



Titanik veri setiyle temel istatistiksel analizler gerçekleştiren ve sonuçları görselleştiren detaylı bir Python kodu bulunmaktadır. Kodda, pandas ve matplotlib gibi kütüphaneler kullanılarak hem istatistiksel analizler hem de görselleştirmeler yapılmaktadır.

a. Temel İstatistiksel analizler (her bir değişkene ait Ortalama, medyan, mod, varyans ve standart sapma gibi temel istatistikleri hesaplanması)

**employee-salaries-analysis Data Setı Kullanılmıştır**

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

*# Veri setini yükleme*

data\_url = "https://raw.githubusercontent.com/datasciencedojo/datasets/master/titanic.csv"

data = pd.read\_csv(data\_url)

*# İlk 5 satırı görüntüleme*

print("**\n**Veri Setinin İlk 5 Satırı:**\n**")

print(data.head())

*# Temel istatistiksel analizler için bir fonksiyon*

def calculate\_statistics(df, column):

stats = {

"Ortalama (Mean)": np.mean(df[column]),

"Medyan (Median)": np.median(df[column]),

"Mod (Mode)": df[column].mode()[0],

"Varyans (Variance)": np.var(df[column], ddof=1),

"Standart Sapma (Std Dev)": np.std(df[column], ddof=1)

}

return stats

*# Sayısal değişkenlerin belirlenmesi*

numeric\_columns = data.select\_dtypes(include=['float64', 'int64']).columns

*# İstatistiksel analizlerin hesaplanması*

all\_statistics = {}

print("**\n**Temel İstatistikler:**\n**")

for col **in** numeric\_columns:

stats = calculate\_statistics(data, col)

all\_statistics[col] = stats

print(f"**\n{**col**}** Değişkeni:")

for key, value **in** stats.items():

print(f" **{**key**}**: **{**value**:**.2f**}**")

*# Grafiksel analizler*

sns.set(style="whitegrid")

*# Histogramlar*

fig, axes = plt.subplots(len(numeric\_columns), 1, figsize=(8, len(numeric\_columns) \* 4))

fig.suptitle("Histogramlar", fontsize=16)

for i, col **in** enumerate(numeric\_columns):

sns.histplot(data[col], kde=True, ax=axes[i], color="skyblue")

axes[i].set\_title(f"**{**col**}** Değişkeni")

axes[i].set\_xlabel(col)

axes[i].set\_ylabel("Frekans")

plt.tight\_layout(rect=[0, 0.03, 1, 0.95])

plt.show()

*# Kategorik değişkenlerin analizleri*

categorical\_columns = data.select\_dtypes(include=['object']).columns

*# Kategorik değişkenlerin sütun grafikleri*

for col **in** categorical\_columns:

plt.figure(figsize=(8, 5))

sns.countplot(data=data, x=col, palette="muted")

plt.title(f"**{**col**}** Değişkeni Frekans Dağılımı")

plt.xlabel(col)

plt.ylabel("Frekans")

plt.xticks(rotation=45)

plt.show()

*# Sayısal değişkenler için scatterplot matrisi*

sns.pairplot(data[numeric\_columns], diag\_kind='kde', corner=True)

plt.suptitle("Sayısal Değişkenler Arasındaki İlişkiler", y=1.02)

plt.show()

*# Sonuçların bir DataFrame'de gösterimi*

statistics\_df = pd.DataFrame(all\_statistics).T

print("**\n**Tüm Değişkenlerin İstatistiksel Özeti:")

print(statistics\_df)

**Çalıştıtırnca Çıkan Çıkan Çıktısı :**

Veri Setinin İlk 5 Satırı:

PassengerId Survived Pclass \

0 1 0 3

1 2 1 1

2 3 1 3

3 4 1 1

4 5 0 3

Name Sex Age SibSp \

0 Braund, Mr. Owen Harris male 22.0 1

1 Cumings, Mrs. John Bradley (Florence Briggs Th... female 38.0 1

2 Heikkinen, Miss. Laina female 26.0 0

3 Futrelle, Mrs. Jacques Heath (Lily May Peel) female 35.0 1

4 Allen, Mr. William Henry male 35.0 0

Parch Ticket Fare Cabin Embarked

0 0 A/5 21171 7.2500 NaN S

1 0 PC 17599 71.2833 C85 C

2 0 STON/O2. 3101282 7.9250 NaN S

3 0 113803 53.1000 C123 S

4 0 373450 8.0500 NaN S

Temel İstatistikler:

PassengerId Değişkeni:

Ortalama (Mean): 446.00

Medyan (Median): 446.00

Mod (Mode): 1.00

Varyans (Variance): 66231.00

Standart Sapma (Std Dev): 257.35

Survived Değişkeni:

Ortalama (Mean): 0.38

Medyan (Median): 0.00

Mod (Mode): 0.00

Varyans (Variance): 0.24

Standart Sapma (Std Dev): 0.49

Pclass Değişkeni:

Ortalama (Mean): 2.31

Medyan (Median): 3.00

Mod (Mode): 3.00

Varyans (Variance): 0.70

Standart Sapma (Std Dev): 0.84

Age Değişkeni:

Ortalama (Mean): 29.70

Medyan (Median): nan

Mod (Mode): 24.00

Varyans (Variance): 211.02

Standart Sapma (Std Dev): 14.53

SibSp Değişkeni:

Ortalama (Mean): 0.52

Medyan (Median): 0.00

Mod (Mode): 0.00

Varyans (Variance): 1.22

Standart Sapma (Std Dev): 1.10

Parch Değişkeni:

Ortalama (Mean): 0.38

Medyan (Median): 0.00

Mod (Mode): 0.00

Varyans (Variance): 0.65

Standart Sapma (Std Dev): 0.81

Fare Değişkeni:

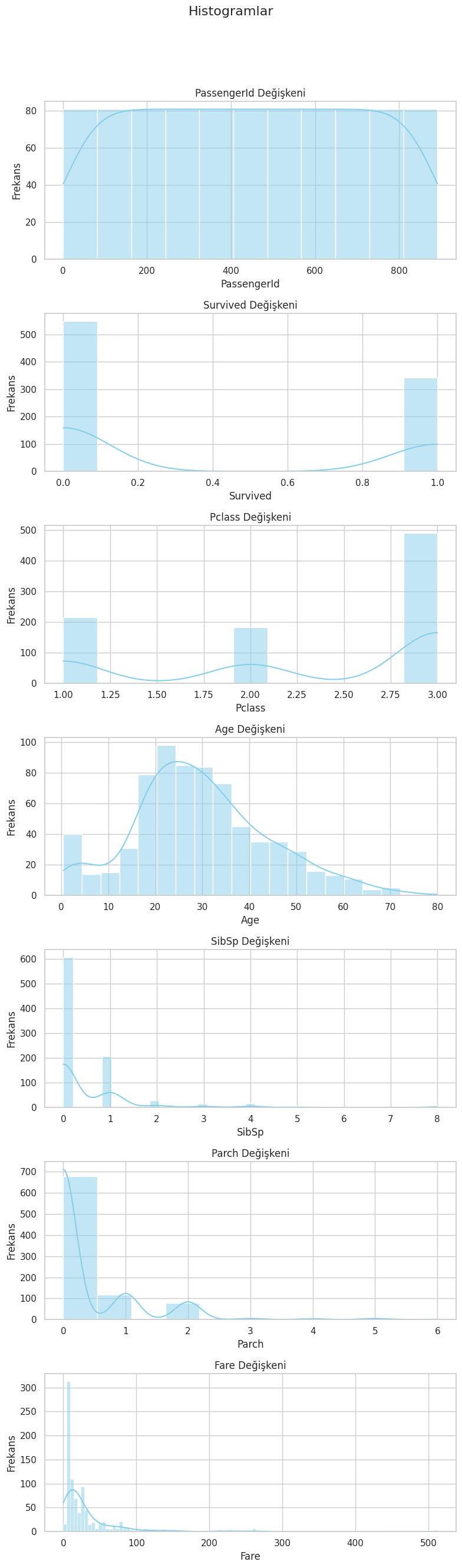
Ortalama (Mean): 32.20

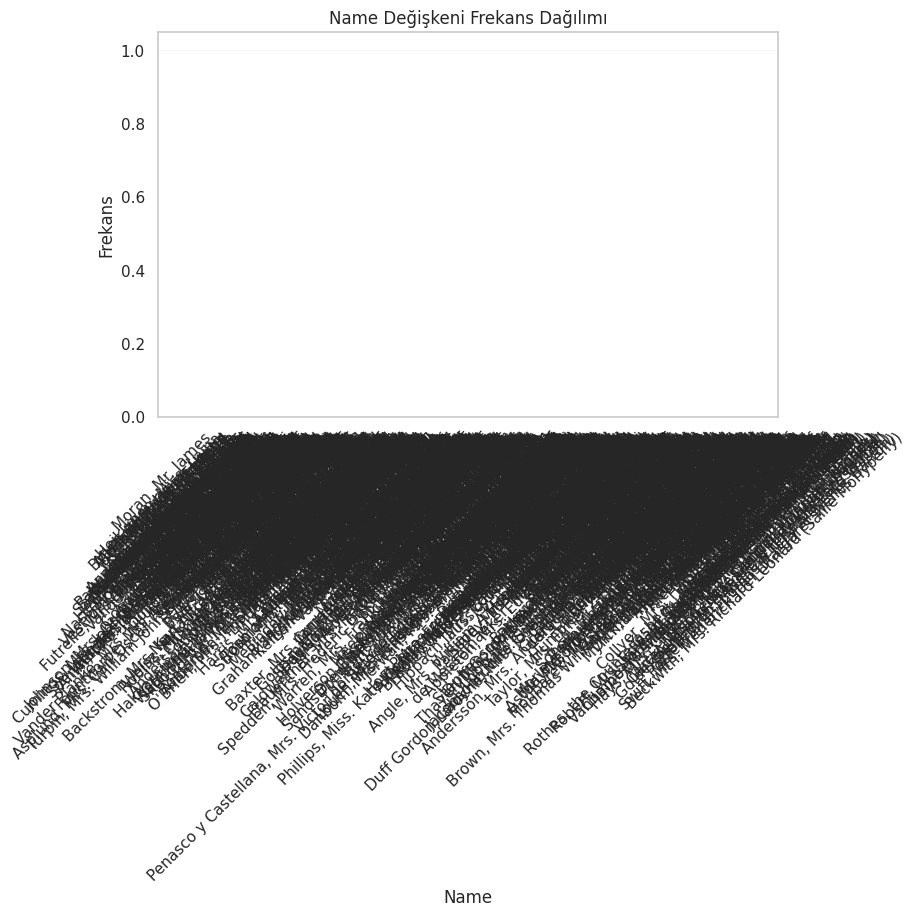
Medyan (Median): 14.45

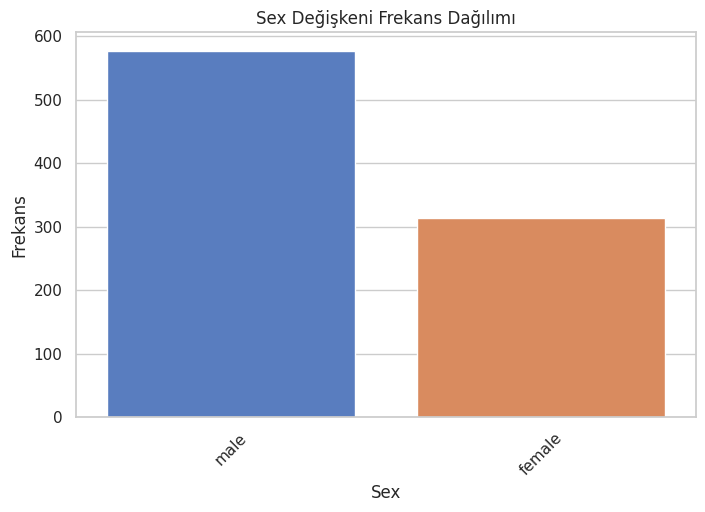
Mod (Mode): 8.05

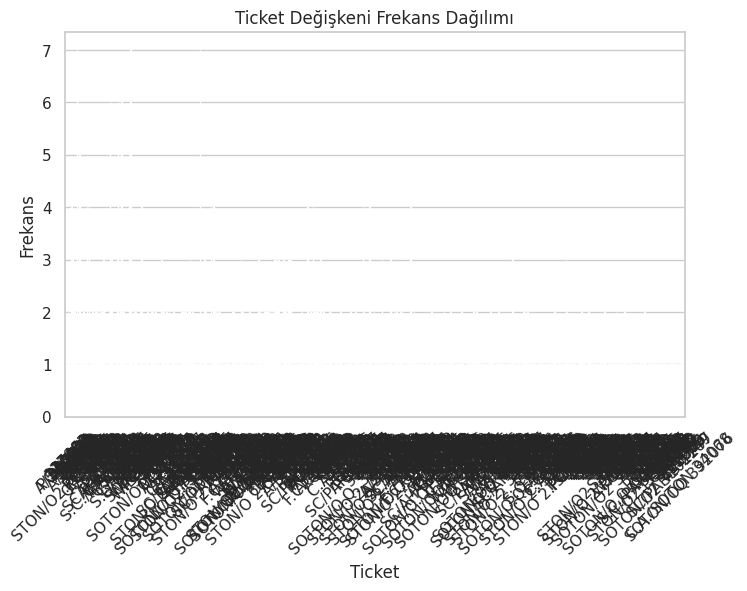
Varyans (Variance): 2469.44

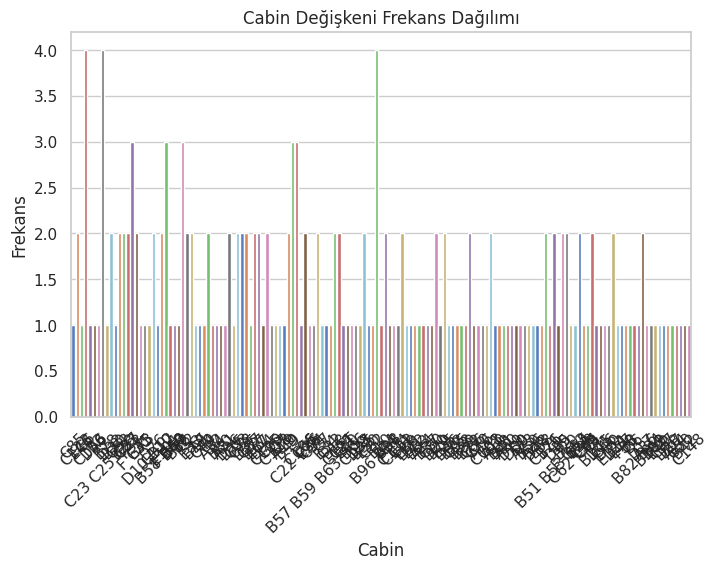
Standart Sapma (Std Dev): 49.69

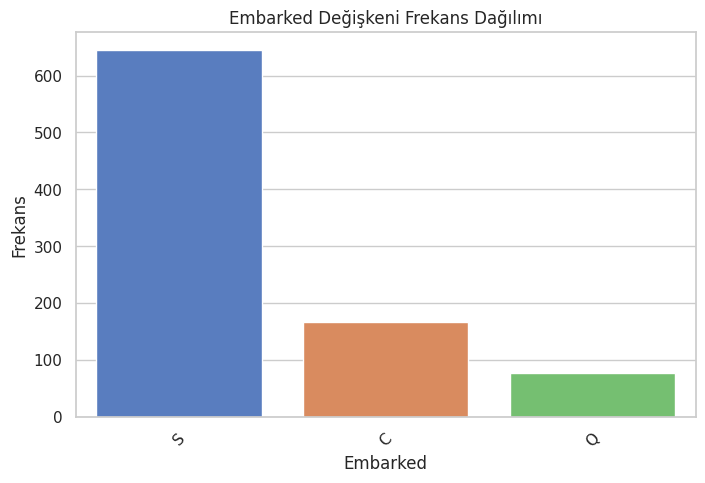
****

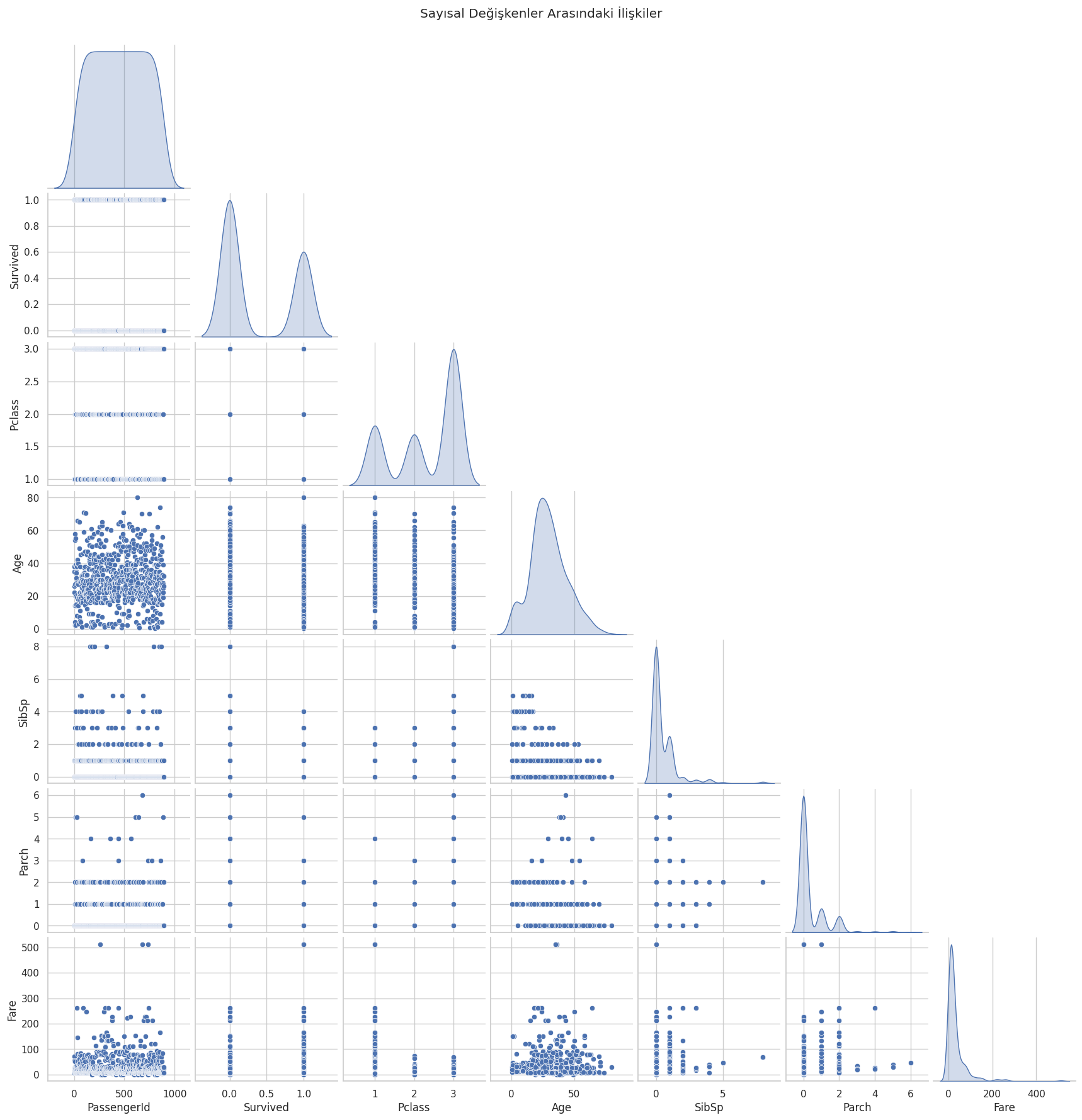
****

****

****

****

****

****

b) Veri görselleştirme (histogram, çubuk grafiği veya dağılım grafiği ile veri setinin özelliklerinin analiz edilmesi ve yorumlanması)

Kaynak Kod:

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

data = {

"Öğrenci": ["Ali", "Ayşe", "Mehmet", "Elif", "Fatma", "Ahmet"],

"Matematik": [85, 90, 70, 88, 95, 76],

"Fizik": [78, 82, 75, 95, 90, 80],

"Kimya": [92, 88, 85, 80, 96, 89]

}

df = pd.DataFrame(data)

#df = pd.read\_csv('train.csv')

# Histogram

plt.figure(figsize=(10, 5))

plt.hist(df["Fizik"], bins=5, color="skyblue", edgecolor="black")

plt.title("Fizik Notları Dağılımı (Histogram)")

plt.xlabel("Not Aralığı")

plt.ylabel("Öğrenci Sayısı")

plt.grid(True)

plt.show()

# Çubuk Grafiği

plt.figure(figsize=(10, 5))

plt.bar(df["Öğrenci"], df["Fizik"], color="orange")

plt.title("Öğrencilerin Fizik Notları (Çubuk Grafiği)")

plt.xlabel("Öğrenci")

plt.ylabel("Fizik Notu")

plt.show()

# Dağılım Grafiği

plt.figure(figsize=(10, 5))

sns.scatterplot(data=df, x="Fizik", y="Kimya", hue="Öğrenci", palette="viridis", s=100)

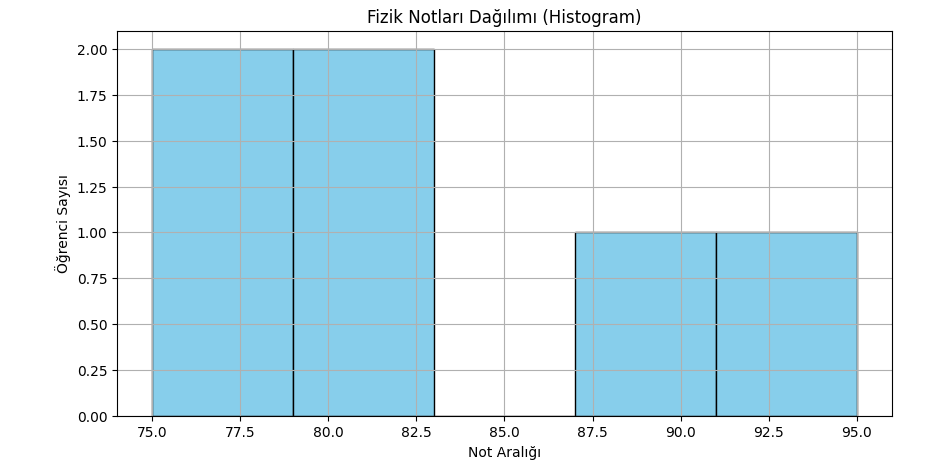
plt.title("Fizik ve Kimya Notları Arasındaki İlişki (Dağılım Grafiği)")

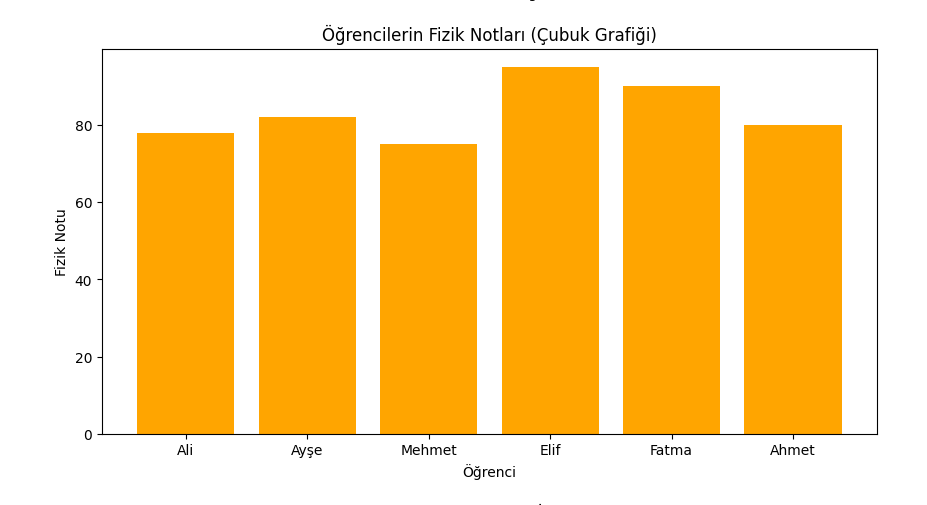
plt.xlabel("Fizik Notu")

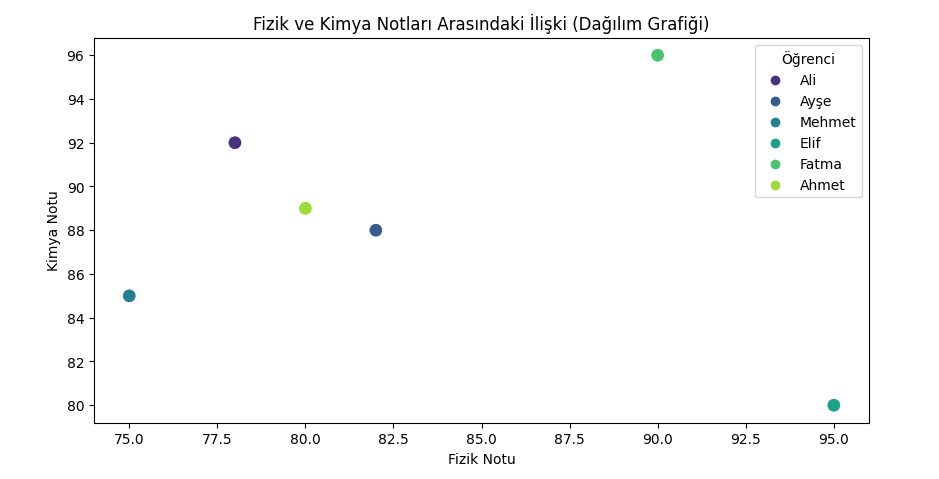
plt.ylabel("Kimya Notu")

plt.show()

**Çalıştıtırnca Çıkan Çıkan Çıktısı :**







1. **Veri Seti Oluşturma**

 **Amaç**: Altı öğrencinin Matematik, Fizik ve Kimya notlarını içeren bir veri çerçevesi (DataFrame) oluşturur.

 **Sonuç**: Öğrenciler ve notlar arasında bir ilişki incelemeyi sağlar.

2. **Histogram (Fizik Notlarının Dağılımı)**

**Amaç**: Fizik notlarının dağılımını incelemek.

**Yapılanlar**:

* Histogram, notların aralıklara (bins) bölünerek her aralıktaki öğrenci sayısını gösterir.
* Grafiğe başlık, etiketler ve kenar çizgileri eklenir.
* **Sonuç**: Notların hangi aralıkta yoğunlaştığını görselleştirir.

3. **Çubuk Grafiği (Öğrencilerin Fizik Notları)**

 **Amaç**: Her öğrencinin Fizik notlarını bireysel olarak göstermek.

 **Yapılanlar**:

* Çubuk grafik, her bir öğrencinin fizik notunu temsil eder.
* Grafiğe başlık, x ve y eksen etiketleri eklenir.
* **Sonuç**: Öğrenciler arasında not farklılıklarını görmeyi sağlar.

4. **Dağılım Grafiği (Fizik ve Kimya Notları İlişkisi)**

 **Amaç**: Fizik ve Kimya notları arasındaki ilişkiyi görselleştirmek.

 **Yapılanlar**:

* Dağılım grafiği her öğrenci için bir nokta oluşturur.
* hue parametresi, öğrencileri farklı renklerle ayırır.
* Nokta boyutu s=100 ile belirlenir.
* **Sonuç**: Fizik ve Kimya notları arasındaki ilişki ve öğrencilerin bireysel katkıları görülür.

**5. Kodun Genel Yapısı**

Kod üç farklı görselleştirme türünü kullanarak veriyi analiz eder. Her bir grafik:

* Veriyi daha anlaşılır hale getirir.
* Başlık ve eksen etiketleri ile açıklanmıştır.
* Farklı renkler ve stillerle zenginleştirilmiştir.

c) Hipotez Testi uygulamaları (Oran Arasındaki Farkın Testi, Bağımsız İki Örneklem T Testi, Ki-Kare Bağımsızlık Testi, Mann-Whitney U Testi, ANOVA…). Bu adımda hipotezler kurularak test sonuçları ve yorumlamalrı verilmelidir.

**Kaynak Kod:**

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

from scipy.stats import ttest\_ind, chi2\_contingency, mannwhitneyu, f\_oneway

*# Veri setini yükleme*

data\_url = "https://raw.githubusercontent.com/datasciencedojo/datasets/master/titanic.csv"

data = pd.read\_csv(data\_url)

*# Veriyi keşfetme*

print("**\n**Veri Setinin İlk 5 Satırı:**\n**")

print(data.head())

*# Eksik verileri doldurma (gerekirse)*

data['Age'].fillna(data['Age'].median(), inplace=True)

data['Embarked'].fillna(data['Embarked'].mode()[0], inplace=True)

data['Fare'].fillna(data['Fare'].median(), inplace=True)

*# Hipotez Testi Uygulamaları*

sns.set(style="whitegrid")

*# 1. Oran Arasındaki Farkın Testi (Survived oranı kadınlar ve erkekler arasında farklı mı?)*

print("**\n**1. Oran Arasındaki Farkın Testi**\n**")

gender\_survived = data.groupby('Sex')['Survived'].mean()

print(gender\_survived)

*# Görselleştirme*

sns.barplot(x='Sex', y='Survived', data=data, palette="pastel")

plt.title("Cinsiyete Göre Hayatta Kalma Oranları")

plt.ylabel("Hayatta Kalma Oranı")

plt.show()

*# 2. Bağımsız İki Örneklem T Testi (Yaş farkı hayatta kalanlar ve ölenler arasında farklı mı?)*

print("**\n**2. Bağımsız İki Örneklem T Testi**\n**")

survived\_age = data[data['Survived'] == 1]['Age']

not\_survived\_age = data[data['Survived'] == 0]['Age']

*# T Testi*

t\_stat, p\_val = ttest\_ind(survived\_age, not\_survived\_age)

print(f"T-istatistiği: **{**t\_stat**:**.2f**}**, P-değeri: **{**p\_val**:**.4f**}**")

*# Görselleştirme*

sns.kdeplot(survived\_age, label='Hayatta Kalanlar', shade=True)

sns.kdeplot(not\_survived\_age, label='Hayatta Kalmayanlar', shade=True)

plt.title("Yaşa Göre Dağılım (Hayatta Kalma Durumuna Göre)")

plt.xlabel("Yaş")

plt.legend()

plt.show()

*# 3. Ki-Kare Bağımsızlık Testi (Cinsiyet ve hayatta kalma bağımsız mı?)*

print("**\n**3. Ki-Kare Bağımsızlık Testi**\n**")

crosstab = pd.crosstab(data['Sex'], data['Survived'])

chi2, p, dof, expected = chi2\_contingency(crosstab)

print(f"Ki-Kare Değeri: **{**chi2**:**.2f**}**, P-değeri: **{**p**:**.4f**}**")

print("Beklenen Frekanslar:**\n**", expected)

*# Görselleştirme*

sns.heatmap(crosstab, annot=True, fmt="d", cmap="coolwarm")

plt.title("Cinsiyet ve Hayatta Kalma Durumu")

plt.xlabel("Hayatta Kalma Durumu")

plt.ylabel("Cinsiyet")

plt.show()

*# 4. Mann-Whitney U Testi (Ücret farkı hayatta kalanlar ve ölenler arasında farklı mı?)*

print("**\n**4. Mann-Whitney U Testi**\n**")

survived\_fare = data[data['Survived'] == 1]['Fare']

not\_survived\_fare = data[data['Survived'] == 0]['Fare']

*# Mann-Whitney U Testi*

u\_stat, p\_val = mannwhitneyu(survived\_fare, not\_survived\_fare)

print(f"U-istatistiği: **{**u\_stat**:**.2f**}**, P-değeri: **{**p\_val**:**.4f**}**")

*# Görselleştirme*

sns.boxplot(x='Survived', y='Fare', data=data, palette="pastel")

plt.title("Hayatta Kalma Durumuna Göre Ücret Dağılımı")

plt.xlabel("Hayatta Kalma Durumu")

plt.ylabel("Ücret")

plt.show()

*# 5. ANOVA Testi (Farklı sınıflar arasında yaş farkı var mı?)*

print("**\n**5. ANOVA Testi**\n**")

class\_age\_groups = [data[data['Pclass'] == i]['Age'].dropna() for i **in** range(1, 4)]

*# ANOVA Testi*

f\_stat, p\_val = f\_oneway(\*class\_age\_groups)

print(f"F-istatistiği: **{**f\_stat**:**.2f**}**, P-değeri: **{**p\_val**:**.4f**}**")

*# Görselleştirme*

sns.boxplot(x='Pclass', y='Age', data=data, palette="pastel")

plt.title("Sınıflara Göre Yaş Dağılımı")

plt.xlabel("Sınıf")

plt.ylabel("Yaş")

plt.show()

*# Genel Sonuçlar*

print("**\n**Hipotez Testleri Tamamlandı.")

**Çalıştıtırnca Çıkan Çıkan Çıktısı :**

Veri Setinin İlk 5 Satırı:

PassengerId Survived Pclass \

0 1 0 3

1 2 1 1

2 3 1 3

3 4 1 1

4 5 0 3

Name Sex Age SibSp \

0 Braund, Mr. Owen Harris male 22.0 1

1 Cumings, Mrs. John Bradley (Florence Briggs Th... female 38.0 1

2 Heikkinen, Miss. Laina female 26.0 0

3 Futrelle, Mrs. Jacques Heath (Lily May Peel) female 35.0 1

4 Allen, Mr. William Henry male 35.0 0

Parch Ticket Fare Cabin Embarked

0 0 A/5 21171 7.2500 NaN S

1 0 PC 17599 71.2833 C85 C

2 0 STON/O2. 3101282 7.9250 NaN S

3 0 113803 53.1000 C123 S

4 0 373450 8.0500 NaN S

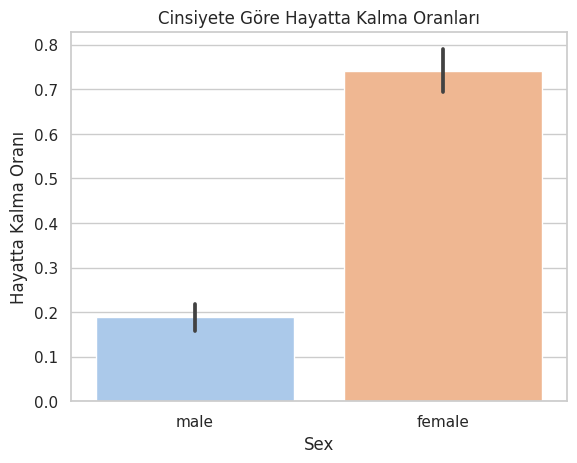
1. Oran Arasındaki Farkın Testi

Sex

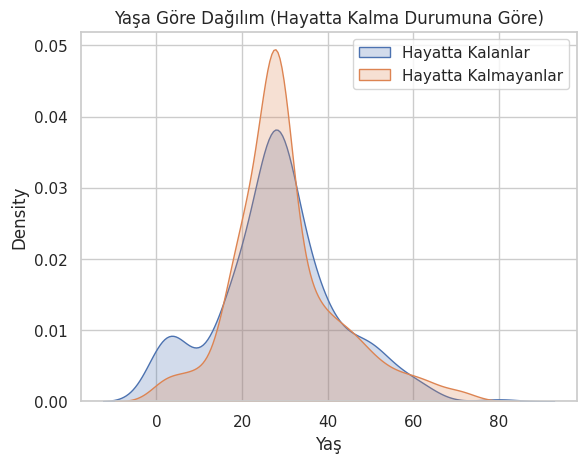
female 0.742038

male 0.188908

Name: Survived, dtype: float64



2. Bağımsız İki Örneklem T Testi



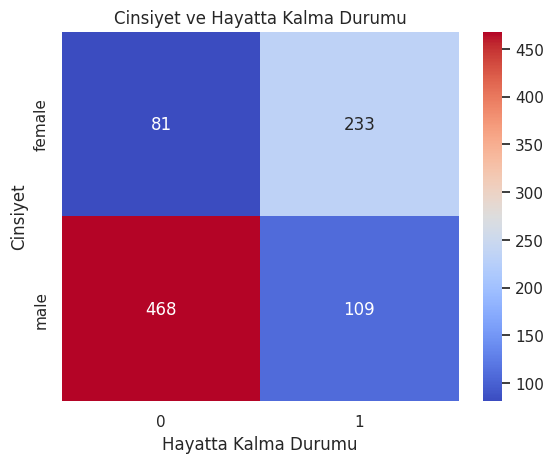
3. Ki-Kare Bağımsızlık Testi

Ki-Kare Değeri: 260.72, P-değeri: 0.0000

Beklenen Frekanslar:

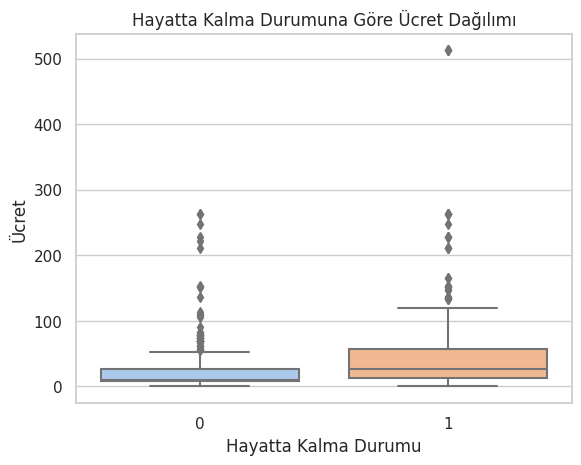
[[193.47474747 120.52525253]

[355.52525253 221.47474747]]



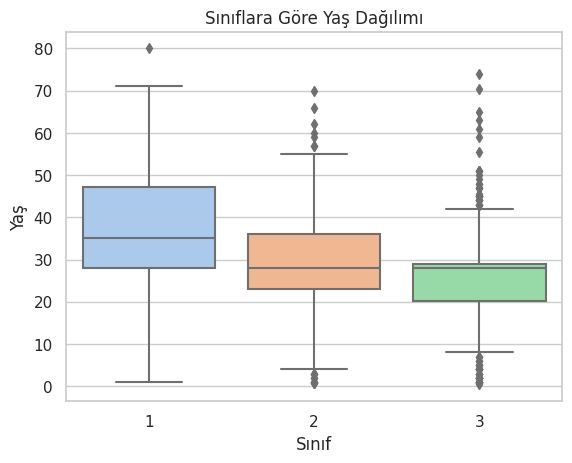
4. Mann-Whitney U Testi

U-istatistiği: 129951.50, P-değeri: 0.0000



5. ANOVA Testi

F-istatistiği: 59.37, P-değeri: 0.0000



 **Oranlar Arasındaki Farkın Testi**: Cinsiyet bazında hayatta kalma oranlarını analiz eder.

 **Bağımsız İki Örneklem T Testi**: Hayatta kalanlar ve ölmeyenler arasındaki yaş farkını test eder.

 **Ki-Kare Bağımsızlık Testi**: Cinsiyet ile hayatta kalma durumu arasındaki ilişkiyi inceler.

 **Mann-Whitney U Testi**: Hayatta kalanlar ve ölmeyenler arasındaki ücret farkını analiz eder.

 **ANOVA Testi**: Yolcu sınıfları arasında yaş farkını test eder.

**d) Regresyon analizi**

**Kaynak Kod:**

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

from sklearn.model\_selection import train\_test\_split

from sklearn.linear\_model import LogisticRegression

from sklearn.metrics import classification\_report, confusion\_matrix, roc\_curve, auc

from sklearn.preprocessing import StandardScaler

*# Veri setini yükleme*

data\_url = "https://raw.githubusercontent.com/datasciencedojo/datasets/master/titanic.csv"

data = pd.read\_csv(data\_url)

*# Veriyi keşfetme*

print("**\n**Veri Setinin İlk 5 Satırı:**\n**")

print(data.head())

*# Eksik verileri doldurma ve kategorik değişkenleri dönüştürme*

data['Age'].fillna(data['Age'].median(), inplace=True)

data['Embarked'].fillna(data['Embarked'].mode()[0], inplace=True)

data['Fare'].fillna(data['Fare'].median(), inplace=True)

data = pd.get\_dummies(data, columns=['Sex', 'Embarked'], drop\_first=True)

*# İlgili sütunların seçimi (hedef: Survived)*

features = ['Pclass', 'Age', 'SibSp', 'Parch', 'Fare', 'Sex\_male', 'Embarked\_Q', 'Embarked\_S']

X = data[features]

y = data['Survived']

*# Eğitim ve test verisinin ayrılması*

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.3, random\_state=42, stratify=y)

*# Verilerin ölçeklenmesi*

scaler = StandardScaler()

X\_train = scaler.fit\_transform(X\_train)

X\_test = scaler.transform(X\_test)

*# Lojistik regresyon modeli oluşturma ve eğitme*

model = LogisticRegression(random\_state=42, max\_iter=1000)

model.fit(X\_train, y\_train)

*# Tahminler*

y\_pred = model.predict(X\_test)

y\_pred\_prob = model.predict\_proba(X\_test)[:, 1]

*# Model performansı değerlendirme*

print("**\n**Sınıflandırma Raporu:**\n**")

print(classification\_report(y\_test, y\_pred))

*# Confusion Matrix*

conf\_matrix = confusion\_matrix(y\_test, y\_pred)

plt.figure(figsize=(6, 4))

sns.heatmap(conf\_matrix, annot=True, fmt="d", cmap="Blues", cbar=False)

plt.title("Confusion Matrix")

plt.xlabel("Tahmin Edilen")

plt.ylabel("Gerçek")

plt.show()

*# ROC Eğrisi ve AUC*

fpr, tpr, thresholds = roc\_curve(y\_test, y\_pred\_prob)

roc\_auc = auc(fpr, tpr)

plt.figure(figsize=(8, 6))

plt.plot(fpr, tpr, color='blue', label=f'ROC Eğrisi (AUC = **{**roc\_auc**:**.2f**}**)')

plt.plot([0, 1], [0, 1], color='gray', linestyle='--')

plt.title("ROC Eğrisi")

plt.xlabel("False Positive Rate")

plt.ylabel("True Positive Rate")

plt.legend()

plt.show()

*# Katsayıların analizi*

coefficients = pd.DataFrame({

'Feature': features,

'Coefficient': model.coef\_[0]

})

coefficients.sort\_values(by='Coefficient', ascending=False, inplace=True)

print("**\n**Model Katsayıları:**\n**")

print(coefficients)

*# Görselleştirme*

plt.figure(figsize=(10, 6))

sns.barplot(x='Coefficient', y='Feature', data=coefficients, palette="viridis")

plt.title("Model Katsayıları")

plt.xlabel("Katsayı")

plt.ylabel("Değişken")

plt.show()

*# Genel Sonuçlar*

print("**\n**Regresyon Analizi Tamamlandı.")

Veri Setinin İlk 5 Satırı:

PassengerId Survived Pclass \

0 1 0 3

1 2 1 1

2 3 1 3

3 4 1 1

4 5 0 3

Name Sex Age SibSp \

0 Braund, Mr. Owen Harris male 22.0 1

1 Cumings, Mrs. John Bradley (Florence Briggs Th... female 38.0 1

2 Heikkinen, Miss. Laina female 26.0 0

3 Futrelle, Mrs. Jacques Heath (Lily May Peel) female 35.0 1

4 Allen, Mr. William Henry male 35.0 0

Parch Ticket Fare Cabin Embarked

0 0 A/5 21171 7.2500 NaN S

1 0 PC 17599 71.2833 C85 C

2 0 STON/O2. 3101282 7.9250 NaN S

3 0 113803 53.1000 C123 S

4 0 373450 8.0500 NaN S

Sınıflandırma Raporu:

precision recall f1-score support

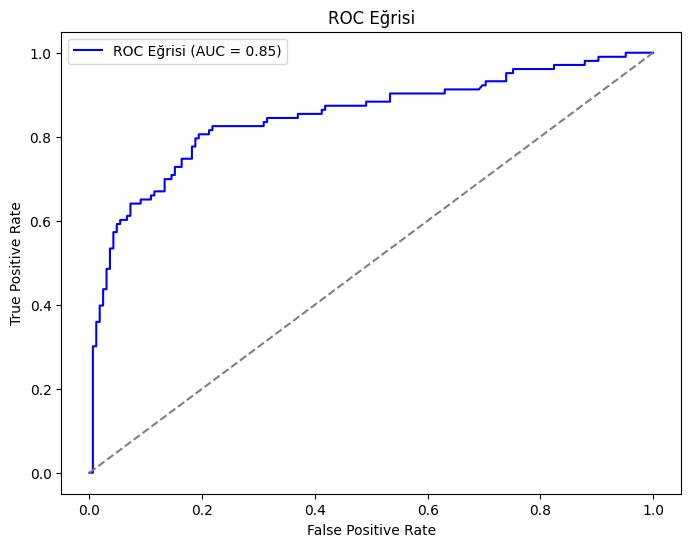
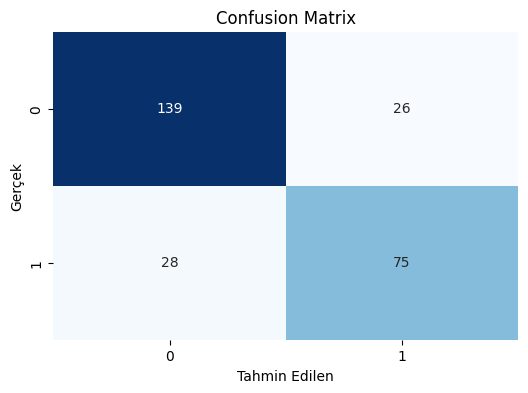
0 0.83 0.84 0.84 165

1 0.74 0.73 0.74 103

accuracy 0.80 268

macro avg 0.79 0.79 0.79 268

weighted avg 0.80 0.80 0.80 268



Model Katsayıları:

Feature Coefficient

4 Fare 0.112919

6 Embarked\_Q 0.082436

3 Parch -0.052008

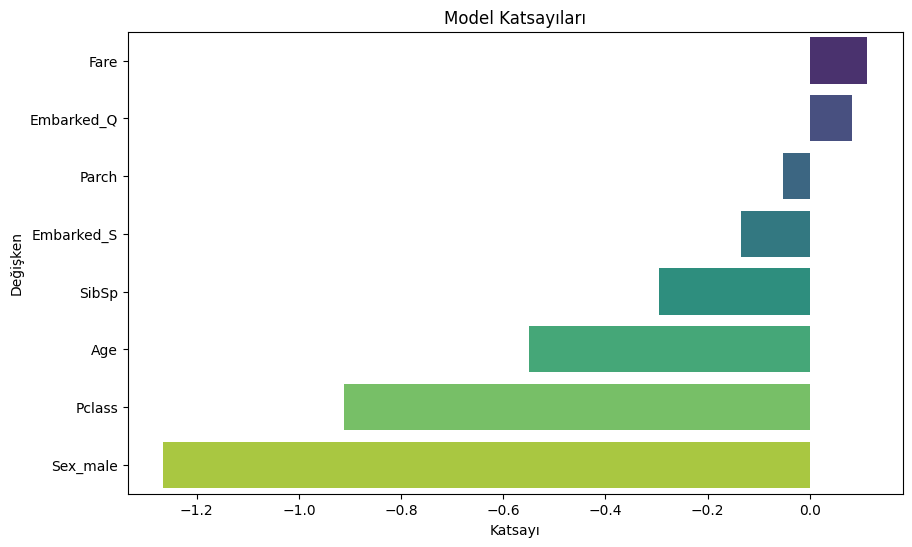
7 Embarked\_S -0.135231

2 SibSp -0.295617

1 Age -0.549540

0 Pclass -0.911516

5 Sex\_male -1.266435



 **Veri İşleme ve Dönüştürme**:

* Eksik veriler dolduruluyor.
* Kategorik değişkenler ("Sex" ve "Embarked") one-hot encoding yöntemiyle sayısal değerlere dönüştürülüyor.

 **Model Eğitimi ve Testi**:

* Veriler eğitim ve test setlerine ayrılıyor.
* Veriler ölçeklendirilerek lojistik regresyon modeli eğitiliyor.

 **Performans Değerlendirme**:

* **Sınıflandırma Raporu**: Precision, recall, F1-score gibi metrikler hesaplanıyor.
* **Confusion Matrix**: Gerçek ve tahmin edilen değerler arasındaki ilişki görselleştiriliyor.
* **ROC Eğrisi ve AUC**: Modelin ayrım gücünü görselleştirmek için ROC eğrisi çiziliyor.

 **Katsayı Analizi**:

* Modelin katsayıları analiz edilerek hangi değişkenlerin etkili olduğu inceleniyor.