knn

June 28, 2024

```
[ ]: import numpy as np
     import pandas as pd
     from sklearn.model_selection import train_test_split
     from sklearn.neighbors import KNeighborsClassifier
     from sklearn.metrics import accuracy_score
     import matplotlib.pyplot as plt
```

```
[ ]: from google.colab import drive
     drive.mount('/content/drive')
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call
drive.mount("/content/drive", force_remount=True).
```

```
[ ]: products= pd.read_csv('/content/drive/MyDrive/ml proj/Message Group - Product.
       ↪csv')
```

```
[ ]: x=products[['MRP','SellPrice']]
     y=products['Discount']
```

```
[ ]: # Assuming 'products' is your DataFrame
     products = products.replace('#REF!', np.nan).dropna()  # Replace '#REF!' with␣
       ↪NaN and drop rows with NaN values

     x = products[['MRP', 'SellPrice']]

     # Extract numerical discount values and handle non-numerical values
     products['Discount'] = products['Discount'].str.extract('(\d+)').astype(float)
     y = products['Discount']

     k = 3
     knn = KNeighborsClassifier(n_neighbors=k)
     knn.fit(x, y)
```

```
[ ]: KNeighborsClassifier(n_neighbors=3)
```

```
[ ]: new_data = np.array([[3900,3120]])
     prediction = knn.predict(new_data)
```

```
if prediction<50:
    print("price is high")
elif prediction>50 and prediction<100:
    print("price is medium")
else:
    print("price is low")
```

```
price is high
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/base.py:439: UserWarning: X does
not have valid feature names, but KNeighborsClassifier was fitted with feature
names
  warnings.warn(
```

[ ]:

```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error

# Assuming 'x' and 'y' are already defined from your previous code
# Split the data into training and testing sets
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.2,␣
 ↪random_state=42)

# Create and fit the LinearRegression model
model = LinearRegression()
model.fit(x_train, y_train)

# Example prediction for new data
new_data = np.array([[3900, 3120]])
prediction = model.predict(new_data)
print("Prediction for new data:", prediction)
```

```
Prediction for new data: [31.59923834]
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/base.py:439: UserWarning: X does
not have valid feature names, but LinearRegression was fitted with feature names
  warnings.warn(
```