**Logistic Regression Report**

**Objective :** Build a binary classifier using **Logistic Regression**.

**Tools Used**

* **Python**
* **Pandas** – for data manipulation
* **Scikit-learn** – for machine learning modeling
* **Matplotlib & Seaborn** – for visualization

**Dataset Summary**

The dataset used is a **binary classification dataset** related to breast cancer diagnosis.

* Target variable: diagnosis
* Class labels:
  + M = Malignant (1)
  + B = Benign (0)
* Features: 30 numeric attributes extracted from cell nuclei in breast mass images.

**Data Preprocessing**

* Dropped irrelevant columns like id and Unnamed: 32.
* Encoded target variable (M → 1, B → 0).
* Checked and removed **missing values**.
* Performed **train-test split** (80-20).
* Standardized features using StandardScaler.

**Model Building**

python

CopyEdit

from sklearn.linear\_model import LogisticRegression

clf = LogisticRegression()

clf.fit(X\_train, y\_train)

**Evaluation Metrics**

* **Confusion Matrix**
* **Precision**
* **Recall**
* **Accuracy**
* **ROC-AUC Score**
* **Classification Report**

**Sigmoid Function Explanation**

The sigmoid function maps any real-valued number into a value between 0 and 1:

σ(z)=11+e−z\sigma(z) = \frac{1}{1 + e^{-z}}σ(z)=1+e−z1​

Used in logistic regression to model probability that a given input belongs to the positive class.

**Threshold Tuning**

You can adjust the default threshold (0.5) to increase **recall** (e.g., for medical use-cases) or **precision** (to reduce false positives).

python

CopyEdit

y\_proba = clf.predict\_proba(X\_test)[:, 1]

y\_pred\_new = (y\_proba >= 0.3).astype(int)

**Conclusion**

* Logistic Regression provided strong baseline performance for binary classification.
* Can be enhanced further with feature selection and model tuning.