

ESKİŞEHİR'DE EMLAK GRUPLANDIRMASI

BATUHAN KIŞLACI



Projenin Amacı

Fiyat algımızı kaybettiğimiz bu günlerde sürekli bir değişim içerisinde olan emlak piyasasını ilk aşamada kategorik olarak ayırıp ortalama fiyat aralığının tahmini.



Veri Toplama

İlan Linklerinin Elde Edilmesi

```
async def fetch_links():
    base_url = "https://www.hepsimlak.com/eskisehir-kiralik"
    max_pages = 100
    async with async_playwright() as p:
        browser = await p.chromium.launch()
        page = await browser.new_page()
        all_links = []
        for n in range(max_pages):
            url = f"{base_url}?page={n}"
            print(f"Fetching: {url}")
            await page.goto(url)
            links = await page.eval_on_selector_all('a[href]', "els => els.map(el => el.href)")
            all_links.extend(links)
        await browser.close()
    return all_links
```

	Link
0	https://www.hepsimlak.com/eskisehir-tepebasi-...
1	https://www.hepsimlak.com/eskisehir-odunpazar-...
2	https://www.hepsimlak.com/eskisehir-tepebasi-...
3	https://www.hepsimlak.com/eskisehir-tepebasi-...
4	https://www.hepsimlak.com/eskisehir-tepebasi-...
5	https://www.hepsimlak.com/eskisehir-tepebasi-...
6	https://www.hepsimlak.com/eskisehir-tepebasi-...
7	https://www.hepsimlak.com/eskisehir-odunpazar-...
8	https://www.hepsimlak.com/eskisehir-tepebasi-...
9	https://www.hepsimlak.com/eskisehir-odunpazar-...

Verinin Toplanması



Dev Foto

1 / 35

121754-545



16.500 TL ⌚

Eskişehir / Tepebaşı / Ömerağa Mah.

İlan no	121754-545
Son Güncelleme Tarihi	27-12-2024
İlan Durumu	Kiralık
Konut Tipi	Daire
Konut Şekli	Daire
Oda + Salon Sayısı	3 + 1
Banyo Sayısı	1
Brüt / Net M2	130 m2 / 110 m2
Kat Sayısı	8 Katlı
Bulunduğu Kat	8. Kat
Bina Yaşı	25 Yaşında
Isınma Tipi	Kombi

```
df = pd.read_excel('hepsiemlaklink.xlsx')
newlist = []
async def extract_val_attributes(myurl,counter):
    async with async_playwright() as p:
        browser = await p.chromium.launch(headless=True)
        page = await browser.new_page()

        await page.goto(myurl)
        await asyncio.sleep(5)
```

```
desired_fields = [
    "Son Güncelleme Tarihi", "Konut Tipi", "Oda + Salon Sayısı",
    "Banyo Sayısı", "Brüt M2", "Kat Sayısı", "Bulunduğu Kat",
    "Bina Yaşı", "Isınma Tipi", "Eşya Durumu", "Kullanım Durumu", "Cephe"
]
```

```
data = await page.evaluate(f'''() => {{
    const items = document.querySelectorAll("li.spec-item");
    const fieldValues = Array.from(items)
        .map(item => {{
            const field = item.querySelector("span.txt")?.textContent.trim();
            const valueElement = item.querySelector("span:not(.txt), a");
            const value = valueElement?.textContent.trim();
            return {{ field, value }};
        }})
        .filter(item => item.field && {desired_fields}.includes(item.field));

    // Extract price value
    const priceElement = document.querySelector("p.fz24-text.price");
    const price = priceElement?.textContent.trim();

    // Extract title value
    const titleElement = document.querySelector("h1.fontRB");
    const title = titleElement?.textContent.trim();

    return {{ fieldValues, price, title }};
}}''')
```

```
newlist.append(data)
```

```
print(f"Counter: {counter}, Extracted data: {data}")
```

İlk Bakış

	fieldValues	price	title	Lokasyon
0	[{"field": "Son Güncelleme Tarihi", "value": "...	12.500 TL	CONKER INSAAT EMLAKTAN ERTUGRULGAZI MAHALLESİ...	Eskişehir / Tepebaşı / Ertuğrulgazi Mah.
1	[{"field": "Son Güncelleme Tarihi", "value": "...	16.000 TL	SAHIBINDEN CARSAMBA PAZARI BASINDA 3+1 BAKIMLI...	Eskişehir / Odunpazarı / Akarbaşı Mah.
2	[{"field": "Son Güncelleme Tarihi", "value": "...	9.000 TL	ILETISIM KAPISI YAKINI 2 ODA ANTRE MUTFAKLI AP...	Eskişehir / Tepebaşı / Şirintepe Mah.
3	[{"field": "Son Güncelleme Tarihi", "value": "...	13.000 TL	SIRINTEPE MAHALLESİ SEMT PAZARI YAKINI ARAKAT ...	Eskişehir / Tepebaşı / Şirintepe Mah.
4	[{"field": "Son Güncelleme Tarihi", "value": "...	10.000 TL	SUTLUCE MAHALLESİ ZIYA GOKALP OKULU YAKINI KIR...	Eskişehir / Tepebaşı / Sütüce Mah.
...
1369	[{"field": "Son Güncelleme Tarihi", "value": "...	9.000 TL	SIRINTEPE MAHALLESİ ECZACILIK YAKINI ESYALI KI...	Eskişehir / Tepebaşı / Şirintepe Mah.
1370	[{"field": "Son Güncelleme Tarihi", "value": "...	13.250 TL	BUYUKDERE MAHALLESİ KIRALIK 1+1 DUBLEKS	Eskişehir / Odunpazarı / Büyükdere Mah.
1371	[{"field": "Son Güncelleme Tarihi", "value": "...	9.000 TL	SUTLUCE'DE GENIS ESYALI KIRALIK 1+1	Eskişehir / Tepebaşı / Sütüce Mah.
1372	[{"field": "Son Güncelleme Tarihi", "value": "...	10.000 TL	ERTUGRULGAZI MH 1+1 ESYALI KIRALIK APART DAIRE	Eskişehir / Tepebaşı / Ertuğrulgazi Mah.
1373	[{"field": "Son Güncelleme Tarihi", "value": "...	12.000 TL	AS YONTEM DEN TUNALI MAHALLESİNDE ESYALI 1+1 DAIRE	Eskişehir / Tepebaşı / Tunalı Mah.

1374 rows × 4 columns

Veri Ön işleme

Verilerin Parçalanması

```
df[['İl', 'İlçe', 'Mahalle']] = df['Lokasyon'].str.split(' / ', expand=True)
df.drop('Lokasyon', axis=1, inplace=True)

split_data = df['fieldValues'].str.split('}', {'', expand=True)

for col in split_data.columns:
    split_data[col] = split_data[col].str.extract(r"'value': (.+)")
    split_data[col] = split_data[col].str.replace("'", "")
    split_data[col] = split_data[col].str.replace('"', '')
    split_data[col] = split_data[col].str.replace("=", "")
    split_data[col] = split_data[col].str.replace("[", "")
    split_data[col] = split_data[col].str.replace("]", "")
    split_data[col] = split_data[col].str.replace("}", "")
    split_data[col] = split_data[col].str.replace("{", "")

split_data.columns = [
    "Son Güncelleme Tarihi", "Konut Tipi", "Oda + Salon Sayısı",
    "Banyo Sayısı", "Brüt M2", "Kat Sayısı", "Bulunduğu Kat", "Bina Yaşı",
    "Isınma Tipi", "Eşya Durumu", "Kullanım Durumu", "Cephe"
]
```

	Son Güncelleme Tarihi	Konut Tipi	Oda + Salon Sayısı	Banyo Sayısı	Brüt M2	Kat Sayısı	Bulunduğu Kat	Bina Yaşı	Isınma Tipi	Eşya Durumu	Kullanım Durumu	Cephe
0	05-12-2024	Daire	2 + 1	1	130 m2	2 Katlı	2. Kat	15 Yaşında	Kombi	Eşyalı Değil	Boş	Doğu, Batı
1	31-10-2024	Daire	3 + 1	2	130 m2	4 Katlı	Ara Kat	35 Yaşında	Merkezi	Eşyalı Değil	Boş	Güney, Doğu
2	05-12-2024	Daire	1 + 1	1	60 m2	3 Katlı	1. Kat	5 Yaşında	Kombi	Eşyalı Değil	Boş	NaN
3	05-12-2024	Daire	2 + 1	1	120 m2	3 Katlı	2. Kat	16 Yaşında	Kombi	Boş	NaN	NaN
4	05-12-2024	Daire	1 + 1	1	50 m2	3 Katlı	3. Kat	4 Yaşında	Kombi	Eşyalı Değil	Boş	NaN

Verilerin Detaylı İncelenmesi

- Kolonların satırlarında kayma bulunmakta
- Veriler m2, Katlı vs gibi anahtar kelimeler içerdikleri için tamsayıya çevrilmiyor

```
object_columns = split_data.select_dtypes(include=['object']).columns

for col in object_columns:
    print(f"Domains for column '{col}':")
    print(split_data[col].unique())
    print("-" * 20)
```

Domains for column 'Oda + Salon Sayısı':

```
['2 + 1' '3 + 1' '1 + 1' '2 + 0' '1 + 0' '4 + 1' '3 + 0' '5 + 1' '1 + 109'
 '7 + 2' nan '1 + 3' '2 + 2']
```

Domains for column 'Banyo Sayısı':

```
['1' '2' '100 m2' '125 m2' '3' '160 m2' '135 m2' '50 m2' '95 m2' '110 m2'
 '4' '90 m2' nan '55 m2' '65 m2' '130 m2' '88 m2' '54 m2' '85 m2' '70 m2']
```

Domains for column 'Brüt M2':

```
['130 m2' '60 m2' '120 m2' '50 m2' '100 m2' '150 m2' '160 m2' '108 m2'
 '140 m2' '125 m2' '138 m2' '90 m2' '83 m2' '155 m2' '145 m2' '55 m2'
 '65 m2' '110 m2' '115 m2' '94 m2' '75 m2' '5 Katlı' '4 Katlı' '158 m2'
 '45 m2' '135 m2' '25 m2' '200 m2' '96 m2' '105 m2' '104 m2' '85 m2'
 '95 m2' '52 m2' '235 m2' '250 m2' '124 m2' '133 m2' '54 m2' '170 m2'
 '127 m2' '190 m2' '185 m2' '195 m2' '79 m2' '275 m2' '70 m2' '165 m2'
 '3 Katlı' '57 m2' '15 m2' '175 m2' '180 m2' '168 m2' '53 m2' '82 m2'
 '122 m2' '106 m2' '68 m2' '80 m2' '40 m2' '240 m2' '118 m2' '35 m2'
 '69 m2' '49 m2' '107 m2' '39 m2' '63 m2' '2 Katlı' '62 m2' '279 m2'
 '230 m2' '128 m2' '30 m2' '6 Katlı' '400 m2' '91 m2' '109 m2' '192 m2'
 '290 m2' '42 m2' '188 m2' '210 m2' '143 m2' '51 m2' nan '111 m2' '320 m2'
 '186 m2' '56 m2' '121 m2' '44 m2' '157 m2' '20 m2' '99 m2' '220 m2'
 '1 Katlı' '72 m2' '98 m2' '4 m2' '117 m2' '225 m2' '201 m2' '48 m2'
 '81 m2' '97 m2' '245 m2' '58 m2' '93 m2' '123 m2' '166 m2' '205 m2'
 '8 Katlı']
```

Domains for column 'Kat Sayısı':

```
['2 Katlı' '4 Katlı' '3 Katlı' '9 Katlı' '7 Katlı' '5 Katlı' '1 Katlı'
 '4. Kat' '12 Katlı' '10 Katlı' '6 Katlı' '8 Katlı' 'En Üst Kat' '2. Kat'
 '3. Kat' '11 Katlı' nan '1. Kat' '18 Katlı' '13 Katlı' 'Yüksek Giriş'
 '33 Katlı' '6. Kat']
```

```
valid_cephe_values = ['Doğu, Batı', 'Güney, Doğu', 'Kuzey, Güney, Batı', 'Kuzey, Doğu, Batı',  
                     'Kuzey, Doğu', 'Güney, Doğu, Batı', 'Kuzey, Batı', 'Kuzey, Güney, Doğu',  
                     'Güney, Batı', 'Güney', 'Batı', 'Doğu', 'Kuzey, Güney', 'Kuzey',  
                     'Kuzey, Güney, Doğu, Batı']
```

```
for index, row in split_data.iterrows():  
    if row['Kullanım Durumu'] in valid_cephe_values:  
        split_data.loc[index, 'Cephe'] = row['Kullanım Durumu']  
        split_data.loc[index, 'Kullanım Durumu'] = None  
    if row['Eşya Durumu'] in valid_cephe_values:  
        split_data.loc[index, 'Cephe'] = row['Eşya Durumu']  
        split_data.loc[index, 'Eşya Durumu'] = None
```

```
valid_kullanin_values = ['Boş', 'Mülk Sahibi', 'Kiracılı']
```

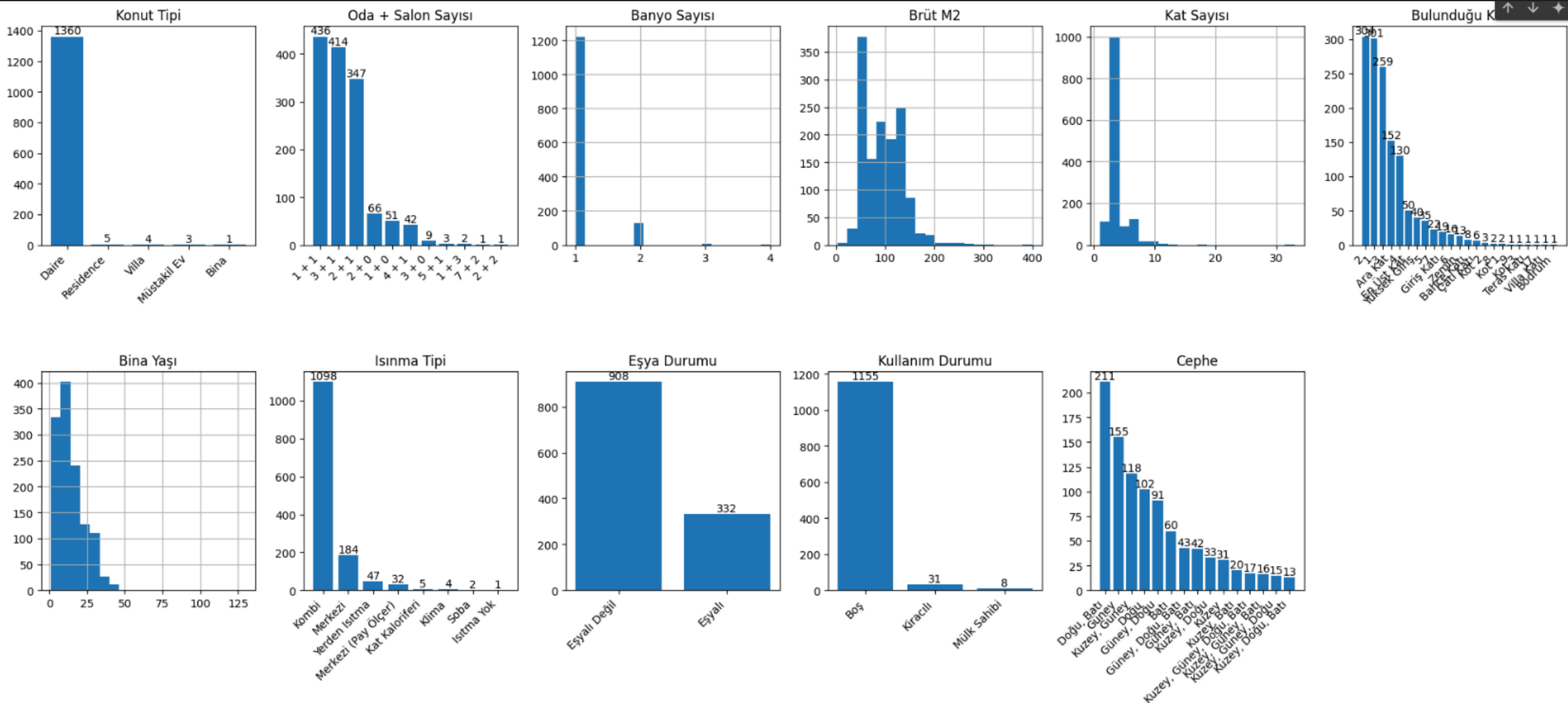
```
for index, row in split_data.iterrows():  
    if row["Eşya Durumu"] in valid_kullanin_values:  
        split_data.loc[index, "Kullanım Durumu"] = row["Eşya Durumu"]  
        split_data.loc[index, "Eşya Durumu"] = None  
    if row['Isınma Tipi'] in valid_kullanin_values:  
        split_data.loc[index, "Kullanım Durumu"] = row['Isınma Tipi']  
        split_data.loc[index, 'Isınma Tipi'] = None
```

```
valid_esya_values = ['Eşyalı Değil', 'Eşyalı']
```

```
for index, row in split_data.iterrows():  
    if row['Isınma Tipi'] in valid_esya_values:  
        split_data.loc[index, 'Eşya Durumu'] = row['Isınma Tipi']  
        split_data.loc[index, 'Isınma Tipi'] = None
```

Kolonlardaki Veri Kaymasını Düzeltilme

Verileri İnceleme



Tüm Verileri Tam Sayı Formatına Çevirme

```
split_data["Brüt M2"] = split_data["Brüt M2"].str.replace(" m2", "")
split_data["Kat Sayısı"] = split_data["Kat Sayısı"].str.replace(" Katlı", "")
split_data["Bulunduğu Kat"] = split_data["Bulunduğu Kat"].str.replace(". Kat", "")
split_data["Bina Yaşı"] = split_data["Bina Yaşı"].str.replace(" Yaşında", "")

split_data.loc[split_data["Konut Tipi"] != "Daire", "Konut Tipi"] = "Others"
split_data.loc[split_data["Isınma Tipi"] == "Merkezi (Pay Ölçer)", "Isınma Tipi"] = "Merkezi"
split_data.loc[~split_data["Isınma Tipi"].isin(["Kombi", "Merkezi"]), "Isınma Tipi"] = "Others"
split_data.loc[split_data["Kullanım Durumu"] != "Boş", "Kullanım Durumu"] = "Dolu"

split_data.loc[split_data["Eşya Durumu"] == "Eşyalı Değil", "Eşya Durumu"] = 0
split_data.loc[split_data["Eşya Durumu"] == "Eşyalı", "Eşya Durumu"] = 1

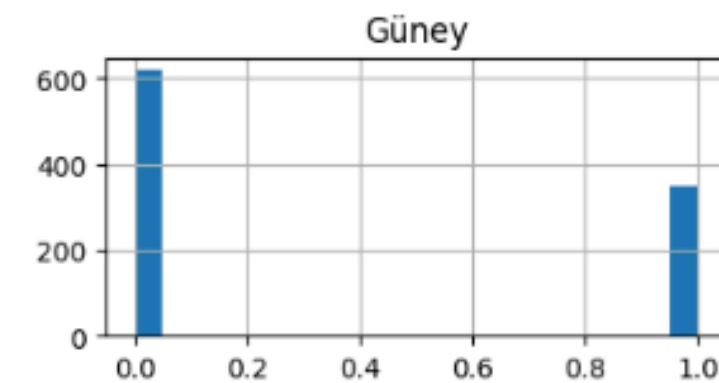
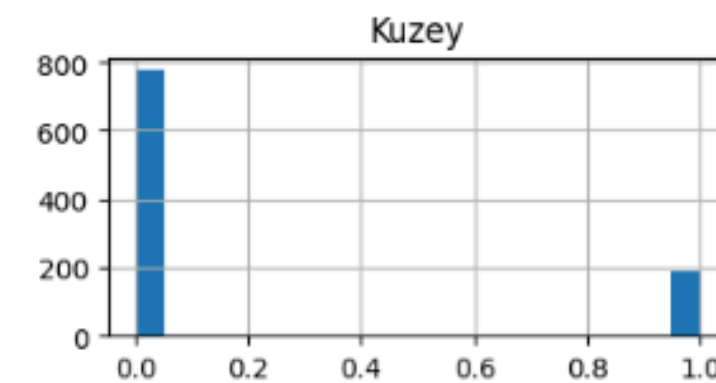
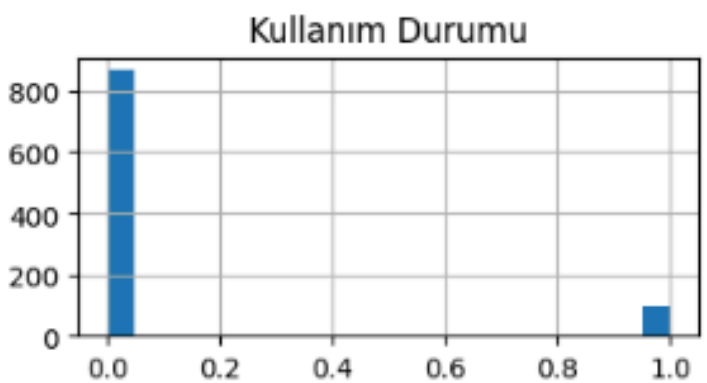
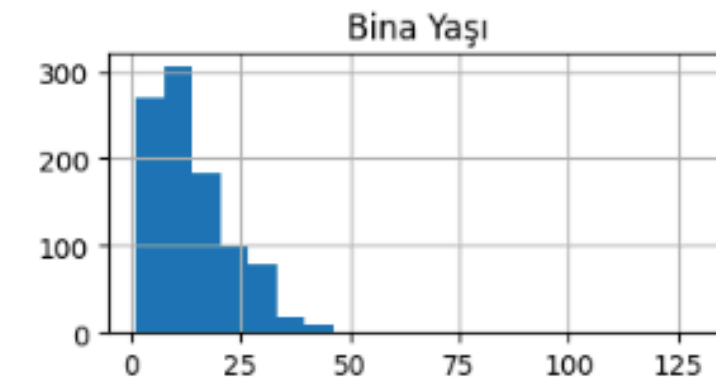
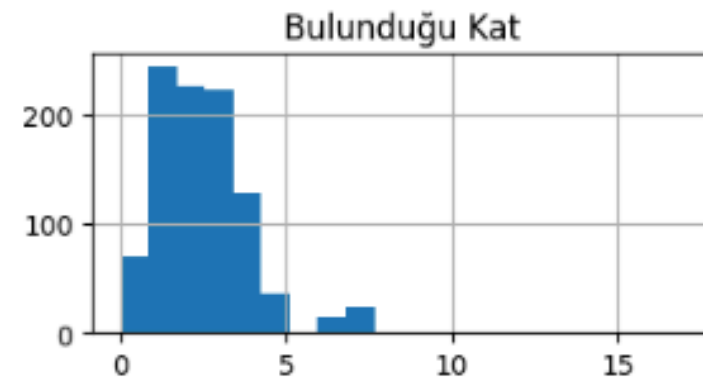
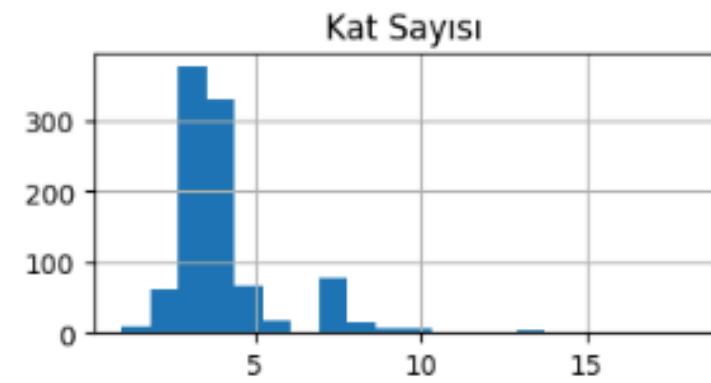
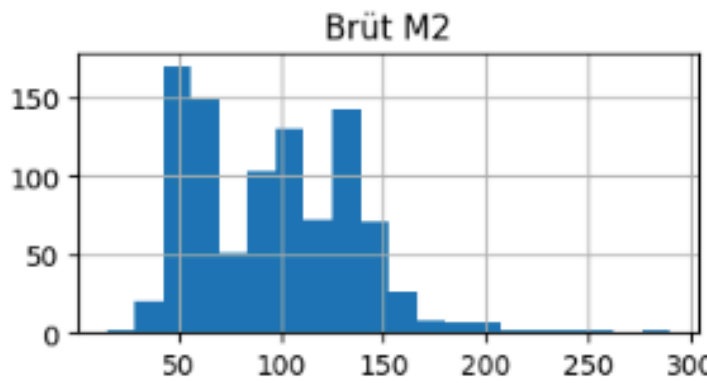
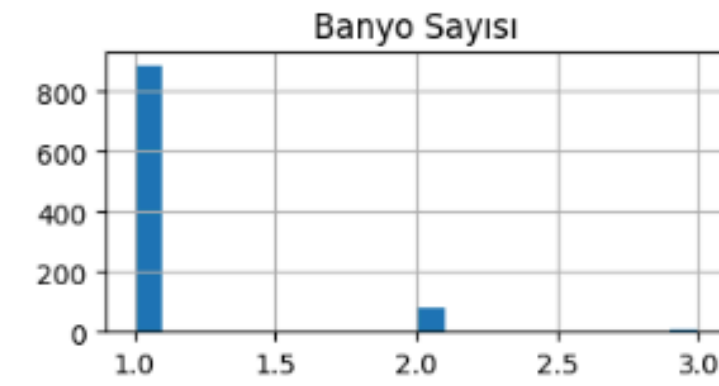
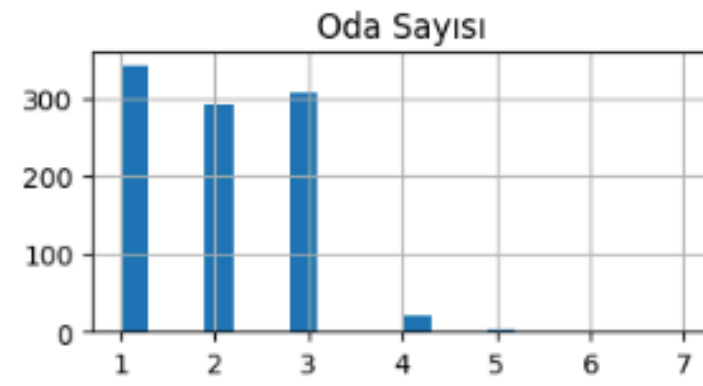
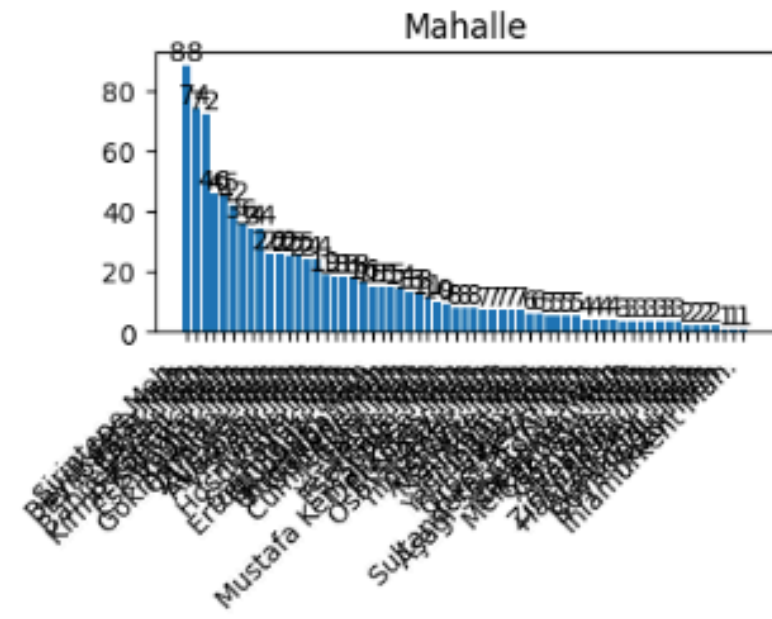
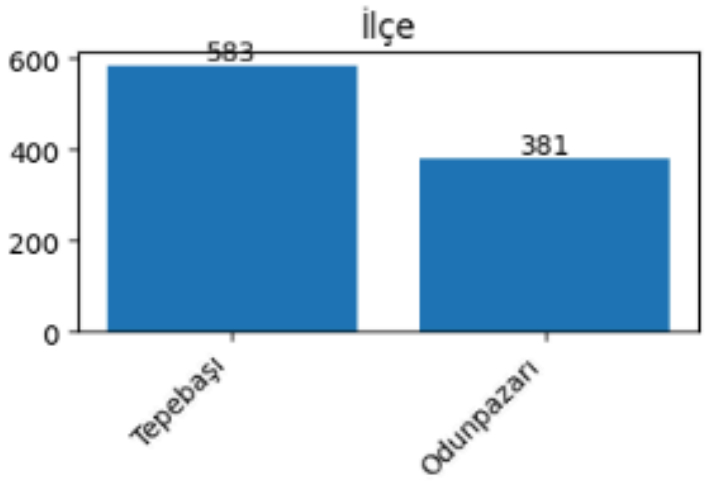
split_data.loc[split_data["Kullanım Durumu"] == "Dolu", "Kullanım Durumu"] = 1
split_data.loc[split_data["Kullanım Durumu"] == "Boş", "Kullanım Durumu"] = 0

merged_df = pd.concat([split_data, df.iloc[:, [1, 4, 5]]], axis=1)
merged_df["price"] = merged_df["price"].astype(str).str.replace(" TL", "")
merged_df["price"] = merged_df["price"].astype(str).str.replace(".", "")

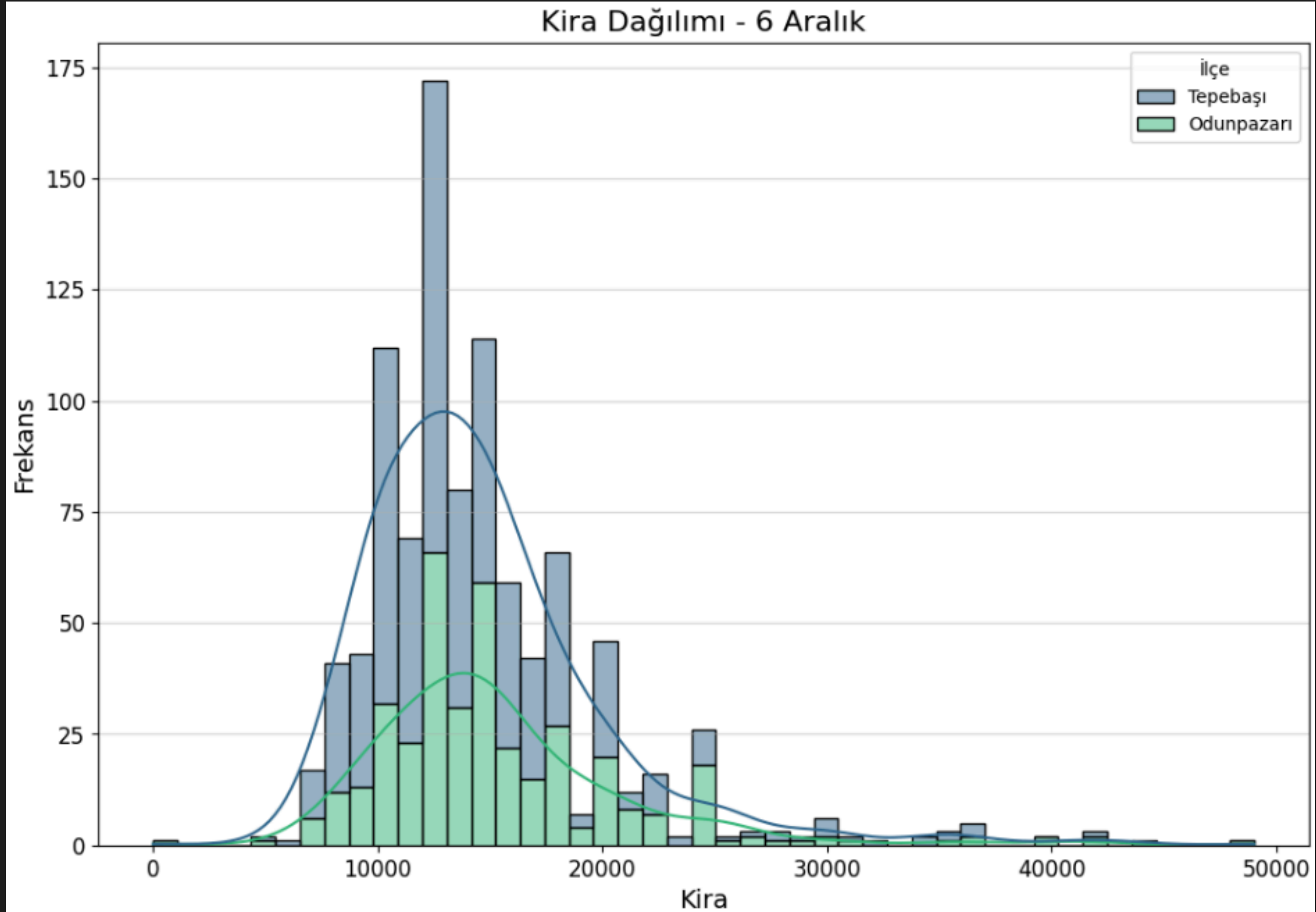
oda_salon = merged_df['Oda + Salon Sayısı'].str.split("+", expand=True)
merged_df['Oda Sayısı'] = oda_salon[0]
merged_df['Salon Sayısı'] = oda_salon[1]
merged_df = merged_df.drop('Oda + Salon Sayısı', axis=1)
```

Column	Non-Null Count	Dtype
-----	-----	-----
İlçe	964 non-null	object
Mahalle	964 non-null	object
Oda Sayısı	964 non-null	Int64
Salon Sayısı	964 non-null	Int64
Banyo Sayısı	964 non-null	Int64
Brüt M2	964 non-null	Int64
Kat Sayısı	964 non-null	Int64
Bulunduğu Kat	964 non-null	Int64
Bina Yaşı	964 non-null	Int64
Eşya Durumu	964 non-null	Int64
Kullanım Durumu	964 non-null	Int64
Doğu	964 non-null	int64
Batı	964 non-null	int64
Kuzey	964 non-null	int64
Güney	964 non-null	int64
Kira	964 non-null	Int64

Ön işleme Sonrası Verilerin Dağılımı



6 Aralık Eskişehir İlçelere Göre Kira Dağılımı



Sınıflandırma Modelinin Oluşturulması

Adım 1

Accuracy: %11

- Temel modelin oluşturulması

```
df = pd.read_csv("FinalDf.csv")
target_column = 'Kira'

X = df.drop(target_column, axis=1)
y = df[target_column]

X = pd.get_dummies(X, drop_first=True)

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

clf = DecisionTreeClassifier(random_state=42)
clf.fit(X_train, y_train)
```


Adım 2

Accuracy : %11 → %17

- 1000'erli gruplandırma

```
df['Kira'] = (df['Kira'] // 1000) * 1000
```

Adım 3

Accuracy : %17 → %50

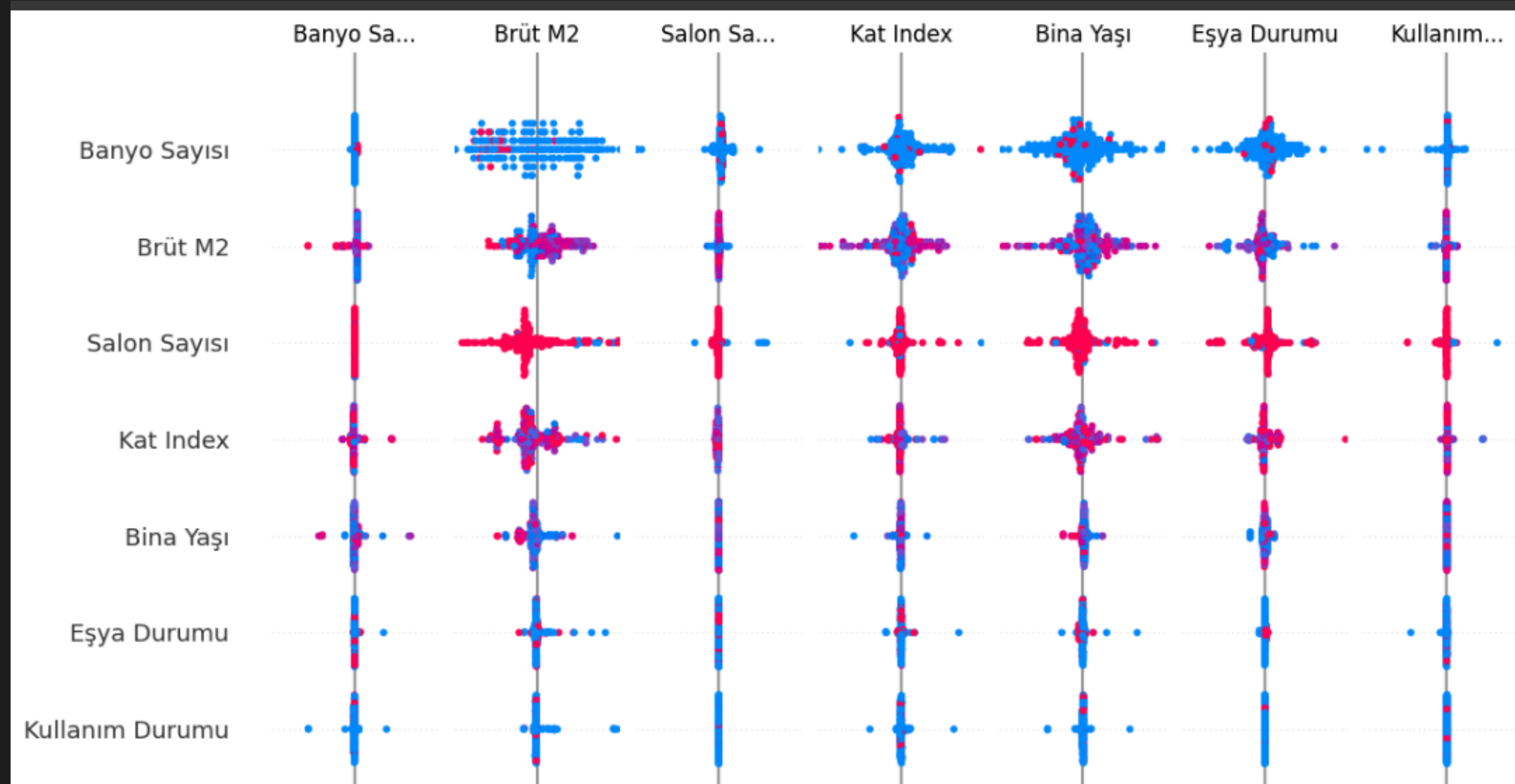
- 5000'erli gruplandırma

```
df['Kira'] = (df['Kira'] // 5000) * 5000
```

Adım 4

- ['Brüt M2', 'Kat Index', 'Bina Yaşı', 'Eşya Durumu'] sonucu en çok etkileyen kolonlar olarak saptandı

```
import shap
explainer = shap.TreeExplainer(clf)
shap_values = explainer.shap_values(X_test)
shap.summary_plot(shap_values, X_test)
```



Adım 5

Accuracy : %50 → %59

- Verilerin normalizasyonu ve hiperparametre optimizasyonu yapılması
- f1-score örneklem boyutu azaldıkça azalıyor

```
scaler = StandardScaler()
X = scaler.fit_transform(X)

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,

param_grid = {
    'n_estimators': [50, 100, 200],
    'max_depth': [None, 10, 20],
    'min_samples_split': [2, 5, 10],
    'min_samples_leaf': [1, 2, 4]
}

rf_classifier = RandomForestClassifier()
grid_search = GridSearchCV(estimator=rf_classifier, param_grid=param_grid, cv=5, scoring='accuracy', n_jobs=-1)
grid_search.fit(X_train, y_train)
best_rf_classifier = grid_search.best_estimator_
```

	precision	recall	f1-score	support
5000	0.44	0.39	0.41	18
10000	0.73	0.66	0.70	92
15000	0.51	0.65	0.57	54
20000	0.31	0.24	0.27	17
accuracy			0.59	181
macro avg	0.50	0.48	0.49	181
weighted avg	0.60	0.59	0.59	181

Adım 6

Accuracy : %59 → %88

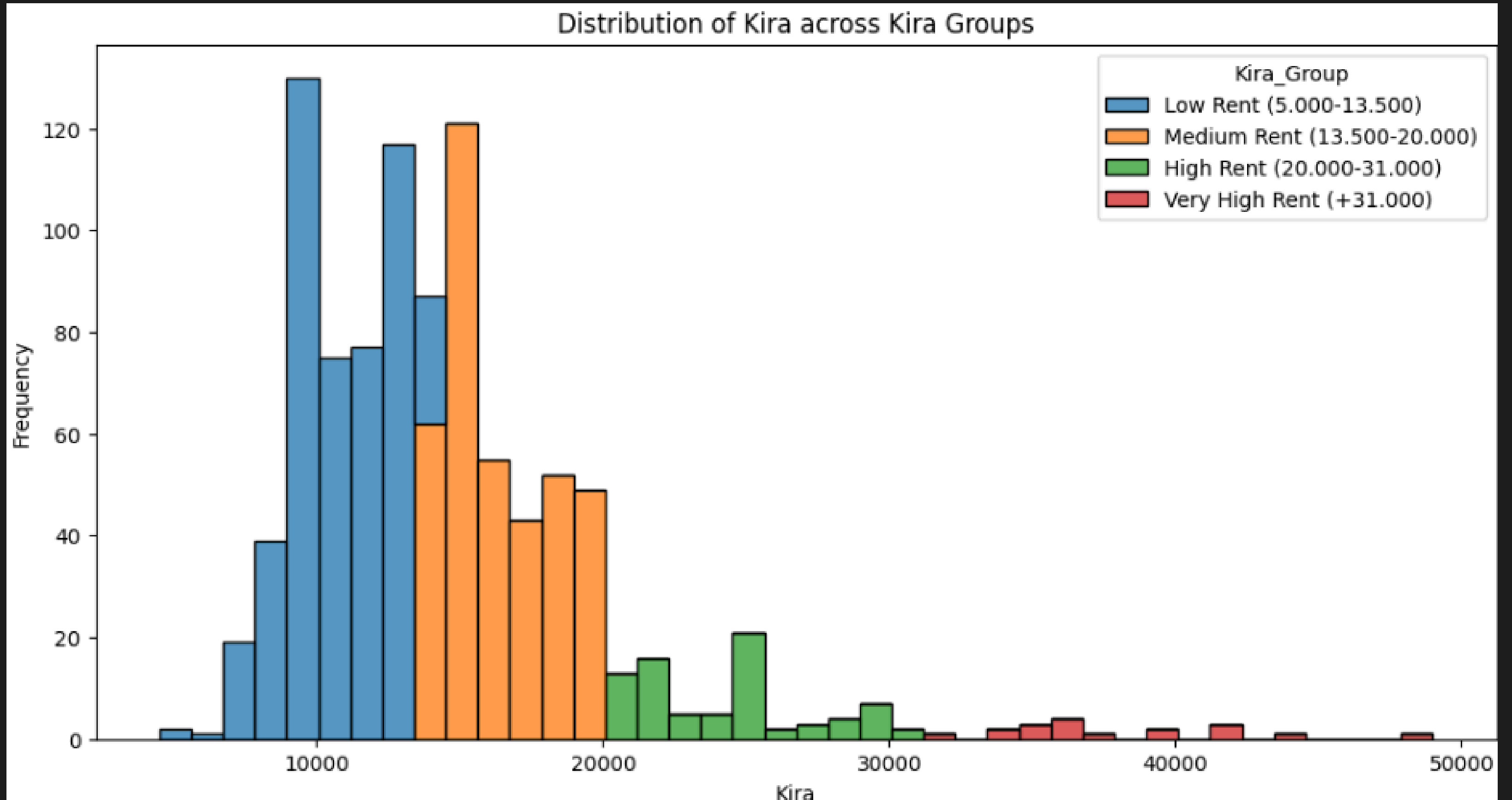
- KMeans ile veri setinin 4'e bölünmesi

```
kira_values = df['Kira'].values.reshape(-1, 1)
kmeans = KMeans(n_clusters=4, random_state=56)
df['Kira_Group_Cluster'] = kmeans.fit_predict(kira_values)
cluster_names = ["Low Rent (5.000-13.500)", "Medium Rent (13.500-20.000)", "High Rent (20.000-30.000)", "Very High Rent (30.000-40.000)"]
sorted_clusters = sorted(df['Kira_Group_Cluster'].unique())
cluster_mapping = {sorted_clusters[i]: cluster_names[i] for i in range(len(sorted_clusters))}
df['Kira_Group'] = df['Kira_Group_Cluster'].map(cluster_mapping)
df.drop(columns=['Kira_Group_Cluster'], inplace=True)
```

- SMOTE ile örneklem sayılarının eşitlenmesi

```
smote = SMOTE(sampling_strategy='all', random_state=56)
X_smote, y_smote = smote.fit_resample(X, y)
```

KMeans Gruplarının Dağılımı



Sonuçların Yorumlanması

- Model gayet iyi çalışıyor
- “Very High Rent”in %100 olasılıkla tahmin edilmesi SMOTE sırasında kullanılan başlangıç örnekleminin çok küçük olmasından kaynaklanıyor

	precision	recall	f1-score	support
High Rent(20.000-31.000)	0.72	0.80	0.76	90
Low Rent (5.000-13.500)	0.85	0.79	0.82	104
Medium Rent (13.500-20.000)	0.96	0.94	0.95	93
Very High Rent(+31.000)	1.00	1.00	1.00	101
accuracy			0.88	388
macro avg	0.88	0.88	0.88	388
weighted avg	0.89	0.88	0.88	388

**Dinlediğiniz için
Teşekkürler**