Yammy or Meh?

Logistic Regression

Prompts

- Create a new column "yammy", where “star_rating” >= 4.5, it's a 1 , otherwise 0.
- Count 0s and 1s in the "yammy" column
- Scale the data
- Build a logistic regression with "yammy" as the target

Strategy

- Create a Teacher
- Check the Balance
- Scale the Features
- Split the Data
- Train the Model
- Check the Accuracy
- Read the Confusion Matrix

Code & Results

1. Create a Teacher

This code checks the "star_rating" and puts a 1 in the new "yammy" column if it is greater than or equal to 4.5.

```python
import pandas as pd

# Assuming you already have a DataFrame named df
data['yammy'] = (data['star_rating'] >= 4.5).astype(int)
data.head()
```

yammy
0
0
1
0
1

Thus, we have the "Yammy" column.

2. Check the Balance

This code counts the number of 0s and 1s in the "Yammy" column.

```python
data['yammy'].value_counts().sort_index()
```

```
0    64
1    57
```

We have 64 "meh" tins and 57 "yummy" ones.

3. Scale the Features

This code puts features between 0 and 1.

```python
from sklearn.preprocessing import MinMaxScaler

# Select only numeric columns (excluding 'yammy' if it's a label)
features = data.drop(columns=['yammy'])  # Drop target column if needed
numeric_cols = features.select_dtypes(include=['float64', 'int64']).columns

scaler = MinMaxScaler()
data[numeric_cols] = scaler.fit_transform(data[numeric_cols])

data[numeric_cols].head()
```

	price	weight	star_rating	total_comments	adult	chicken	wet	from_ocean	company_farmina	company_me-o	company_meat	company_purepet	comp
0	0.022888	0.113797	0.60	0.638163	1.0	0.0	1.0	1.0	0.0	0.0	0.0	1.0	
1	0.128879	0.697885	0.65	0.904091	1.0	0.0	1.0	1.0	0.0	0.0	0.0	1.0	
2	0.285714	0.697885	0.75	0.620522	1.0	0.0	1.0	1.0	0.0	0.0	0.0	0.0	
3	0.059754	0.234642	0.50	0.348278	1.0	0.0	1.0	1.0	0.0	0.0	1.0	0.0	
4	0.050538	0.103726	0.75	1.000000	0.5	0.0	1.0	1.0	0.0	0.0	0.0	0.0	

4. Split the Data

This code first splits the teacher from the features and then splits the data into training and testing sets.

```python
from sklearn.model_selection import train_test_split

# Separate features and target
X = data.drop(columns=['yammy','star_rating'])
y = data['yammy']

# Split into training and test sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

5. Train the Model

This code creates a logistic regression model and trains it using the training set.

```python
from sklearn.linear_model import LogisticRegression

# Initialize and train model
model = LogisticRegression()
model.fit(X_train, y_train)
```

6. Check the Accuracy & Confusion Matrix

This code evaluates model accuracy and calculates a confusion matrix.

```python
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report

# Make predictions
y_pred = model.predict(X_test)

# Evaluation
print("Accuracy:", accuracy_score(y_test, y_pred))
print("Confusion Matrix:\n", confusion_matrix(y_test, y_pred))
print("Classification Report:\n", classification_report(y_test, y_pred))
```

```
Accuracy: 0.76
Confusion Matrix:
 [[13  2]
 [ 4  6]]
```

✅ The model got 76% of the tins right.

❌ It made some mistakes:

- It mislabeled 2 out of 15 meh tins as yammy
- It mislabeled 4 out of 10 yammy tins as meh

So it's doing well overall, but still has room to learn!

Made by: @katerynakononova

Watch the short: https://youtube.com/shorts/Uu5xhdya1oo

Website: https://katerynakononova.github.io/meowlearning/

Machine Learning: for Humans on Cats on Amazon!

📘👉 https://www.amazon.com/dp/B0CW9SFYXF