Logistic Regression Model for Breast Cancer Prediction

NAME- Ankita Satapathy
ROLL NO- 22053400

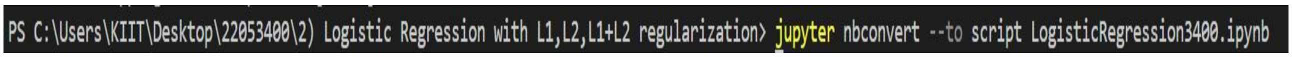
1. Introduction

This project implements a logistic regression model to predict the likelihood of breast cancer based on the mean radius of a tumor. The model is built using Python and the scikit-learn library, with a custom logistic regression class. The application provides an API using FastAPI and a front-end interface for user interaction.

2. Files in the Submission

i. LogisticRegression3400.ipynb: Implements a logistic regression model using gradient descent in Jupyter Notebook and also contains the model training process.

ii. LogisticRegression3400.py: Contains the implementation of a custom logistic regression model using Python, including options for L1, L2, and L1+L2 regularization. Generated in VsCode terminal using command-



```
PS C:\Users\KIIT\Desktop\22053400\2) Logistic Regression with L1,L2,L1+L2 regularization> jupyter nbconvert --to script LogisticRegression3400.ipynb
```

iii. main.py: Defines the FastAPI server to handle prediction requests.

iv. model.pkl: The trained model serialized using pickle.

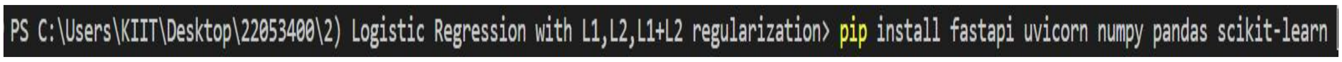
v. index.html: Front-end UI for user input and displaying predictions.

3. Installation and Setup

i. Prerequisites :- Required Python packages: Numpy, scikit-learn, fastapi, pickle, uvicorn.

- NumPy (*numpy)* is used for numerical computations and matrix operations in model training.
- Scikit-learn (*sklearn*) is used for loading the dataset and provides ML utilities.
- FastAPI (*fastapi*) is used to create a web API for serving predictions.
- Pickle (*pickle*) is used to save and load the trained model for reuse.
- Uvicorn (*uvicorn*) is used to run the FastAPI server asynchronously.

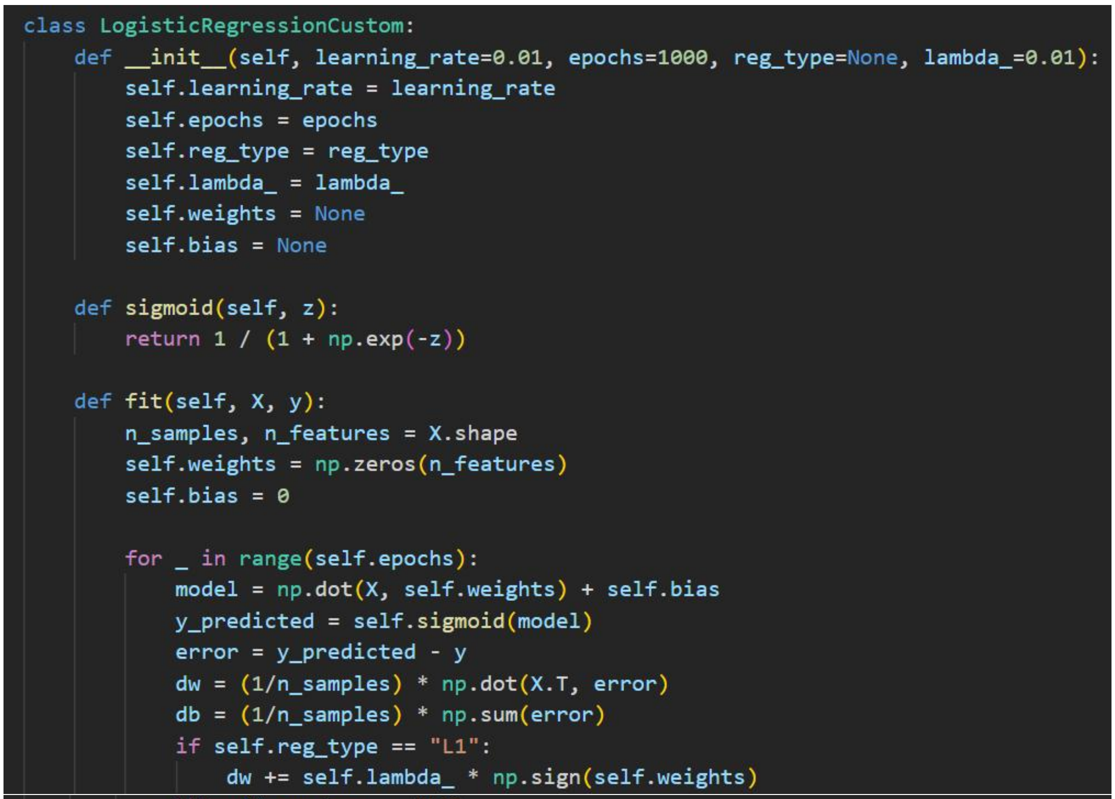
ii. Installing Dependencies:- The following command is used in vscode terminal to install dependencies



```
PS C:\Users\KIIT\Desktop\22053400\2) Logistic Regression with L1,L2,L1+L2 regularization> pip install fastapi uvicorn numpy pandas scikit-learn
```

4. Model Implementation

The LogisticRegressionCustom class in LogisticRegression3400.py implements a logistic regression model with gradient descent optimization and optional regularization (L1, L2, or L1+L2). The model predicts the probability of breast cancer based on tumor features.

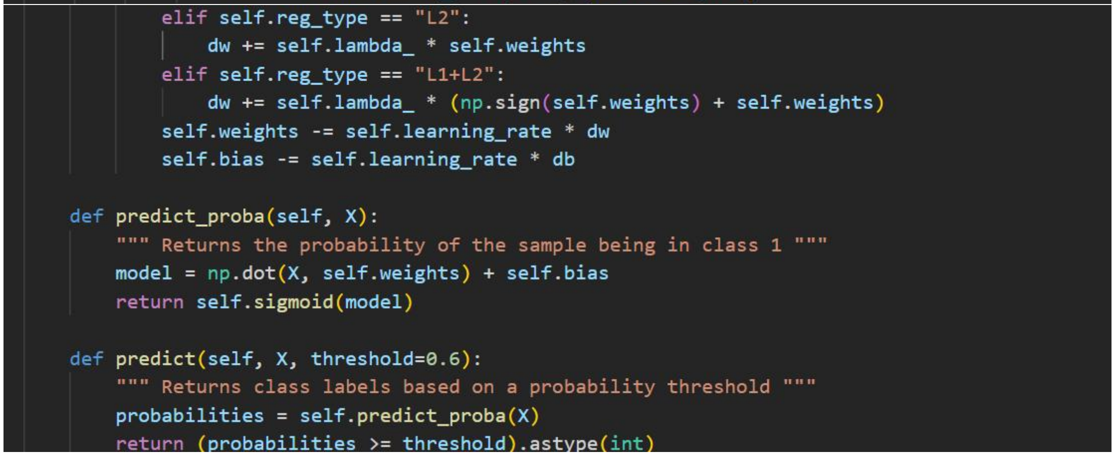


```python
class LogisticRegressionCustom:
    def __init__(self, learning_rate=0.01, epochs=1000, reg_type=None, lambda_=0.01):
        self.learning_rate = learning_rate
        self.epochs = epochs
        self.reg_type = reg_type
        self.lambda_ = lambda_
        self.weights = None
        self.bias = None

    def sigmoid(self, z):
        return 1 / (1 + np.exp(-z))

    def fit(self, X, y):
        n_samples, n_features = X.shape
        self.weights = np.zeros(n_features)
        self.bias = 0

        for _ in range(self.epochs):
            model = np.dot(X, self.weights) + self.bias
            y_predicted = self.sigmoid(model)
            error = y_predicted - y
            dw = (1/n_samples) * np.dot(X.T, error)
            db = (1/n_samples) * np.sum(error)
            if self.reg_type == "L1":
                dw += self.lambda_ * np.sign(self.weights)
```



```python
            elif self.reg_type == "L2":
                dw += self.lambda_ * self.weights
            elif self.reg_type == "L1+L2":
                dw += self.lambda_ * (np.sign(self.weights) + self.weights)
            self.weights -= self.learning_rate * dw
            self.bias -= self.learning_rate * db

    def predict_proba(self, X):
        """ Returns the probability of the sample being in class 1 """
        model = np.dot(X, self.weights) + self.bias
        return self.sigmoid(model)

    def predict(self, X, threshold=0.6):
        """ Returns class labels based on a probability threshold """
        probabilities = self.predict_proba(X)
        return (probabilities >= threshold).astype(int)
```

5. Training the Model

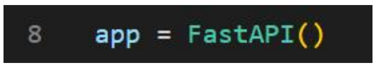
- The dataset used is the Breast Cancer Dataset from sklearn.datasets.
- The model is trained on the Mean Radius feature (X = data.data[:, 0]) to predict the target (y = data.target).
- The training process uses gradient descent with 1000 epochs and L2 regularization.

- The input feature is normalized using Min-Max scaling.
- After training, the model is saved using pickle.

6. API Implementation

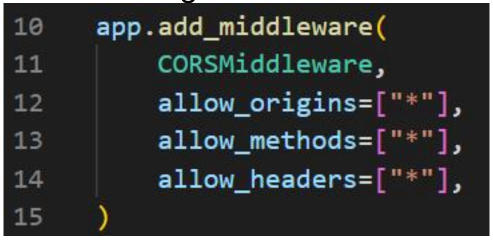
The FastAPI-based backend (main.py) loads the trained model and provides an endpoint to make predictions:

i. API Setup



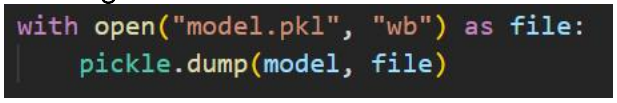
```python
app = FastAPI()
```

ii. CORS Configuration- To allow front-end requests:



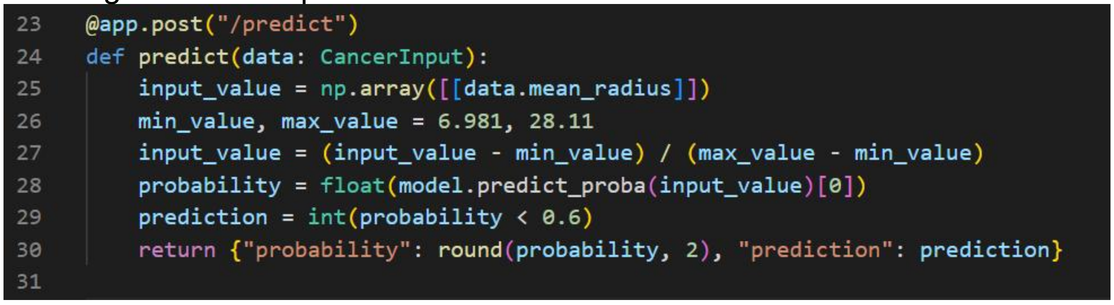
```python
app.add_middleware(
    CORSMiddleware,
    allow_origins=["*"],
    allow_methods=["*"],
    allow_headers=["*"],
)
```

iii. Loading the Trained Model



```python
with open("model.pkl", "wb") as file:
    pickle.dump(model, file)
```

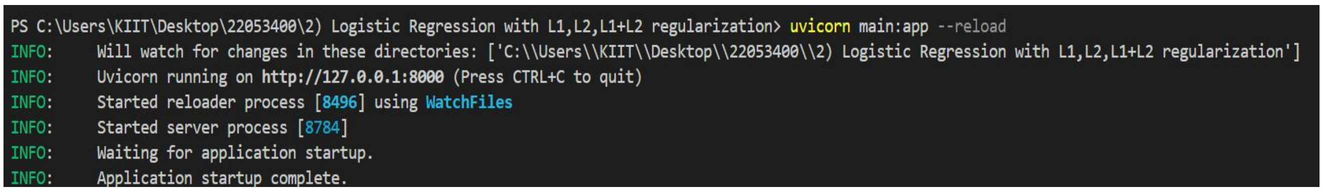
iv. Defining the API Endpoint



```python
@app.post("/predict")
def predict(data: CancerInput):
    input_value = np.array([[data.mean_radius]])
    min_value, max_value = 6.981, 28.11
    input_value = (input_value - min_value) / (max_value - min_value)
    probability = float(model.predict_proba(input_value)[0])
    prediction = int(probability < 0.6)
    return {"probability": round(probability, 2), "prediction": prediction}
```

7. Running the Application (FastAPI Server)

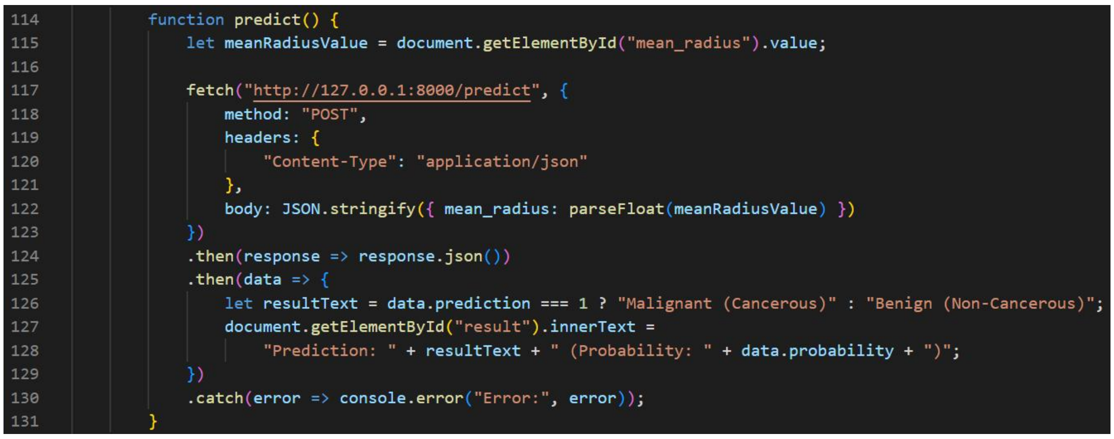
The following command is used in vscode to start the API server:



```
PS C:\Users\KIIT\Desktop\22053400\2) Logistic Regression with L1,L2,L1+L2 regularization> uvicorn main:app --reload
INFO:     Will watch for changes in these directories: ['C:\\Users\\KIIT\\Desktop\\22053400\\2) Logistic Regression with L1,L2,L1+L2 regularization']
INFO:     Uvicorn running on http://127.0.0.1:8000 (Press CTRL+C to quit)
INFO:     Started reloader process [8496] using WatchFiles
INFO:     Started server process [8784]
INFO:     Waiting for application startup.
INFO:     Application startup complete.
```

8. Front-End Implementation

The index.html file provides a simple UI for users to input mean radius values and get predictions using the API. It has inline CSS and JavaScript. The JavaScript function sends a request to the FastAPI backend:

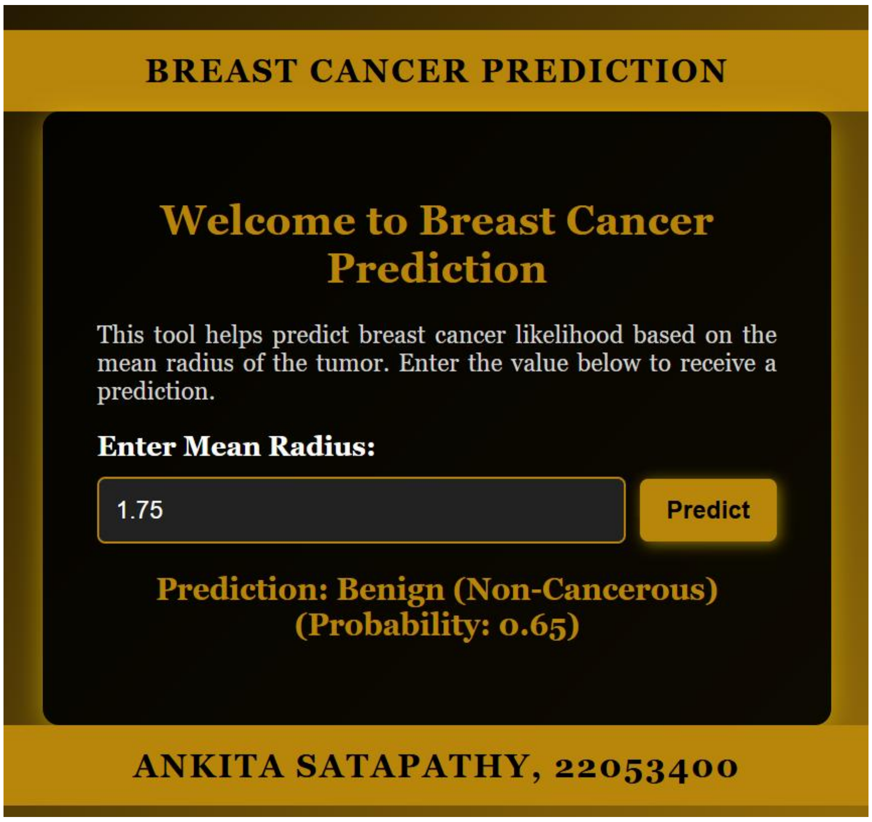


```
        function predict() {
            let meanRadiusValue = document.getElementById("mean_radius").value;

            fetch("http://127.0.0.1:8000/predict", {
                method: "POST",
                headers: {
                    "Content-Type": "application/json"
                },
                body: JSON.stringify({ mean_radius: parseFloat(meanRadiusValue) })
            })
            .then(response => response.json())
            .then(data => {
                let resultText = data.prediction === 1 ? "Malignant (Cancerous)" : "Benign (Non-Cancerous)";
                document.getElementById("result").innerText =
                    "Prediction: " + resultText + " (Probability: " + data.probability + ")";
            })
            .catch(error => console.error("Error:", error));
        }
```

9. Conclusion

This project successfully implements a logistic regression model to predict breast cancer based on the mean radius of a tumor. The model is integrated with a FastAPI backend, using NumPy for numerical computations and Scikit-learn for dataset handling. The trained model is saved using Pickle and deployed via Uvicorn for real-time predictions.The frontend layout features a centered card with a dark-themed background, rounded corners, and a shadow effect. At the top, a bold gold header displays the title "Breast Cancer Prediction," followed by a welcoming message and a brief description of the tool. Below, a user input field for Mean Radius is provided, along with a "Predict" button, which displays the predicted cancer likelihood underneath upon submission.



BREAST CANCER PREDICTION

Welcome to Breast Cancer Prediction

This tool helps predict breast cancer likelihood based on the mean radius of the tumor. Enter the value below to receive a prediction.

Enter Mean Radius: 1.75

Prediction: Benign (Non-Cancerous) (Probability: 0.65)

ANKITA SATAPATHY, 22053400