

CO2 Emisyon Analizi ve Regresyon Modelleri ile Tahmin

Projenin Amacı

Bu proje, otomobillerin **CO2 emisyonlarını** etkileyen faktörleri analiz etmek ve tahmin etmek için geliştirilmiştir. Proje kapsamında veri görselleştirme, normalizasyon ve model eğitimi algoritmalarının performanslarını değerlendirme adımları uygulanmıştır. Amacımız, *motor hacmi*, *yakit tüketimi* ve *araç özellikleri* gibi değişkenlerin CO2 emisyonlarına etkisini anlamak ve bu emisyonları en iyi şekilde tahmin edebilen modeli belirlemektir.

```
In [46]: import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.preprocessing import MinMaxScaler
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.tree import DecisionTreeRegressor
from sklearn.ensemble import RandomForestRegressor, GradientBoostingRegressor
from sklearn.svm import SVR
from sklearn.neural_network import MLPRegressor
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score
```

```
In [47]: # Veri yükleme ve kontroller

data = pd.read_csv("co2.csv")
print(data.head())
print("*" * 45)
print(f"Veri setinin boyutu: {data.shape}")
print("*" * 45)
print(data.describe())
print("*" * 45)
print("Eksik Veri Sayıları:")
print(data.isnull().sum())
print("*" * 45)
```

	Make	Model	Vehicle Class	Engine Size(L)	Cylinders	Transmission	\
0	ACURA	ILX	COMPACT	2.0	4	AS5	
1	ACURA	ILX	COMPACT	2.4	4	M6	
2	ACURA	ILX HYBRID	COMPACT	1.5	4	AV7	
3	ACURA	MDX 4WD	SUV - SMALL	3.5	6	AS6	
4	ACURA	RDX AWD	SUV - SMALL	3.5	6	AS6	

	Fuel Type	Fuel Consumption City (L/100 km)	\
0	Z	9.9	
1	Z	11.2	
2	Z	6.0	
3	Z	12.7	
4	Z	12.1	

	Fuel Consumption Hwy (L/100 km)	Fuel Consumption Comb (L/100 km)	\
0	6.7	8.5	
1	7.7	9.6	
2	5.8	5.9	
3	9.1	11.1	
4	8.7	10.6	

	Fuel Consumption Comb (mpg)	CO2 Emissions(g/km)	
0	33	196	
1	29	221	
2	48	136	
3	25	255	
4	27	244	

Veri setinin boyutu: (7385, 12)

	Engine Size(L)	Cylinders	Fuel Consumption City (L/100 km)	\
count	7385.000000	7385.000000	7385.000000	
mean	3.160068	5.615030	12.556534	
std	1.354170	1.828307	3.500274	
min	0.900000	3.000000	4.200000	
25%	2.000000	4.000000	10.100000	
50%	3.000000	6.000000	12.100000	
75%	3.700000	6.000000	14.600000	
max	8.400000	16.000000	30.600000	

	Fuel Consumption Hwy (L/100 km)	Fuel Consumption Comb (L/100 km)	\
count	7385.000000	7385.000000	
mean	9.041706	10.975071	
std	2.224456	2.892506	
min	4.000000	4.100000	
25%	7.500000	8.900000	
50%	8.700000	10.600000	
75%	10.200000	12.600000	
max	20.600000	26.100000	

	Fuel Consumption Comb (mpg)	CO2 Emissions(g/km)	
count	7385.000000	7385.000000	
mean	27.481652	250.584699	
std	7.231879	58.512679	
min	11.000000	96.000000	
25%	22.000000	208.000000	
50%	27.000000	246.000000	
75%	32.000000	288.000000	
max	69.000000	522.000000	

Eksik Veri Sayıları:

Make	0
Model	0
Vehicle Class	0
Engine Size(L)	0
Cylinders	0
Transmission	0
Fuel Type	0
Fuel Consumption City (L/100 km)	0
Fuel Consumption Hwy (L/100 km)	0

```
Fuel Consumption Comb (L/100 km)      0
Fuel Consumption Comb (mpg)            0
CO2 Emissions(g/km)                  0
dtype: int64
*****
```

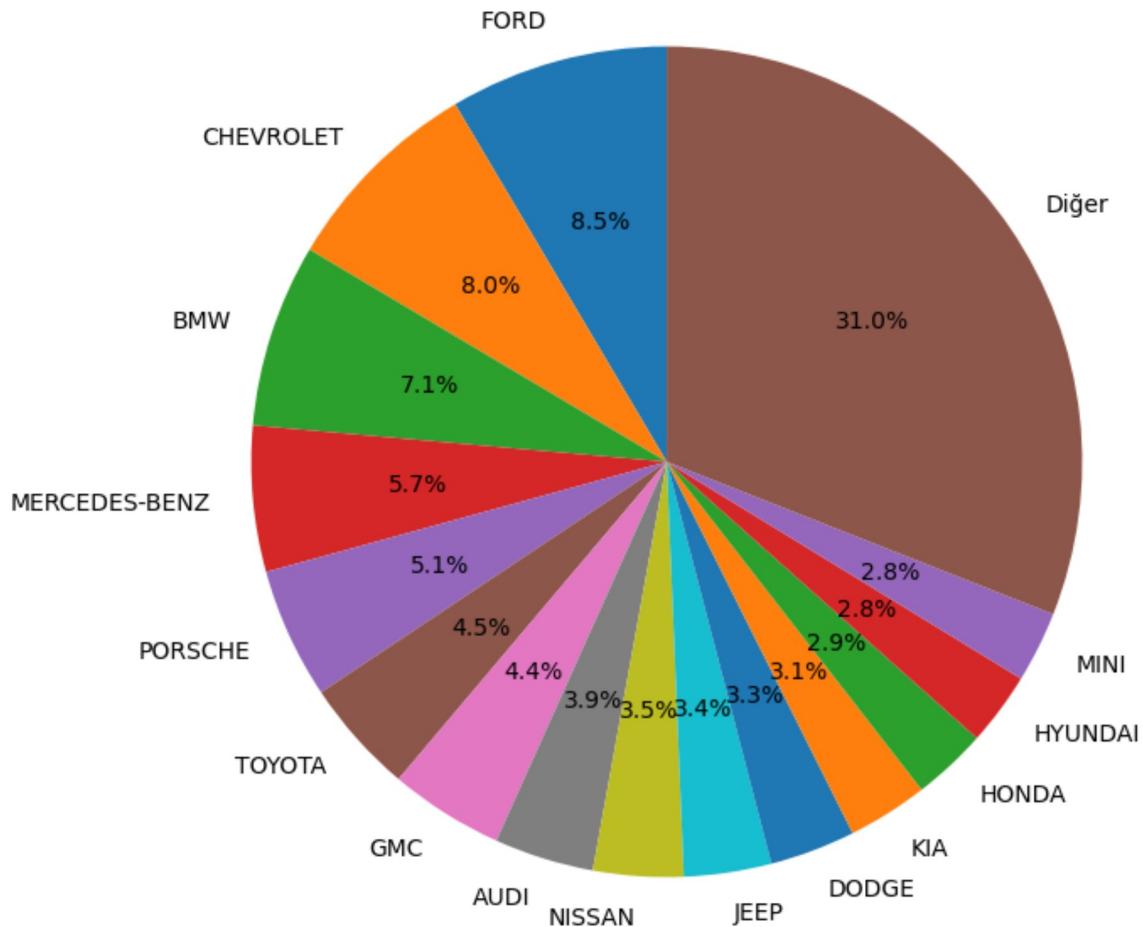
```
In [48]: # Eksik veri oranlarını görmek için:
print("Eksik Veri Oranları (%):")
print((data.isnull().mean() * 100).round(2))
```

```
Eksik Veri Oranları (%):
Make                      0.0
Model                     0.0
Vehicle Class              0.0
Engine Size(L)              0.0
Cylinders                  0.0
Transmission                0.0
Fuel Type                   0.0
Fuel Consumption City (L/100 km) 0.0
Fuel Consumption Hwy (L/100 km) 0.0
Fuel Consumption Comb (L/100 km) 0.0
Fuel Consumption Comb (mpg)    0.0
CO2 Emissions(g/km)          0.0
dtype: float64
```

```
In [49]: # İlk 15 markayı al, diğerlerini 'Diğer' olarak topla
marka_15 = data['Make'].value_counts().head(15)
diger_markalar = data['Make'].value_counts().iloc[15:].sum()

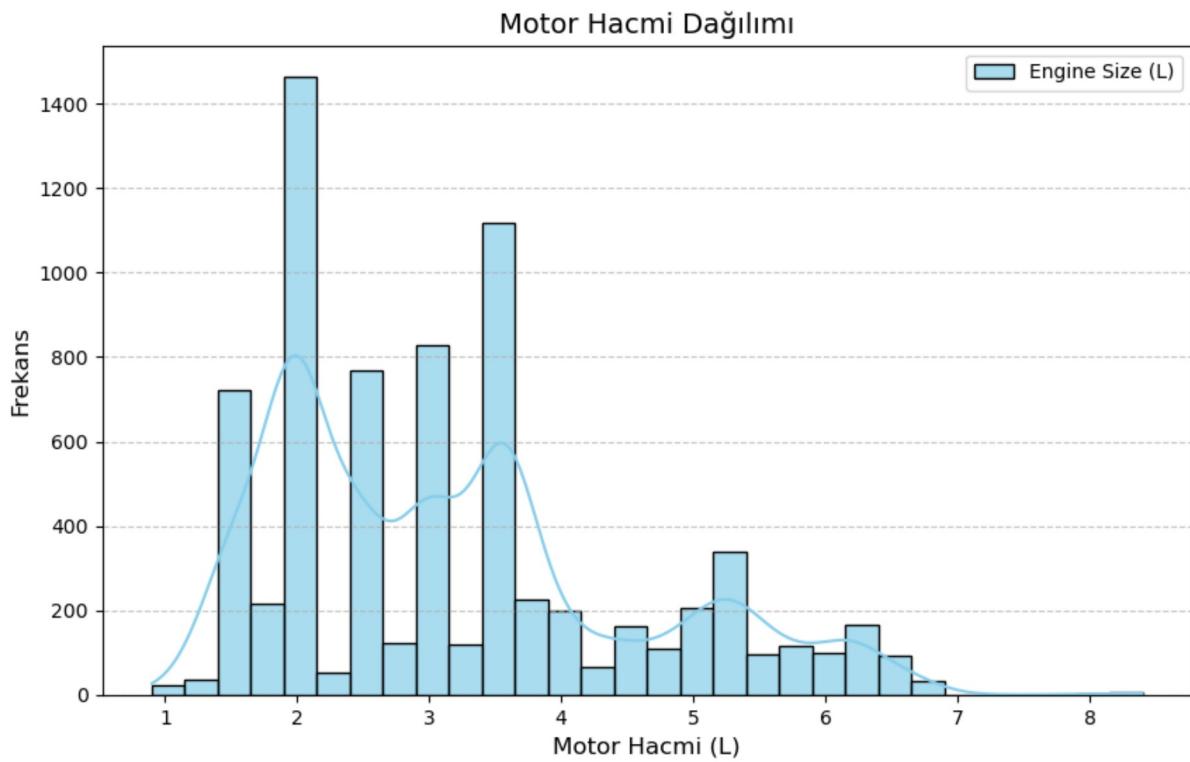
top_10_makes = pd.concat([marka_15, pd.Series({'Diğer': diger_markalar})])
plt.figure(figsize=(10, 8))
plt.pie(top_10_makes, labels=top_10_makes.index, autopct='%1.1f%%', startangle=90)
plt.title('Araba Markaları')
plt.show()
```

Araba Markaları



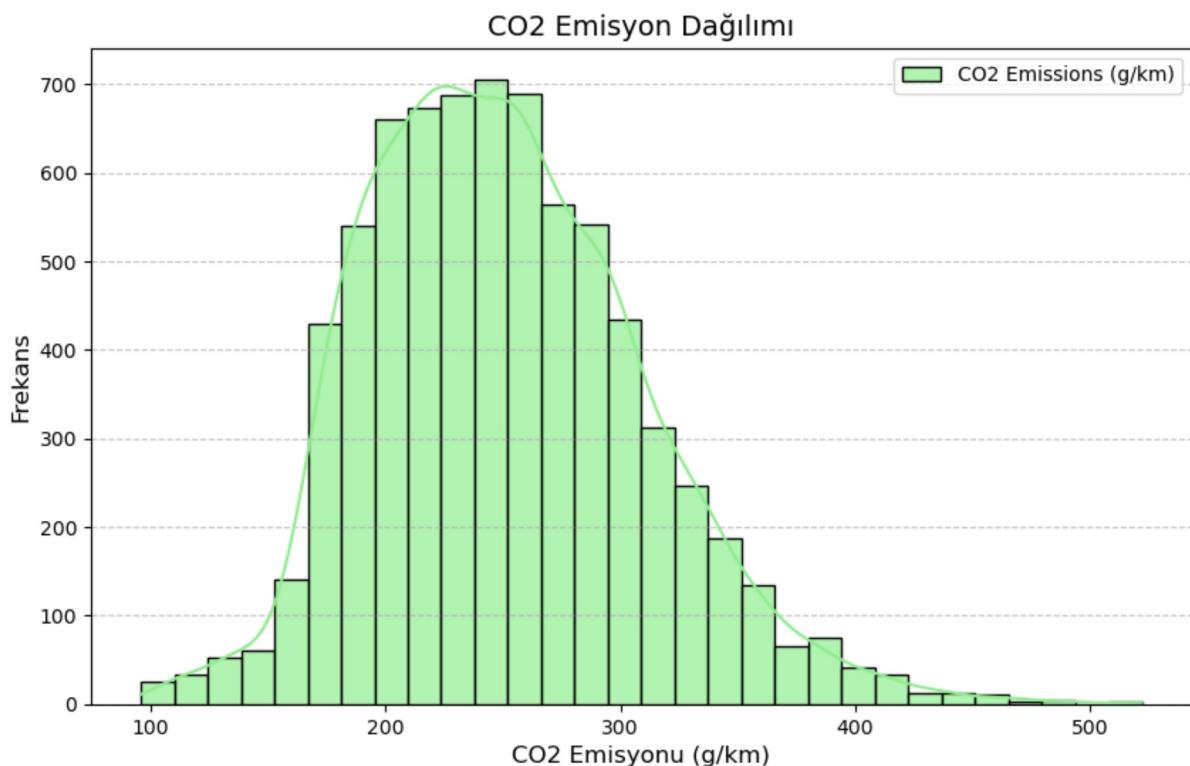
Bu görselleştirme, veri setindeki otomobil markalarının dağılımını analiz etmektedir. İlk 15 marka, araç sayısına göre sıralanmış ve ayrı ayrı gösterilmiştir. Geriye kalan tüm markalar bir araya getirilerek "Diğer" kategorisi altında toplanmıştır.

```
In [50]: plt.figure(figsize=(10, 6))
sns.histplot(data['Engine Size(L)'], kde=True, bins=30, color='skyblue', alpha=0.7, label='Engin
plt.title('Motor Hacmi Dağılımı', fontsize=14)
plt.xlabel('Motor Hacmi (L)', fontsize=12)
plt.ylabel('Frekans', fontsize=12)
plt.grid(axis='y', linestyle='--', alpha=0.7)
plt.legend()
plt.show()
```



Bu görselleştirme, araçların motor hacmi (litre cinsinden) dağılımını göstermektedir. Motor hacimlerinin yoğun olduğu aralıklar, veri setindeki araçların ortalama motor büyüklüklerini anlamak için önemlidir.

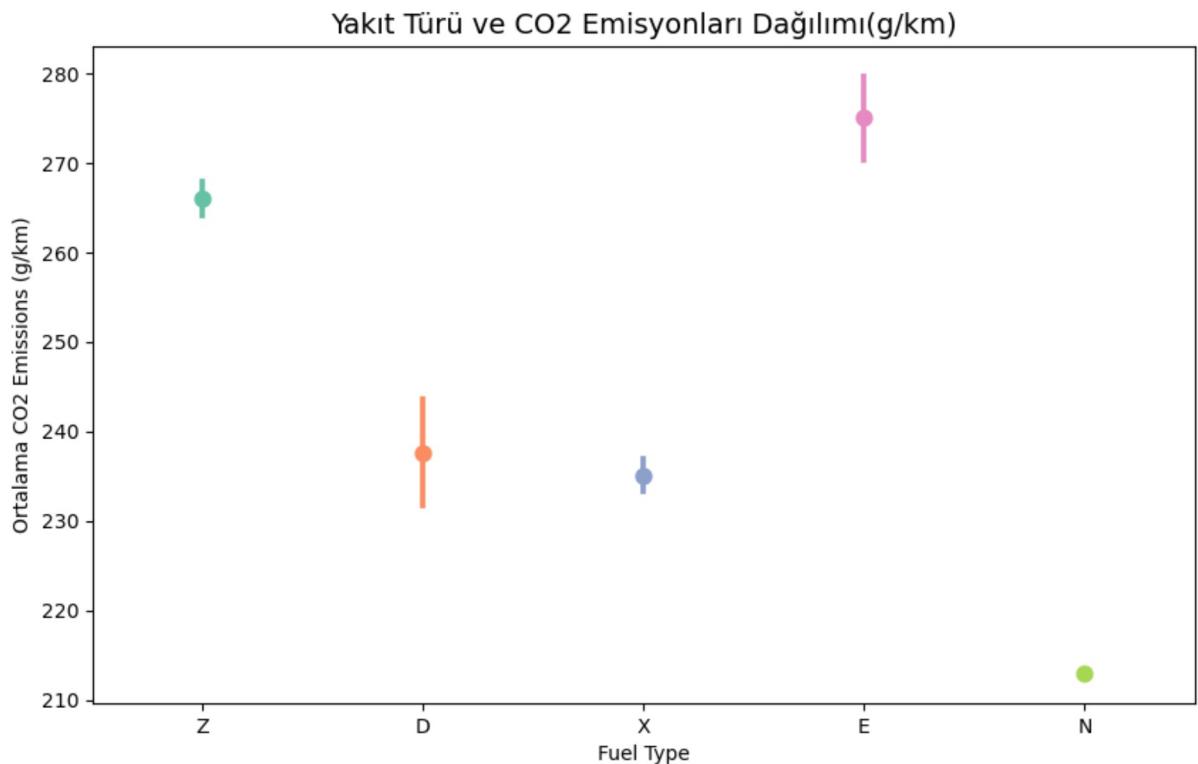
```
In [51]: # CO2 Emisyon Dağılımı
plt.figure(figsize=(10, 6))
sns.histplot(data['CO2 Emissions(g/km)'], kde=True, bins=30, color='lightgreen', alpha=0.7, label='CO2 Emission Distribution')
plt.title('CO2 Emisyon Dağılımı', fontsize=14)
plt.xlabel('CO2 Emisyonu (g/km)', fontsize=12)
plt.ylabel('Frekans', fontsize=12)
plt.grid(axis='y', linestyle='--', alpha=0.7)
plt.legend()
plt.show()
```



Bu görselleştirme, araçların CO2 emisyonlarının (g/km cinsinden) dağılımını göstermektedir. Grafik, araçların

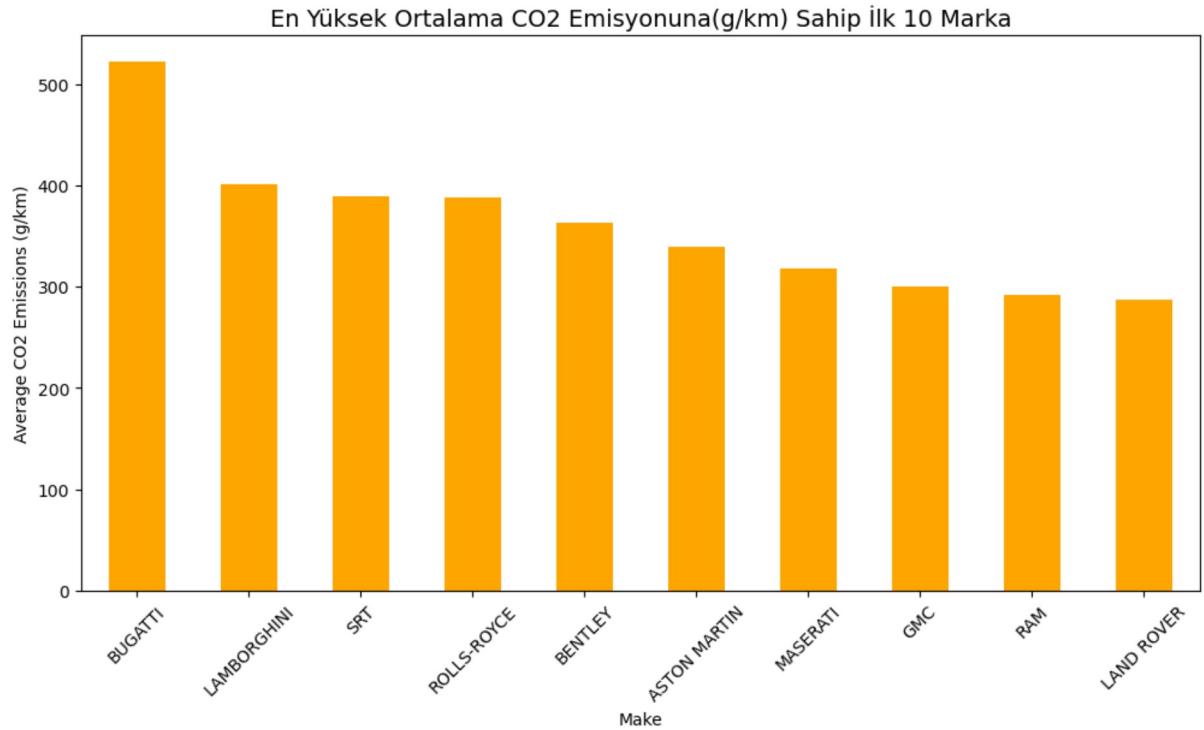
çoğunlukla hangi CO₂ emisyon seviyelerinde bulunduğu analiz etmek için kullanılır ve çevresel etkiler açısından önemli bilgiler sağlar.

```
In [52]: plt.figure(figsize=(10, 6))
sns.pointplot(x='Fuel Type', y='CO2 Emissions(g/km)', data=data, hue='Fuel Type', palette="Set2"
plt.title("Yakıt Türü ve CO2 Emisyonları Dağılımı(g/km)", fontsize=14)
plt.xlabel("Fuel Type")
plt.ylabel("Ortalama CO2 Emissions (g/km)")
plt.show()
```



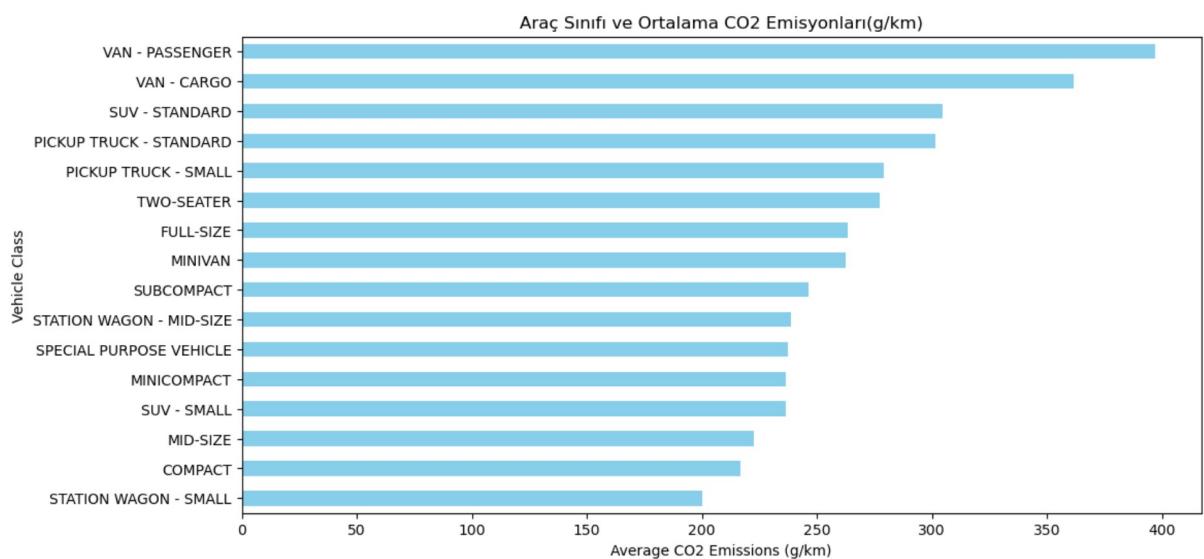
Bu görselleştirme, farklı yakıt türlerine göre araçların ortalama CO₂ emisyonlarını (g/km cinsinden) göstermektedir. Pointplot, her yakıt türü için CO₂ emisyonlarının ortalamasını işaretleyerek, hangi yakıt türlerinin çevreye daha fazla karbon salınımına yol açtığını görselleştirir.

```
In [53]: arac_markalari = data.groupby('Make')[['CO2 Emissions(g/km)']].mean().sort_values(ascending=False)
plt.figure(figsize=(12, 6))
arac_markalari.plot(kind='bar', color='orange')
plt.title("En Yüksek Ortalama CO2 Emisyonuna(g/km) Sahip İlk 10 Marka", fontsize=14)
plt.xlabel("Make")
plt.ylabel("Average CO2 Emissions (g/km)")
plt.xticks(rotation=45)
plt.show()
```



Bu görselleştirme, araç markalarına göre ortalama CO₂ emisyonlarının (g/km cinsinden) sıralamasını göstermektedir. En yüksek ortalama emisyon sahibi ilk 10 marka, bar grafiğiyle görselleştirilmiştir. Bu grafik, hangi markaların daha yüksek karbon salımına sahip olduğunu göstererir.

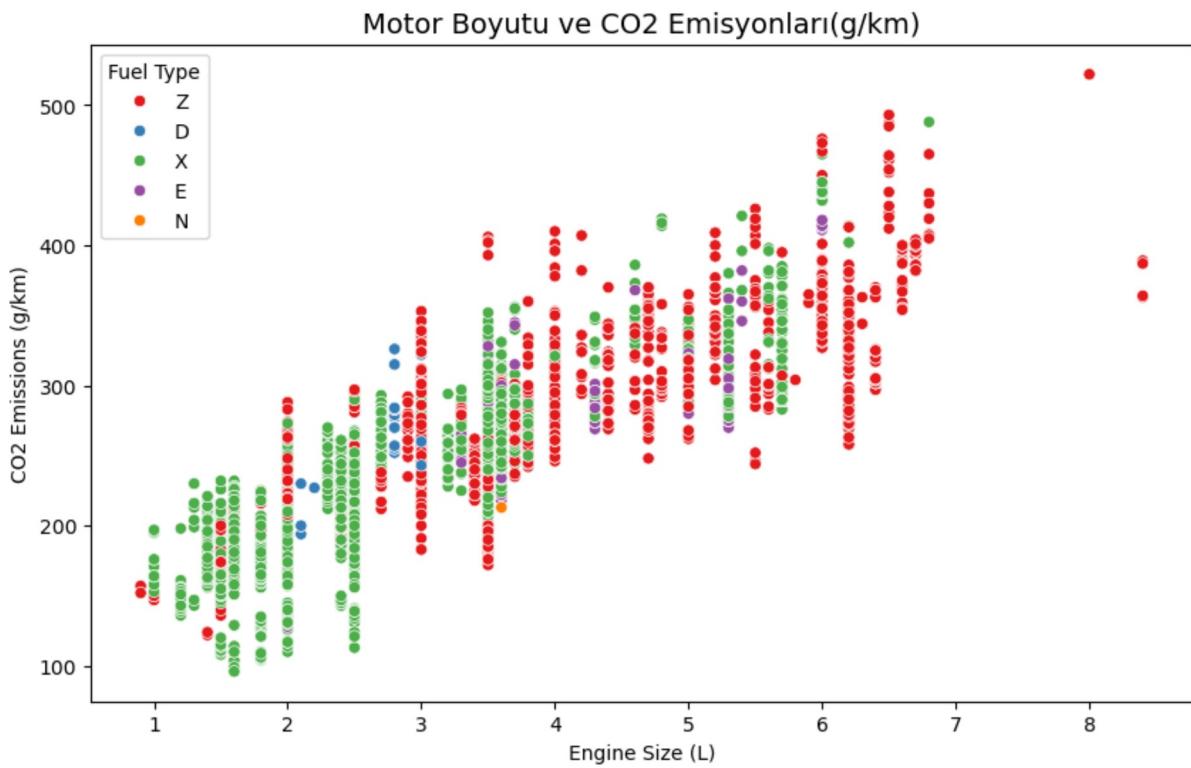
```
In [54]: arac_sinifi = data.groupby('Vehicle Class')['CO2 Emissions(g/km)'].mean().sort_values()
plt.figure(figsize=(12, 6))
arac_sinifi.plot(kind='barh', color='skyblue')
plt.title("Araç Sınıfı ve Ortalama CO2 Emisyonları(g/km)", fontsize=12)
plt.xlabel("Average CO2 Emissions (g/km)")
plt.ylabel("Vehicle Class")
plt.show()
```



Bu görselleştirme, araç sınıflarına göre ortalama CO₂ emisyonlarını (g/km cinsinden) göstermektedir. Bar grafik, farklı araç sınıflarının çevresel etkilerini karşılaştırarak hangi sınıfların daha fazla karbon salımına yol açtığını açıkça ortaya koyar.

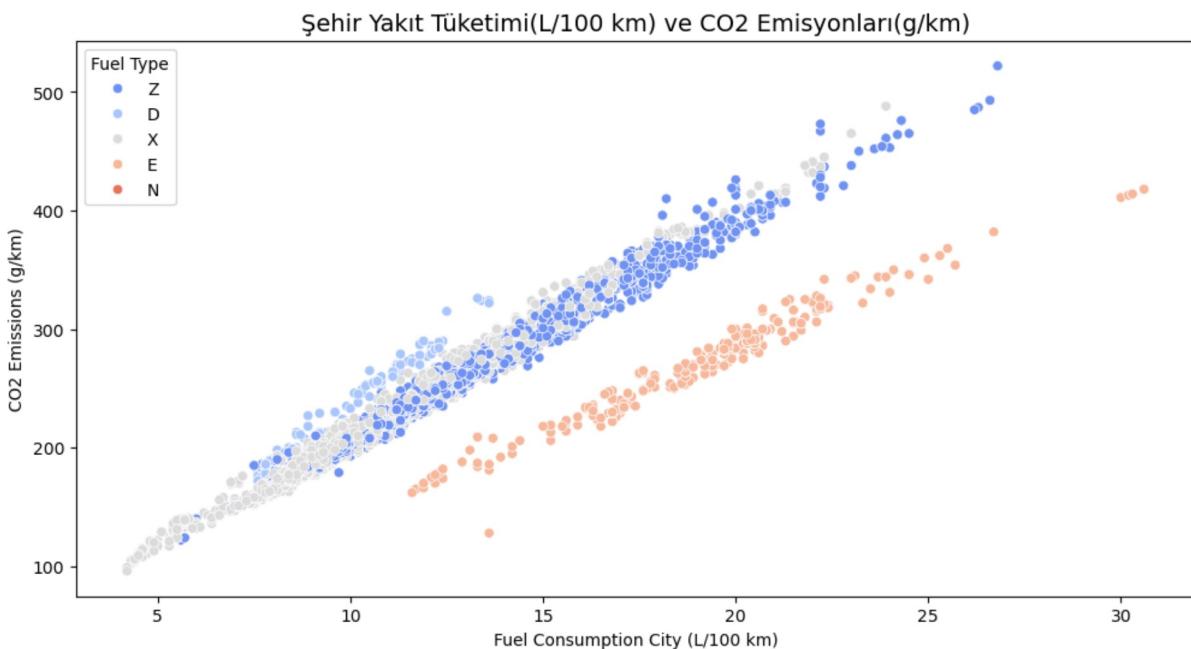
```
In [55]: plt.figure(figsize=(10, 6))
sns.scatterplot(x='Engine Size(L)', y='CO2 Emissions(g/km)', hue='Fuel Type', data=data, palette='viridis')
plt.title("Motor Boyutu ve CO2 Emisyonları(g/km)", fontsize=14)
plt.xlabel("Engine Size (L)")
plt.ylabel("CO2 Emissions (g/km)")
```

```
plt.legend(title="Fuel Type")
plt.show()
```



Bu görselleştirme, motor hacmi (L) ile CO₂ emisyonları (g/km) arasındaki ilişkiyi incelemektedir. Nokta grafiği, motor boyutunun CO₂ emisyonlarına nasıl etki ettiğini gösterirken, renkler aracılığıyla farklı yakıt türlerinin emisyon üzerindeki etkisi de vurgulanmaktadır. Bu analiz, motor hacminin büyümesiyle genellikle CO₂ emisyonlarının arttığını, ancak yakıt türünün bu ilişkide belirleyici bir faktör olabileceğini gözler önüne seriyor.

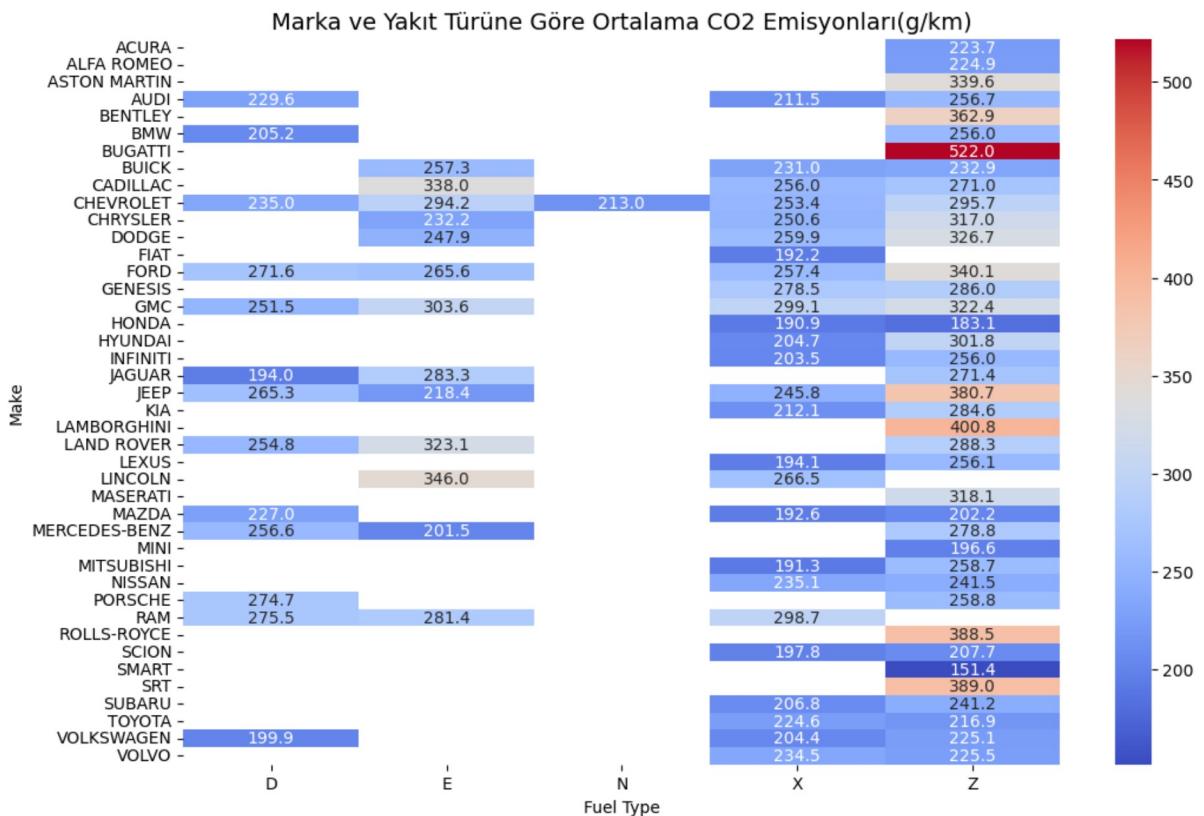
```
In [56]: plt.figure(figsize=(12, 6))
sns.scatterplot(x='Fuel Consumption City (L/100 km)', y='CO2 Emissions(g/km)', hue='Fuel Type',
plt.title("Şehir Yakıt Tüketimi(L/100 km) ve CO2 Emisyonları(g/km)", fontsize=14)
plt.xlabel("Fuel Consumption City (L/100 km)")
plt.ylabel("CO2 Emissions (g/km)")
plt.legend(title="Fuel Type")
plt.show()
```



Bu grafik, şehir içi yakıt tüketimi (L/100 km) ile CO₂ emisyonları (g/km) arasındaki ilişkiyi gösterir. Farklı

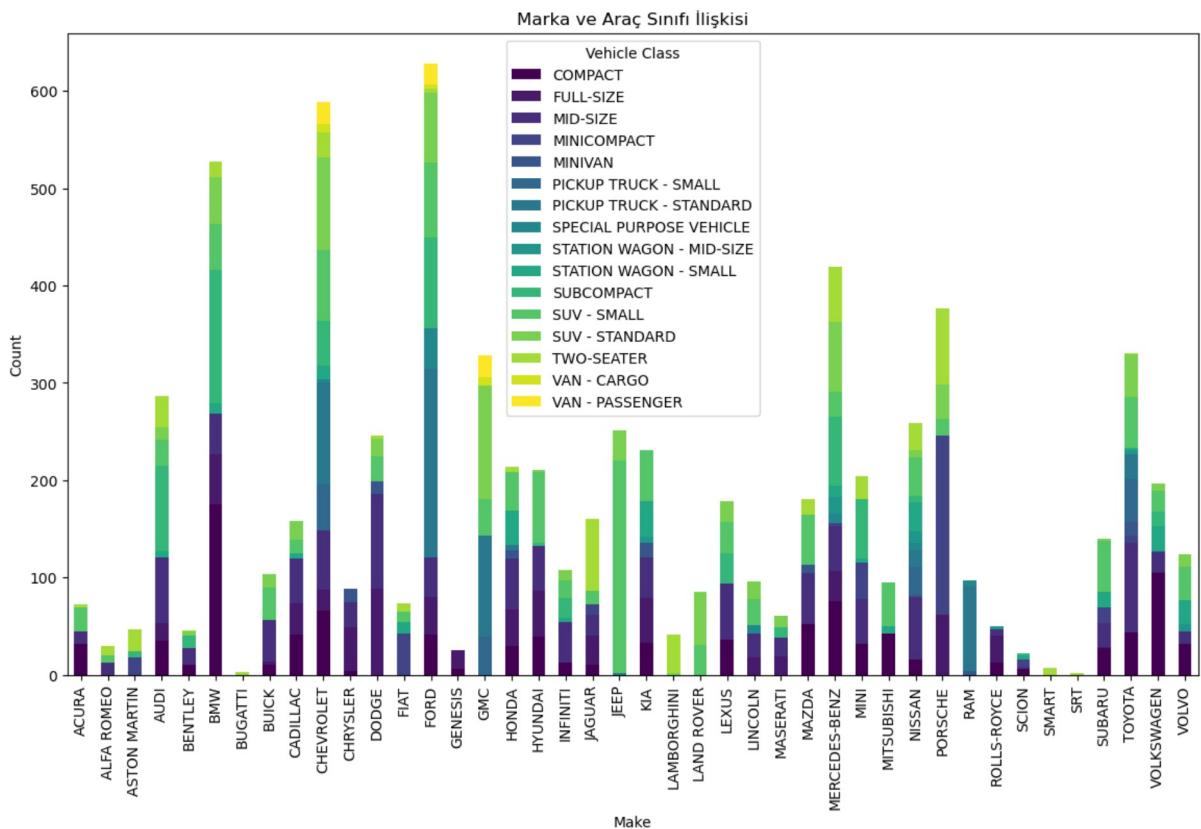
renkler aracılığıyla yakıt türleri vurgulanmakta olup, şehir içi yakıt tüketiminin artışıyla CO₂ emisyonlarının da genellikle arttığı gözlemlenmektedir.

```
In [57]: fuel_make_co2 = data.pivot_table(values='CO2 Emissions(g/km)', index='Make', columns='Fuel Type'
plt.figure(figsize=(12, 8))
sns.heatmap(fuel_make_co2, cmap='coolwarm', annot=True, fmt='.1f')
plt.title("Marka ve Yakıt Türüne Göre Ortalama CO2 Emisyonları(g/km)", fontsize=14)
plt.xlabel("Fuel Type")
plt.ylabel("Make")
plt.show()
```



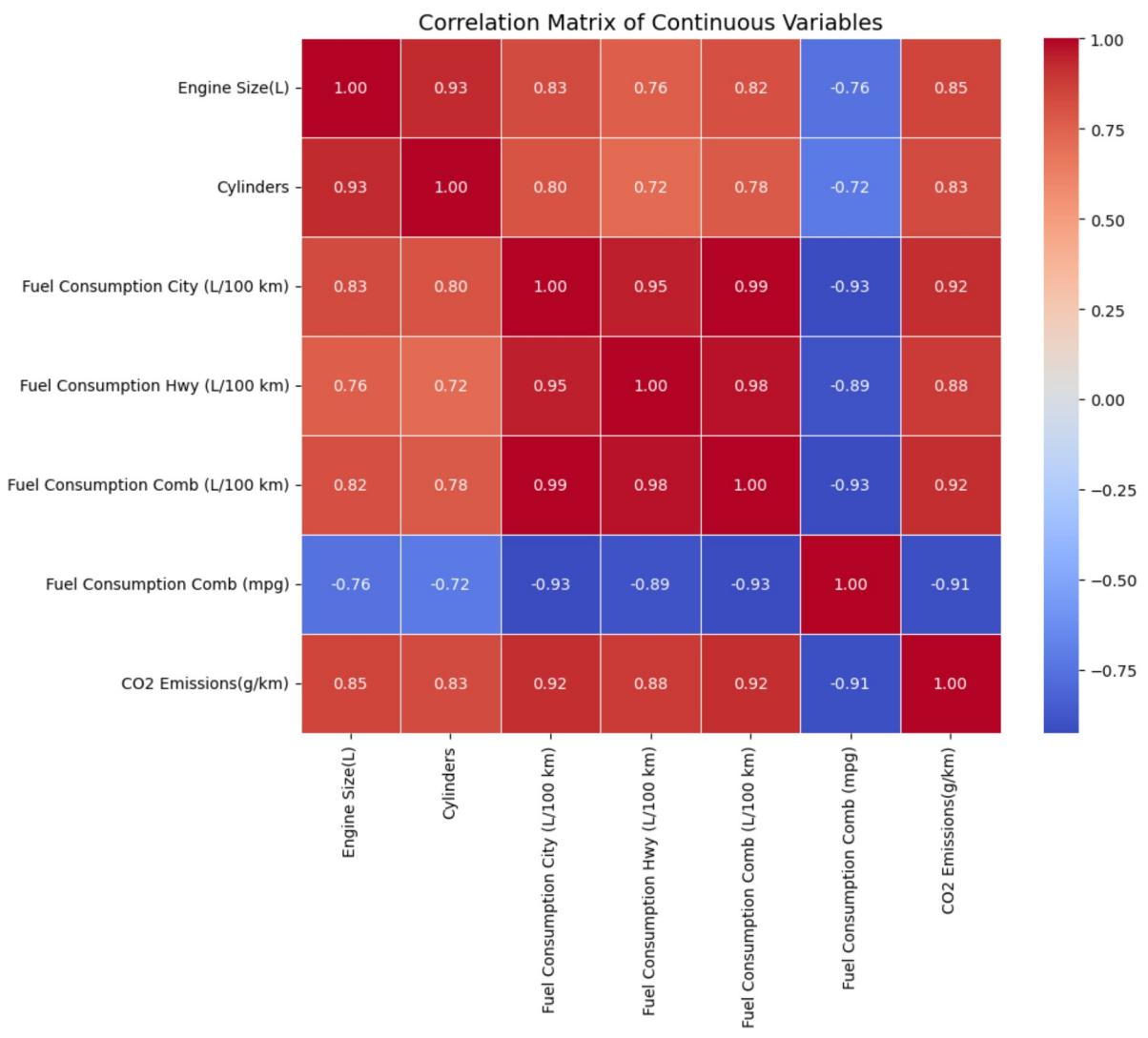
Bu ısı haritası, araba markalarına ve yakıt türlerine göre ortalama CO₂ emisyonlarını (g/km) gösterir. Farklı renkler, emisyon seviyelerini temsил etmekte olup, belirli markalar ve yakıt türlerinin CO₂ emisyonları arasındaki farkları görsel olarak vurgular. İşı haritası, her markanın farklı yakıt türlerine göre ortalama emisyon düzeylerini karşılaştırmaya olanak sağlar.

```
In [58]: veri_arac_marka = pd.crosstab(data['Make'], data['Vehicle Class'])
veri_arac_marka.plot(kind='bar', stacked=True, figsize=(14, 8), colormap='viridis')
plt.title('Marka ve Araç Sınıfı İlişkisi')
plt.xlabel('Make')
plt.ylabel('Count')
plt.show()
```



Bu grafik, araç markaları ve araç sınıfları arasındaki ilişkiyi gösterir. Yiğinlı çubuk grafik, her bir marka için farklı araç sınıflarının sayısını görsel olarak sunar. Bu sayede, markaların hangi araç sınıflarına daha fazla hitap ettiğini ve hangi sınıflarda yoğunlaştığını kolayca gözlemlayabilirsiniz.

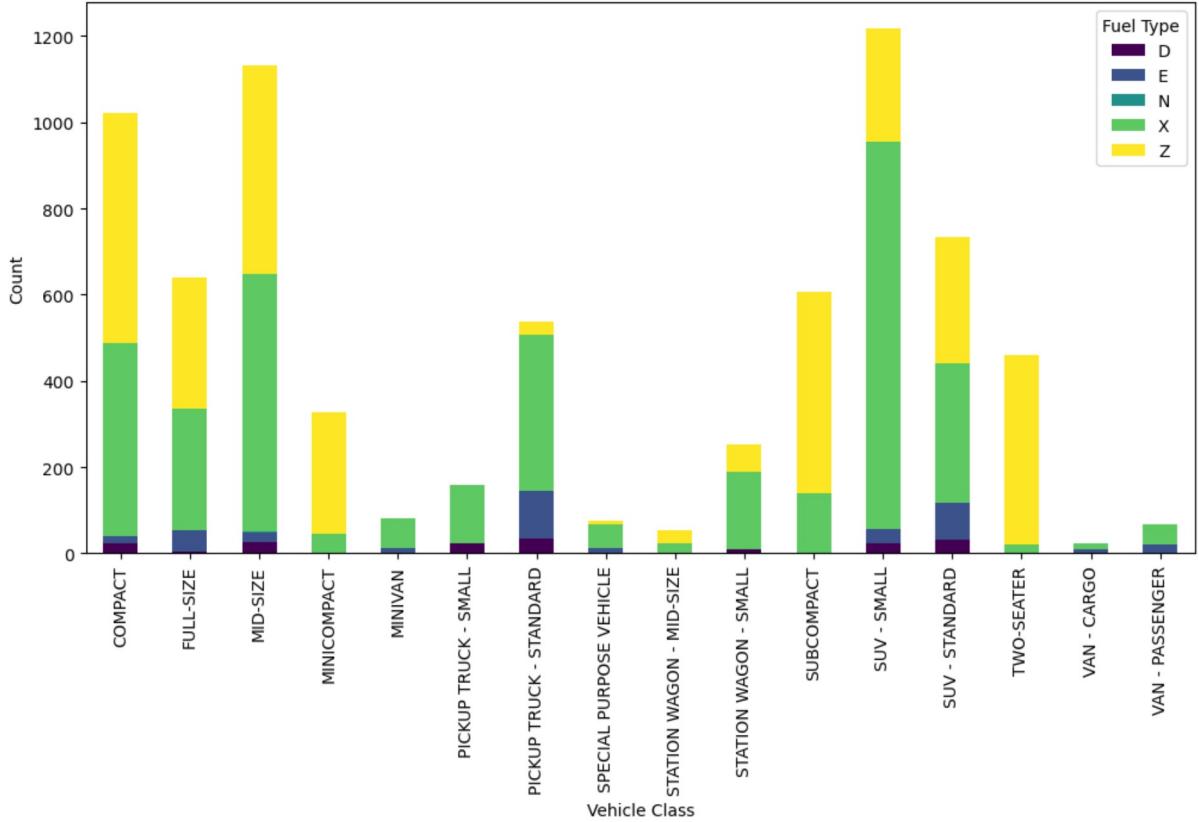
```
In [59]: sutunlar = ['Engine Size(L)', 'Cylinders', 'Fuel Consumption City (L/100 km)', 'Fuel Consumption Highway (L/100 km)']
kolerasyon = data[sutunlar].corr()
plt.figure(figsize=(10, 8))
sns.heatmap(kolerasyon, annot=True, fmt='.2f', cmap='coolwarm', linewidths=0.5)
plt.title("Correlation Matrix of Continuous Variables", fontsize=14)
plt.show()
```



Bu ısı haritası, sürekli değişkenler arasındaki korelasyonları göstermektedir. Grafik, motor hacmi, silindir sayısı, şehir içi ve şehir dışı yakıt tüketimi gibi farklı değişkenlerin birbirleriyle olan ilişkilerini sayısal olarak sunar. Korelasyon değerleri, -1 ile 1 arasında değişir.

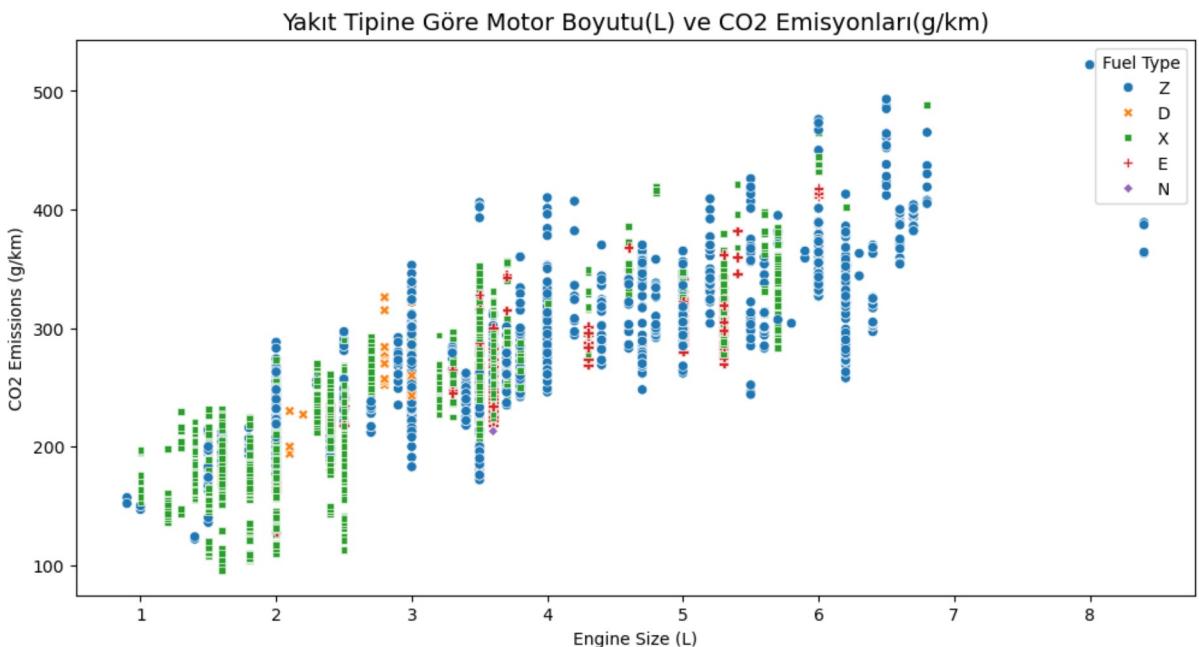
```
In [60]: vehicle_fuel = pd.crosstab(data['Vehicle Class'], data['Fuel Type'])
vehicle_fuel.plot(kind='bar', stacked=True, figsize=(12, 6), colormap='viridis')
plt.title("Araç Sınıfı ve Yakıt Türü", fontsize=14)
plt.xlabel("Vehicle Class")
plt.ylabel("Count")
plt.legend(title="Fuel Type")
plt.show()
```

Araç Sınıfı ve Yakıt Türü



Bu grafik, araç sınıfı ile yakıt türü arasındaki ilişkiyi görselleştirmektedir. Araç sınıflarına göre farklı yakıt türlerinin kullanım oranlarını gösteren yiğilmiş bar grafiği, her sınıfın yakıt türlerinin sayısal dağılımını net bir şekilde ortaya koyar. Bu tür analizler, her bir araç sınıfında hangi yakıt türlerinin daha yaygın kullanıldığını anlamaya yardımcı olur ve araç sınıfı ve yakıt türü arasındaki potansiyel ilişkileri incelemeye olanak tanır.

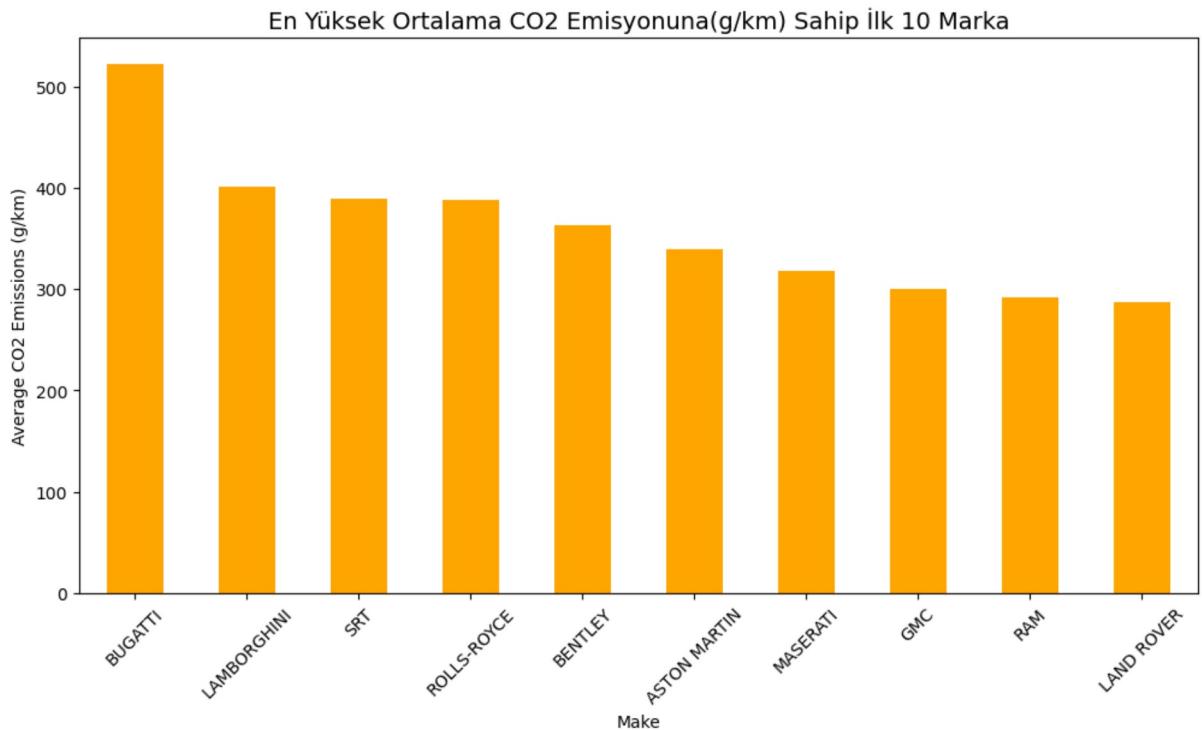
```
In [61]: plt.figure(figsize=(12, 6))
sns.scatterplot(x='Engine Size(L)', y='CO2 Emissions(g/km)', hue='Fuel Type', style='Fuel Type',
plt.title("Yakıt Tipine Göre Motor Boyutu(L) ve CO2 Emisyonları(g/km)", fontsize=14)
plt.xlabel("Engine Size (L)")
plt.ylabel("CO2 Emissions (g/km)")
plt.legend(title="Fuel Type")
plt.show()
```



Bu grafik, yakıt türüne göre motor hacmi (L) ile CO2 emisyonları (g/km) arasındaki ilişkiyi

görselleştirmektedir. Scatter plot, farklı yakıt türleri için motor hacminin artışı ile CO2 emisyonlarının nasıl değiştiğini gösterir.

```
In [62]: araclarin_ortalama_emisyonlari = data.groupby('Make')[['CO2 Emissions(g/km)']].mean().sort_values()
plt.figure(figsize=(12, 6))
araclarin_ortalama_emisyonlari.plot(kind='bar', color='orange')
plt.title("En Yüksek Ortalama CO2 Emisyonuna(g/km) Sahip İlk 10 Marka", fontsize=14)
plt.xlabel("Make")
plt.ylabel("Average CO2 Emissions (g/km)")
plt.xticks(rotation=45)
plt.show()
```

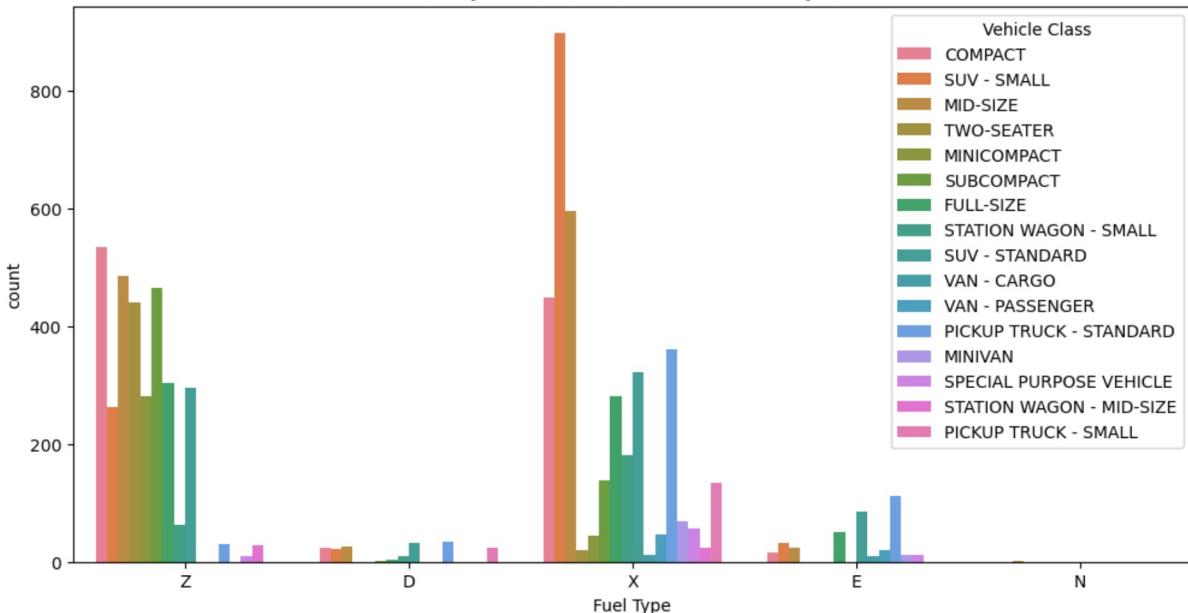


Bu grafik, en yüksek ortalama CO2 emisyonuna sahip ilk 10 arabanın markalarını göstermektedir. Grafik, her bir markanın ortalama CO2 emisyon seviyelerini (g/km) bar grafik ile sunmaktadır.

```
In [63]: scaler = MinMaxScaler()
data[['Engine Size(L)', 'CO2 Emissions(g/km)']] = scaler.fit_transform(data[['Engine Size(L)', 'CO2 Emissions(g/km)'])
# MinMaxScaler ile 'Engine Size(L)' ve 'CO2 Emissions(g/km)' sütunlarını 0 ile 1 arasında normal
```

```
In [64]: plt.figure(figsize=(12, 6))
sns.countplot(x='Fuel Type', data=data, hue='Vehicle Class')
plt.title('Araç Sınıfı ve Yakıt Türü Arasındaki İlişki')
plt.show()
```

Araç Sınıfı ve Yakıt Türü Arasındaki İlişki



Bu görselde, araç sınıfı ve yakıt türü arasındaki ilişkiyi incelemek için bir sayısal görselleştirme kullanılmıştır. 'Fuel Type' (Yakıt Türü) ve 'Vehicle Class' (Araç Sınıfı) arasındaki dağılımı bir countplot ile gösterilmektedir.

```
In [65]: # Veri setini böleme
X = pd.get_dummies(data.drop(['CO2 Emissions(g/km)'], axis=1), drop_first=True)
y = data['CO2 Emissions(g/km)']

# Eğitim ve test setlerine ayırma
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

```
In [66]: def model_egitimi(model, X_train, X_test, y_train, y_test, model_name):
    model.fit(X_train, y_train)

    y_pred = model.predict(X_test)

    # Performans metriklerini hesapla
    mae = mean_absolute_error(y_test, y_pred)
    mse = mean_squared_error(y_test, y_pred)
    r2 = r2_score(y_test, y_pred)

    print(f"{model_name} Performansı:")
    print(f"MAE: {mae}")
    print(f"MSE: {mse}")
    print(f"R²: {r2}")
    print("-" * 40)

    return {'Model': model_name, 'MAE': mae, 'MSE': mse, 'R²': r2}
```

```
In [67]: # Kullanılacak modellerin listesi
models = [
    (LinearRegression(), "Linear Regression"),
    (RandomForestRegressor(random_state=42), "Random Forest Regressor"),
    (GradientBoostingRegressor(random_state=42), "Gradient Boosting Regressor"),
    (MLPRegressor(hidden_layer_sizes=(50, 50), max_iter=500, random_state=42), "Neural Networks")
]

results = []

for model, name in models:
    result = model_egitimi(model, X_train, X_test, y_train, y_test, name)
    results.append(result)

results_df = pd.DataFrame(results)

# MAE Karşılaştırması
```

```

plt.figure(figsize=(10, 5))
plt.bar(results_df['Model'], results_df['MAE'], color='skyblue')
plt.title("Model MAE Karşılaştırması")
plt.ylabel("MAE")
plt.xticks(rotation=45)
plt.show()

# R2 Karşılaştırması
plt.figure(figsize=(10, 5))
plt.bar(results_df['Model'], results_df['R2'], color='lightgreen')
plt.title("Model R2 Karşılaştırması")
plt.ylabel("R2 Score")
plt.xticks(rotation=45)
plt.show()

```

Linear Regression Performansı:

MAE: 4912508.70150789

MSE: 747826380287460.8

R²: -3.945559324699434e+16

Random Forest Regressor Performansı:

MAE: 0.0051966866240791805

MSE: 0.00011768699305135873

R²: 0.9937907912174596

Gradient Boosting Regressor Performansı:

MAE: 0.007875092454446867

MSE: 0.0001681654693874989

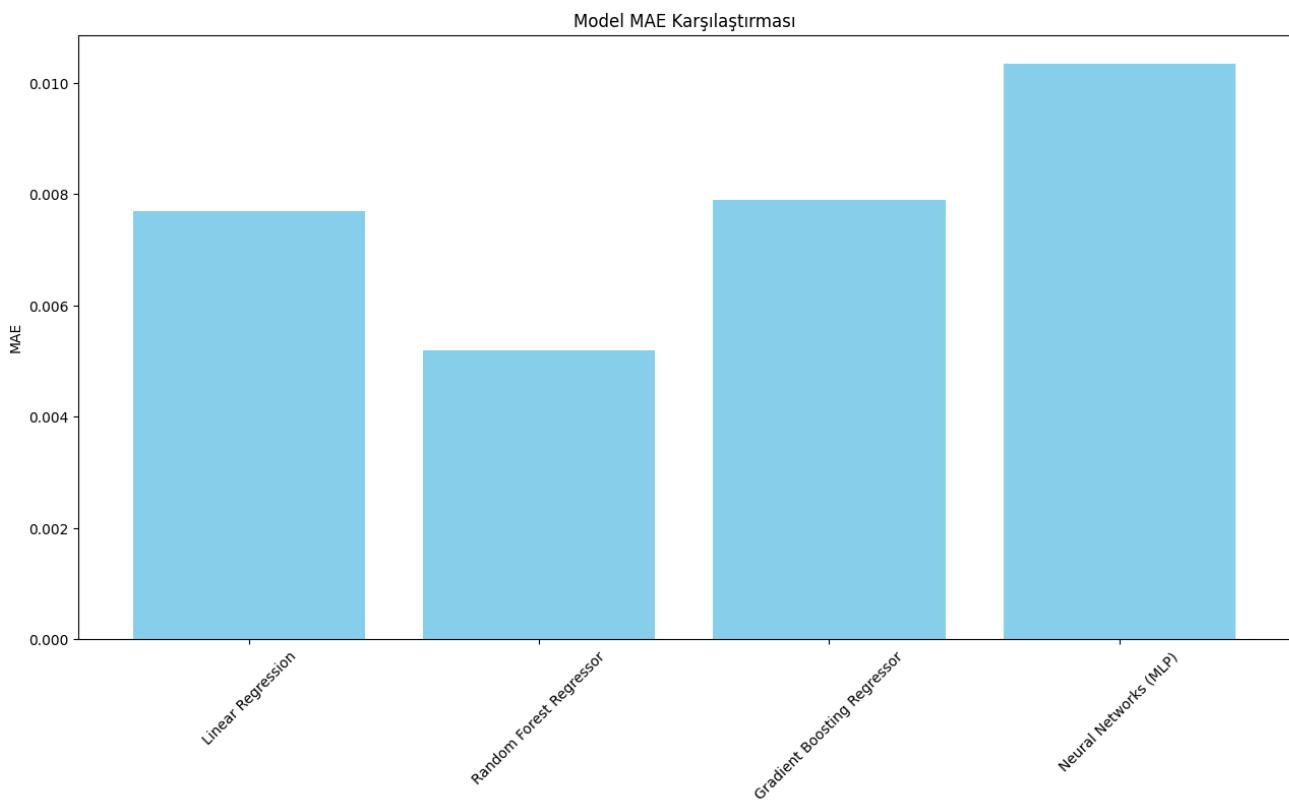
R²: 0.991127528349839

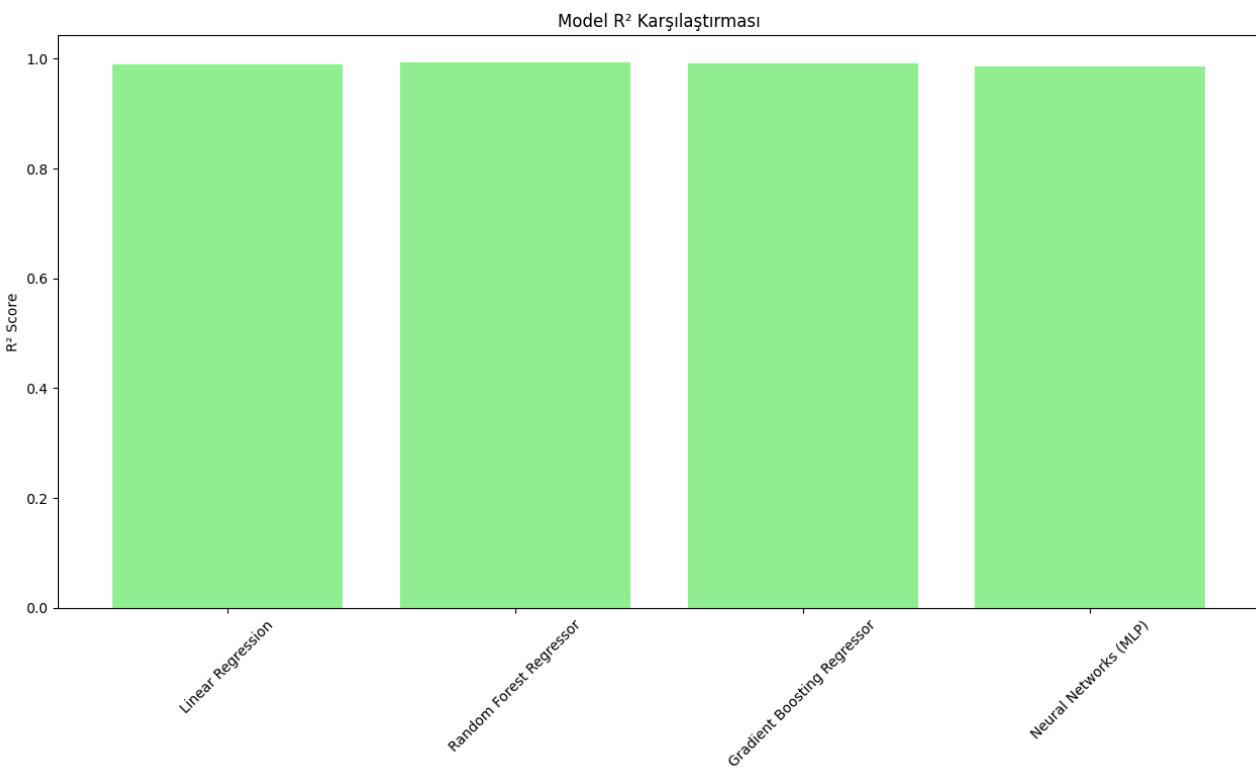
Neural Networks (MLP) Performansı:

MAE: 0.010340169812499711

MSE: 0.0002580693550640582

R²: 0.9863841664705548





Bu projede dört farklı makine öğrenmesi modelinin CO2 emisyon tahminlerinde ne kadar başarılı olduğunu görmek amacıyla performansları değerlendirilmiştir. Sonuçlar, her modelin Ortalama Mutlak Hata (MAE), Ortalama Kare Hata (MSE) ve R² skorları üzerinden analiz edilmiştir.

Linear Regression modeli, CO2 emisyonlarını tahmin etme konusunda çok zayıf bir performans sergilemiştir. Modelin R² skoru negatif bir değere ulaşmış, bu da modelin veriye uygunluğunun oldukça düşük olduğunu ve tahminlerin rastgelelige yakın olduğunu gösteriyor. Ayrıca, MAE ve MSE değerleri çok yüksek çıkmıştır, bu da modelin ciddi hata paylarına sahip olduğunu göstermektedir.

Random Forest Regressor modeli ise çok daha başarılı bir performans sergilemiştir. R² skoru 0.99'a yakın olup, model veriyi çok iyi öğrenmiştir ve oldukça düşük hata payları ile tahminler yapabilmiştir. MAE ve MSE değerleri de oldukça düşüktür, bu da modelin doğruluğunun yüksek olduğunu gösteriyor. Bu model, CO2 emisyon tahminleri için en etkili sonuçları elde etmiştir.

Gradient Boosting Regressor modeli de oldukça iyi bir performans sergilemiştir. R² skoru 0.99'a yakın olup, MAE ve MSE değerleri Random Forest'a yakın seviyelerde kalmıştır. Bu model de yüksek doğrulukla tahminler yapabilmiştir, ancak Random Forest'ın performansı kadar etkili olmamıştır.

Son olarak, **Neural Networks (MLP)** modeli, biraz daha düşük bir doğruluk ile sonuçlanmıştır. R² skoru 0.99'a oldukça yakın olmakla birlikte, Random Forest ve Gradient Boosting'in performansına kıyasla biraz daha yüksek hata paylarına sahiptir. Ancak yine de modelin tahmin gücü oldukça iyi seviyededir.

Genel olarak, **Random Forest Regressor** modeli, en düşük hata ve en yüksek doğrulukla CO2 emisyonlarını tahmin etme konusunda en iyi performansı gösteren model olmuştur. Diğer modeller de başarılı olmasına rağmen, Random Forest bu alanda öne çıkmaktadır.