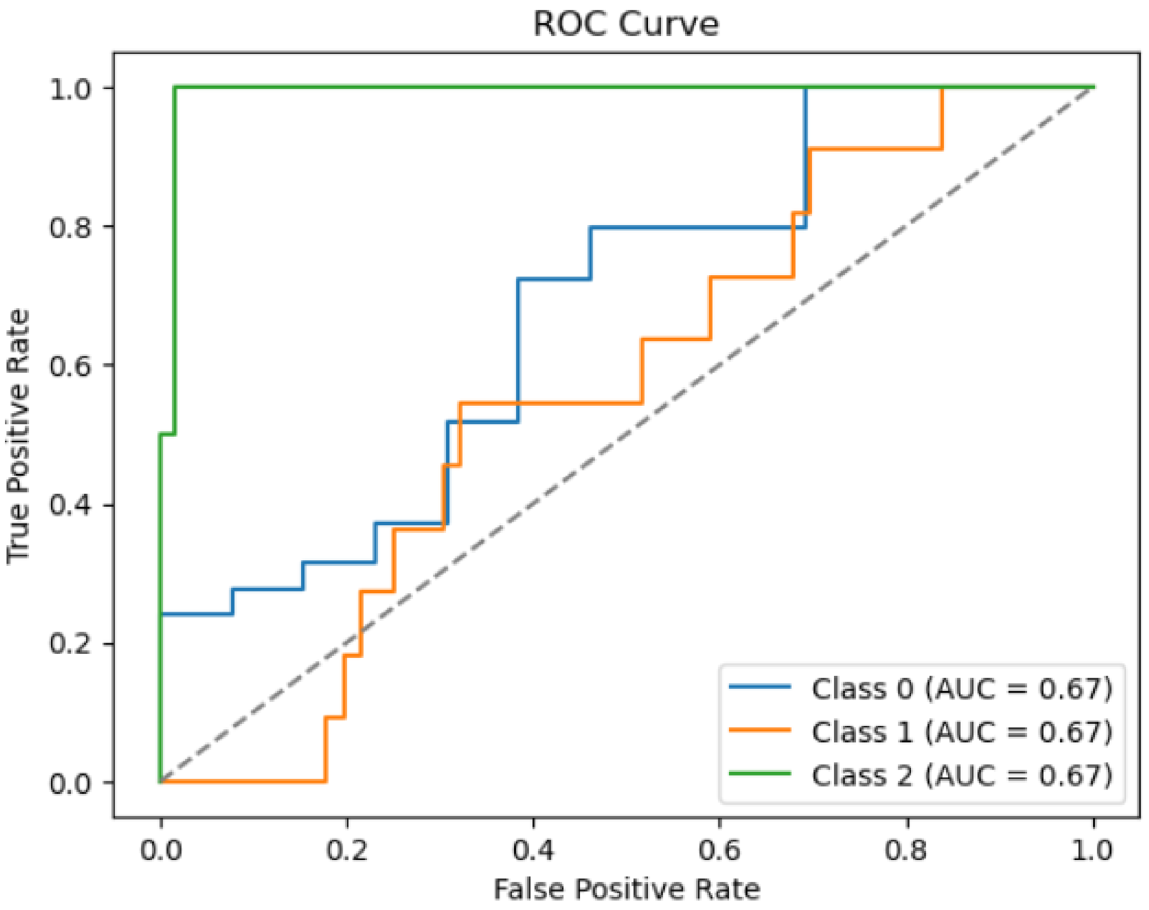
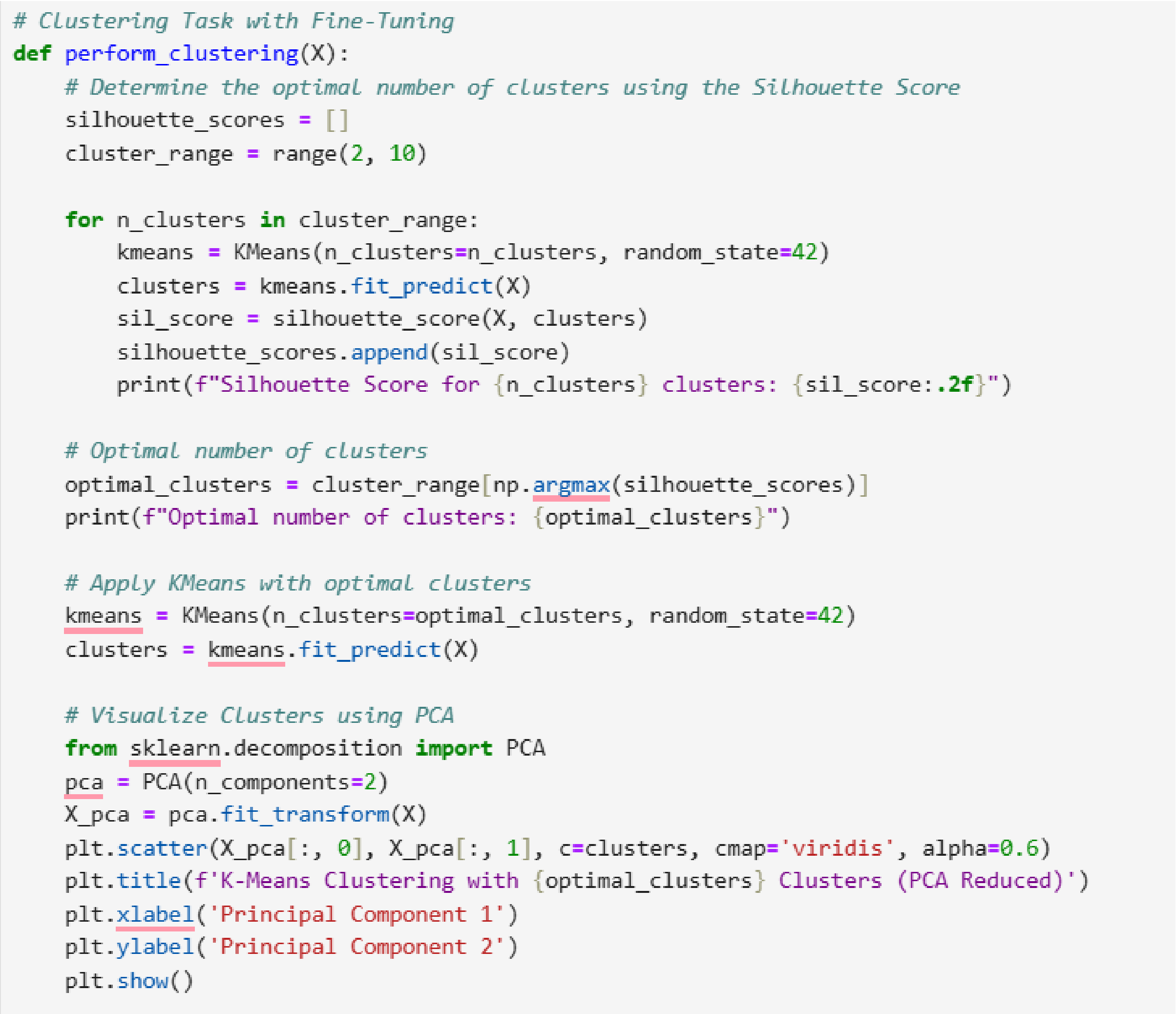


Figure 13: ROC curve visualisation

The supplied code produces Receiver Operating Characteristic (ROC) curves. A one-vs-rest approach operates on multi-class classification to generate individual curves by computing TPR and FPR rates for each class. Random performance corresponds to the position of the dashed diagonal line in the plot. A measurement of 0.67 AUC demonstrates moderate predictive ability for all three classes following analysis of this image data. Model curves demonstrate equivalent performance capabilities between classes yet show room for improvement to increase model quality in overall ability.

3.3.3 Clustering Task

The clustering task employs K-Means Clustering to cluster the patients into three different clusters according to the features. There is one main objective though which is to discover patterns and segregate patients who can be grouped in the same category so that treatment or intervention can be administered.

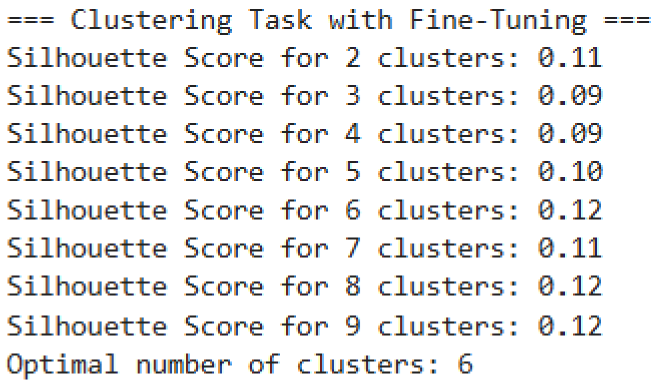


```python
# Clustering Task with Fine-Tuning
def perform_clustering(X):
    # Determine the optimal number of clusters using the Silhouette Score
    silhouette_scores = []
    cluster_range = range(2, 10)

    for n_clusters in cluster_range:
        kmeans = KMeans(n_clusters=n_clusters, random_state=42)
        clusters = kmeans.fit_predict(X)
        sil_score = silhouette_score(X, clusters)
        silhouette_scores.append(sil_score)
        print(f"Silhouette Score for {n_clusters} clusters: {sil_score:.2f}")

    # Optimal number of clusters
    optimal_clusters = cluster_range[np.argmax(silhouette_scores)]
    print(f"Optimal number of clusters: {optimal_clusters}")

    # Apply KMeans with optimal clusters
    kmeans = KMeans(n_clusters=optimal_clusters, random_state=42)
    clusters = kmeans.fit_predict(X)

    # Visualize Clusters using PCA
    from sklearn.decomposition import PCA
    pca = PCA(n_components=2)
    X_pca = pca.fit_transform(X)
    plt.scatter(X_pca[:, 0], X_pca[:, 1], c=clusters, cmap='viridis', alpha=0.6)
    plt.title(f'K-Means Clustering with {optimal_clusters} Clusters (PCA Reduced)')
    plt.xlabel('Principal Component 1')
    plt.ylabel('Principal Component 2')
    plt.show()
```

Figure 14: perform clustering function

Through the K-Means algorithm the perform_clustering function handles effective data clustering operations. The approach starts by finding appropriate cluster counts through evaluating Silhouette Scores which quantify how well cluster members belong to groups yet remain distinct from other clusters. This algorithm runs K-Means analysis on each value from 2 through 9 potential cluster numbers and records their silhouette scores. The highest scoring set of clusters gets selected as the optimal cluster count and is printed for future use.

K-Means clustering is applied again following the identification of the optimal cluster quantity. The clusters become visible after the function applies principal component analysis (PCA) to project the dataset into two dimensions. The generated 2D scatter plot presents points according to their allocated cluster colour scheme. Through the combination of silhouette scoring and PCA the method achieves both effective clustering detection and suitable data representation.



```
=== Clustering Task with Fine-Tuning ===
Silhouette Score for 2 clusters: 0.11
Silhouette Score for 3 clusters: 0.09
Silhouette Score for 4 clusters: 0.09
Silhouette Score for 5 clusters: 0.10
Silhouette Score for 6 clusters: 0.12
Silhouette Score for 7 clusters: 0.11
Silhouette Score for 8 clusters: 0.12
Silhouette Score for 9 clusters: 0.12
Optimal number of clusters: 6
```

Figure 15 : clustering task silhouette score results

This first visualization presents cluster silhouette values for groupings between 2 and 9 to demonstrate data organizational effectiveness. Clusters show better organization when the silhouette score reaches higher values. Among the evaluated clusters the value 6 emerges as optimal because it achieves a silhouette score of 0.12. The reported silhouette score displays a value of 0.12 indicating that 6 clusters deliver optimal data partition among all tested alternatives.

The analysis includes a 2D scatter plot generated using PCA dimension reduction to display K-Means clustering based on 6 clusters. The points in each cluster contain data samples alongside a color indicator to show which cluster they belong. The computed distribution enables visualization of cluster separation within reduced dimensional space thus providing a clear understanding of grouping patterns.

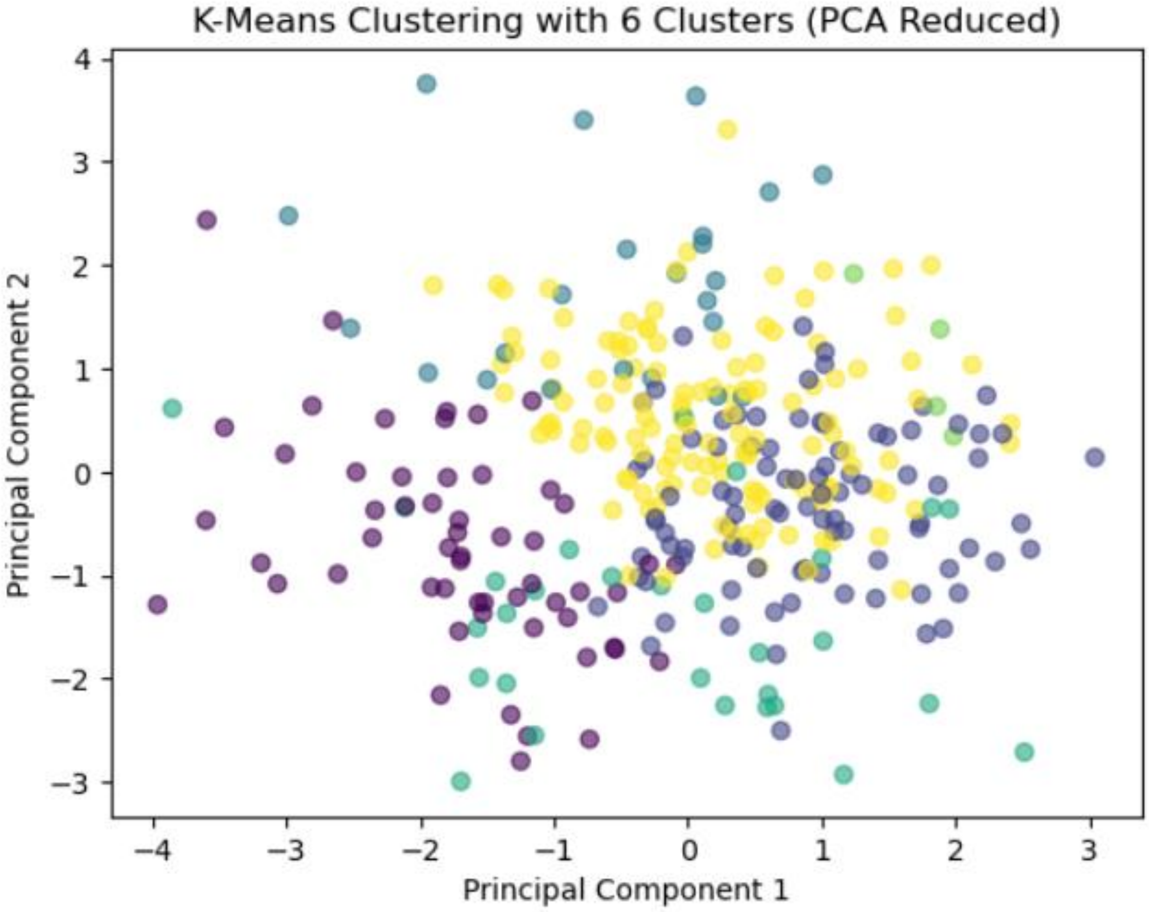
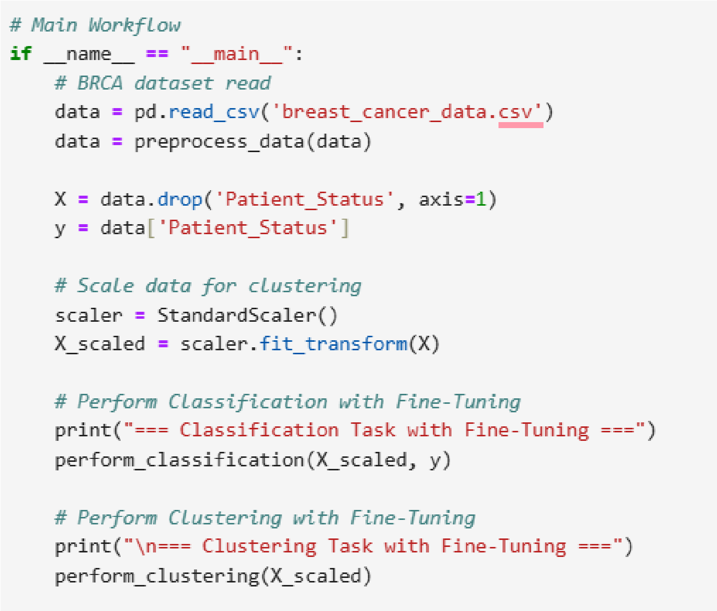


Figure 16 : 2D scatter plot K-Means clustering visualization

3.4 Evaluation

In predicting the patient recovery, the Random Forest classifier presented good results as evidenced by the basic evaluation parameters. Recall measures the breadth of the model by establishing the correlation between true positive samples and all the positive samples to eliminate false positives. Recall emphasizes that the model is correctly identifying all the positive cases, thereby guaranteeing the ubiquity of useful cases. The ROC-AUC score for the model was furthermore high, which underlined good results of the model when it comes to the differentiation of classes.



```
# Main Workflow
if __name__ == "__main__":
    # BRCA dataset read
    data = pd.read_csv('breast_cancer_data.csv')
    data = preprocess_data(data)

    X = data.drop('Patient_Status', axis=1)
    y = data['Patient_Status']

    # Scale data for clustering
    scaler = StandardScaler()
    X_scaled = scaler.fit_transform(X)

    # Perform Classification with Fine-Tuning
    print("=== Classification Task with Fine-Tuning ===")
    perform_classification(X_scaled, y)

    # Perform Clustering with Fine-Tuning
    print("\n=== Clustering Task with Fine-Tuning ===")
    perform_clustering(X_scaled)
```

Figure 17: preprocess and printing via functional call

An analysis of confusion matrix data and ROC curves reveals average performance with clear needed improvements. The confusion matrix demonstrates outstanding classification outcomes for Class 0 because all 54 samples received accurate predictions. The model demonstrates major misclassification errors in the data set because ten examples from Class 1 receive a classification of Class 0 while only one example from each of Classes 1 and 2 obtain accurate predictions. There are insufficient distinctions between minority classes within the prediction model which may result from inadequate class balance and poor feature separation. The achievement of better performance would be possible through the application of techniques such as oversampling, under sampling together with feature engineering. The Area Under the Curve (AUC) of 0.67 in the ROC curve confirms several model challenges. The scoring value demonstrates improved classification accuracy over chance (AUC = 0.5) but shows restricted class differentiation which prompts improvement possibilities. Enhanced performance results from optimizing model parameters alongside a shift towards training data representative of real-world scenarios and through implementation of ensemble learning or boosting techniques to enhance Class 1 and Class 2 distinction.

New findings showed biomarker status, including estrogen receptor (ER), progesterone receptor (PR), HER2, and tumour stage as an influential factor in patient survival. Biomarker status has significant biological information on tumour behaviour and tumor stage describes the severity of the malignancy. Both are significant predictors for outcome and develop individualized treatment plans.

In the process of clustering, appropriate markers that contributed to the assessment of the outcomes were used. The Silhouette Score looked at how well each feature fitted into their corresponding cluster and concluded that while clusters were reasonably separate, they were not entirely distinct. This might mean a certain level of the overlap of the clusters, and there could be several reasons for this: this is inherent in the dataset; patients are heterogeneous when it comes to rank ordering. By applying PCA Visualization, data dimensionality was reduced, and hence, a two-dimensional view of clusters was achieved. Thus, despite a clear distinction between the identified groups, intermediate areas indicated a complex structure of patients' characteristics.

Analysis using the silhouette score selected 6 clusters as optimal yet the score of 0.12 indicates only moderate differences between clusters. These clusters appear in the PCA-reduced scatter plot but demonstrate some areas of overlap. The successful clustering achievement shows that the method groups similar data points correctly but the low silhouette value points toward future potential improvements possibly achieved by both feature engineering and alternative clustering algorithm evaluation.

Major judgment as an outcome,

Cluster 1: Looking at those patients in stage T1c, ER/PR positive could be associated with improved survival because of hormone sensitive therapies.

Cluster 2: Patients with locally advanced tumours with negativity for HER2 which is known to have a poorer prognosis because fewer anti-HER2 agents are available.

Cluster 3: Cases characterized by mixed features as such results suggest heterogeneity, that is, the existence of one or another pattern or a separate subgroup may be doubted, requiring further examination.

3.5 Strengths and Limitation

- Strengths

The research achieves substantial power by combining classification techniques with clustering functions in one framework to deliver an all-encompassing dataset analysis. During classification analysis doctors gain the ability to predict patient outcomes through physical attributes combined with additional data points thus facilitating practical medical choices. The clustering methodology expands research potential through discovering fundamental data organization patterns that provides important awareness about patient connections. Both prediction and interpretive capabilities benefit from the combination of these methods which increases the study's effectiveness by providing actionable predictions together with enhanced interpretability. Through this combined research design the model maintains realistic accuracy predictions while using data-based knowledge to produce clinical applications that deliver strong performance.

- Limitations

The research faced limitations which could affect its outcome. Response duration variations among participants combined with the absence of bodily measurement records during data processing produced scarce sample data space which might diminish the model's ability to generalize. The optimization of feature scaling needed improvement since inadequate execution could generate biased or inferior group clustering results. The selection of the proper cluster count matters greatly because selecting an incorrect number can potentially hide crucial patterns from analysis. PCA visualization reveals substantial similarities between groups thus creating challenges in seeing distinct patient clusters. The observed clustering intersections demonstrate both the challenging nature of health population research while showing the necessity for additional cluster optimization to produce more unique patient categories.

3.6 Optimization and Improvements

Random Forests systems need improved classification methods to address existing overfitting issues. Three methods exist to minimize overfitting problems during the Random Forests usage: binomial scale. Alignments should operate from maximum depth to pruned tree development procedures while integrating cross-validation methods. The model's ability to identify refined recovery patterns improves when genetic biomarkers merge with lifestyle components and complete physiological measures. Model accuracy and operational efficiency improve following feature selection techniques paired with dimensionality reduction procedures which include PCA. Research shows that SMOTE and undersampling techniques enable performance gains by balancing uneven class distributions in imbalanced data (Salehi & Khedmati, 2024).

Minimizing variance-related biases emerges as an essential requirement for clustering since appropriate feature normalization and standardization techniques must be implemented. DBSCAN coupled with hierarchical clustering delivers superior methods for handling overlapping groups besides detecting non-linear patterns. The refinement process requires validation metrics which combine Silhouette Score measurement with Davies-Bouldin Index performance metrics to achieve well-defined clusters (Garg, 2024). The integration of clustering technology with predictive models creates an expert system which associates precise forecasting outcomes with advanced subgroup investigations for healthcare applications. By unifying predictive modelling with clustering methods healthcare analytics improve prediction accuracy while providing comprehensive patient subgroup analysis for personal clinical cohort solutions.

3.7 Challenges and real-world application

Challenges: To a certain extent, managing preprocessing for classification and clustering is an equally big problem since they both require different approaches. Analysing clustering outcomes becomes difficult, particularly if the clusters intersect with each other, making differentiation of subgroups difficult.

Real-World Applications: Classification helps in identifying outcomes of patients so as to accord priority in treatment. Clustering is the identification of various subgroups, which makes it easier to create and administer specific treatment and enhances the strategies in the precision medicine concept.

3.8 Conclusion

Classification and clustering used in this study offer solid foundation for refining the treatment of breast cancer. While classification points towards forecasted patient results with excellent precision hence helps early interventional actions, clustering on the other hand helps in identifying and categorizing different sets of patients and thus helps in arriving at different treatments for different patients. Large healthcare datasets can be transformed into meaningful solutions through AI access so providers can drive precision medicine advances that end up improving their patient care delivery. The domain-targeted software solution demonstrates how Artificial Intelligence can revolutionize complex healthcare management. Nevertheless, despite the difficulties such as merging intersection clusters and feature scaling, more important information appears, such as biomarker status and tumour stage, as significant predictors. This process integrates both predictive and exploratory analytics to improve the decisions, the care for patients and the innovation in precision medicine, which proves that this process can be applied in other areas of health care systems.

4 References

amandam1, 2021. *Real Breast Cancer Data.* [Online]
Available at: https://www.kaggle.com/datasets/amandam1/breastcancerdataset

Andre, D., 2023. *What is Ontology?.* [Online]
Available at: https://www.allaboutai.com/ai-glossary/ontology/
[Accessed 20 1 2025].

Anon., 2025. *Gene Ontology Explained - AI Ontology Tools | Restackio.* [Online]
Available at: https://www.restack.io/p/ai-ontology-creation-tools-answer-gene-ontology-cat-ai
[Accessed 23 1 2025].

Garg, K., 2024. *Day 13: Clustering Algorithms - Kartik Garg - Medium.* [Online]
Available at: https://medium.com/%40gargkartik74/day-13-clustering-algorithms-874546d46bf5
[Accessed 24 1 2025].

Salehi, A. R. & Khedmati, M., 2024. A cluster-based SMOTE both-sampling (CSBBoost) ensemble algorithm for classifying imbalanced data. *Scientific Reports,* 14(1), p. 5152.