

Task Report: Breast Cancer Classification Using Logistic Regression

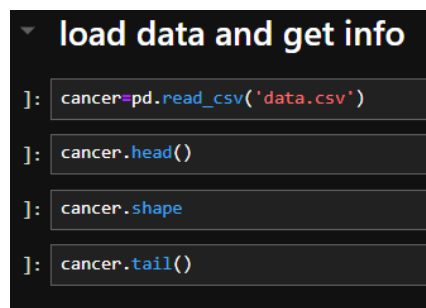
1. Introduction

The task is to predict whether a breast cancer tumour is malignant (M) or benign (B) using features computed from digitised images of a fine needle aspirate (FNA) of a breast mass. This is a binary classification problem where the target variable (diagnosis) can take one of two values: malignant (M) or benign (B).

The goal is to build a machine-learning model using Logistic Regression to predict the diagnosis based on these features.

2. Data Exploration and Understanding

Before building the model, we explored the dataset to understand its structure and identify potential issues, such as missing data.



```
load data and get info

]: cancer=pd.read_csv('data.csv')

]: cancer.head()

]: cancer.shape

]: cancer.tail()
```

- The dataset has 569 instances (rows) and 32 attributes (columns).
- Diagnosis: Target column (Malignant/Benign).
- Features such as radius_mean, texture_mean, and smoothness_mean describe the characteristics of the cell nuclei.

Checking for missing values:

```
Analysis and Visualization

: cancer.isna().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
  Unnamed: 32      569
  dtype: int64
```

No missing data was found in the dataset.

3. Data Preprocessing

We started by encoding the target variable (diagnosis), which is categorical, into numerical values using Label Encoding. This step is necessary because machine learning models work with numeric data.

```
# Create a LabelEncoder object
le = LabelEncoder()

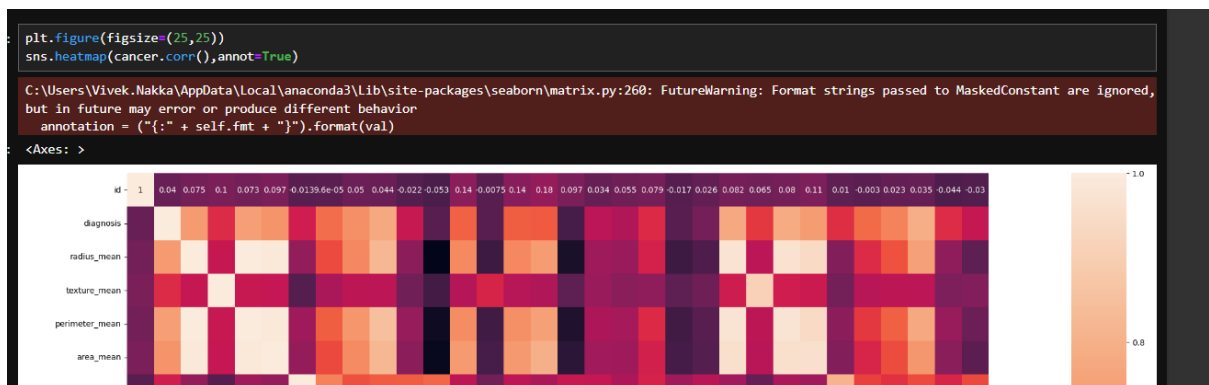
# Fit and transform the categorical data
cancer['diagnosis'] = le.fit_transform(cancer['diagnosis'])
```

diagnosis	
0	M
1	B

- **Malignant (M)** is encoded as 1.
- **Benign (B)** is encoded as 0.

4. Correlation Analysis

We generated a correlation matrix to visualise the relationships between features and the target variable. Features with a high correlation with the target could be important predictors in the model.



Key observations:

- Some features (e.g., radius_mean, perimeter_mean) have strong correlations with the target variable (diagnosis).
- This helps us understand which features are likely to be the most informative for prediction.

5. Splitting the Data into Train and Test Sets

The dataset was divided into training and test sets. We used 80% of the data for training the model and 20% for testing its performance.

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

[ ]: y=cancer['diagnosis']

[ ]: x

[ ]: y

[ ]: x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=0.2,random_state=42)

[ ]: x_train
```

- `x_train, x_test`: Features for training and testing.
- `y_train, y_test`: Target variable for training and testing.

6. Feature Scaling

Logistic Regression requires feature scaling for optimal performance. We used the `StandardScaler` to scale the features, so they all have a mean of 0 and a standard deviation of 1.

```
s=StandardScaler()

x_train=s.fit_transform(x_train)
x_test=s.fit_transform(x_test)
```

7. Training the Logistic Regression Model

We used the Logistic Regression model to train the data.

```
] : model=LogisticRegression()
] : model.fit(x_train,y_train)
] : LogisticRegression
    LogisticRegression()
```

- The model is trained using the `fit()` method, where the training data (`x_train, y_train`) is passed in.
- Logistic Regression is a suitable algorithm for binary classification problems like this one, as it estimates the probability that a given instance belongs to one of two classes.

8. Model Evaluation

After training the model, we evaluated its performance on both the training and test sets using accuracy as the metric.

```
predict_test=model.predict(x_test)

predict_test

array([0, 1, 1, 0, 0, 1, 1, 1, 0, 0, 0, 1, 0, 1, 0, 1, 0, 0, 0, 1, 0, 0,
       1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 1, 0, 0,
       0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 1, 1, 0, 0,
       1, 1, 0, 0, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 1, 1, 1, 1, 1,
       0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 1, 1, 0, 1, 1, 0, 0, 0, 1, 0,
       1, 0, 0, 1])
```

Accuracy on Test Data:

To assess the performance of the model, we calculated the accuracy on the test set.

```
accuracy_score(y_test,predict_test)
0.9824561403508771
```

Accuracy on Training Data:

We also calculated the accuracy of the model on the training data to check if the model was overfitting or underfitting.

```
predict_train=model.predict(x_train)

predict_train

array([0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 1, 1, 0, 0, 1, 1, 1, 0, 0, 0, 1,
       0, 0, 0, 1, 0, 1, 0, 0, 1, 0, 1, 1, 1, 0, 1, 0, 0, 0, 0, 1, 1, 0,
       0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0,
       0, 0, 0, 1, 1, 1, 1, 0, 1, 0, 1, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0,
       1, 0, 1, 1, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 1, 1,
       0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 0, 0, 1, 0, 0, 1, 0, 1, 0,
       1, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 1, 0, 1, 0, 1, 0, 0, 1, 0, 0, 0,
       0, 1, 0, 0, 0, 1, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0,
       0, 1, 0, 1, 0, 0, 0, 1, 0, 1, 1, 0, 0, 1, 0, 1, 1, 0, 0, 0, 0, 1,
       0, 0, 1, 0, 1, 0, 0, 0, 1, 0, 1, 0, 0, 1, 1, 0, 0, 1, 0, 1, 1, 0,
       1, 1, 0, 0, 1, 1, 1, 0, 0, 0, 0, 1, 0, 1, 1, 1, 1, 0, 0, 0, 0, 0,
       0, 0, 0, 1, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 1, 0, 1, 0,
       0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 1,
       1, 1, 0, 1, 0, 0, 1, 1, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0,
       0, 1, 1, 0, 0, 0, 1, 1, 0, 0, 1, 0, 1, 1, 0, 1, 1, 0, 0, 1, 0, 1,
       1, 1, 0, 0, 0, 1, 0, 0, 1, 1, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 1,
       1, 1, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 1, 1, 1, 0, 0, 1, 0,
       0, 1, 1, 1, 1, 0, 0, 1, 1, 0, 0, 0, 1, 1, 1, 0, 0, 1, 0, 0, 0,
       1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 1, 0, 1, 0,
       0, 1, 1, 1, 0, 1, 1, 0, 1, 0, 0, 0, 0, 1, 0])

accuracy_score(y_train,predict_train)
0.9868131868131869
```

8. Summary of Results

The logistic regression model demonstrated strong performance in predicting breast cancer diagnosis. Key metrics from the model are as follows:

- **Training Accuracy:** The model achieved high accuracy on the training set, suggesting a good fit to the training data.
- **Test Accuracy:** The model's accuracy on the test data was similarly high, indicating good generalization to unseen data.