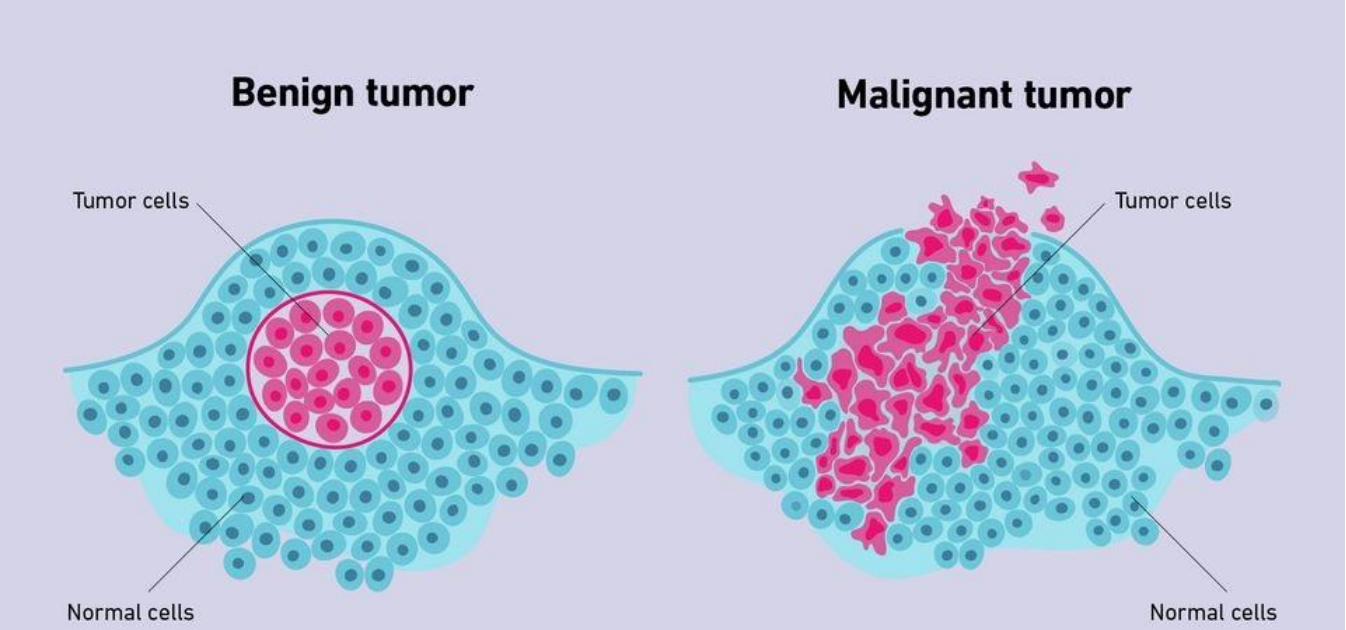
Module Title: Machine Learning and Autonomous Systems
Module Code: CS4S771
Module Leader/Tutor: Carl Jones
Assignment Title: Medical Image Classification

Student Name: Salma Javid
Student ID: 30107961

Table of Contents

I. Abstract

Breast cancer, a prevalent health concern globally, demands advanced diagnostic tools to help in the early detection and classification of tissue abnormalities. This assignment delves into the realm of machine learning for the classification of breast tissue mass images derived from fine needle aspirations. The dataset encompasses a diverse array of features characterizing cell nuclei properties, offering insights into the intricacies of tissue structure. With a focus on classifying masses as either malignant or benign, our endeavor involves implementing a systematic approach encompassing data preprocessing, feature selection, model training, and rigorous evaluation.



The Objectives: The significance of this work lies in its potential to contribute to the development of a robust and accurate system for breast cancer classification. Accurate classification is pivotal for timely intervention, thereby enhancing patient outcomes and reducing the impact of this prevalent and potentially life-threatening condition. In this context, we aim to advance our understanding of the efficacy of machine learning in breast cancer classification and contribute to the ongoing efforts in the intersection of healthcare and technology.

Worldwide, female breast cancer has now surpassed lung cancer as the most commonly diagnosed cancer. Also, the second leading cause of death among women is breast cancer, directly after lung cancer.

Tests/procedures for breast cancer diagnosis:	**Main Treating for Breast Cancer**:
● An oral examination ● Mammographic screening [X-ray] ● Biopsy / Fine Needle Aspiration ● Computerized Tomography (CT) or Ultrasound Scans	● Surgery ● Radiotherapy ● Chemotherapy ● Hormone therapy ● Targeted therapy

— Global Breast Cancer Statistics

- Breast cancer occurs in every country in the world. It is now the world's most prevalent cancer.
- About 1 in 7 women are diagnosed with breast cancer during their lifetime.
- In 2020, 11.7% of all cancers diagnosed were female breast cancer. There were 2.26 million women diagnosed with breast cancer and 685,000 deaths.
- There were 7.8 million women alive who were diagnosed with breast cancer.
- Approximately 0.5–1% of breast cancers occur in men. There were 25,100 men diagnosed with breast cancer and 12,100 deaths.

— Scope of the problem

Breast cancer occurs in every country of the world in women at any age after puberty but with increasing rates in later life. It is a disease in which abnormal breast cells grow out of control and form tumours. If left unchecked, the tumours can spread throughout the body and become fatal.

— Breast Cancer Mortality in the UK

- Breast cancer is the 4th most common cause of cancer death.
- 2nd most common cause of any death in women
- There are around 56,000 new cases of breast cancer every year: that's over 150 cases every day
- Around 11,400 women and 85 men die from breast cancer every year. That's 32 deaths every day.
- 48% of deaths from breast cancer are in those aged 75 and over
- Since the mid-1980s, breast cancer mortality rates have decreased by 45%

— Importance of Early Detection

Breast cancer is a life-threatening disease, and early detection can certainly reduce the rate of mortality. An analysis of the most recent data has shown that the survival rate is 88% after 5 years of diagnosis and 80% after 10 years of Diagnosis. If cancer is detected early, it can be treated before it spreads to other parts of the body. Early diagnosis can also greatly increase the rate of recovery, thereby increasing the chances of survival.

— Importance of Correct Classification

Detection of breast cancer is the preliminary phase in cancer diagnosis. So, classifiers with higher accuracy are always desired. Accurate classification can prevent patients from undergoing unnecessary treatments. Thus, the correct diagnosis and classification of patients into malignant or benign groups is the subject of much research.

II. Introduction

This assignment attempts to address the classification of breast tissue mass images using the different machine-learning techniques available. The dataset comprises features (real value Mean, Standard error, and "worst" or largest value) derived from digitized images of fine needle aspirations of a breast tissue mass. The objective is to develop a system for accurately distinguishing between malignant (M) and benign (B) cases.

The implementation involves data preprocessing, feature selection, model training, and evaluation using various classifying algorithms to find the best one. Design considerations include the choice of algorithm, hyperparameter tuning, and ethical considerations in handling medical data. The results are analyzed using standard metrics. The report critically discusses the design, implementation, training methodology, results, and potential improvements, providing insights into the efficacy of the developed system for breast tissue mass classification.

— Understanding Fine Needle Aspiration

Fine needle aspiration (FNA) is a procedure used to obtain a small amount of breast tissue or fluid or a sample of cells from a suspicious area with a thin, hollow needle (21 to 25 gauge) which will be examined under a microscope by a pathologist. The results help determine whether it’s a cyst, an infection, a benign tumor, or cancer. This is how doctors find out whether an abnormal area or lump (tumour) is cancerous (malignant) or non-cancerous (benign).

— Discussing the Database

The dataset used is publicly available and was created by Dr. William H. Wolberg, physician at the University Of Wisconsin Hospital at Madison, Wisconsin, USA. To create the dataset Dr. Wolberg used fluid samples, taken from patients with solid breast masses and an easy-to-use graphical computer program called Xcyt. The program uses a curve-fitting algorithm, to compute ten features from each one of the cells in the sample, than it calculates the mean value, extreme value and standard error of each feature for the image, returning a 30 real-valued vector

The details of the attributes found in WDBC dataset

ID number, Diagnosis (M = malignant, B = benign) and ten real-valued features are computed for each cell nucleus: Radius, Texture, Perimeter, Area, Smoothness, Compactness, Concavity, Concave points, Symmetry and Fractal dimension. These features are computed from a digitized image of a breast mass's fine needle aspirations (FNA). They describe characteristics of the cell nuclei present in the image *(Wolberg,William, Mangasarian,Olvi, Street,Nick, and Street,W.. (1995). Breast Cancer Wisconsin (Diagnostic). UCI Machine Learning Repository. https://doi.org/10.24432/C5DW2B)*.

1. Preliminary Setup and Dataset Overview

1.1: Importing Required Libraries, Packages and Modules

```
#First, let's import the libraries required for our use. We will be doing this as and when required.
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np
import seaborn as sns
```

1.2: Database Upload

Use the pandas library to load the dataset into a DataFrame.

```
#Now let's load the dataset from Google Drive
breast_cancer_dataset = pd.read_csv('/content/drive/MyDrive/MSc_in_AI/Machine_Learning/breast-cancer-wisconsin.csv')
```

1.3: Inspecting The Dataset

1.3.1: Knowing the Shape

Using the **shape method** to find out the total number of rows/records and columns/features in the dataset.

```
[4] breast_cancer_dataset.shape

    (569, 33)
```

1.3.2: Knowing the Column Length

Using the **len function** to find out the total number of columns/features in the dataset.

```
len(breast_cancer_dataset.columns)

33
```

1.3.3: Knowing the Features

Listing out the names of all the columns in the dataset.

```
breast_cancer_dataset.columns

Index(['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', 'Unnamed: 32'],
      dtype='object')
```

Result: This result shows an "Unnamed: 32" column which doesn't exist in the dataset (though showing in the DataFrame). Let me DROP it.

Dropping Unnamed Column

```
[7]  breast_cancer_dataset = breast_cancer_dataset.drop('Unnamed: 32', axis=1)
```

Now let's check the names of columns again to verify if the Unnamed:32 column is dropped (or not!).

```
breast_cancer_dataset.columns
```

```
Index(['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'],
      dtype='object')
```

Success! It is dropped.

1.3.4: Knowing the Row Length

Using the **len function** to find out the total number of rows/records in the dataset.

```
len(breast_cancer_dataset)
```

```
569
```

1.3.5: Knowing the Dataset Size

Using the **Size function** to find out the size of the dataset.

```
breast_cancer_dataset.size
```

```
18208
```

Displaying the First Five Rows (Head)

breast_cancer_dataset.head(): Using the head() to print the first 5 records from the dataframe.

```
breast_cancer_dataset.head()
```

	id	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean	compactness_mean	concavity_mean	concave points_mean	...	radius_worst	t
0	842302	M	17.99	10.38	122.80	1001.0	0.11840	0.27760	0.3001	0.14710	...	25.38	
1	842517	M	20.57	17.77	132.90	1326.0	0.08474	0.07864	0.0869	0.07017	...	24.99	
2	84300903	M	19.69	21.25	130.00	1203.0	0.10960	0.15990	0.1974	0.12790	...	23.57	
3	84348301	M	11.42	20.38	77.58	386.1	0.14250	0.28390	0.2414	0.10520	...	14.91	
4	84358402	M	20.29	14.34	135.10	1297.0	0.10030	0.13280	0.1980	0.10430	...	22.54	

5 rows × 32 columns

Now we can notice the total column number shown is 32, not 33 as we dropped the Unnamed: 32.

Displaying the Last Five Rows (Tail)

breast_cancer_dataset.tail(): Using the tail() to print the last 5 records from the DataFrame

```
breast_cancer_dataset.tail()
```

	id	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean	compactness_mean	concavity_mean	concave points_mean	...	radius_worst
564	926424	M	21.56	22.39	142.00	1479.0	0.11100	0.11590	0.24390	0.13890	...	25.450
565	926682	M	20.13	28.25	131.20	1261.0	0.09780	0.10340	0.14400	0.09791	...	23.690
566	926954	M	16.60	28.08	108.30	858.1	0.08455	0.10230	0.09251	0.05302	...	18.980
567	927241	M	20.60	29.33	140.10	1265.0	0.11780	0.27700	0.35140	0.15200	...	25.740
568	92751	B	7.76	24.54	47.92	181.0	0.05263	0.04362	0.00000	0.00000	...	9.456

5 rows × 32 columns

1.3.6: Describing Statistics of our DataFrame

breast_cancer_dataset.describe(): Generates the descriptive statistics providing a quick summary of the distribution of values (count, mean, standard deviation, minimum value, maximum value, etc.) in each numerical column of the DataFrame.

```
# Using describe() to generate descriptive statistics for our DataFrame.
breast_cancer_dataset.describe()
```

	id	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean	compactness_mean	concavity_mean	concave points_mean	symmetry_mean
count	5.690000e+02	569.000000	569.000000	569.000000	569.000000	569.000000	569.000000	569.000000	569.000000	569.000000
mean	3.037183e+07	14.127292	19.289649	91.969033	654.889104	0.096360	0.104341	0.088799	0.048919	0.181162
std	1.250206e+08	3.524049	4.301036	24.298981	351.914129	0.014064	0.052813	0.079720	0.038803	0.027414
min	8.670000e+03	6.981000	9.710000	43.790000	143.500000	0.052630	0.019380	0.000000	0.000000	0.106000
25%	8.692180e+05	11.700000	16.170000	75.170000	420.300000	0.086370	0.064920	0.029560	0.020310	0.161900
50%	9.060240e+05	13.370000	18.840000	86.240000	551.100000	0.095870	0.092630	0.061540	0.033500	0.179200
75%	8.813129e+06	15.780000	21.800000	104.100000	782.700000	0.105300	0.130400	0.130700	0.074000	0.195700
max	9.113205e+08	28.110000	39.280000	188.500000	2501.000000	0.163400	0.345400	0.426800	0.201200	0.304000

8 rows × 31 columns

1.3.7: Finding Unique Values in Target Feature

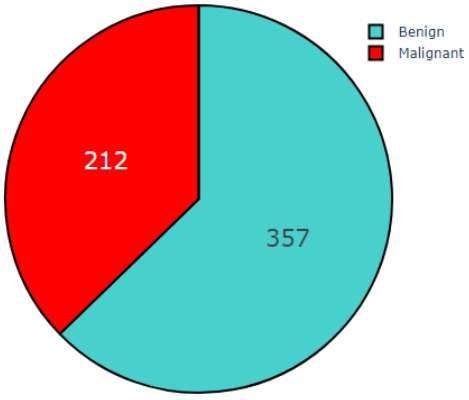
Now let's find out the unique values from the 'Diagnosis' column/feature of the DataFrame.

```
[ ]  breast_cancer_dataset['diagnosis'].unique()

     array(['M', 'B'], dtype=object)
```

2. Data Visualization

Checking distribution of the "Diagnosis" feature with a Pie chart to compare the number of Malignant VS Benign Tumours.



3. Data Preprocessing

It includes checking for missing values, anomalies, or outliers in the dataset. It also involves deciding on strategies for handling categorical variables, scaling numerical features, and encoding labels. Preprocessing steps also include standardization, normalization, or handling imbalanced classes.

Checking for Missing Values: Now let's check if our DataFrame has any missing values using breast_cancer_dataset.isnull().sum() method.

```
breast_cancer_dataset.isnull().sum()
```

```
id                         0
diagnosis                  0
radius_mean                0
texture_mean               0
perimeter_mean             0
area_mean                  0
smoothness_mean            0
compactness_mean           0
concavity_mean             0
concave points_mean        0
symmetry_mean              0
fractal_dimension_mean     0
radius_se                  0
texture_se                 0
perimeter_se               0
area_se                    0
smoothness_se              0
compactness_se             0
concavity_se               0
concave points_se          0
symmetry_se                0
fractal_dimension_se       0
radius_worst               0
texture_worst              0
perimeter_worst            0
area_worst                 0
smoothness_worst           0
compactness_worst          0
concavity_worst            0
concave points_worst       0
symmetry_worst             0
fractal_dimension_worst    0
dtype: int64
```

Result: There are no missing values in our dataset.

4. Feature Engineering

Feature engineering transforms raw data into a format suitable for machine learning algorithms. Effective feature engineering can significantly impact the success of a model, often leading to improved accuracy, robustness, and interpretability.

4.1: Feature Mapping

I am using this technique to convert M and B into numbers. Our predictive models can better understand numbers so this will help with data analysis/classification.

```
# catogorical column - Diagnosis (M = malignant = 1 , B = benign = 0)
breast_cancer_dataset['diagnosis']=breast_cancer_dataset['diagnosis'].map({"M":1,"B":0})
#other way to convert categorical values to numeric: dataset['binary_variable'].replace({'yes':1, 'no':0},inplace=True)
```

After feature mapping, now let's use the tail [breast_cancer_dataset.tail()] method to check if the diagnosis feature is converted to binary variables or not.

```
#Now let's check the last 5 rows using tail() method.
breast_cancer_dataset.tail()
```

	id	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean	compactness_mean	concavity_mean	concave points_mean	...	radius_worst
564	926424	1	21.56	22.39	142.00	1479.0	0.11100	0.11590	0.24390	0.13890	...	25.450
565	926682	1	20.13	28.25	131.20	1261.0	0.09780	0.10340	0.14400	0.09791	...	23.690
566	926954	1	16.60	28.08	108.30	858.1	0.08455	0.10230	0.09251	0.05302	...	18.980
567	927241	1	20.60	29.33	140.10	1265.0	0.11780	0.27700	0.35140	0.15200	...	25.740
568	92751	0	7.76	24.54	47.92	181.0	0.05263	0.04362	0.00000	0.00000	...	9.456

5 rows × 32 columns

Success! Now our diagnosis feature has binary values.

4.2: Feature Selection

Feature selection is an important step in building a classification model. It is advantageous to limit the number of input attributes in a classifier to have good predictive and less computationally intensive models.

I have selected the **"worst" or largest value** of the given 10 features computed for each cell nucleus in the breast cancer dataset. It is based on my understanding that the worst-case scenario might be more indicative of the underlying pathology or severity of the condition being studied.

My Reasons:
Clinical Relevance: In medical contexts, the "worst" or largest values may represent the most severe or aggressive characteristics of the cell nuclei which can be crucial for prognosis and treatment decisions.

Pathological Significance: The "worst" values might be more indicative of pathological abnormalities or malignant behavior in the cells.

Risk Assessment: Focusing on the worst-case scenario helps identify cases that may have a higher risk of progression or recurrence.

Decision-Making: Clinicians prioritize information about the most aggressive features to decide on the type of treatment required.

For the above listed reasons, the worst or largest values seem the most relevant of each feature for the classification task at hand.

4.3: Feature Scaling

Feature scaling is a preprocessing technique used to standardize or normalize independent variables (features) of the dataset. Real-world datasets such as ours, the breast cancer database, usually contain features of varying degrees. For our ML model/algorithm to interpret accurately these features need to be on the same scale/range. This is why we need to perform feature scaling.

5. Train-Test Split

I am splitting my data into training and testing sets in the ratio of 70:30 where 70 is the percentage of the training set, and 30 is the testing set. Splitting a dataset into training and testing sets is a fundamental practice in machine learning, and it serves several important purposes. Some of them are:

Model Evaluation: It provides a way to check how well a machine learning model can perform on unseen data. The model is trained on the training set, and its performance is then tested on the testing set.

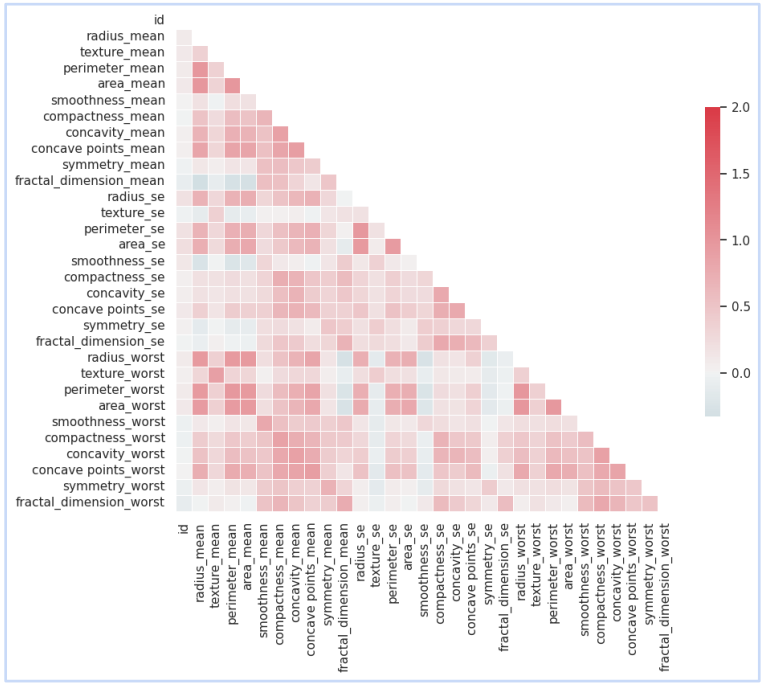
Building Trust in the Model: The ability of a model to perform well on an independent testing set instills confidence in its predictive capabilities. It demonstrates that the model has learned patterns that generalize to new instances.

Simulation of Real-world Deployment: In real-world scenarios, a machine learning model is deployed to make predictions on new, unseen data. The testing set simulates this deployment scenario, providing insights into how the model is likely to perform in practice.

Result:

```
Training set - Feature Length (x, y):  (398, 31) diagnosis:  (398,)
Test set - Feature Length  (x, y):  (171, 31) diagnosis:  (171,)
```

6. Correlation Matrix



Each cell in the table represents the correlation between two variables. The coefficients quantify the strength and direction of a linear relationship. The values range from -1 to 1, where 1 indicates a perfect positive correlation, -1 indicates a perfect negative correlation and 0 indicates no correlation.

Colors indicate the strength and direction of the relationship. Blue represents negative, red represents positive, and white indicates no correlation. The deeper the shade, the stronger the correlation.

Axes: Each axis of the heatmap corresponds to a feature in the dataset.

Based on the heatmap, here are some observations:

- There appear to be several strong positive correlations, represented by dark red squares.
- There are also some strong negative correlations, represented by dark blue squares.
- The majority of the squares are light in color, indicating weaker correlations between features.

7. Analyzing Classification Algorithms

In this project, I plan to use the following Machine Learning algorithms for classification:

1. Decision Tree Classifier
2. Logistic Regression
3. Convolutional Neural Network

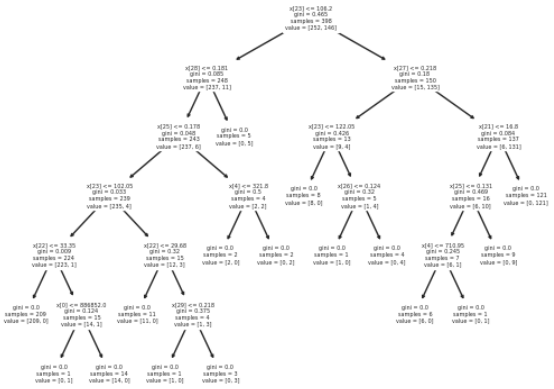
7.1: Testing Algorithm #1 — Decision Tree Classification

Discussing my design considerations for Decision Tree Classification: It is a type of tree-based supervised machine learning algorithm used for both classification and regression tasks. It offers a transparent and intuitive approach to solving classification problems. It builds a tree-like structure where each internal node represents a decision based on a feature, each branch represents an outcome of that decision, and each leaf node represents a class label. **Interpretability:** Decision trees provide a clear and interpretable decision-making process. **Feature Importance:** Decision trees can reveal important features for classification.

Train the Model: Fit the selected model on the training data.
Model Evaluation/Evaluation Metrics: I will evaluate the model's performance using metrics like accuracy, precision, recall, and F1-score.

```
classifier.score(x_test, y_test)
```

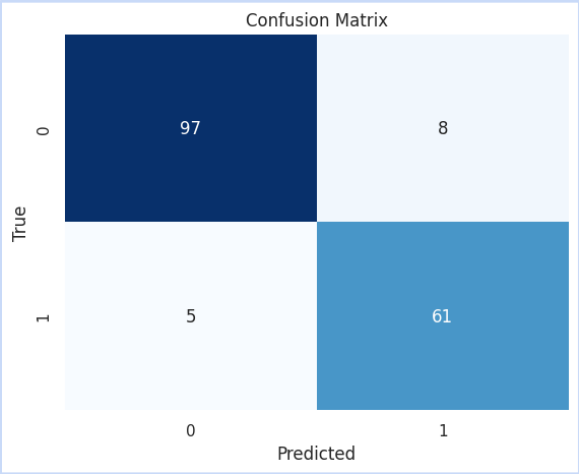
```
0.9181286549707602
```



An accuracy of 91.81% (a higher accuracy) suggests that the decision tree classifier correctly predicted the class labels for approximately 91.81% of the samples in the test set. However additional metrics like precision, recall, F1-score, or a confusion matrix might be needed for a more comprehensive evaluation of the model's performance.

7.1.1: Confusion Matrix for DecisionTreeClassifier Model

Confusion matrices are calculated for accuracy, sensitivity, and specificity to visualize predicted outcomes and actual results. It is a table that categorizes predictions according to whether they match the actual value. It is a good way to look at how a model is performing.



This figure gives the details of True Positive, True Negative, False Positive and False Negative of cancer detection:
True Positive (TP): 97% of Malignant tumours correctly classified as Malignant
True Negative (TN): 61% of Benign tumours correctly classified as benign
False Positive (FP): 8% of Benign tumour incorrectly classified as malignant
False Negative (FN): 5% of Malignant tumour incorrectly classified as benign

Classification Report:
The classification report provides a summary of the performance of a classification model.

Classification Report:

	precision	recall	f1-score	support
B	0.95	0.92	0.94	105
M	0.88	0.92	0.90	66
accuracy			0.92	171
macro avg	0.92	0.92	0.92	171
weighted avg	0.93	0.92	0.92	171

Here's a brief explanation of the metrics:

- Precision: Measures the accuracy of positive predictions made by the model.
  - For class B (Benign) 95% were correctly predicted.
  - For class M (Malignant) 88% were correctly predicted.
- Recall (Sensitivity): Recall is the ratio of correctly predicted positive observations.
  - For class B (Benign) 92% of all actual Benign cases were correctly identified by the model.
  - For class M (Malignant) 92% of all actual Malignant cases were correctly identified.
- F1-Score: Is the harmonic mean of precision & recall. It provides a balance between both.
  - For class B (Benign), the score is 94%.
  - For class M (Malignant), the score is 90%.
- Accuracy: The model correctly predicted the class for 92% across all classes.
- Macro Avg: Macro Avg. Precision, Recall, and F1-score are all 0.92.
- Weighted Avg: Weighted Avg Precision, Recall, and F1-score are all 0.92.

The classification report provides a comprehensive overview of the model's performance, indicating high precision, recall, and F1-score for both classes, as well as a high overall accuracy. The model demonstrates good ability to distinguish between Benign and Malignant cases in the test data. **The result** is reasonably good, but let me try and find a better algorithm which can fetch increased accuracy.

7.2: Testing Algorithm #2 — Logistic Regression

Logistic Regression is a statistical method used for predicting probability for classification problems. It is a classification algorithm used for binary or multiclass classification problems. In our case, it is predicting if the tumour is Malignant or Benign.

```
Classifier Score: 0.935672514619883

Classification Report:
              precision    recall  f1-score   support

           0       0.94      0.96      0.95       105
           1       0.94      0.89      0.91        66

    accuracy                           0.94       171
   macro avg       0.94      0.93      0.93       171
weighted avg       0.94      0.94      0.94       171


Accuracy: 0.935672514619883
```

Result:
The Accuracy is 93.5% — Second best!

Precision: 94% for both classes. High precision relates to a low false positive rate.

Recall (Sensitivity): For class 0, recall is 0.96, and for class 1, recall is 0.89. High recall relates to a low false negative rate.

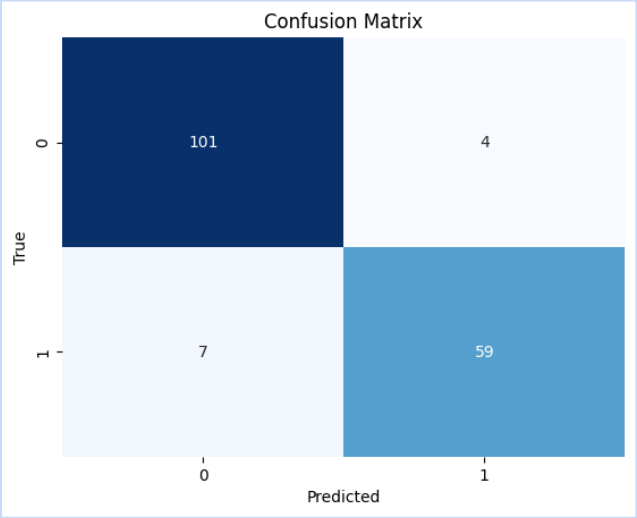
F1-score: For Class 0: 0.95. For class 1: 0.91.

Accuracy is approximately 93.56%.

Interpretation: The model is performing very well, with high precision, recall, and F1-score for both classes (0 and 1). The accuracy indicates that the model is making correct predictions for the majority of the samples in the test set indicating robust performance across different classes.

Conclusions: The model appears to be effective and **second best** in predicting the diagnosis, achieving high precision, recall, and F1-score for both classes.

7.2.1: Confusion Matrix for LogisticRegression



Here's a report:
True Positives: The model correctly predicted 101 instances of the positive class (Malignant).
False Negatives: The model incorrectly predicted 7 instances as negative class (Benign) when they were actually positive (Malignant).
False Positives: The model incorrectly predicted 4 instances as the positive class (Malignant) when they were actually negative (Benign).
True Negatives: The model correctly predicted 59 instances of the negative class (Benign).

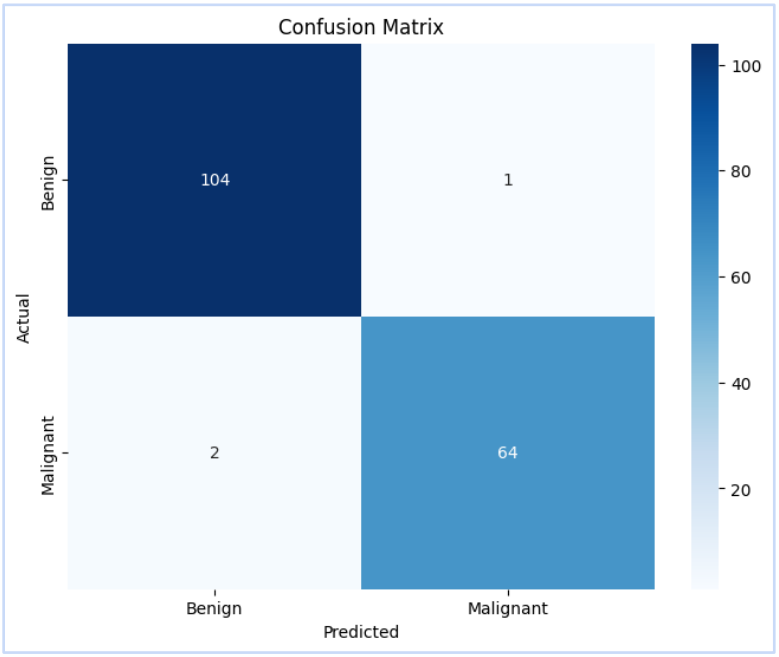
7.3: Test Algorithm #3 — Convolutional Neural Network (CNN)

Building a Convolutional Neural Network (CNN) for breast cancer classification. **Design Considerations:** CNNs are deep learning models designed for image-related tasks, capturing spatial hierarchies through convolutional layers.

```
Epoch 1/7
13/13 [==============================] - 8s 64ms/step - loss: 0.5101 - accuracy: 0.9020 - val_loss: 0.3769 - val_accuracy: 0.9591
Epoch 2/7
13/13 [==============================] - 0s 10ms/step - loss: 0.3080 - accuracy: 0.9296 - val_loss: 0.2377 - val_accuracy: 0.9649
Epoch 3/7
13/13 [==============================] - 0s 10ms/step - loss: 0.2009 - accuracy: 0.9397 - val_loss: 0.1736 - val_accuracy: 0.9708
Epoch 4/7
13/13 [==============================] - 0s 11ms/step - loss: 0.1450 - accuracy: 0.9497 - val_loss: 0.1491 - val_accuracy: 0.9708
Epoch 5/7
13/13 [==============================] - 0s 10ms/step - loss: 0.1127 - accuracy: 0.9623 - val_loss: 0.1366 - val_accuracy: 0.9708
Epoch 6/7
13/13 [==============================] - 0s 10ms/step - loss: 0.0930 - accuracy: 0.9698 - val_loss: 0.1318 - val_accuracy: 0.9766
Epoch 7/7
13/13 [==============================] - 0s 13ms/step - loss: 0.0789 - accuracy: 0.9749 - val_loss: 0.1307 - val_accuracy: 0.9825
6/6 [==============================] - 0s 6ms/step - loss: 0.1307 - accuracy: 0.9825
Test Accuracy: 0.9824561476707458
```

With the test accuracy of 98.24% CNN is by far the best model for Breast Cancer classification.

7.3.1: Confusion Matrix for the above CNN Model



The confusion matrix will help in understanding how well my model is performing in terms of true positives, true negatives, false positives, and false negatives.

Here's a brief explanation of the metrics:

- Precision: For both the classes 0 (Benign) and 1 (Malignant) 98% were correctly predicted.
- Recall: For class 0 (Benign) 99% of cases were correctly identified; for class 1 (Malignant) 97%.
- F1-score: 99% for class 0 (Benign) and 98% for class 1 (Malignant) representing a balanced performance.
- Accuracy: 98% meaning the model correctly predicted the class for 98% of instances in the test set.

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

           0       0.98      0.99      0.99       105
           1       0.98      0.97      0.98        66

    accuracy                           0.98       171
   macro avg       0.98      0.98      0.98       171
weighted avg       0.98      0.98      0.98       171
```

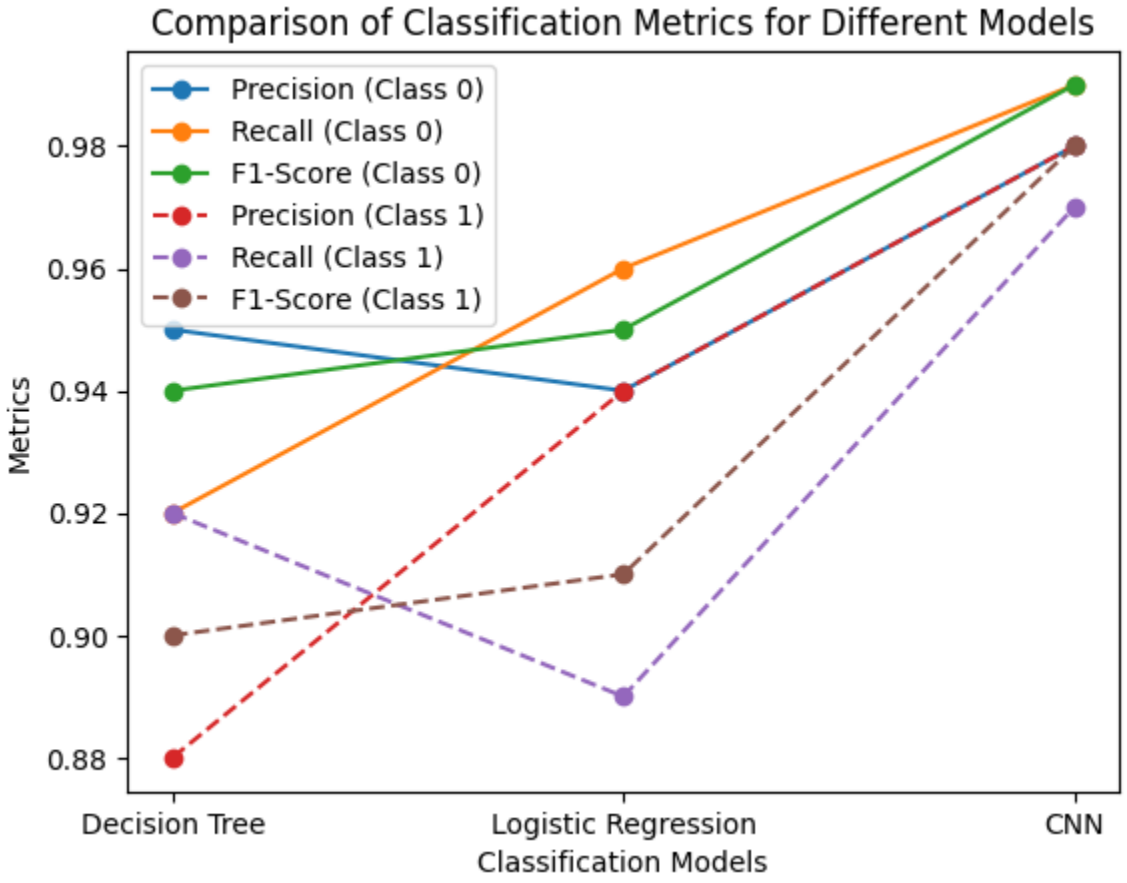
Importance of False Negatives: The presence of false negatives is a cause for concern. False negatives are particularly critical in scenarios that could lead to severe consequences such as in medical diagnosis --- A false negative might mean failing to identify a condition that may require immediate treatment.

8. Comparison and Conclusion

Evaluation Metrics: The metrics I am using for evaluation are:

- Precision
- Recall
- F1-Score

Let's plot a chart to compare the evaluation metrics of different machine learning algorithms we tested: 'Decision Tree', 'Logistic Regression', and 'CNN'.



The accuracy scores of the algorithms tested is:

- Decision Tree Classifier was good with 91.81% accuracy.
- Logistic Regression was better with 93.56% accuracy.
- Convolutional Neural Network turned out to be the best with 98.24% accuracy.

— My Conclusion

The evaluation of various classification models on the Breast Cancer Dataset demonstrates that the Convolutional Neural Network (CNN) model/algorithm **outperforms** other models with a remarkable test accuracy of 98.24%. The CNN exhibits superior performance in distinguishing between benign and malignant cases, showcasing its effectiveness in handling the complex patterns present in the digitized images of fine needle aspirations of breast tissue masses. The robustness and accuracy of the CNN make it

the optimal choice for the classification task, underscoring its potential as a powerful tool in the domain of breast cancer diagnosis and classification.

However, it is crucial to collaborate with domain experts, stakeholders, and end-users to align model performance with the specific requirements and implications of the application. Continuous monitoring and iterative improvements will contribute to refining the model's ability to accurately classify positive instances, thereby enhancing its overall effectiveness and reliability in real-world scenarios.

— How can the system be improved?

- **Data Quality:** By ensuring that the medical image data is of high quality, properly labeled, and free from artifacts that may impact model performance.
- **Class Imbalance:** By exploring techniques such as oversampling, undersampling, or using different evaluation metrics that account for class distribution.
- **Ensemble Methods:** By building an ensemble of models to combine their predictions, potentially improving overall accuracy and robustness.
- **Cross-Validation:** Use techniques like k-fold cross-validation to get a more robust estimate of the model's performance.
- **Handling Overfitting:** Implement strategies to avoid overfitting, such as regularization techniques, early stopping during training, or using ensemble methods that naturally reduce overfitting, like Random Forests.

— Societal and Ethical Issues

The development and deployment of machine learning (ML) autonomous systems raise several societal and ethical issues. Here are some key considerations:

1. **Bias and Fairness:** ML models can inherit biases present in training data, leading to unfair and discriminatory outcomes. Unfair treatment of certain groups or individuals can reinforce and perpetuate societal inequalities.
2. **Transparency and Explainability:** Many advanced ML models, particularly deep learning models, are often seen as "black boxes" because their decision-making processes are not easily interpretable. Lack of transparency can lead to distrust, hinder accountability, and make it challenging to understand and rectify erroneous decisions.
3. **Privacy Concerns:** ML systems often require vast amounts of data, and the collection and use of personal information raise privacy concerns. Invasive data collection and potential misuse of personal information can compromise individuals' privacy rights.
4. **Security Risks:** ML models can be vulnerable to adversarial attacks, where malicious actors manipulate input data to deceive the system. Security vulnerabilities can lead to unintended consequences, such as misclassification or exploitation of the system.
5. **Job Displacement and Economic Impact:** The widespread adoption of autonomous systems could lead to job displacement in certain industries, potentially impacting the job market. The economic impact may result in socioeconomic disparities.
6. **Accountability and Liability:** Determining accountability and liability for the actions of autonomous systems, especially in cases of system failures or accidents, is challenging. Lack of

clear accountability can hinder legal and ethical responsibility for the consequences of ML system behavior.

7. **Data Governance and Ownership:** The ownership and governance of data used to train ML models are often unclear, leading to questions about who controls and benefits from the data. Unequal access to and control over data can exacerbate existing power imbalances.

By carefully considering these aspects, we can lay the foundation for a well-thought-out machine learning model that can perform better and provide meaningful insights. Addressing these issues requires collaboration among technologists, policymakers, ethicists, and the broader society to ensure the responsible development and deployment of machine learning autonomous systems.

Ethical Considerations: We must ensure that our process is fair, unbiased, and does not reinforce existing disparities in healthcare. Also, we should ensure ethical considerations, such as patient privacy, data security, and transparency in model decision-making, are adequately addressed throughout the improvement process. Ethical frameworks, regulations, and ongoing dialogue are essential to navigating the complex landscape of societal and ethical considerations in this rapidly evolving field.

III. References

William Wolberg, O.M. (1993) 'Breast Cancer Wisconsin (Diagnostic)'. Available at: https://archive.ics.uci.edu/dataset/17 (Accessed: 12 January 2024).

Breast cancer. Available at: https://www.who.int/news-room/fact-sheets/detail/breast-cancer (Accessed: 12 January 2024).

'Facts and figures'. *Breast Cancer UK*. Available at: https://www.breastcanceruk.org.uk/about-breast-cancer/facts-figures-and-qas/facts-and-figures/ (Accessed: 12 January 2024).

Shen, L., Margolies, L.R., Rothstein, J.H., Fluder, E., McBride, R. and Sieh, W. (2019) 'Deep Learning to Improve Breast Cancer Detection on Screening Mammography'. *Scientific Reports* 9(1), p. 12495. Available at: https://www.nature.com/articles/s41598-019-48995-4 (Accessed: 12 January 2024).

Kwak, D., Choi, J. and Lee, S. (2023) 'Rethinking Breast Cancer Diagnosis through Deep Learning Based Image Recognition'. *Sensors (Basel, Switzerland)* 23(4), p. 2307. Available at: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC9958611/ (Accessed: 12 January 2024).

Liu, L., Feng, W., Chen, C., Liu, M., Qu, Y. and Yang, J. (2022) 'Classification of breast cancer histology images using MSMV-PFENet'. *Scientific Reports* 12(1), p. 17447. Available at: https://www.nature.com/articles/s41598-022-22358-y (Accessed: 12 January 2024).