COSC 522 UTK

Project 1

Student Name:Richa Patel

```
import pandas as pd

# Load the data
url = 'https://gist.github.com/rhema/3b4b729d98978b9020d85c9b9e7c9dd6/raw/e0831e9
all_data = pd.read_csv(url)

# Split the data into training and testing sets
train_data = all_data.iloc[:275, :]
test_data = all_data.iloc[275:, :]

# Function to calculate prior probabilities
def calculate_priors(train_data):
    class_counts = train_data['Type'].value_counts()
    priors = class_counts / len(train_data)
    return priors

# Function to calculate likelihoods with Laplace smoothing
def calculate_likelihoods(train_data):
    likelihoods = {}
    for type_name in train_data['Type'].unique():
        type_data = train_data[train_data['Type'] == type_name]
        feature_likelihoods = {}
        for feature in ['HP', 'Attack', 'Defense']:
            feature_counts = type_data[feature].value_counts()
            unique_values = len(train_data[feature].unique())  # Count unique fea
            feature_likelihoods[feature] = {
                value: (feature_counts.get(value, 0) + 1) / (len(type_data) + uni
                for value in train_data[feature].unique()
            }
        likelihoods[type_name] = feature_likelihoods
    return likelihoods

# Function to make predictions
def predict(row, priors, likelihoods):
    probabilities = {}
    for type_name in priors.index:
        # Start with the prior probability
        probabilities[type_name] = priors[type_name]
        # Multiply by the likelihood of each feature
        for feature in ['HP', 'Attack', 'Defense']:
```

```
        for feature in ["HP", "Attack", "Defense"]:
            feature_value = row[feature]
            probabilities[type_name] *= likelihoods[type_name][feature].get(featu
    # Return the type with the highest probability
    predicted_type = max(probabilities, key=probabilities.get)
    return {"score": probabilities[predicted_type], "class": predicted_type}

# Calculate priors and likelihoods
priors = calculate_priors(train_data)
likelihoods = calculate_likelihoods(train_data)

# Use this loop to make predictions for the test data
for index, row in test_data.iterrows():
    result = predict(row, priors, likelihoods)
    print(f"Pokémon: {row['Name']}, Predicted Type: {result['class']}, Probabilit
```

```
Pokémon: Pidgeotto, Predicted Type: Normal, Probability: 0.0157
Pokémon: Zubat, Predicted Type: Water, Probability: 0.0165
Pokémon: Igglybuff, Predicted Type: Normal, Probability: 0.0248
Pokémon: Watchog, Predicted Type: Normal, Probability: 0.0120
Pokémon: Pyroar, Predicted Type: Water, Probability: 0.0212
Pokémon: Sawsbuck, Predicted Type: Flying, Probability: 0.0158
Pokémon: Suicune, Predicted Type: Water, Probability: 0.0294
Pokémon: Zangoose, Predicted Type: Normal, Probability: 0.0077
Pokémon: Azumarill, Predicted Type: Water, Probability: 0.0246
Pokémon: Squirtle, Predicted Type: Water, Probability: 0.0157
Pokémon: Togetic, Predicted Type: Water, Probability: 0.0218
Pokémon: Gliscor, Predicted Type: Flying, Probability: 0.0280
Pokémon: Noibat, Predicted Type: Water, Probability: 0.0165
Pokémon: Skiploom, Predicted Type: Water, Probability: 0.0165
Pokémon: Snorlax, Predicted Type: Flying, Probability: 0.0158
Pokémon: Simipour, Predicted Type: Flying, Probability: 0.0158
Pokémon: Chansey, Predicted Type: Normal, Probability: 0.0248
Pokémon: Butterfree, Predicted Type: Normal, Probability: 0.0157
Pokémon: Spearow, Predicted Type: Water, Probability: 0.0165
Pokémon: Delibird, Predicted Type: Water, Probability: 0.0165
Pokémon: Feraligatr, Predicted Type: Flying, Probability: 0.0280
Pokémon: Bunnelby, Predicted Type: Water, Probability: 0.0165
```