DEGREE PROGRAMMES

INDIVIDUAL ASSIGNMENT_40%

FEBRUARY 2025 SEMESTER

DATE:12th March 2025, 4.00pm-5.00pm

MODULE NAME	**: DATA ANALYTICS AND MACHINE LEARNING**
MODULE CODE	**: ITS69304**
EXAM DURATION	**: 45 MINUTES**

This paper consists of THREE (3) printed pages, inclusive of this page.

Candidate Number : 0364483

Name : Hng Qi Yean

Signature : *Hng*

Instruction to Candidates:

1. Answer **ALL** questions
2. The breakdown of exam questions by Module Learning Outcome(s) and its associate weightage is as follows:

MLO	Section(s)/ Question(s)	Marks
MLO3	ALL QUESTIONS	/ 40
	TOTAL	/ 40

MLO3:Design and evaluate applications of data analytics in various industries

3. Non-programmable electronic calculators may be used.
4. This is an OPEN BOOK assessment.
5. Severe disciplinary action will be taken against those caught violating examination rules.

FEB 2025 SEMESTER INDIVIDUAL ASSIGNMENT	40%

Classification- Breast Cancer using Random Forest

Write code with python programming and machine learning algorithms for the given instructions:

1) Import necessary libraries. Explain (4marks)

Code:

Question 1 - Import libraries

```
[1]  # Import essential libraries
     import pandas as pd  # For data manipulation
     import numpy as np  # For numerical operations
     import matplotlib.pyplot as plt  # For visualization
     import seaborn as sns  # For better visualization
     from sklearn.model_selection import train_test_split  # To split data
     from sklearn.ensemble import RandomForestClassifier  # ML algorithm
     from sklearn.preprocessing import LabelEncoder, MinMaxScaler  # Encoding & Scaling
     from sklearn.metrics import accuracy_score, classification_report, confusion_matrix  # Evaluation Metrics
```

Explanation:

1. pandas: Used for handling datasets.
2. numpy: Helps with numerical computations.
3. matplotlib.pyplot & seaborn: Used for visualizing the data.
4. sklearn.model_selection.train_test_split: Splits data into training & testing sets.
5. sklearn.ensemble.RandomForestClassifier: Implements Random Forest Classifier model.
6. sklearn.preprocessing: Handles encoding categorical features and scaling numerical features.
7. sklearn.metrics: Evaluates model performance.

FEB 2025 SEMESTER INDIVIDUAL ASSIGNMENT	40%

2) Import dataset. Explain (4marks)

Code:

Question 2 - Import Dataset

```
# Load dataset
df = pd.read_csv("/content/breast-cancer.csv")  # Change path if needed

# Display first 5 rows
print(df.head())
```

Output:

```
         id diagnosis  radius_mean  texture_mean  perimeter_mean  area_mean  \
0    842302         M        17.99         10.38          122.80     1001.0
1    842517         M        20.57         17.77          132.90     1326.0
2  84300903         M        19.69         21.25          130.00     1203.0
3  84348301         M        11.42         20.38           77.58      386.1
4  84358402         M        20.29         14.34          135.10     1297.0

   smoothness_mean  compactness_mean  concavity_mean  concave points_mean  \
0          0.11840           0.27760          0.3001              0.14710
1          0.08474           0.07864          0.0869              0.07017
2          0.10960           0.15990          0.1974              0.12790
3          0.14250           0.28390          0.2414              0.10520
4          0.10030           0.13280          0.1980              0.10430

   ...  radius_worst  texture_worst  perimeter_worst  area_worst  \
0  ...         25.38          17.33           184.60      2019.0
1  ...         24.99          23.41           158.80      1956.0
2  ...         23.57          25.53           152.50      1709.0
3  ...         14.91          26.50            98.87       567.7
4  ...         22.54          16.67           152.20      1575.0

   smoothness_worst  compactness_worst  concavity_worst  concave points_worst  \
0            0.1622             0.6656           0.7119                0.2654
1            0.1238             0.1866           0.2416                0.1860
2            0.1444             0.4245           0.4504                0.2430
3            0.2098             0.8663           0.6869                0.2575
4            0.1374             0.2050           0.4000                0.1625

   symmetry_worst  fractal_dimension_worst
0          0.4601                  0.11890
1          0.2750                  0.08902
2          0.3613                  0.08758
3          0.6638                  0.17300
4          0.2364                  0.07678

[5 rows x 32 columns]
```

Explanation:

1. pd.read_csv(): Reads the Breast Cancer dataset.
2. df.head(): Displays the first 5 rows to understand the data.

FEB 2025 SEMESTER INDIVIDUAL ASSIGNMENT	40%

3) Determine your X and Y variables. Write the code. Assign X variables to all except diagnosis feature- # diagnosis labels (0 for malignant, 1 for benign) (4marks)

Code:

Question 3 - Determine X and Y variables

```
[3]  # Identify target variable
     y = df['diagnosis']  # Target: 'diagnosis' (0 = Malignant, 1 = Benign)

     # Encode the target variable (if it's categorical)
     le = LabelEncoder()
     y = le.fit_transform(y)

     # Identify feature variables (drop 'diagnosis' column)
     X = df.drop(columns=['diagnosis'])

     # Feature scaling using MinMaxScaler
     scaler = MinMaxScaler()
     X = pd.DataFrame(scaler.fit_transform(X), columns=X.columns)
```

Explanation:

1. y = df['diagnosis']: Target variable (Malignant = 0, Benign = 1).
2. LabelEncoder(): Converts categorical target to numerical (if necessary).
3. X = df.drop(columns=['diagnosis']): Assigns all other columns as features.
4. MinMaxScaler(): Scales feature values between 0 and 1.

FEB 2025 SEMESTER INDIVIDUAL ASSIGNMENT	40%

4) Print some information about the dataset and the output looks like below:

```
Breast Cancer dataset information:
Number of samples: *****
Number of features: *****
Number of malignant tumors (class 0): *****
Number of benign tumors (class 1): *****
```

(10marks)

Code:

Question 4

```
# Print dataset information
print("\nBreast Cancer Dataset Information:")
print(f"Number of samples: {df.shape[0]}")
print(f"Number of features: {df.shape[1] - 1}")  # Exclude the target column
print(f"Number of malignant tumors (class 0): {(y == 0).sum()}")
print(f"Number of benign tumors (class 1): {(y == 1).sum()}")
```

Output:

```
Breast Cancer Dataset Information:
Number of samples: 569
Number of features: 31
Number of malignant tumors (class 0): 357
Number of benign tumors (class 1): 212
```

Explanation:

1. df.shape[0]: Total number of rows (samples).
2. df.shape[1] - 1: Number of features (excluding target column).
3. (y == 0).sum(): Counts malignant cases (class 0).
4. (y == 1).sum(): Counts benign cases (class 1).

FEB 2025 SEMESTER INDIVIDUAL ASSIGNMENT	40%

5) Fit the dataset into Random Forest Classifier- show output (4marks)

Code:

Question 5 - Fit the dataset

```
# Split dataset into training (80%) and testing (20%)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Train Random Forest model
rf_model = RandomForestClassifier(n_estimators=100, random_state=42)
rf_model.fit(X_train, y_train)

# Show training output
print("\nModel Training Completed!")
```

Output:

```
Model Training Completed!
```

Explanation:

1. train_test_split(): Splits dataset into 80% training, 20% testing.
2. RandomForestClassifier(): Creates Random Forest model with 100 trees.
3. rf_model.fit(): Trains the model using the training dataset.

FEB 2025 SEMESTER INDIVIDUAL ASSIGNMENT	40%

6) Print accuracy value as per below:

```
Accuracy on the training subset:(:.3f) *****
Accuracy on the testing subset:(:.3f) *****
```

(4marks)

Code:

Question 6

```
# Predictions
y_train_pred = rf_model.predict(X_train)
y_test_pred = rf_model.predict(X_test)

# Calculate Accuracy
train_acc = accuracy_score(y_train, y_train_pred)
test_acc = accuracy_score(y_test, y_test_pred)

# Print Accuracy
print(f"\nAccuracy on the training subset: {train_acc:.3f}")
print(f"Accuracy on the testing subset: {test_acc:.3f}")
```

Output:

```
Accuracy on the training subset: 1.000
Accuracy on the testing subset: 0.965
```

Explanation:

1. rf_model.predict(): Predicts outcomes on training & testing sets.
2. accuracy_score(): Computes accuracy for training & testing.
3. Prints formatted accuracy with :.3f (3 decimal places).

FEB 2025 SEMESTER INDIVIDUAL ASSIGNMENT	40%

7) Give your insight about the result on the accuracy. What do you think? (8marks)

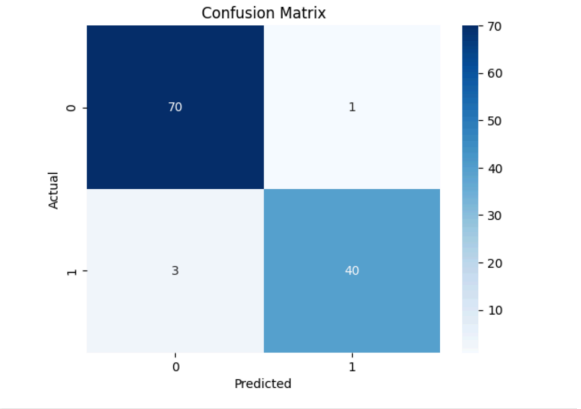
Code:

Question 7 - Insights

```
# Print Classification Report
print("\nClassification Report:\n", classification_report(y_test, y_test_pred))

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

Output:

```
Classification Report:
              precision    recall  f1-score   support

           0       0.96      0.99      0.97        71
           1       0.98      0.93      0.95        43

    accuracy                           0.96       114
   macro avg       0.97      0.96      0.96       114
weighted avg       0.97      0.96      0.96       114
```



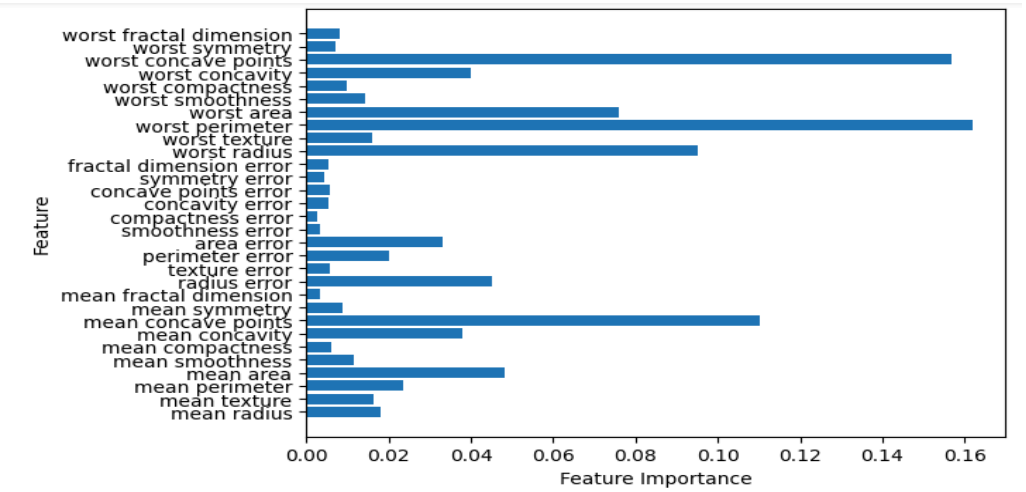
Explanation:

The accuracy results indicate that the Random Forest model has achieved a perfect training accuracy of 1.000 and a test accuracy of 0.965. While this high performance suggests that the model is highly effective at classification, the perfect score on the training data raises concerns about overfitting. Overfitting occurs when the model memorizes the training data rather than learning generalizable patterns, which can reduce its effectiveness on unseen data.

The confusion matrix shows that the model correctly classified most samples, with only one

false positive and three false negatives. This indicates that the model is slightly better at detecting malignant cases (class 0) compared to benign cases (class 1). In a medical context, false negatives (misclassifying malignant tumors as benign) are more concerning because they can lead to delayed treatment. To address this, adjustments such as tuning hyperparameters, reducing tree depth, or using different classification thresholds could be beneficial.

8) Plot a bar graph as per below: (2marks)



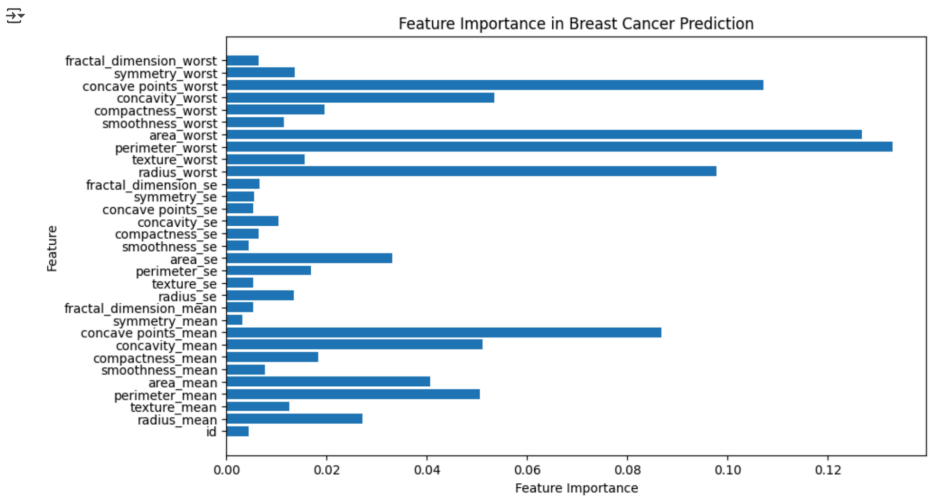
Code:

Question 8 - Plot a bar graph

```
# Get the number of features
n_features = X.shape[1]

# Plot feature importance
plt.figure(figsize=(10, 6))
plt.barh(range(n_features), rf_model.feature_importances_, align='center')
plt.yticks(np.arange(n_features), X.columns)  # Use column names from DataFrame
plt.xlabel('Feature Importance')
plt.ylabel('Feature')
plt.title('Feature Importance in Breast Cancer Prediction')
plt.show()
```

Output:

FEB 2025 SEMESTER INDIVIDUAL ASSIGNMENT	40%



Explanation:

1. X.shape[1] – Retrieves the number of features (columns) in the dataset.
2. plt.figure(figsize=(10, 6)) – Creates a new figure for the plot with a specified size.
3. plt.barh(range(n_features), rf_model.feature_importances_, align='center') – Plots a horizontal bar chart showing feature importance scores from the Random Forest model.
4. plt.yticks(np.arange(n_features), X.columns) – Labels the y-axis with feature names from the dataset.
5. plt.xlabel('Feature Importance') – Sets the label for the x-axis.
6. plt.ylabel('Feature') – Sets the label for the y-axis.
7. plt.title('Feature Importance in Breast Cancer Prediction') – Adds a title to the plot.
8. plt.show() – Displays the generated feature importance plot.

FEB 2025 SEMESTER INDIVIDUAL ASSIGNMENT	40%

Submission Requirements

1. Font type : Times New Roman
2. Font size : 12
3. Line spacing : 1.5
4. Alignment : Justify Text
5. **Document type : .pdf, .ipynb (both files must be submitted)**
6. Number of pages : 5 – 20 pages (do not exceed the page limit)
7. Your full report should consist of the following:
    a) Cover page (Name, ID, Date, Signature, Score)
    b) Appendixes (line spacing = 1.0)
        - List of references (APA format)
        - Python script
8. Start each question with number of the questions.
9. All figures and tables are labelled properly.
10. File naming conventions: StudentName_ID

Notes:

- Student is not allowed to transcribe directly (copy and paste) any material from another source into their submission. It is an open book assignment- you can refer to notes and websites as references only.
- Show the relevant Python code, the program outputs and explanation to each answer.

FEB 2025 SEMESTER INDIVIDUAL ASSIGNMENT	40%

- All process/functions must be clearly explained.
- Include in-text citation to support your answers and add the list of references at the end of your report (APA format). The list of references is to be alphabetized by the first author's last name, or (if no author is listed) the organization or title.