

Malatya3

December 26, 2024

```
[1]: # Gerekli kütüphaneleri import et
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

# Grafik stili ayarları
sns.set(style="whitegrid")
plt.rcParams["figure.figsize"] = (10, 6)

# Veri setini yükle
df = pd.read_csv("Fraud.csv")

# Veri setinin ilk 5 satırını görüntüle
print("Veri setinin ilk 5 satırı:")
print(df.head())

# Veri setinin genel bilgileri
print("\nVeri setinin genel bilgileri:")
print(df.info())

# Veri setindeki eksik değerleri kontrol et
print("\nEksik değer analizi:")
print(df.isnull().sum())

# Temel istatistiksel özet
print("\nVeri setinin istatistiksel özeti:")
print(df.describe())

# Sütunlardaki benzersiz değerleri incele
print("\nBenzersiz değerler:")
print(df.nunique())
```

Veri setinin ilk 5 satırı:

	step	type	amount	nameOrig	oldbalanceOrg	newbalanceOrig	\
0	1	PAYMENT	9839.64	C1231006815	170136.0	160296.36	
1	1	PAYMENT	1864.28	C1666544295	21249.0	19384.72	
2	1	TRANSFER	181.00	C1305486145	181.0	0.00	

3	1	CASH_OUT	181.00	C840083671	181.0	0.00
4	1	PAYMENT	11668.14	C2048537720	41554.0	29885.86

		nameDest	oldbalanceDest	newbalanceDest	isFraud	isFlaggedFraud
0	M1979787155		0.0	0.0	0	0
1	M2044282225		0.0	0.0	0	0
2	C553264065		0.0	0.0	1	0
3	C38997010		21182.0	0.0	1	0
4	M1230701703		0.0	0.0	0	0

Veri setinin genel bilgileri:

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 6362620 entries, 0 to 6362619

Data columns (total 11 columns):

#	Column	Dtype
0	step	int64
1	type	object
2	amount	float64
3	nameOrig	object
4	oldbalanceOrg	float64
5	newbalanceOrig	float64
6	nameDest	object
7	oldbalanceDest	float64
8	newbalanceDest	float64
9	isFraud	int64
10	isFlaggedFraud	int64

dtypes: float64(5), int64(3), object(3)

memory usage: 534.0+ MB

None

Eksik değer analizi:

step	0
type	0
amount	0
nameOrig	0
oldbalanceOrg	0
newbalanceOrig	0
nameDest	0
oldbalanceDest	0
newbalanceDest	0
isFraud	0
isFlaggedFraud	0

dtype: int64

Veri setinin istatistiksel özeti:

	step	amount	oldbalanceOrg	newbalanceOrig	\
count	6.362620e+06	6.362620e+06	6.362620e+06	6.362620e+06	

mean	2.433972e+02	1.798619e+05	8.338831e+05	8.551137e+05
std	1.423320e+02	6.038582e+05	2.888243e+06	2.924049e+06
min	1.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00
25%	1.560000e+02	1.338957e+04	0.000000e+00	0.000000e+00
50%	2.390000e+02	7.487194e+04	1.420800e+04	0.000000e+00
75%	3.350000e+02	2.087215e+05	1.073152e+05	1.442584e+05
max	7.430000e+02	9.244552e+07	5.958504e+07	4.958504e+07

	oldbalanceDest	newbalanceDest	isFraud	isFlaggedFraud
count	6.362620e+06	6.362620e+06	6.362620e+06	6.362620e+06
mean	1.100702e+06	1.224996e+06	1.290820e-03	2.514687e-06
std	3.399180e+06	3.674129e+06	3.590480e-02	1.585775e-03
min	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00
25%	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00
50%	1.327057e+05	2.146614e+05	0.000000e+00	0.000000e+00
75%	9.430367e+05	1.111909e+06	0.000000e+00	0.000000e+00
max	3.560159e+08	3.561793e+08	1.000000e+00	1.000000e+00

Benzersiz değerler:

```
step          743
type          5
amount        5316900
nameOrig      6353307
oldbalanceOrg 1845844
newbalanceOrig 2682586
nameDest      2722362
oldbalanceDest 3614697
newbalanceDest 3555499
isFraud        2
isFlaggedFraud 2
dtype: int64
```

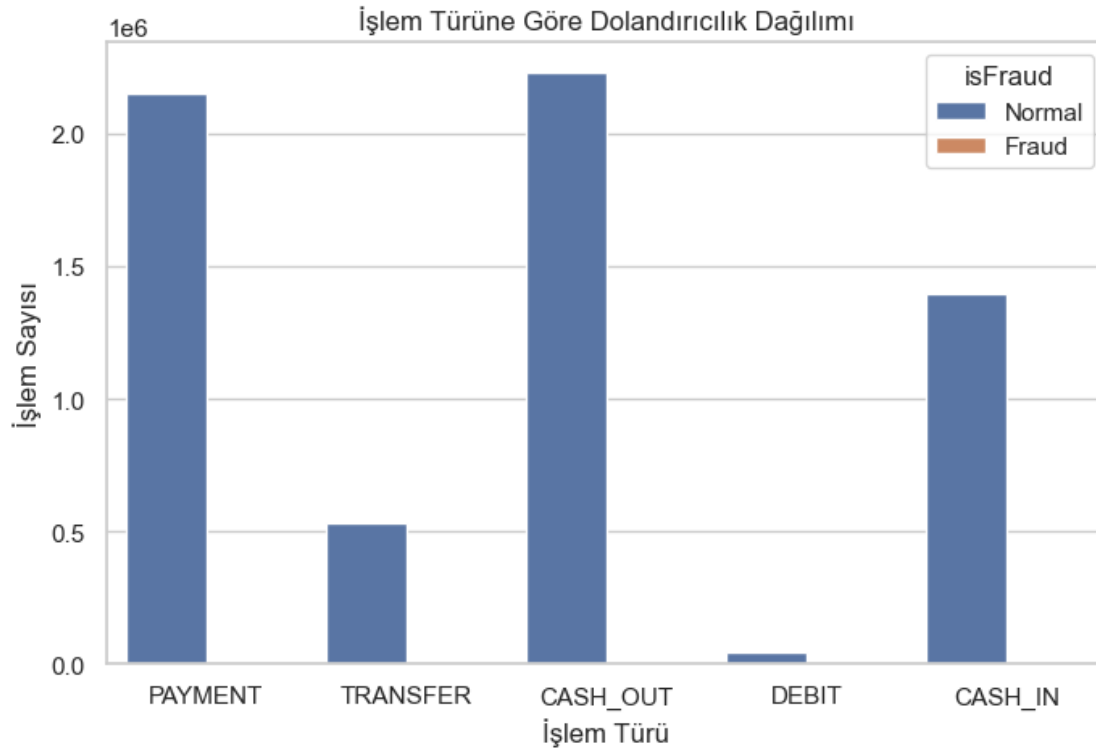
```
[2]: # İşlem türlerine göre veri dağılımı
print("\nİşlem türlerine göre dağılım:")
print(df['type'].value_counts())
```

İşlem türlerine göre dağılım:

```
type
CASH_OUT    2237500
PAYMENT     2151495
CASH_IN     1399284
TRANSFER    532909
DEBIT        41432
Name: count, dtype: int64
```

```
[3]: # İşlem türü ve dolandırıcılık arasındaki ilişkiyi görselleştir
plt.figure(figsize=(8, 5))
```

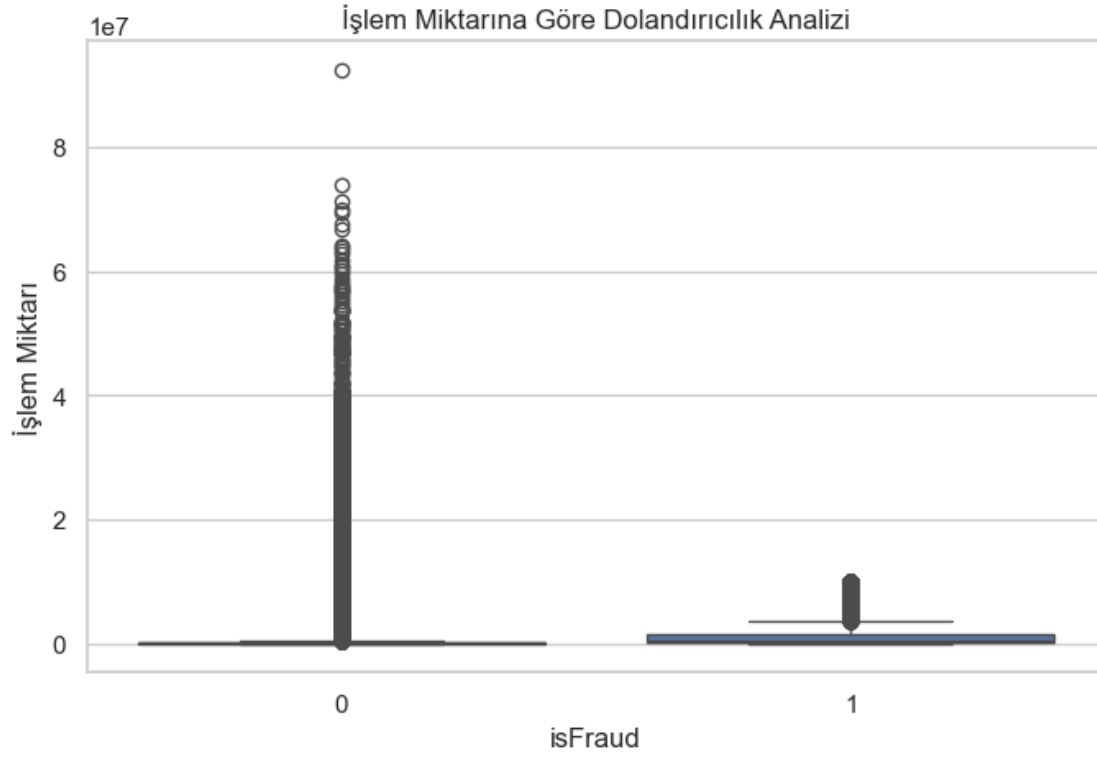
```
sns.countplot(data=df, x='type', hue='isFraud')
plt.title("İşlem Türüne Göre Dolandırıcılık Dağılımı")
plt.xlabel("İşlem Türü")
plt.ylabel("İşlem Sayısı")
plt.legend(title="isFraud", labels=["Normal", "Fraud"])
plt.show()
```



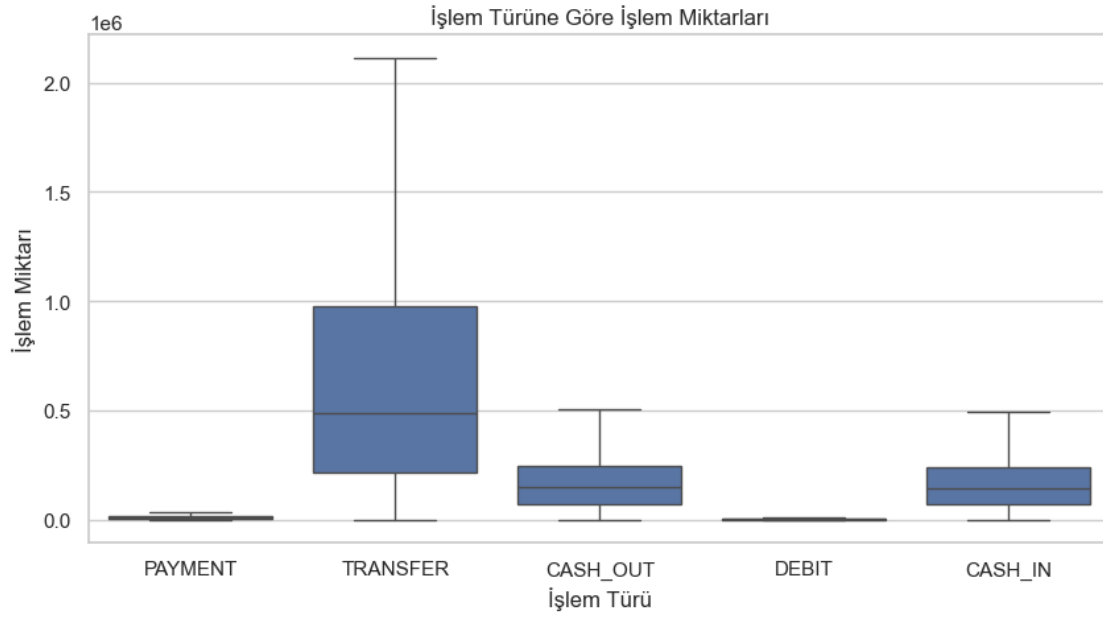
```
[4]: # Zaman dilimine (step) göre dolandırıcılık olaylarını incele
fraud_by_step = df[df['isFraud'] == 1].groupby('step')['isFraud'].count()
plt.figure(figsize=(10, 5))
fraud_by_step.plot(kind='line', color='red', label='Fraud')
plt.title("Zaman Dilimine Göre Dolandırıcılık Dağılımı")
plt.xlabel("Zaman Dilimi (Step)")
plt.ylabel("Dolandırıcılık Olay Sayısı")
plt.legend()
plt.show()
```



```
[5]: # İşlem miktarı ve dolandırıcılık ilişkisinin görselleştirilmesi
plt.figure(figsize=(8, 5))
sns.boxplot(data=df, x='isFraud', y='amount')
plt.title("İşlem Miktarına Göre Dolandırıcılık Analizi")
plt.xlabel("isFraud")
plt.ylabel("İşlem Miktarı")
plt.show()
```

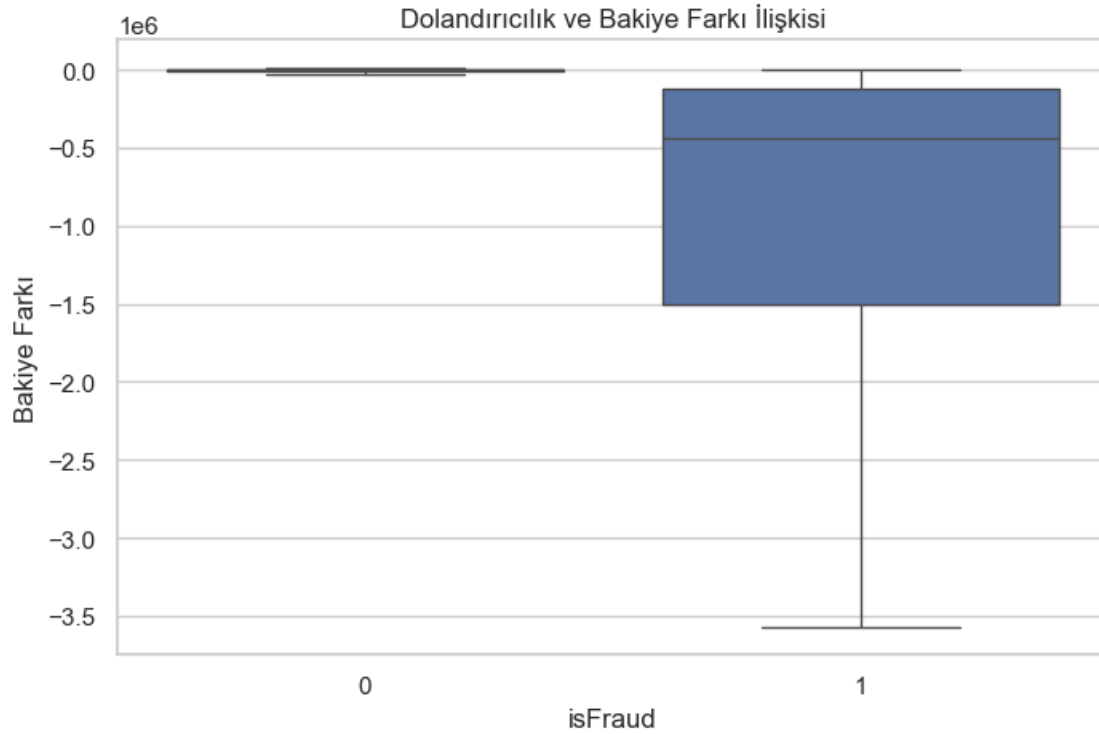


```
[6]: # İşlem türüne göre miktarların dağılımı
plt.figure(figsize=(10, 5))
sns.boxplot(data=df, x='type', y='amount', showfliers=False)
plt.title("İşlem Türüne Göre İşlem Miktarları")
plt.xlabel("İşlem Türü")
plt.ylabel("İşlem Miktarı")
plt.show()
```

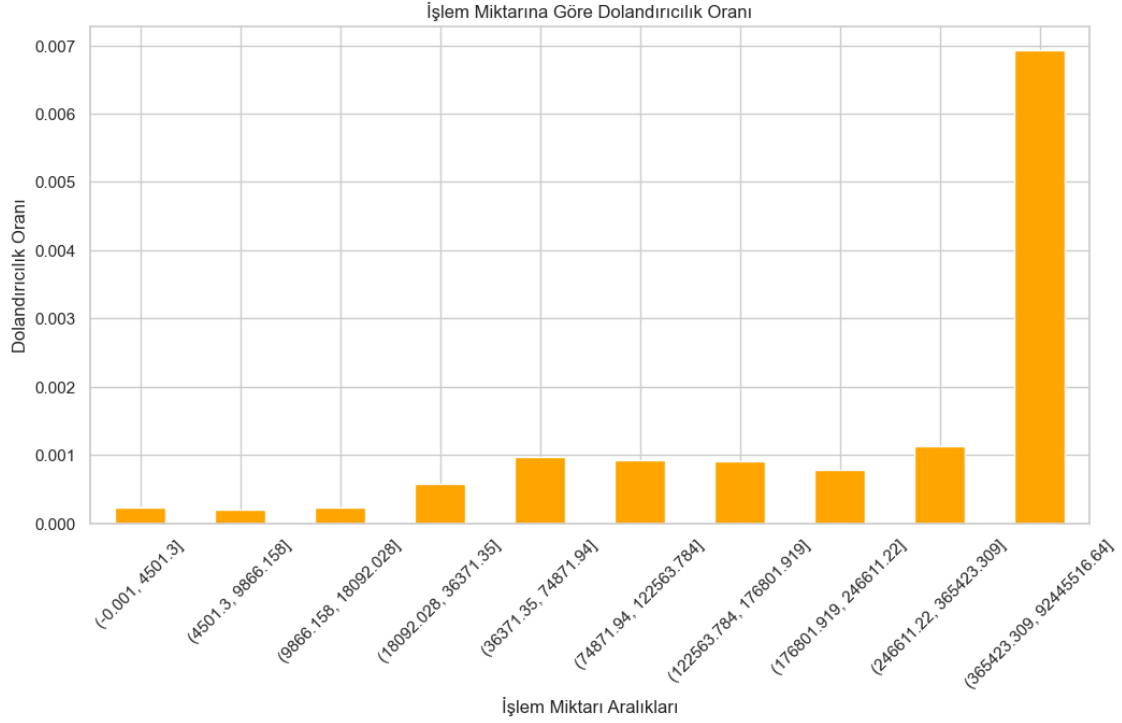


```
[7]: # Oldbalance ve Newbalance arasındaki farkı ekle
df['balance_diff'] = df['newbalanceOrig'] - df['oldbalanceOrg']

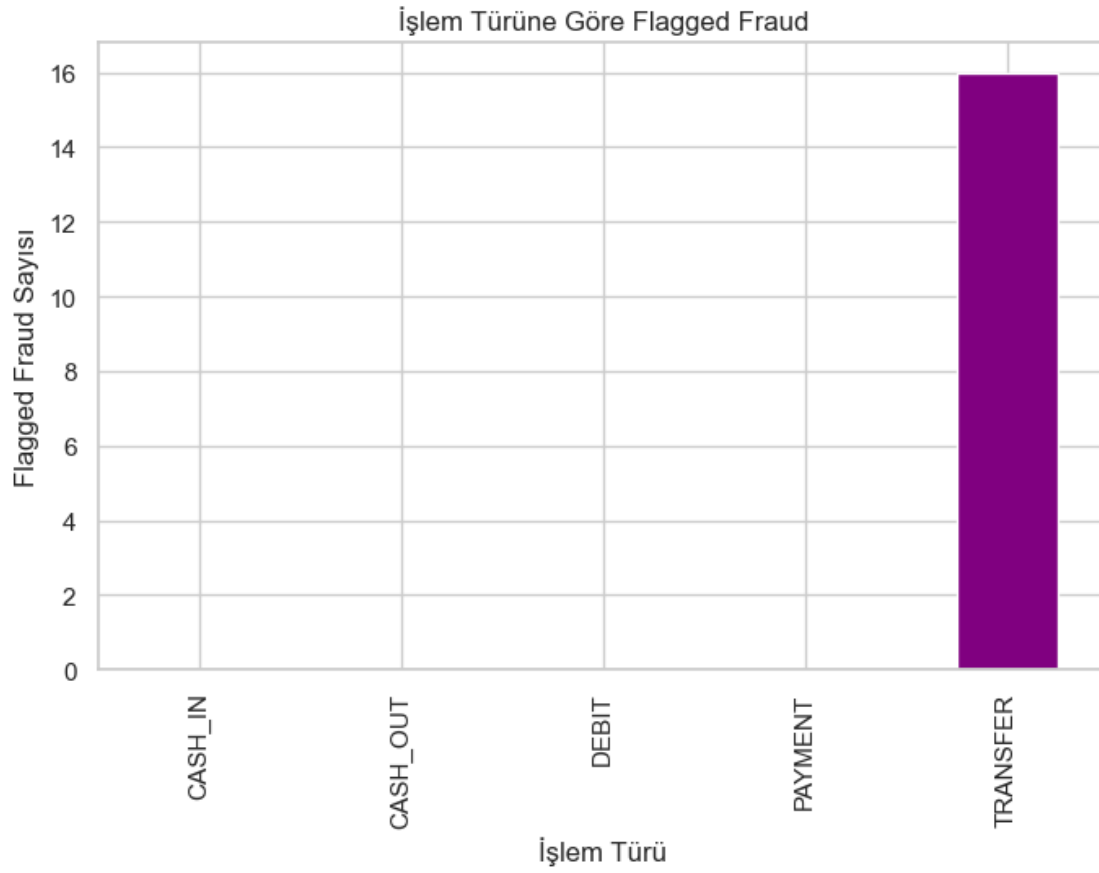
# Dolandırıcılık ve balance farkı ilişkisi
plt.figure(figsize=(8, 5))
sns.boxplot(data=df, x='isFraud', y='balance_diff', showfliers=False)
plt.title("Dolandırıcılık ve Bakiye Farkı İlişkisi")
plt.xlabel("isFraud")
plt.ylabel("Bakiye Farkı")
plt.show()
```



```
[8]: # İşlem tutarına bağlı dolandırıcılık oranı
df['amount_bins'] = pd.qcut(df['amount'], q=10, duplicates='drop')
fraud_rate = df.groupby('amount_bins', observed=False)['isFraud'].mean()
plt.figure(figsize=(12, 6))
fraud_rate.plot(kind='bar', color='orange')
plt.title("İşlem Miktarına Göre Dolandırıcılık Oranı")
plt.xlabel("İşlem Miktarı Aralıkları")
plt.ylabel("Dolandırıcılık Oranı")
plt.xticks(rotation=45)
plt.show()
```

```
[9]: # İşlem türüne göre flaggedFraud analizi
flagged_analysis = df.groupby('type')['isFlaggedFraud'].sum()
plt.figure(figsize=(8, 5))
flagged_analysis.plot(kind='bar', color='purple')
plt.title("İşlem Türüne Göre Flagged Fraud")
plt.xlabel("İşlem Türü")
plt.ylabel("Flagged Fraud Sayısı")
plt.show()
```



```
[10]: # Veri setindeki korelasyon matrisini incele
numeric_cols = df.select_dtypes(include=['number'])
corr = numeric_cols.corr()
plt.figure(figsize=(10, 8))
sns.heatmap(corr, annot=True, cmap='coolwarm', fmt='.2f')
plt.title("Korelasyon Matrisi")
plt.show()
```

