def load\_data(filename):

    data = pd.read\_csv(filename)

    data.drop(columns=["Unnamed: 32", "id"], inplace=True)

    data["diagnosis"] = data["diagnosis"].map({"M": 1, "B": 0})

    return data

def handle\_outliers(data):

    # Separate features and target variable

    X = data.drop(columns=["diagnosis"])

    y = data["diagnosis"]

    # Calculate IQR

    Q1 = X.quantile(0.25)

    Q3 = X.quantile(0.75)

    IQR = Q3 - Q1

    # Find outlier indices

    outlier\_indices = ((X < (Q1 - 1.5 \* IQR)) | (X > (Q3 + 1.5 \* IQR))).any(axis=1)

    # Remove outliers from the features

    X\_no\_outliers = X[~outlier\_indices]

    # Remove the corresponding target values

    y\_no\_outliers = y[~outlier\_indices]

    # Concatenate the features without outliers with the target variable

    data\_no\_outliers = pd.concat([X\_no\_outliers, y\_no\_outliers.reset\_index(drop=True)], axis=1)

    return data\_no\_outliers

def handle\_class\_imbalance(data):

    smote = SMOTE(sampling\_strategy='auto', random\_state=42)

    X = data.drop(columns=["diagnosis"])

    y = data["diagnosis"]

    X\_resampled, y\_resampled = smote.fit\_resample(X, y)

    # Combine resampled features and target into one DataFrame

    resampled\_data = pd.concat([X\_resampled, y\_resampled], axis=1)

    return resampled\_data

def scale\_features(data):

    scaler = StandardScaler()

    features = data.drop(columns=["diagnosis"])

    scaled\_features = scaler.fit\_transform(features)

    return pd.DataFrame(scaled\_features, columns=features.columns)

def handle\_multicollinearity(data):

    pca = PCA(n\_components=0.95)

    transformed\_data = pca.fit\_transform(data)

    return pd.DataFrame(transformed\_data)

def main():

    data = load\_data("data.csv")

    data = handle\_class\_imbalance(data)

    data = scale\_features(data)

    data = handle\_multicollinearity(data)

    data = handle\_outliers(data)

    # Save preprocessed data to a new CSV file

    data.to\_csv("preprocessed\_cancer\_data.csv", index=False)

if \_\_name\_\_ == "\_\_main\_\_":

    main()