

```
In [1]: import pandas as pd
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score

In [2]: # load the dataset
df = pd.read_csv('/Users/srilathasirigala/Documents/LogisticRegressionnData/candy-data.csv')
```

```
In [3]: df.describe()
```

	chocolate	fruity	caramel	peanutyalmondy	nougat	crispedricewafer	hard	bar	pluribus	sugarpercent	pricepercent	winpercent
count	85.000000	85.000000	85.000000	85.000000	85.000000	85.000000	85.000000	85.000000	85.000000	85.000000	85.000000	85.000000
mean	0.435294	0.447059	0.164706	0.164706	0.082353	0.082353	0.176471	0.247059	0.517647	0.478647	0.468882	50.316764
std	0.498738	0.500140	0.373116	0.373116	0.276533	0.276533	0.383482	0.433861	0.502654	0.282778	0.285740	14.714357
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.011000	0.011000	22.445341
25%	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.220000	0.255000	39.141056
50%	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	1.000000	0.465000	0.465000	47.829754
75%	1.000000	1.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	1.000000	0.732000	0.651000	59.863998
max	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	0.988000	0.976000	84.180290

```
In [4]: df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 85 entries, 0 to 84
Data columns (total 13 columns):
#   Column                Non-Null Count  Dtype
---  -
0   competitorname         85 non-null    object
1   chocolate              85 non-null    int64
2   fruity                 85 non-null    int64
3   caramel                85 non-null    int64
4   peanutyalmondy         85 non-null    int64
5   nougat                 85 non-null    int64
6   crispedricewafer       85 non-null    int64
7   hard                   85 non-null    int64
8   bar                    85 non-null    int64
9   pluribus               85 non-null    int64
10  sugarpercent            85 non-null    float64
11  pricepercent            85 non-null    float64
12  winpercent              85 non-null    float64
dtypes: float64(3), int64(9), object(1)
memory usage: 8.8+ KB
```

```
In [5]: print(df["competitorname"])

0          100 Grand
1           3 Musketeers
2             One dime
3          One quarter
4           Air Heads
...
80          Twizzlers
81          Warheads
82    Welch's Fruit Snacks
83  Werther's Original Caramel
84          Whoppers
Name: competitorname, Length: 85, dtype: object
```

```
In [6]: import matplotlib.pyplot as plt

# Select the relevant columns from the DataFrame
features = ['chocolate', 'fruity', 'caramel', 'peanutyalmondy', 'nougat', 'crispedricewafer', 'hard', 'bar', 'pluribus']
data = df[features]

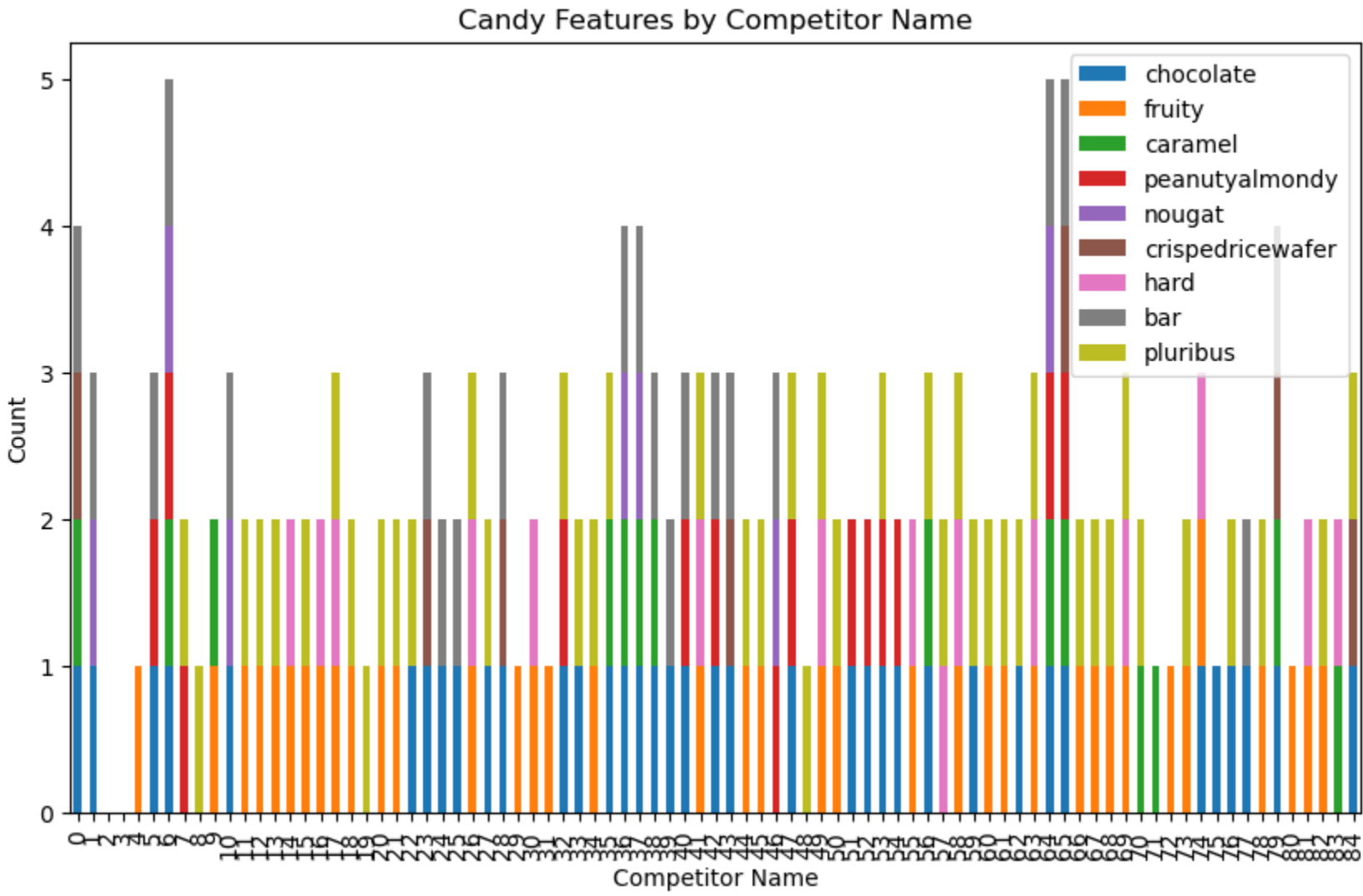
# Create the stacked bar plot
ax = data.plot(kind='bar', stacked=True, figsize=(10, 6))

# Set the x-axis label
ax.set_xlabel('Competitor Name')

# Set the y-axis label
ax.set_ylabel('Count')

# Set the title of the plot
ax.set_title('Candy Features by Competitor Name')

# Show the plot
plt.show()
```



```
In [7]: # define the independent and dependent variables
X = df.drop(['competitorname', 'winpercent'], axis=1) # independent variables
y = df['winpercent'] > 50 # dependent variable (binary classification)

# split the dataset into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)

# create the logistic regression model
model = LogisticRegression()

# fit the model to the training data
model.fit(X_train, y_train)

# make predictions on the testing data
y_pred = model.predict(X_test)

# evaluate the accuracy of the model
accuracy = accuracy_score(y_test, y_pred)
print(f'Accuracy: {accuracy:.2f}')

Accuracy: 0.69
```

```
In [8]: pickle.dump(model, open("model123.pkl", 'wb'))

-----
NameError                                Traceback (most recent call last)
/var/folders/g1/734tgsrx39q4gpdsgyvfnp840000gn/T/ipykernel_67088/545286204.py in <module>
----> 1 pickle.dump(model, open("model123.pkl", 'wb'))

NameError: name 'pickle' is not defined
```

```
In [ ]: from flask import Flask, jsonify, request, render_template
import pickle

app = Flask(__name__, template_folder='/Users/srilathasirigala/Documents/Intern/LogisticRegressionnData/Templates')
modele = pickle.load(open("model123.pkl", 'rb'))

@app.route("/")
def Home():
    return render_template("index.html")

@app.route('/predict', methods=['POST'])
def predict():
    # Get the input features from the request
    data = request.get_json()
    features = [data["feature1"], data["feature2"], data["feature3"], data["feature4"], data["feature5"], data["feature6"], data["feature7"], data["feature8"], data["feature9"], data["feature10"], data["feature11"], data["feature12"], data["feature13"]]
    # Make a prediction with the model
    prediction = modele.predict([features])[0]
    # Return the prediction as a JSON object
    return jsonify({'prediction': prediction})

# Start the app
if __name__ == "__main__":
    app.run(debug=True, port=5002, use_reloader=False)

* Serving Flask app "__main__" (lazy loading)
* Environment: production
  WARNING: This is a development server. Do not use it in a production deployment.
  Use a production WSGI server instead.
* Debug mode: on

* Running on http://127.0.0.1:5002/ (Press CTRL+C to quit)
127.0.0.1 - - [11/Apr/2023 10:20:48] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [11/Apr/2023 10:20:48] "GET /static/css/style.css HTTP/1.1" 404 -
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```