Milestone3

May 12, 2024

1 Milestone 3

2 Analyzing the Demand for Bike Sharing

Bike-sharing systems are a means of renting bicycles where the process of obtaining membership, rental, and bike return is automated via a network of kiosk locations throughout a city. Using these systems, people are able rent a bike from a one location and return it to a different place on an as-needed basis. Currently, there are over 500 bike-sharing programs around the world.

[1]:
```
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
```

[2]:
```
df = pd.read_csv('/Users/mohamedalbasuony/Downloads/bike+sharing+dataset/day.
 ↪csv')
df['dteday'] = pd.to_datetime(df['dteday'])
df.head()
```

[2]:

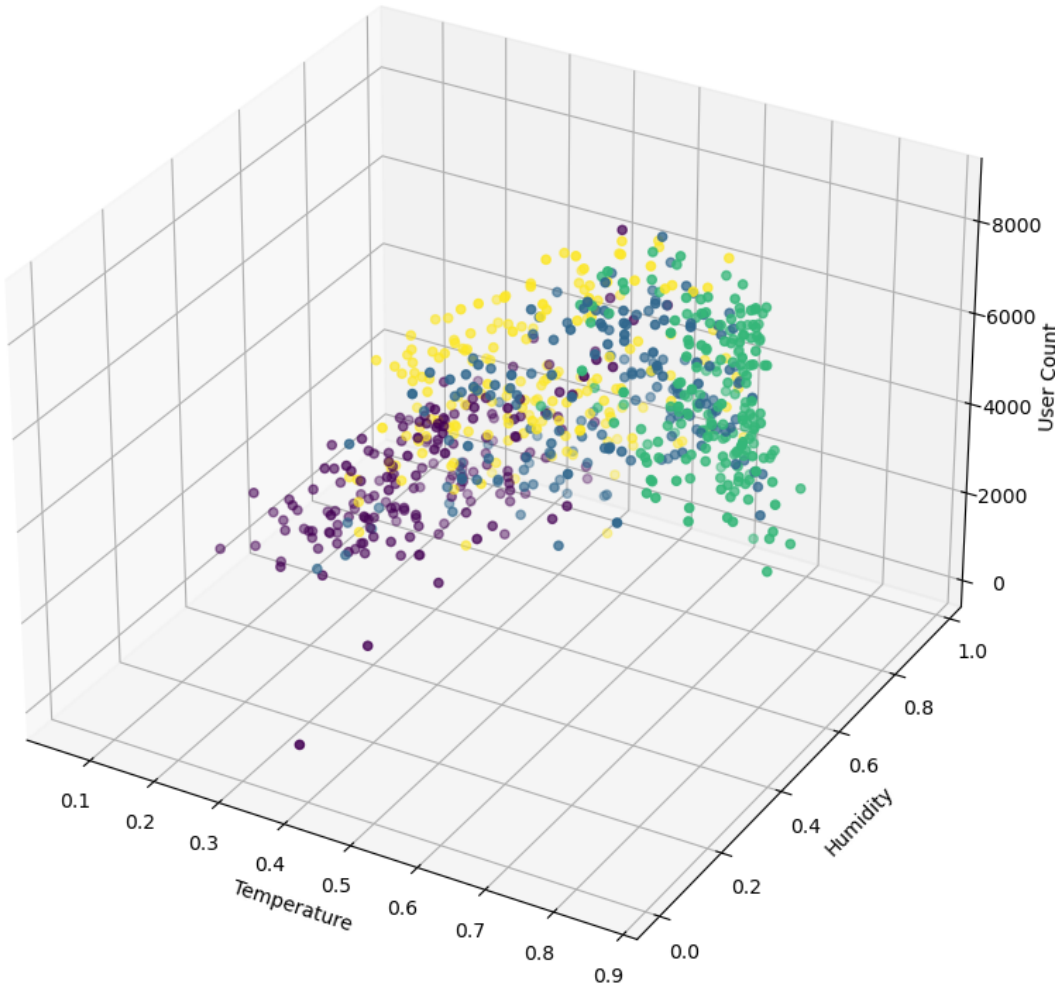
	instant	dteday	season	yr	mnth	holiday	weekday	workingday	\
0	1	2011-01-01	1	0	1	0	6	0	
1	2	2011-01-02	1	0	1	0	0	0	
2	3	2011-01-03	1	0	1	0	1	1	
3	4	2011-01-04	1	0	1	0	2	1	
4	5	2011-01-05	1	0	1	0	3	1	

	weathersit	temp	atemp	hum	windspeed	casual	registered	\
0	2	0.344167	0.363625	0.805833	0.160446	331	654	
1	2	0.363478	0.353739	0.696087	0.248539	131	670	
2	1	0.196364	0.189405	0.437273	0.248309	120	1229	
3	1	0.200000	0.212122	0.590435	0.160296	108	1454	
4	1	0.226957	0.229270	0.436957	0.186900	82	1518	

	cnt
0	985
1	801
2	1349
3	1562

```
4  1600
```

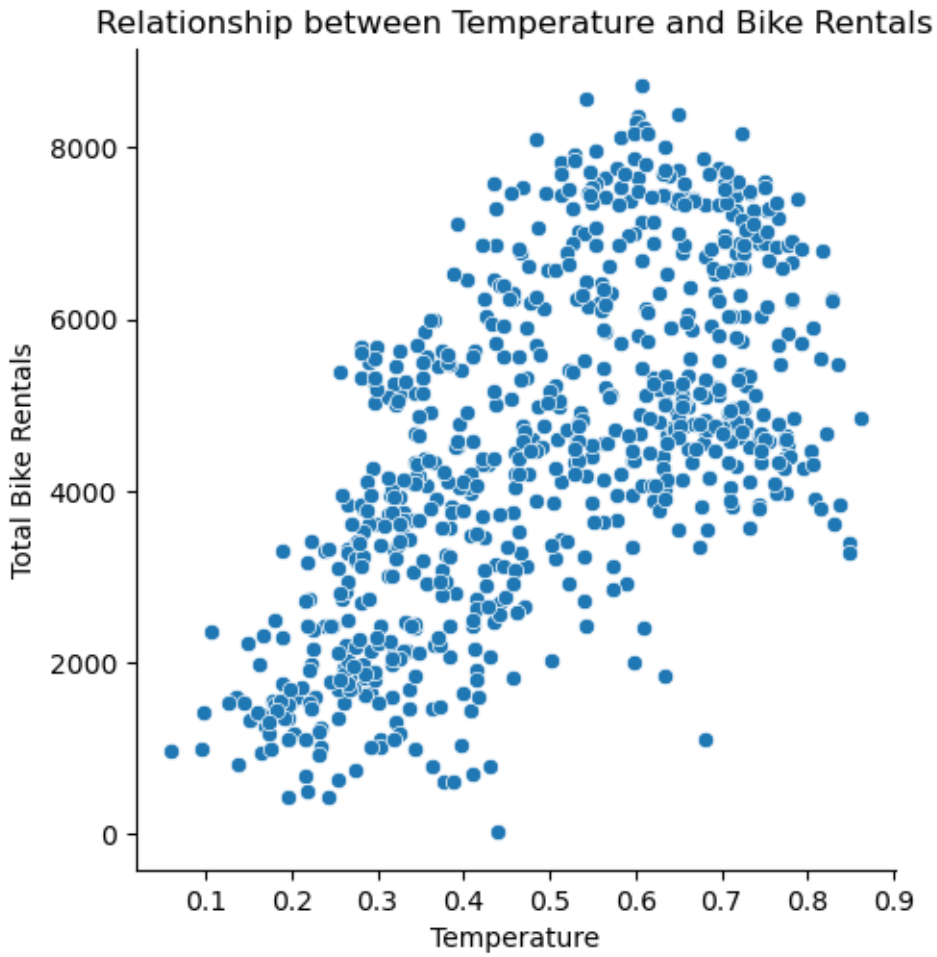
[3]:
```
from mpl_toolkits.mplot3d import Axes3D

fig = plt.figure(figsize=(10, 10))
ax = fig.add_subplot(111, projection='3d')

ax.scatter(df["temp"], df["hum"], df["cnt"], c=df["season"], cmap="viridis")

ax.set_xlabel("Temperature")
ax.set_ylabel("Humidity")
ax.set_zlabel("User Count")

plt.show()
```



This 3D scatter plot allows us to visualize the complex interplay between temperature, humidity, and user count. We can observe distinct clusters of user count based on temperature and humidity combinations. Additionally, the color-coding by season reveals seasonal variations in these relationships.

[4]:
```
# Bivariate Analysis using Seaborn's relplot:

import seaborn as sns

# Create a relplot to visualize the relationship between 'temp' and 'cnt'
sns.relplot(x="temp", y="cnt", data=df)
plt.title("Relationship between Temperature and Bike Rentals")
plt.xlabel("Temperature")
plt.ylabel("Total Bike Rentals")
plt.show()
```

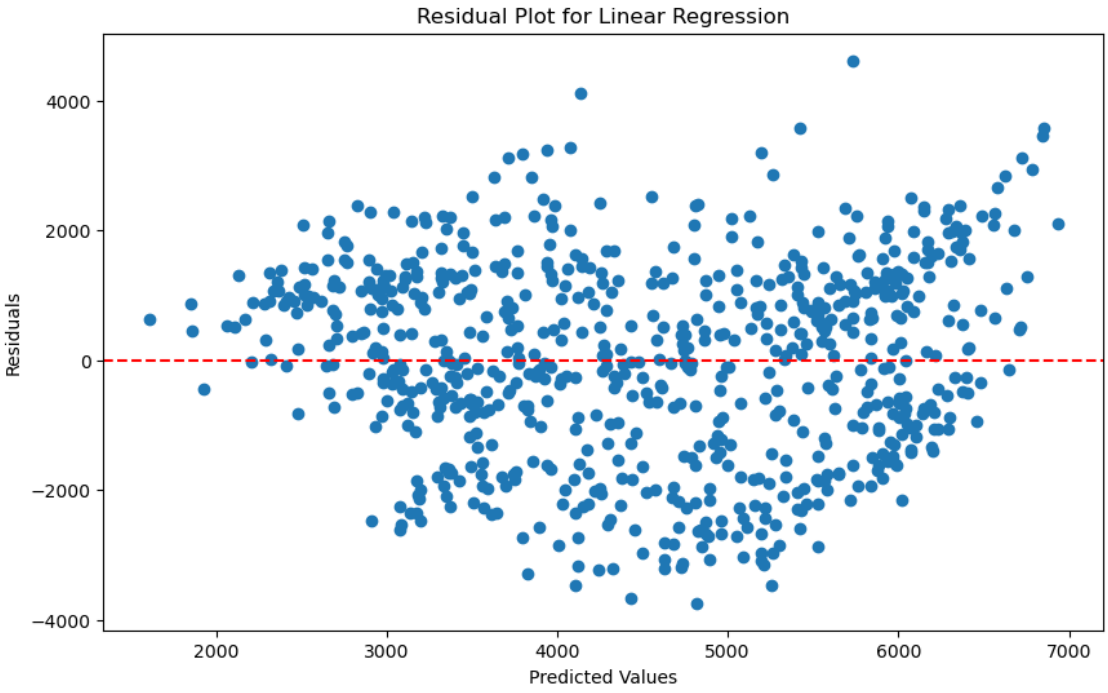
```
/Users/mohamedalbasuony/anaconda3/lib/python3.11/site-
packages/seaborn/axisgrid.py:118: UserWarning: The figure layout has changed to
tight
  self._figure.tight_layout(*args, **kwargs)
```



This relplot helps us understand the relationship between temperature and the total number of bike rentals. We can see a positive correlation, indicating that as the temperature increases, the number of bike rentals also tends to increase.

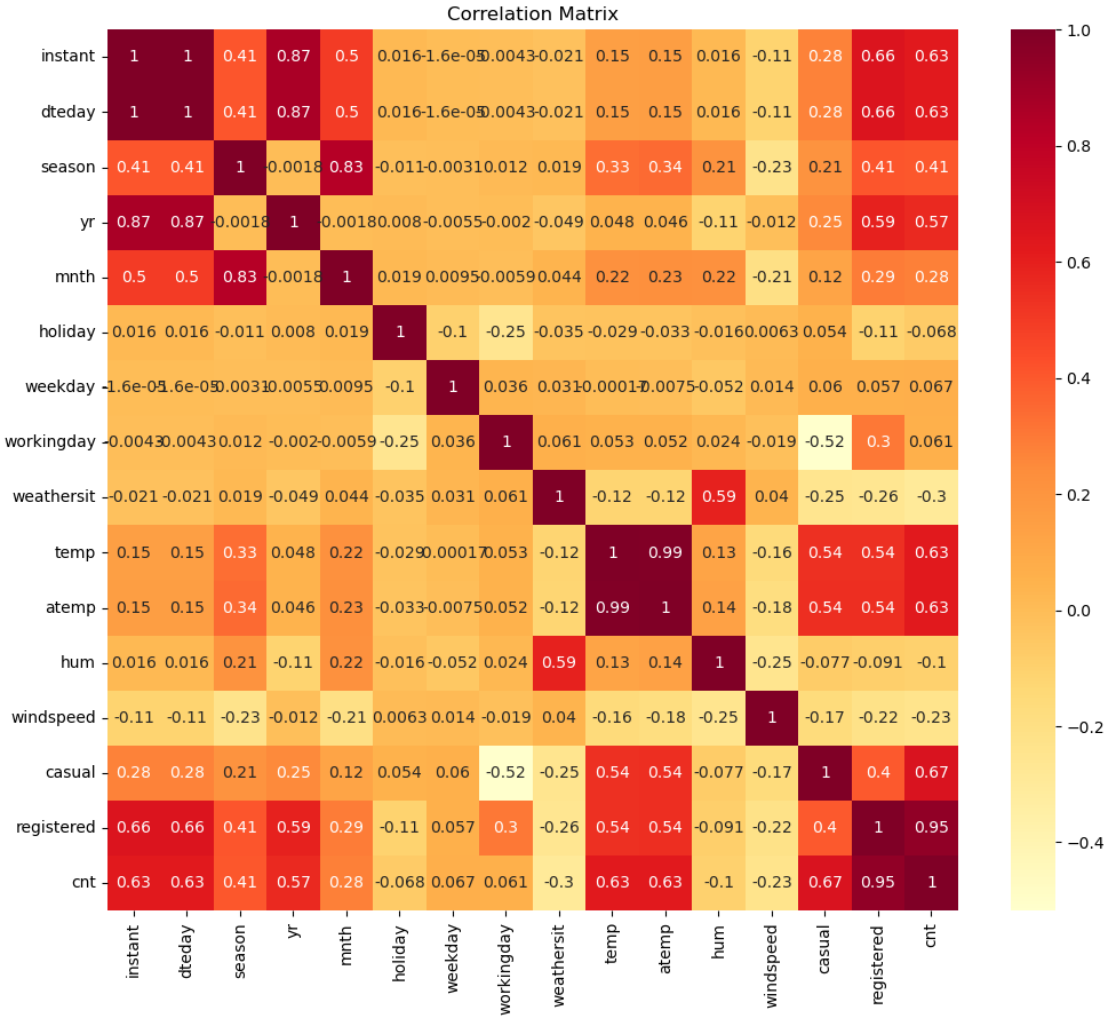
[5]:
```
#Residual Plot for Linear Regression:
from sklearn.linear_model import LinearRegression

# Fit a linear regression model
X = df[['temp']]
y = df['cnt']
model = LinearRegression()
model.fit(X, y)

# Create a residual plot
plt.figure(figsize=(10, 6))
plt.scatter(model.predict(X), model.predict(X) - y)
plt.axhline(y=0, color='red', linestyle='--')
```

```
plt.title("Residual Plot for Linear Regression")
plt.xlabel("Predicted Values")
plt.ylabel("Residuals")
plt.show()
```



The residual plot helps us assess the validity of the linear regression model. The random scatter of the residuals around the horizontal line at 0 suggests that the model is a good fit for the data.

[6]:
```
# Create a correlation matrix
corr_matrix = df.corr()

# Plot the correlation matrix using Seaborn's heatmap
plt.figure(figsize=(12, 10))
sns.heatmap(corr_matrix, annot=True, cmap='YlOrRd')
plt.title("Correlation Matrix")
plt.show()
```

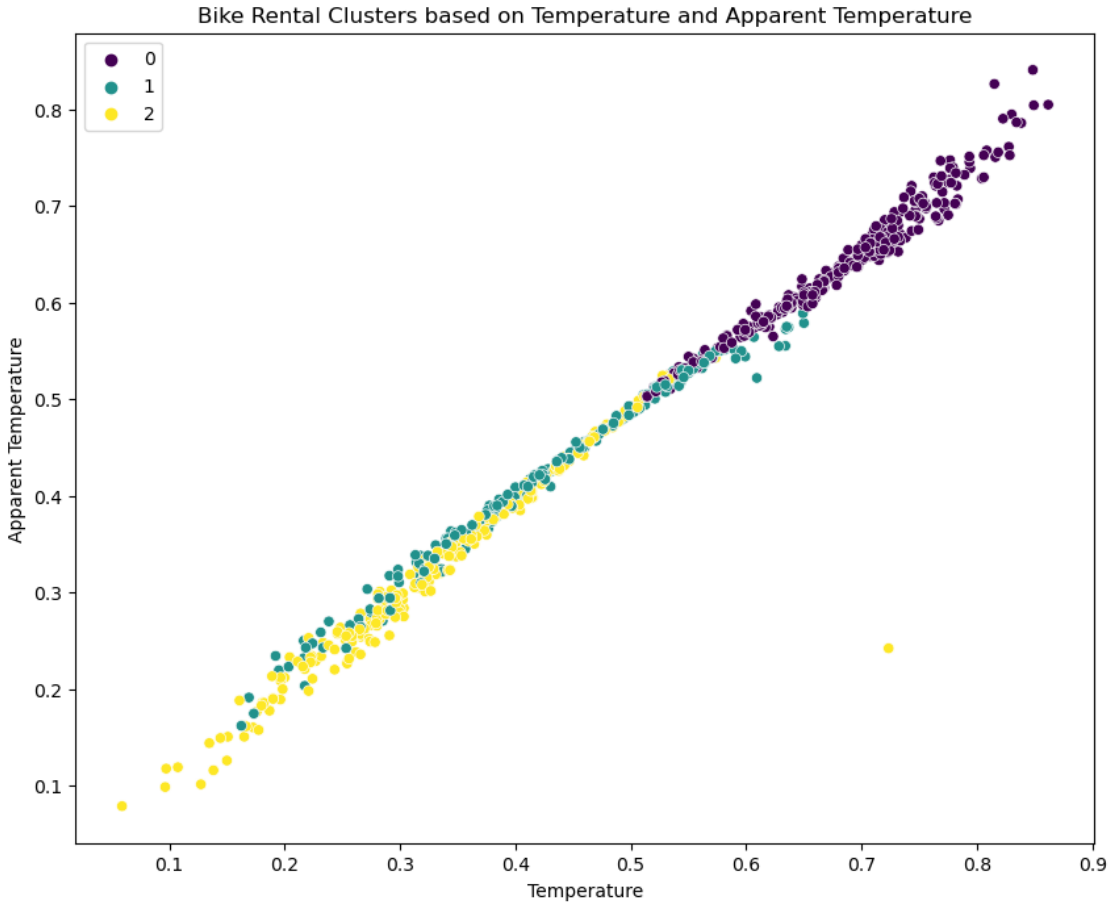


This correlation matrix provides insights into the relationships between different variables in the dataset. It helps us identify which variables are strongly correlated with the target variable 'cnt'.

[7]:
```
#Clustering Analysis using K-Means:

from sklearn.cluster import KMeans

# Standardize the data
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
X_scaled = scaler.fit_transform(df[['temp', 'atemp', 'hum', 'windspeed']])

# Perform K-Means clustering
kmeans = KMeans(n_clusters=3, random_state=42)
labels = kmeans.fit_predict(X_scaled)
```

```
# Visualize the clusters using Seaborn's scatterplot
plt.figure(figsize=(10, 8))
sns.scatterplot(x='temp', y='atemp', data=df, hue=labels, palette='viridis')
plt.title("Bike Rental Clusters based on Temperature and Apparent Temperature")
plt.xlabel("Temperature")
plt.ylabel("Apparent Temperature")
plt.show()
```

```
/Users/mohamedalbasuony/anaconda3/lib/python3.11/site-
packages/sklearn/cluster/_kmeans.py:1412: FutureWarning: The default value of
`n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_init`
explicitly to suppress the warning
  super()._check_params_vs_input(X, default_n_init=10)
```



This clustering analysis groups the bike rental data based on the temperature, apparent temperature, humidity, and windspeed. The resulting clusters can be used to better understand the patterns in bike rental demand under different environmental conditions.