

MAKİNE ÖĞRENMESİ ÖĞRETİCİSİ-PROJE 2
MARKA BİLİNİRLİK ANALİZİ
PYTHON İLE



MAKİNE ÖĞRENMESİNE YOLCULUK 2

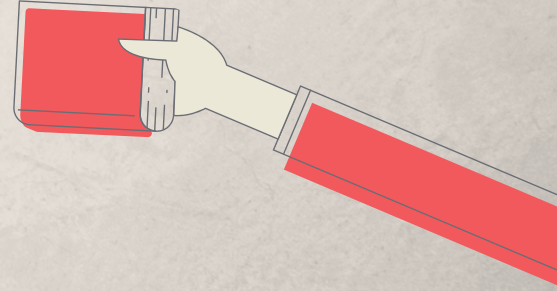


BERKANT ASLAN

ÖNSÖZ



Makine Öğrenmesini Python dili ve proje ile öğrenilmesini sağlamak için hazırlamış olduğum bu ikinci e-kitabımı sizlerle paylaşıyorum. Sorularınız ve tavsiyeleriniz için hello@berkantaslan.com olan mail adresime mail atabilirsiniz. Bu kitap üzerinde zamanla bazı güncellemeler olabilecek ve güncellemeler oldukça sizlerle paylaşacağım. Haberdar olabilmek için beni takip edebilirsiniz.



EŞİME...



Bu kitabın sizlere Makine Öğrenmesi konusunda pusula olması dileklerimle...



KİM BU BERKANT ASLAN YAA?

Selçuk Üniversitesi'nde Mekatronik Mühendisliği alanında Yapay Zeka konusunda yüksek lisans yapmakta; Yapay Zeka, Veri Bilimi, Görüntü İşleme, IoT, Büyük Veri, İş Zekası ve Siber Güvenlik ilgi alanları ve bu konularda tecrübeleri olan; insanları çok seven ve insanlığa kendisini adanmış; Türkiye aşığ; saygılı ve sevgili; Isparta'nın gülü :) ve eşiyle çok mutlu :)

 www.berkantaslan.com

 hello@berkantaslan.com

 [/berkantaslan](https://www.linkedin.com/company/berkantaslan)

 [@berkantaslan](https://twitter.com/berkantaslan)

 [/berkantaslan](https://www.youtube.com/channel/UCberkantaslan)

 [/berkantaslan](https://www.instagram.com/berkantaslan)

 [@berkantaslan](https://www.facebook.com/berkantaslan)

İÇİNDEKİLER

İÇİNDEKİLER.....	1
1. GİRİŞ	2
2. VERİ HAKKINDA	2
3. PROJE TANIMI.....	2
4. METODOLOJİ.....	2
4.1. Araştırma Metodoloji Özeti.....	2
4.2. CRISP-DM Metodu.....	3
4.3. Veriyi Anlamak	3
4.4. Veri Ön İşleme	4
4.3.1. Kütüphane ve Veri İçerik Aktarma	4
4.3.2. Veri Temizleme	5
4.3.3. Kullanıma Uygun Olarak Veriyi Düzenleme	5
4.3.4. Veri Setine Veri Ekleme.....	5
4.5. Keşif Analizi	5
4.6. Kullanılacak Yöntemlere Karar Verme	6
4.7. Test ve Eğitim Kümesi Olarak Veriyi Bölme.....	6
4.8. Özellik Ölçekleme.....	6
4.9. Sahte Değişken Tuzağına Dikkat Edin.....	7
5. VERİ ve PAZAR ANALİZİ	7
6. SONUÇ	73
7. KAYNAKLAR.....	73

1. GİRİŞ

A markasının stratejik hedefleri doğrultusunda hedef kitledeki algısını ölçümlemek ve elde edilen bulgular doğrultusunda markasını en iyi şekilde yönetebilmeyi planlamaktadır.

2. VERİ HAKKINDA

Anketörler aracılığıyla düzenlenen ankete katılan kullanıcıların sorularının yanıtlarını içeren dosyadaki veriler kullanılacaktır.

3. PROJE TANIMI

Gerçekleştirilen bu araştırma ile; hedef kitle nezdindeki A MARKASI markasının bilinirliği, rakiplerine göre konumu ve iletişim kampanyalarının marka üzerindeki etkisini ölçümlemek için makine öğrenimi ve istatistik ile yaklaşılmaktadır.

4. METODOLOJİ

4.1. Araştırma Metodoloji Özeti

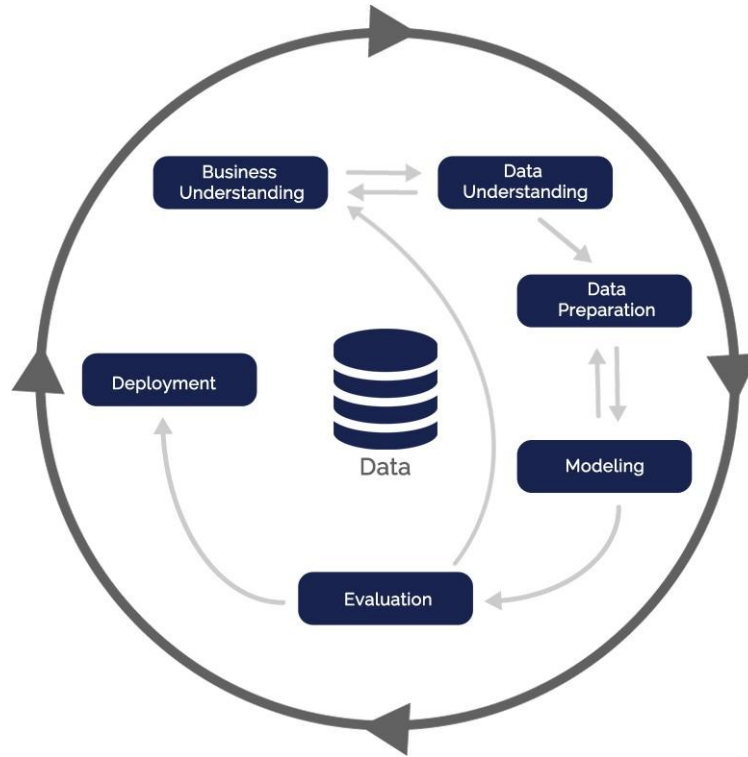
A markasının stratejik hedefleri doğrultusunda hedef kitledeki algısını ölçümlemek ve elde edilen bulgular doğrultusunda markasını en iyi şekilde yönetebilmeyi planlamaktadır. Gerçekleştirilen bu araştırma ile; hedef kitle nezdindeki A markasının bilinirliği, rakiplerine göre konumu ve iletişim kampanyalarının marka üzerindeki etkisini ölçümlemek mümkün olacaktır.



Görsel-1: Araştırma ile İlgili Bilgiler

4.2. CRISP-DM Metodu

Veri madenciliği projelerinin daha etkili, daha hızlı, daha güvenli, daha az maliyetli hale gelmesi için veri madenciliği için standart süreçleri tanımlayan bir model olan CRISP-DM (Cross Industry Standard Process for Data Mining) yöntemi. Bu yöntem, problemi ve ne yapmamız gerektiğini anlamakla başlar ve Makine Öğrenimi için veri hazırlama, modelleme ve değerlendirme ile devam eder.



Görsel-2: CRISP-DM Metodu Yapısı

4.3. Veriyi Anlamak

18-65 yaş aralığı, A, B, C1, C2, D, E SES grubu, %50 kadın, %50 erkek olmak üzere 1000 kişi ile görüşülmüştür. 21 Temmuz – 4 Ağustos 2022 tarih aralığında Kantitatif Araştırma ve CAWI (Bilgisayar İnternet Anketi) ile veriler toplanmıştır. Ancak bu çalışmada verilerin bir kısmı bulunmamaktadır. O yüzden veri seti bilgileri ilerleyen sayfalarda gösterilecektir.

4.4. Veri Ön İşleme

Veri ön işleme kısmı, verilerin işlenmesi için önemlidir. Bu bölüm kütüphaneyi ve veri içe aktarmayı, veri temizlemeyi, kullanılacak uygun veri düzenlemeyi, Makine Öğrenimi bölümüne geçmek için veri çerçevesine veri eklemeyi içerir. Makine Öğrenimi bölümü için ön işlemlere ihtiyacımız var.

4.4.1. Kütüphane ve Veri İçe Aktarma

Süreçlere göre hangi kütüphanelerin kullanılacağına karar veriyoruz. Python'da veri ve veri çerçevesi işlemleri için Pandas kütüphanesi, hesaplamalar için NumPy kütüphanesi, çizim için Matplotlib ve Seaborn kütüphanesi, "import" komutu ile Makine Öğrenimi işlemi için Sci-Kit Learn kütüphanesi kullanılır. İhtiyaç olan tüm kütüphaneler ihtiyaç oldukça da eklenebilecektir. Veri kullanımı csv (virgülle ayrılmış değerler), excel, html vb. olabilir ve "read_csv / excel (" dosya_adı ")" komutu ile alınır. Veri dosyası çalışma dizini olmalıdır. Çalışma dizininde değilse, dosya_adı için dosya dizinini yazabilirsiniz. Ve bu herhangi bir değişken olarak tanımlanır.

"AMarkas_data_need.xlsx" dosyası veri analizi için kullanıldı. Bu veriler üzerinden önemli olan ve sonucumuzu etkileyebilecek veriler bırakılarak gereksiz ve etkisiz değerler olduğunu tecrübem ve tahminen kaldırdığım veriler sonrasında güncel veri seti excel dosyası olarak AMarkas_data_need.xlsx ismiyle kaydedildi.

```
import numpy as np
import pandas as pd
import matplotlib
import matplotlib.pyplot as plt
import matplotlib.ticker as mtick
import matplotlib.gridspec as grid_spec
import seaborn as sns
import plotly.graph_objs as go
import plotly.offline as po
from bubbly.bubbly import bubbleplot
from plotly.offline import init_notebook_mode, iplot
from sklearn.impute import SimpleImputer
from sklearn.preprocessing import StandardScaler
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import classification_report
import plotly.graph_objects as go
```

```
from sklearn.metrics import confusion_matrix
from sklearn.metrics import roc_auc_score, roc_curve
import math
import re
from wordcloud import WordCloud

init_notebook_mode(connected=True)
data = pd.read_excel("AMarkas_data_needs.xlsx")
print(data.head())
```

4.4.2. Veri Temizleme

Bazı Makine Öğrenimi algoritmaları, eksik değerlere sahip verileri çalıştıramaz. İşlemler için düzeltilmesi gerekir. Eksik değerler NaN (sayı değil), "?", boşluk vb. görülebilir. Eksik sayısal değerler problemi sabit istatistiksel yöntemler (ortalama vb.), belirli sayılar koyarak veya satırları silerek çözülebilir.

Büyük-küçük harf veya kelimeler veya farklı diller arasındaki boşluk gibi sütun adı tutarsızlığı varsa, keşfetmeden önce verileri teşhis etmemiz gerekir. Büyük-küçük harf veya boşluk sorunu düzeltilmeli. Verilerimizin sütun adlarında büyük-küçük harf veya boşluk varsa, sütun adlarını değiştirmeliyiz. IDE'de kodlayarak veya Excel'de yazarak sütun adlarını değiştirebilirsiniz. Bu çalışmada böyle bir problem olmadığından bu konuda bir aksiyon alınmamıştır.

4.4.3. Kullanıma Uygun Olarak Veriyi Düzenleme

Veriler kategorik (nominal veya sıralı) veya sayısal (oran veya aralık) olabilir. Yalnızca sayısal verilere sahipsek, uygun veriler için düzenlememiz gerekmez. Kategorik veriler varsa bunlar nicel verilere dönüştürülmelidir.

4.4.4. Veri Setine Veri Ekleme

Eğer farklı bir veri eklemek gerekiyor olsaydı veri setimize uygun bir şekilde eklenmesi gerekecekti.

4.5. Keşif Analizi

Verilerimizi anlamak ve üzerinde düşünmek istiyoruz. İlk olarak, istatistiksel bilgiler, veri çerçevelerinin kaç satır ve sütun olduğu vb. verilerden özet ve bilgi alıyoruz. Ve verileri teşhis etmek için keşif analizine ihtiyaç vardır.

4.6. Kullanılacak Yöntemlere Karar Vermek

Birçok Makine Öğrenme Algoritması var ve projeye göre seçilmektedir. Tecrübelerime güvenerek iyi sonuç alacağımı düşündüğüm için projede RandomForestClassifier Makine Öğrenmesi methodu kullanılmıştır.

RandomForestClassifier, özellikle sınıflandırma problemleri için kullanılır. Eğer veri setinizde hedef değişken olarak sınıflar (kategorik değerler) bulunuyorsa ve bu sınıfları tahmin etmeye çalışıyorsanız (örneğin, "Düşük", "Yüksek" gibi), RandomForest uygun bir modeldir.

Avantajları:

- Kategorik hedef değişkenlerle iyi çalışır. Veri setindeki önemli özellikleri belirleyebilir (Feature Importance).
- Sınıflandırma problemlerinde çok güçlüdür ve genellikle yüksek doğruluk verir.

Ne Zaman Kullanılmalı?

- Hedef değişkeniniz sınıflar içeriyorsa (örneğin, müşteri segmentleri, satın alma kararları, güven ve tavsiye).
- Karar vermek için farklı özelliklerin önemini anlamak istiyorsanız.

4.7. Test ve Eğitim Kümesi Olarak Veriyi Bölme

Başarıyı ölçmek için test ve train kümesi olarak veri bölme yapılır. Makine Öğrenimi Algoritmasında amaç değerlendirme geliştirmektir. Genel olarak 1/3 testi ve 2/3 train için rastgele kullanılır, yüzdelik bölünme olarak bilinir. Makine öğrenimi algoritması train verileri üzerinde çalışır ve test verileri üzerinde tahmin yapar, tahmin ve test verilerinin sonuçları karşılaştırılır. Ve başarı görülür ve Makine Öğrenimi Yönteminin iyi olup olmadığına karar verebiliriz. Başarıya göre kullanılan Makine Öğrenimi Yöntemi değiştirebiliriz.

4.8. Özellik Ölçekleme

Özellik Ölçekleme süreci Makine Öğrenimi için önemlidir. Farklı sütunların farklı veri hareketleri ve istatistiksel özellikleri (ortalama, minimum / maksimum değerler, standart sapma vb.) vardır. Yani sütunların etkileri farklıdır ve bu bir problemdir. Özellik ölçeklendirmeyi kullanarak bu sorunu çözmeliyiz. Özellik Ölçeklendirme iki klasik yöntemle yapılır: Standardizasyon ve Normalleştirme. Normalleştirme yöntemini kullanıyoruz. Sayısal veriler 0 ile 1 arasında ölçeklenir.

$$Z = \frac{x - \mu}{\sigma} \quad Z = \frac{x - \min(x)}{[\max(x) - \min(x)]}$$

Formül-1: Standardizasyon ve Normalleştirme Formülleri

4.9. Sahte Değişken Tuzağına Dikkat Edin

Kukla değişken, veri çerçevesinin değişkenlerinin aynı değerlere sahip olduğu durumdur. Bazı Makine Öğrenimi Algoritmaları kukla değişkenden etkilenebilir. Veri çerçevesinde herhangi bir durum varsa, birini iptal etmek zorundayız. Örneğin, dize değerini sayısal değere dönüştürmek için erkek ve kadın değerleri 0 ve 1 olarak kodlanabilir. Ve bu iki kodlanmış sütunu kullanırsak sorun yaşayabiliriz. Çıktı, erkek veya kadın olabilen durumdan iki kez etkilenir. Sadece bir sütun yeterlidir ve ikincisi iptal edilir. Veri çerçevemiz için gerekli değildir.

5. VERİ ve PAZAR ANALİZİ

```
print(data.columns)
print(data.info())
print(data.describe())
```

Verilerimiz ile ilgili özet bilgilere (önizleme, count, mean, std, min, 25%, 50%, 75%, max) kod çıktısından ulaşıyoruz.

```
cinsiyet_kota = data['CinsiyetKota'].value_counts()
yas_kota = data['YasKota'].value_counts()
ses_quota = data['SESQuota'].value_counts()
maas_coded = data['MaasCoded'].value_counts()
il_kota = data['ilKota'].value_counts()

def plot_bar_chart(data, title, xtick_labels, rotate_xticks=False):
    total = data.sum()
    bar_positions = np.arange(len(data))

    plt.figure(figsize=(8, 6))
    bars = plt.bar(bar_positions, data.values,
                   align='center')
```

```

    if rotate_xticks:
        plt.xticks(bar_positions, labels=xtick_labels,
                    rotation=90)

    else:
        plt.xticks(bar_positions, labels=xtick_labels)

plt.title(title)

for bar in bars:
    height = bar.get_height()
    percentage = (height / total) * 100
    plt.text(bar.get_x() + bar.get_width() / 2, height,
             f'{percentage:.1f}%', ha='center', va='bottom')

plt.tight_layout()
plt.show()

yas_kota_labels = ['25-34', '35-44', '45-54', '18-24', '55+']
maas_coded_labels = ['7501-10000TL', '10001-15000TL', '5001-7500TL', '15001-20000TL',
                    '3001-5000TL', '3000 TL'den az', '20001-30000TL', '30001 TL'den fazla']
cinsiyet_kota_labels = ['Erkek', 'Kadın']
ses_quota_labels = ['D', 'C2', 'C1', 'AB']
il_kota_labels = ['İstanbul', 'Ege', 'Akdeniz', 'Güneydoğu Anadolu', 'Batı Anadolu',
                  'Doğu Marmara', 'Batı Karadeniz', 'Orta Anadolu', 'Ortadoğu Anadolu', 'Batı Marmara', 'Doğu Karadeniz', 'Kuzeydoğu Anadolu']

plt.figure(figsize=(8, 6))
plt.pie(cinsiyet_kota, labels=cinsiyet_kota_labels,
        autopct='%1.1f%%', startangle=90)
plt.title('Cinsiyet Kota Distribution')
plt.show()
plot_bar_chart(yas_kota, 'Yas Kota Distribution', yas_kota_labels)

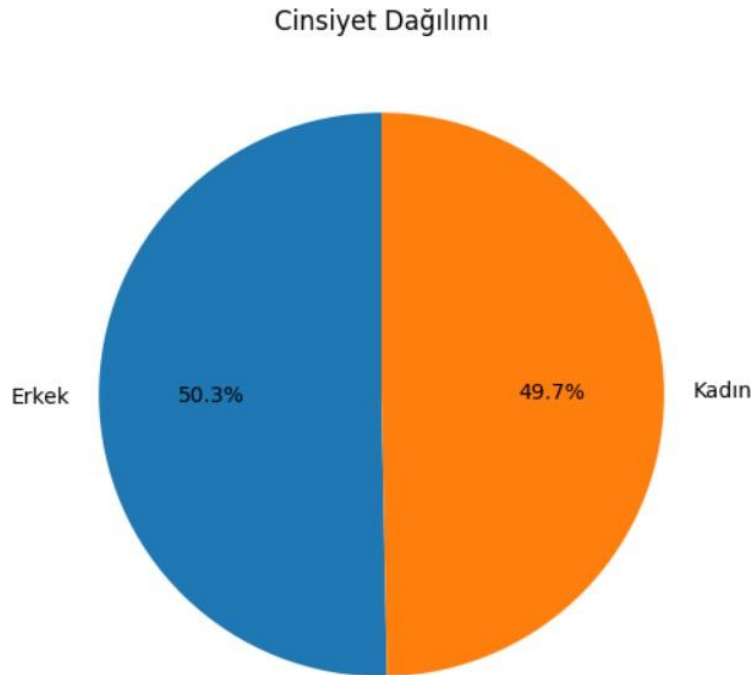
```

```
plt.figure(figsize=(8, 6))
plt.pie(ses_quota, labels=ses_quota_labels, autopct='%1.1f%%', startangle=90)
plt.title('SES quota Distribution')
plt.show()

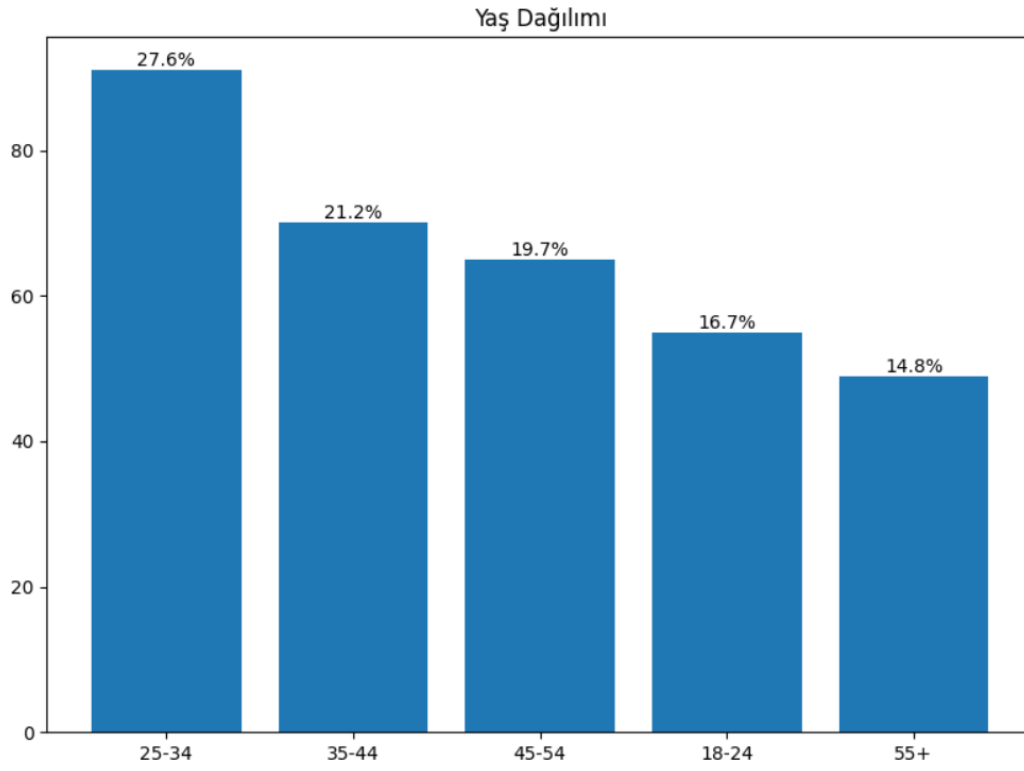
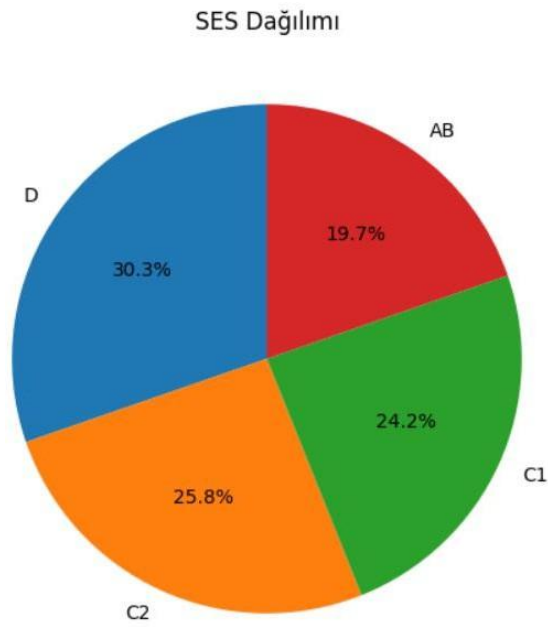
plot_bar_chart(maas_coded, 'Maas Coded Distribution',
               maas_coded_labels, rotate_xticks=True)

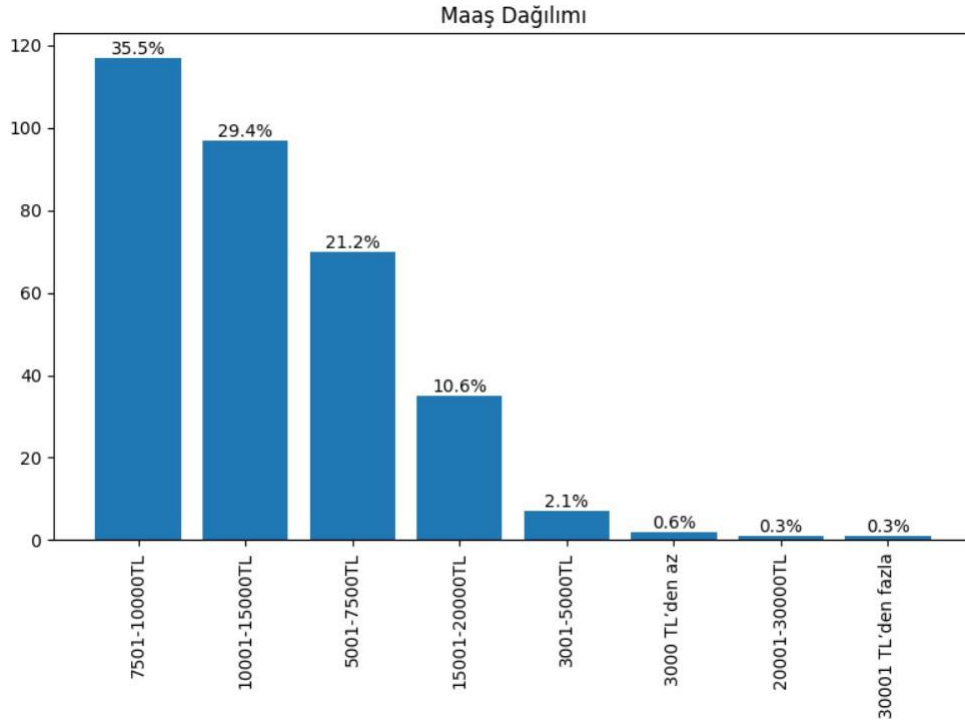
plot_bar_chart(il_kota, 'il Kota Distribution',
               il_kota_labels, rotate_xticks=True)
```

Bu kod çıktısı ile ankete katılan kişilerin demografik özelliklerine bir bakış atmaktayız. Aşağıdaki grafikler ve bilgilerden de görüldüğü üzere katılımcıların %49.7'si kadın, %50.3'ü erkek'tir. Katılımcıların %27.6'sı «25-34 yaş aralığında», %21.2'si «35-44» yaş aralığındadır. Görüşülen katılımcılar ortalama 37 yaşındadır.

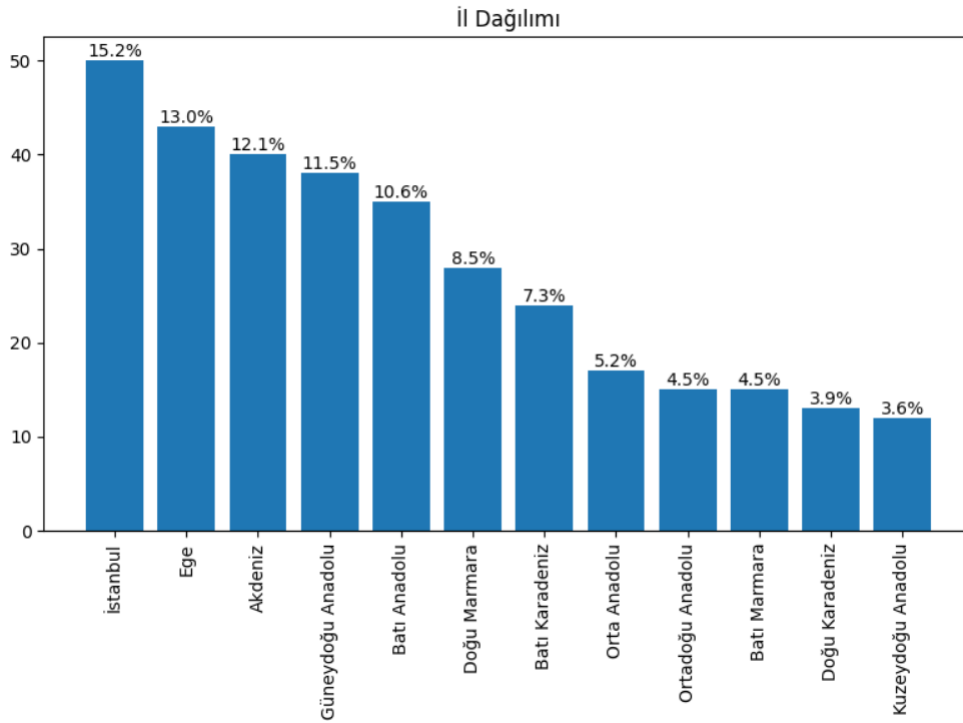


Grafik-1: Cinsiyet Dağılımı Pasta Grafiği

**Grafik-2:** Yaş Dağılımını Bar Grafiği**Grafik-3:** SES Quota Dağılımını Bar Grafiği



Grafik-4: Maaş Dağılımı Bar Grafiği



Grafik-5: İl Dağılımı Bar Grafiği

```
x1 = data.iloc[:, 6:].values
y1 = data.iloc[:, 4].values
y2 = data.iloc[:, 5].values
```

x1 A Markası'na olan güveni etkileyen girdiler ve y1 sonuç olan A Markası'na olan güveni 1-5 arasında puanlanan değerdir. y2 ise sonuç olan A Markası'nı ne kadar tavsiye ettiğini 1-10 arasında puanlanan değerdir.

```
y1histogram = pd.cut(y1, bins=[1, 3, 5], labels=['Düşük', 'Yüksek'])
y2histogram = pd.cut(y2, bins=[1, 7, 10], labels=['Düşük', 'Yüksek'])
```

```
print(pd.DataFrame(x1).isnull().sum())
print(pd.DataFrame(y1).isnull().sum())
print(pd.DataFrame(y2).isnull().sum())
```

```
imputer = SimpleImputer(missing_values=np.nan, strategy='mean')
x1 = imputer.fit_transform(x1)
```

```
y1 = pd.Series(y1histogram).ffill()
y2 = pd.Series(y2histogram).ffill()
```

Boş olan değerler sonucun fazla etkilenmemesi için ortalama değer ile doldurulmuştur.

```
from sklearn.model_selection import train_test_split
x_train1, x_test1, y_train1, y_test1 = train_test_split(x1, y1, test_size=0.33,
random_state=42)
x_train2, x_test2, y_train2, y_test2 = train_test_split(x1, y2, test_size=0.33,
random_state=42)
```

Kod çıktısı olarak test ve eğitim kümeleri oluşturulur.

```
model = RandomForestClassifier()
model2 = RandomForestClassifier()
model.fit(x_train1, y_train1)
model2.fit(x_train2, y_train2)
```

```

y_pred = model.predict(x_test1)
y_pred2 = model2.predict(x_test2)

print(classification_report(y_test1, y_pred))
print(classification_report(y_test2, y_pred2))

```

	precision	recall	f1-score	support
Düşük	1.00	0.30	0.46	10
Yüksek	0.93	1.00	0.97	99
accuracy			0.94	109
macro avg	0.97	0.65	0.71	109
weighted avg	0.94	0.94	0.92	109

	precision	recall	f1-score	support
Düşük	0.81	0.74	0.77	23
Yüksek	0.93	0.95	0.94	86
accuracy			0.91	109
macro avg	0.87	0.85	0.86	109
weighted avg	0.91	0.91	0.91	109

Tablo-1: Model Doğruluğu Analizi

Sonuçları incelediğimizde F1-score ve precision değerlerinin 1'e yakın olması kullandığımız algoritmamızın ve modelimizin doğru olduğunu göstermektedir.

Feature Importance (Özelliklerin Önemi):

RandomForestClassifier, her bir özelliğin sınıflandırmaya katkısını ölçer ve bu katkıyı özellik önem sıralaması (Feature Importance) ile sunar. Bu sayede, hangi demografik verilerin veya marka bilinirliklerinin tahminlerde daha etkili olduğunu anlayabiliriz. Özellikle pazar analizi yaparken, hangi faktörlerin müşteri tercihlerini en çok etkilediğini görmek çok değerlidir.

```

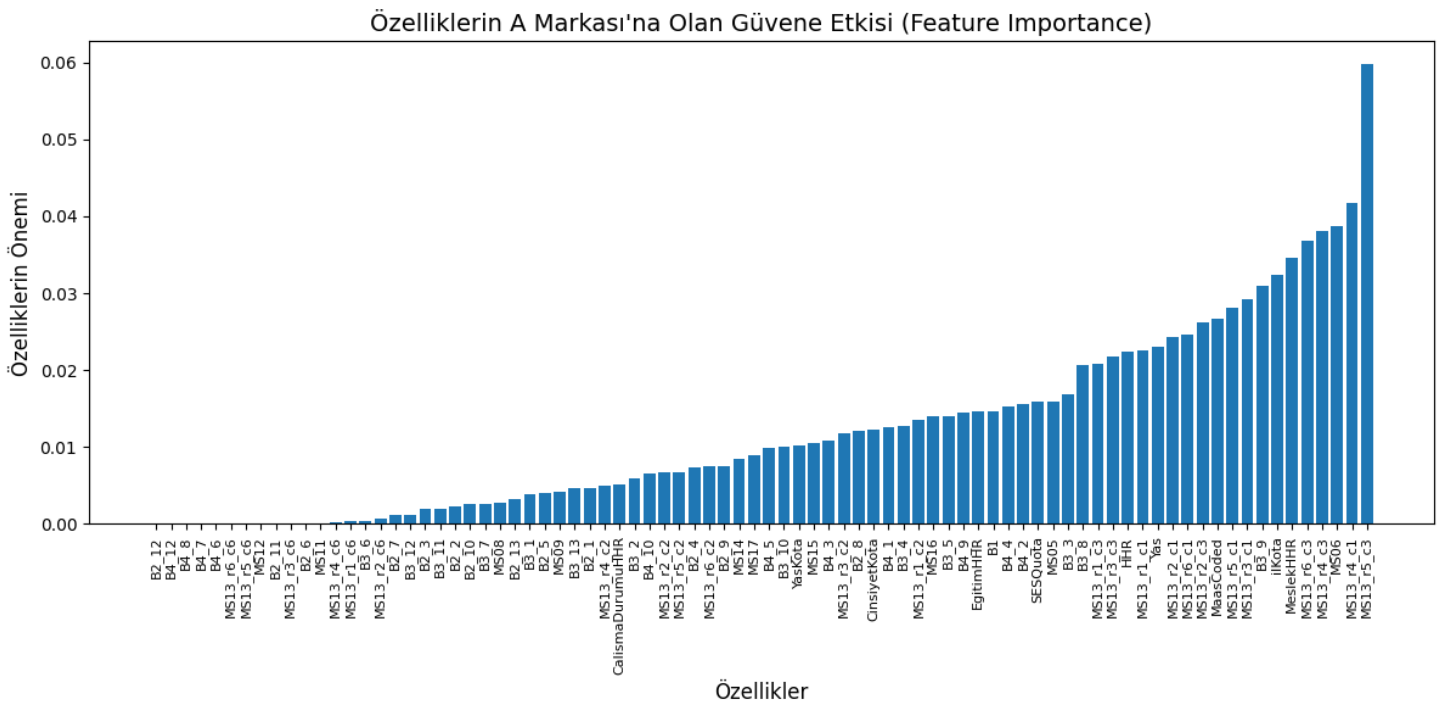
importances = model.feature_importances_
sorted_idx = np.argsort(importances)
features = data.columns[6:]

plt.figure(figsize=(12, 6))
plt.bar(np.array(features)[sorted_idx], importances[sorted_idx])

```

```
plt.ylabel('Özelliklerin Önemi', fontsize=12)
plt.xlabel('Özellikler', fontsize=12)
plt.title('Özelliklerin A Markası'na Olan Güvene Etkisi (Feature Importance)',
          fontsize=14)
plt.xticks(rotation=90, fontsize=8)

plt.tight_layout()
plt.show()
```



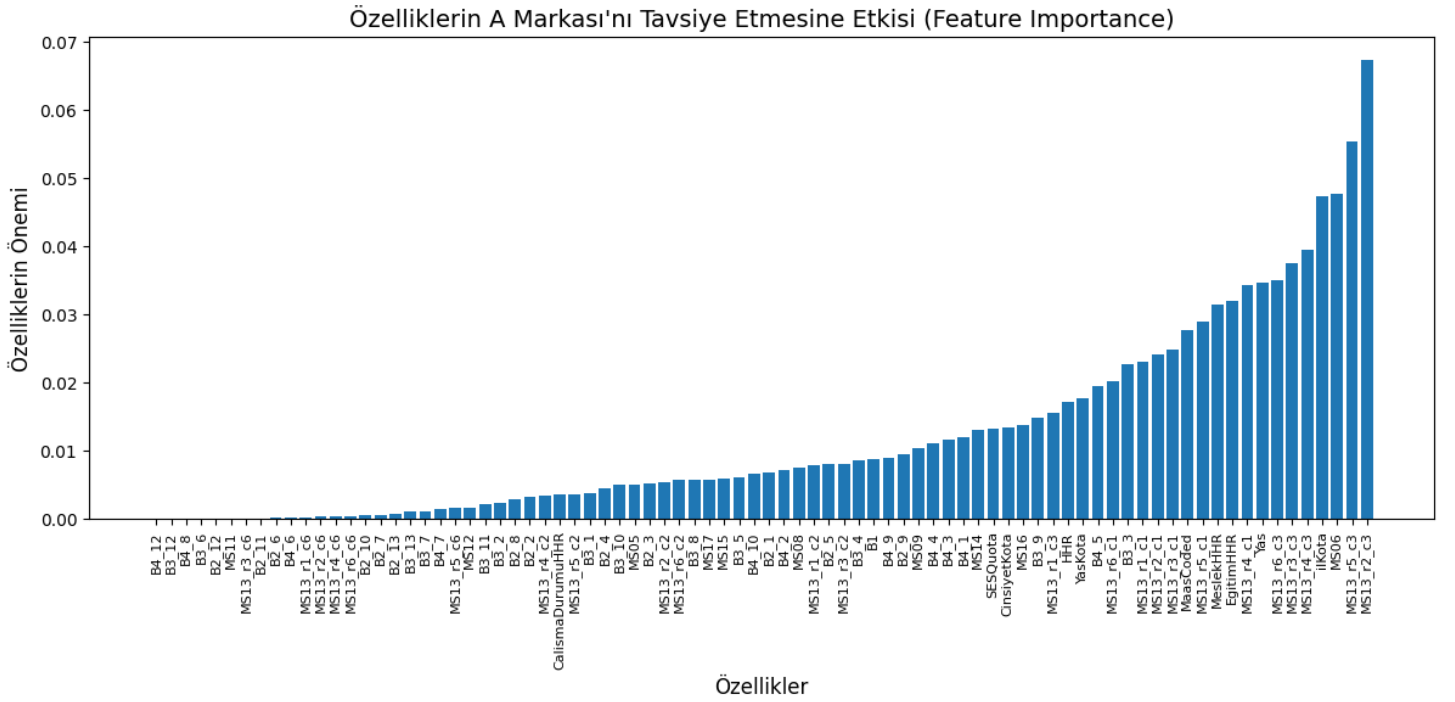
Grafik-6: Özelliklerin A Markası'na Olan Güvene Etkisi Bar Grafiği

```
importances2 = model2.feature_importances_
sorted_idx2 = np.argsort(importances2)
features = data.columns[6:]

plt.figure(figsize=(12, 6))
plt.bar(np.array(features)[sorted_idx2], importances2[sorted_idx2])
plt.ylabel('Özelliklerin Önemi', fontsize=12)
plt.xlabel('Özellikler', fontsize=12)
```

```
plt.title("Özelliklerin A Markası'nı Tavsiye Etmesine Etkisi (Feature Importance)",
        fontsize=14)
plt.xticks(rotation=90, fontsize=8)

plt.tight_layout()
plt.show()
```



Grafik-7: Özelliklerin A Markası'nı Tavsiye Etmesine Etkisi Bar Grafiği

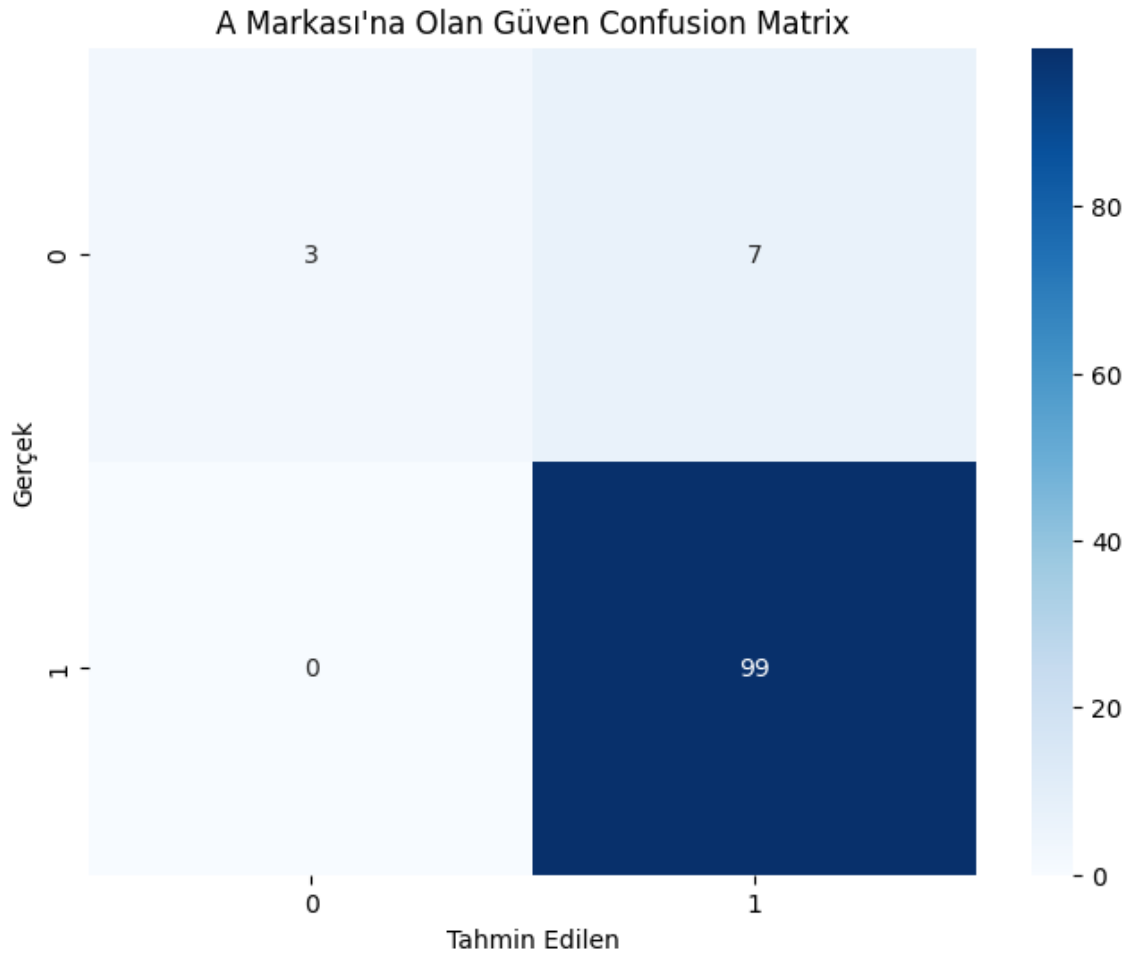
Bu grafikler, hangi özelliklerin modelde önemli olduğunu gösterir. Örneğin, müşteri yaşı, cinsiyet, gelir durumu gibi özelliklerin pazar analizine olan etkisini daha net görebilirsiniz. Eğer belirli bir özellik diğerlerinden daha fazla öne çıkıyorsa, bu özellik pazar stratejisi için kritik olabilir.

Confusion Matrix (Karışıklık Matrisi):

Confusion matrix, modelin yanlış ve doğru tahminlerini sınıflandırır. Sınıflar arasında modelin nerelerde hata yaptığını, hangi sınıfların birbiriyle karıştığını gösterir. Özellikle pazar analizinde belirli müşteri segmentleri arasındaki farkları incelemek için faydalı olabilir.

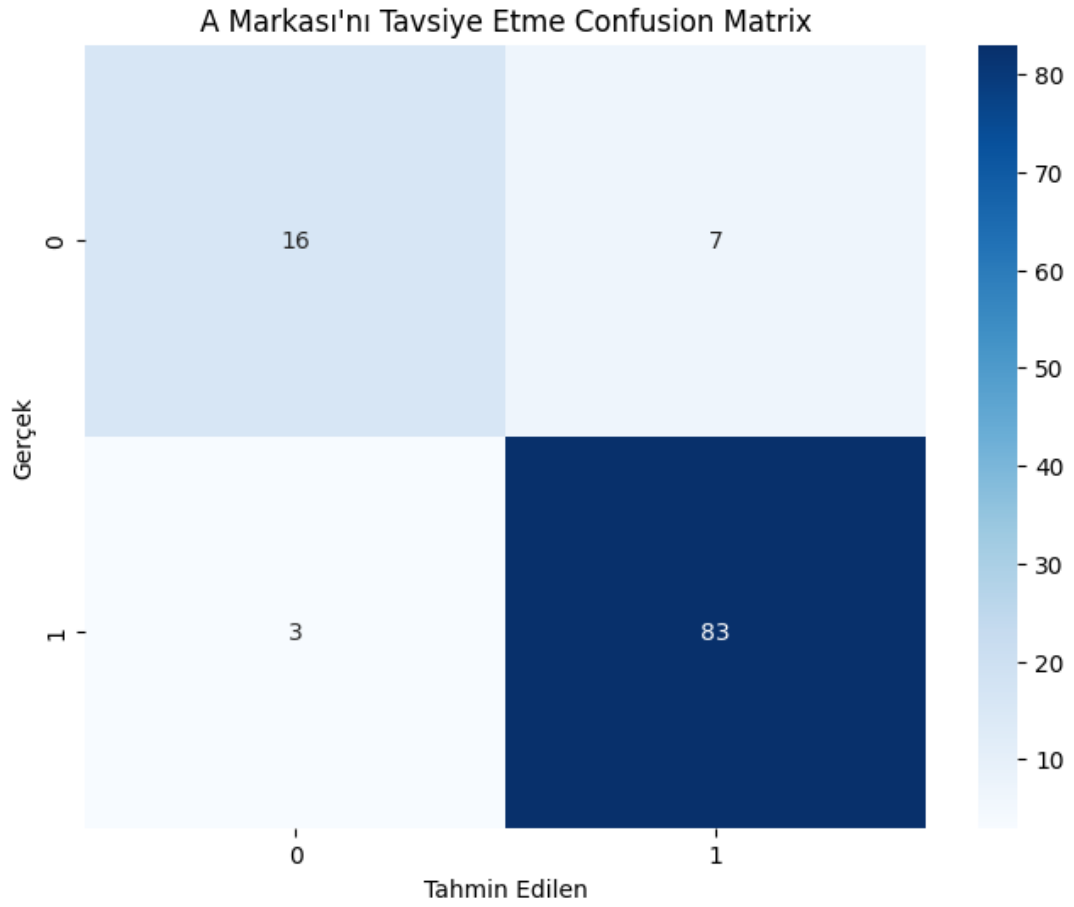
```
cm = confusion_matrix(y_test1, y_pred)
plt.figure(figsize=(8, 6))
```

```
sns.heatmap(cm, annot=True, fmt='d', cmap='Blues')  
plt.xlabel('Tahmin Edilen')  
plt.ylabel('Gerçek')  
plt.title("A Markası'na Olan Güven Confusion  
Matrix") plt.show()
```



Grafik-8: A Markası'na Olan Güven Confusion Matrix Grafiği

```
cm2 = confusion_matrix(y_test2, y_pred2)  
plt.figure(figsize=(8, 6))  
sns.heatmap(cm2, annot=True, fmt='d', cmap='Blues')  
plt.xlabel('Tahmin Edilen')  
plt.ylabel('Gerçek')  
plt.title("A Markası'nı Tavsiye Etme Confusion  
Matrix") plt.show()
```

Grafik-9: A Markası'nı Tavsiye Etme Confusion Matrix Grafiği

Confusion matrix, hangi sınıflarda modelin iyi tahmin yaptığını ve hangi sınıflarda hatalı olduğunu gösterir. Bu da pazar analizi sırasında hangi segmentlerin karıştığını veya yanlış sınıflandırıldığını anlamaya yardımcı olur.

ROC ve AUC (Receiver Operating Characteristic ve Area Under Curve):

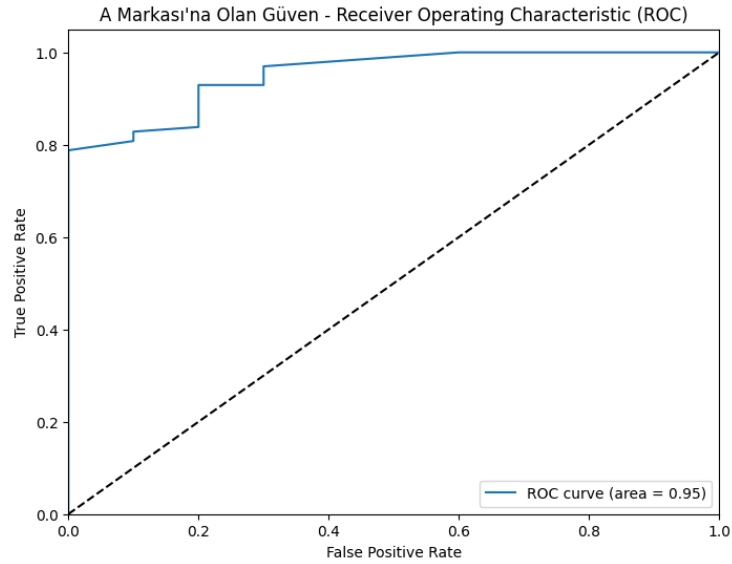
ROC eğrisi, modelin sınıflandırma performansını ölçen bir araçtır. AUC (Area Under Curve) ise bu eğrinin altındaki alanı ifade eder ve sınıflandırmanın ne kadar başarılı olduğunu gösterir. Pazar analizinde, müşteri sınıflandırmalarının ne kadar iyi yapıldığını ölçmek için kullanılabilir.

```
fpr, tpr, _ = roc_curve(y_test1, model.predict_proba(x_test1)[:, 1],
pos_label='Yüksek')
roc_auc = roc_auc_score(y_test1, model.predict_proba(x_test1)[:, 1])
plt.figure(figsize=(8, 6))
plt.plot(fpr, tpr, label=f'ROC curve (area = {roc_auc:.2f})')
```

```

plt.plot([0, 1], [0, 1], 'k--')
plt.xlim([0.0, 1.0])
plt.ylim([0.0, 1.05])
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title("A Markası'na Olan Güven - Receiver Operating Characteristic (ROC)")
plt.legend(loc="lower right")
plt.show()

```

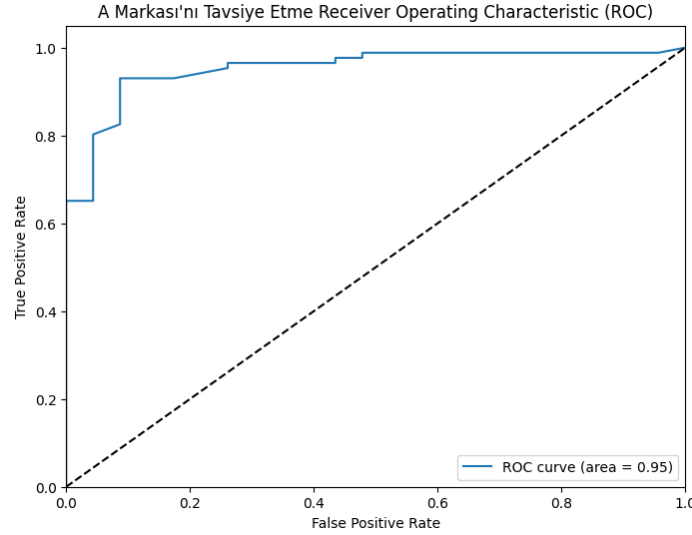


Grafik-10: A Markası'na Olan Güven - Receiver Operating Characteristic (ROC) Grafiği

```

fpr2, tpr2, _ = roc_curve(y_test2, model2.predict_proba(x_test2)[:, 1],
pos_label='Yüksek')
roc_auc2 = roc_auc_score(y_test2, model2.predict_proba(x_test2)[:, 1])
plt.figure(figsize=(8, 6))
plt.plot(fpr2, tpr2, label=f'ROC curve (area = {roc_auc2:.2f})')
plt.plot([0, 1], [0, 1], 'k--')
plt.xlim([0.0, 1.0])
plt.ylim([0.0, 1.05])
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title("A Markası'nı Tavsiye Etme Receiver Operating Characteristic (ROC)")
plt.legend(loc="lower right")
plt.show()

```



Grafik-11: A Markası'nı Tavsiye Etme - Receiver Operating Characteristic (ROC) Grafiği

ROC (Receiver Operating Characteristic) eğrisi, bir sınıflandırma modelinin performansını değerlendiren ve modelin True Positive Rate (Gerçek Pozitif Oran-TPR) ile False Positive Rate (Yanlış Pozitif Oran-FPR) arasındaki ilişkiyi gösteren bir görselleştirme aracıdır. ROC eğrisi, sınıflandırıcı modelin çeşitli eşik değerlerindeki performansını incelemek için kullanılır.

True Positive Rate (TPR): TPR, aynı zamanda Recall (Duyarlılık) veya Hassasiyet olarak bilinir. Bu oran, modelin doğru bir şekilde pozitif sınıfı (örneğin "Yüksek" sınıfı) tahmin ettiği örneklerin, gerçekten pozitif olan tüm örneklerle oranıdır.

$$TPR = \frac{\text{Doğru Pozitif (TP)}}{\text{Doğru Pozitif (TP)} + \text{Yanlış Negatif (FN)}}$$

Formül-2: True Positive Rate (TPR) Formülü

False Positive Rate (FPR): FPR, negatif sınıfa ait olan örneklerin yanlış bir şekilde pozitif sınıfa (örneğin "Yüksek" sınıfı) tahmin edilme oranıdır. Bu oran, modelin pozitif sınıfa dair yaptığı yanlış tahminlerin, aslında negatif olan tüm örneklerle oranını gösterir.

$$FPR = \frac{\text{Yanlış Pozitif (FP)}}{\text{Yanlış Pozitif (FP)} + \text{Doğru Negatif (TN)}}$$

Formül-3: False Positive Rate (FPR) Formülü

AUC, ROC eğrisinin altındaki alanı temsil eder ve modelin genel performansını değerlendirmek için kullanılır. AUC, 0 ile 1 arasında bir değere sahiptir:

- **AUC = 1:** Mükemmel model
- **AUC = 0.5:** Model rastgele tahmin yapıyor demektir.
- **AUC ne kadar yüksekse,** model o kadar iyi performans gösterir.

ROC Eğrisi Nasıl Yorumlanır?

- **Sol üst köşeye yakın eğri:** Modelin yüksek doğrulukla sınıflandırma yaptığı anlamına gelir. Yani hem düşük FPR hem de yüksek TPR.
- **Köşegen çizgiye yakın eğri:** Modelin rastgele tahmin yaptığını gösterir (kötü model performansı).

```
low_c = '#dd4124'
high_c = '#009473'
background_color = '#fbfbfb'

newdata1 = data.iloc[:, 6:]
continuous = [var for var in newdata1.columns if newdata1[var].dtype != 'O']

n_vars = len(continuous)
n_cols = 3
n_rows = math.ceil(n_vars / n_cols)
fig, axes = plt.subplots(n_rows, n_cols, figsize=(
    12, n_rows * 4), dpi=150, facecolor=background_color)
fig.subplots_adjust(wspace=0.2, hspace=0.5)

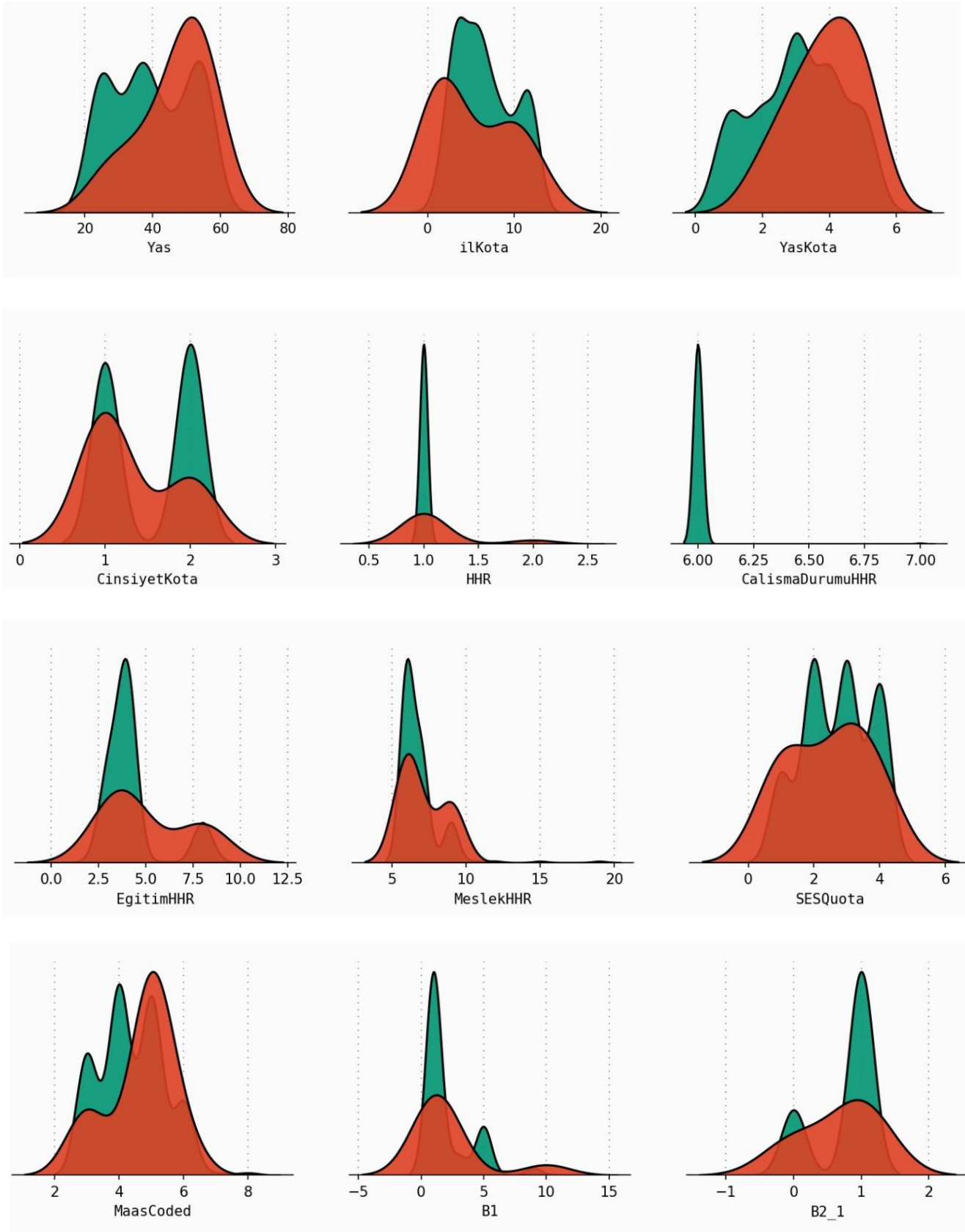
y1 = data.iloc[:, 4].values
Yes = data[y1 >= 4]
```

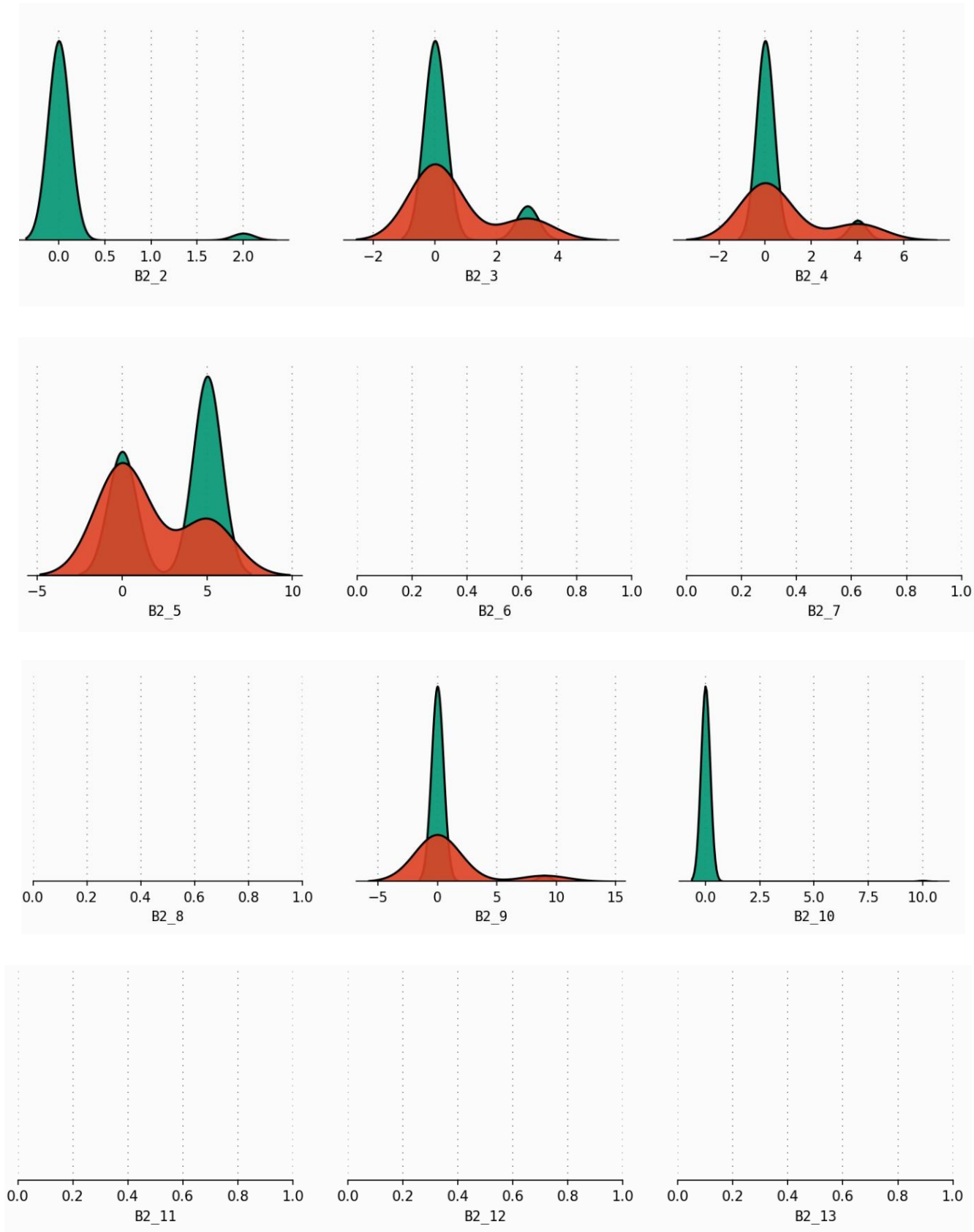
```
No = data[y1 <= 3]

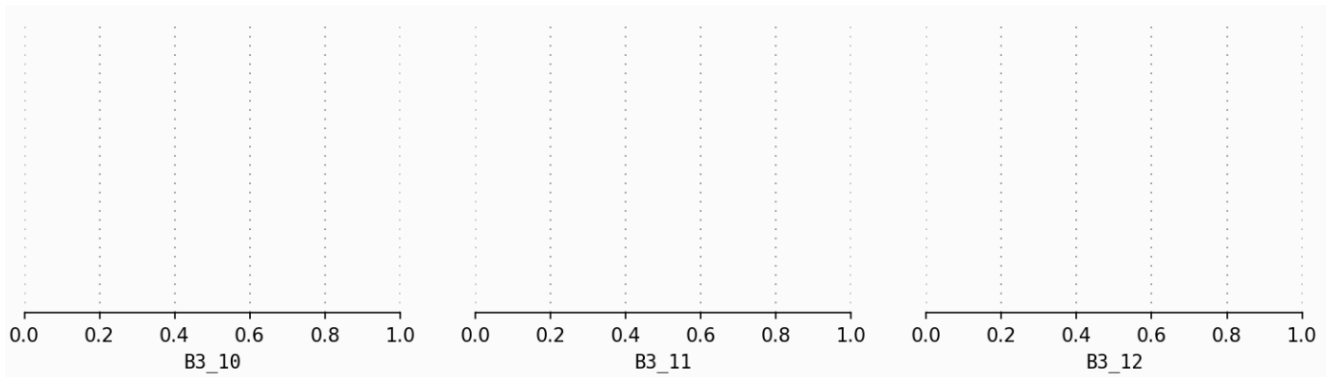
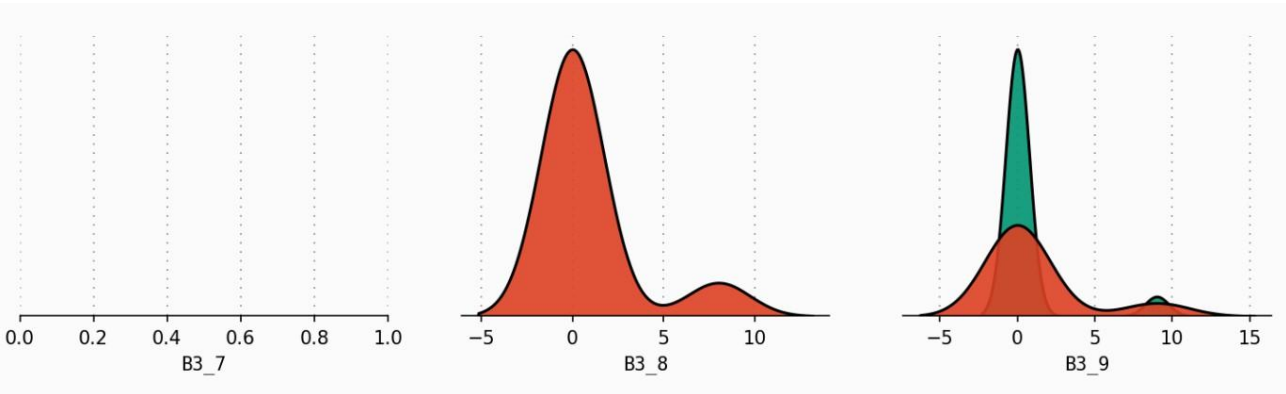
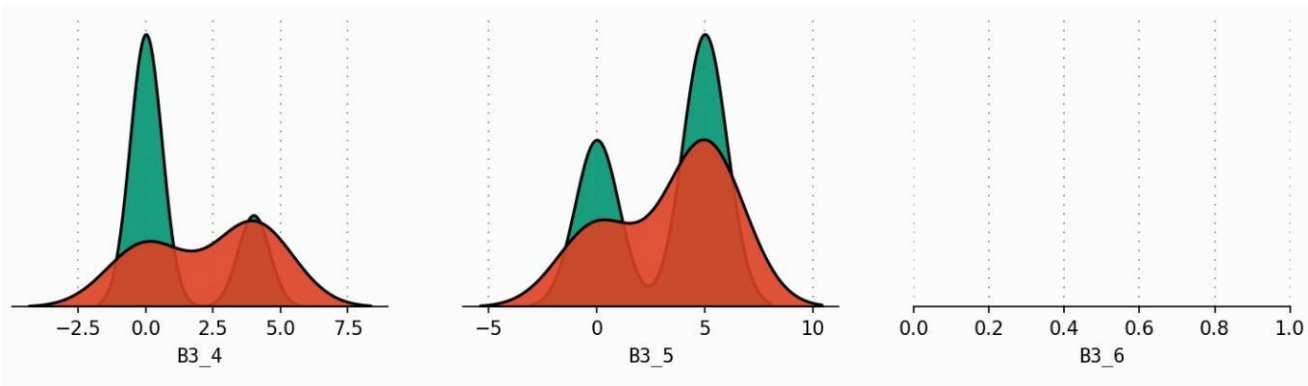
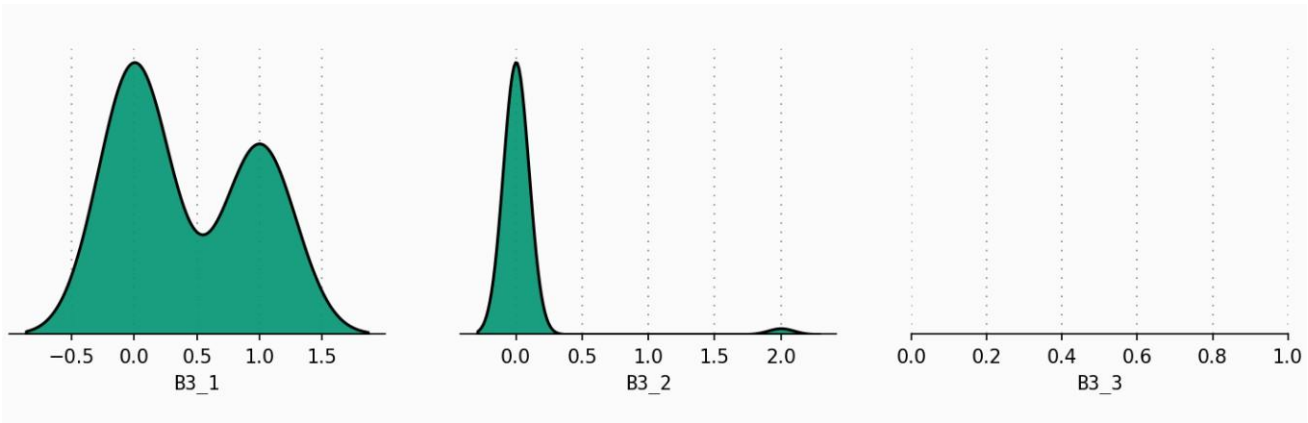
plot = 0
for row in range(n_rows):
    for col in range(n_cols):
        if plot < n_vars:
            ax = axes[row, col]
            ax.set_facecolor(background_color)
            ax.tick_params(axis='y', left=False)
            ax.get_yaxis().set_visible(False)
            ax.set_axisbelow(True)
            for s in ["top", "right", "left"]:
                ax.spines[s].set_visible(False)

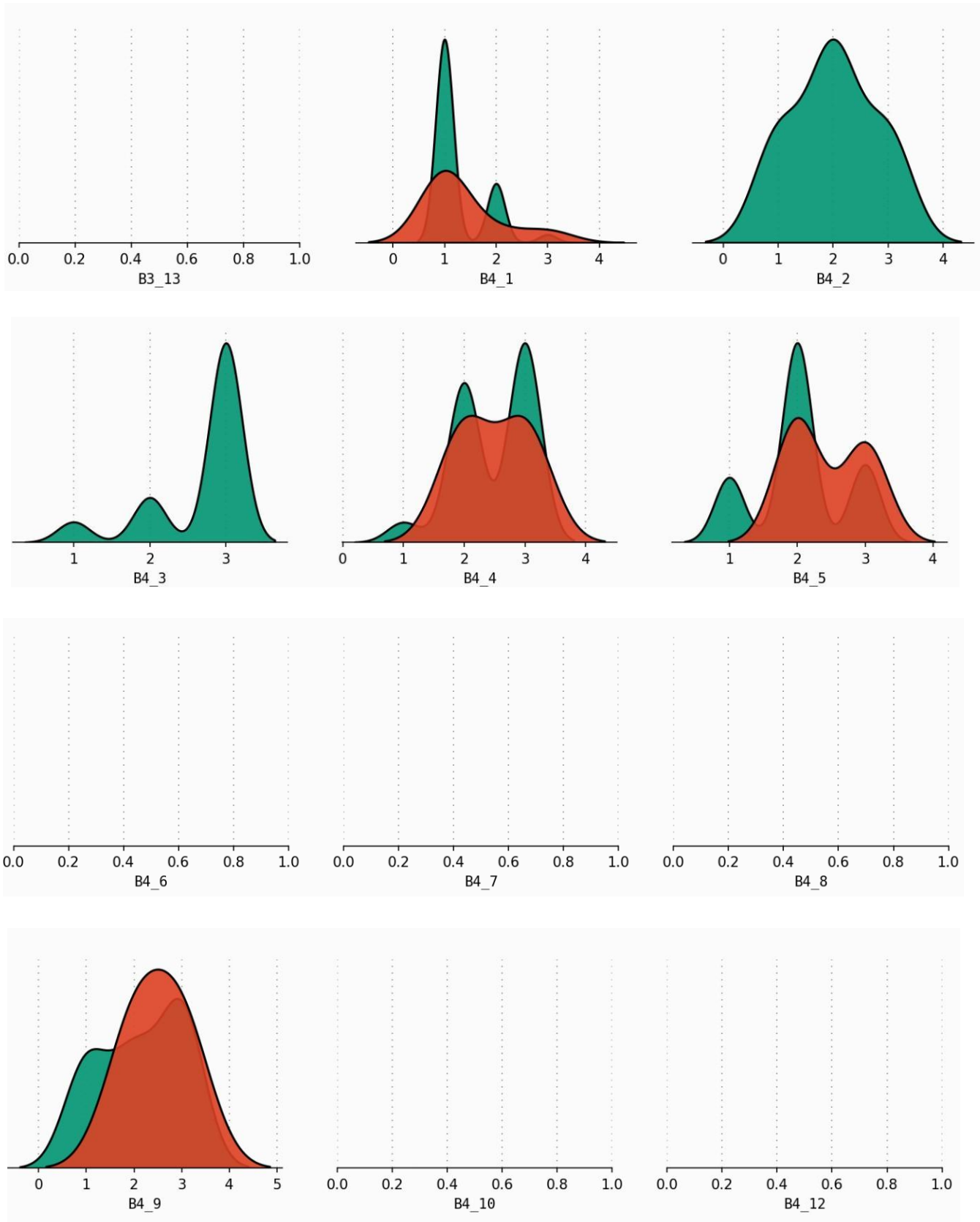
            sns.kdeplot(Yes[continuous[plot]], ax=ax, color=high_c,
                        ec='black', fill=True, linewidth=1.5, alpha=0.9, zorder=3,
                        legend=False, warn_singular=False)
            sns.kdeplot(No[continuous[plot]], ax=ax, color=low_c,
                        fill=True, ec='black', linewidth=1.5, alpha=0.9, zorder=3,
                        legend=False, warn_singular=False)
            ax.grid(which='major', axis='x', zorder=0,
                    color='gray', linestyle=':', dashes=(1, 5))
            ax.set_xlabel(continuous[plot], fontfamily='monospace')

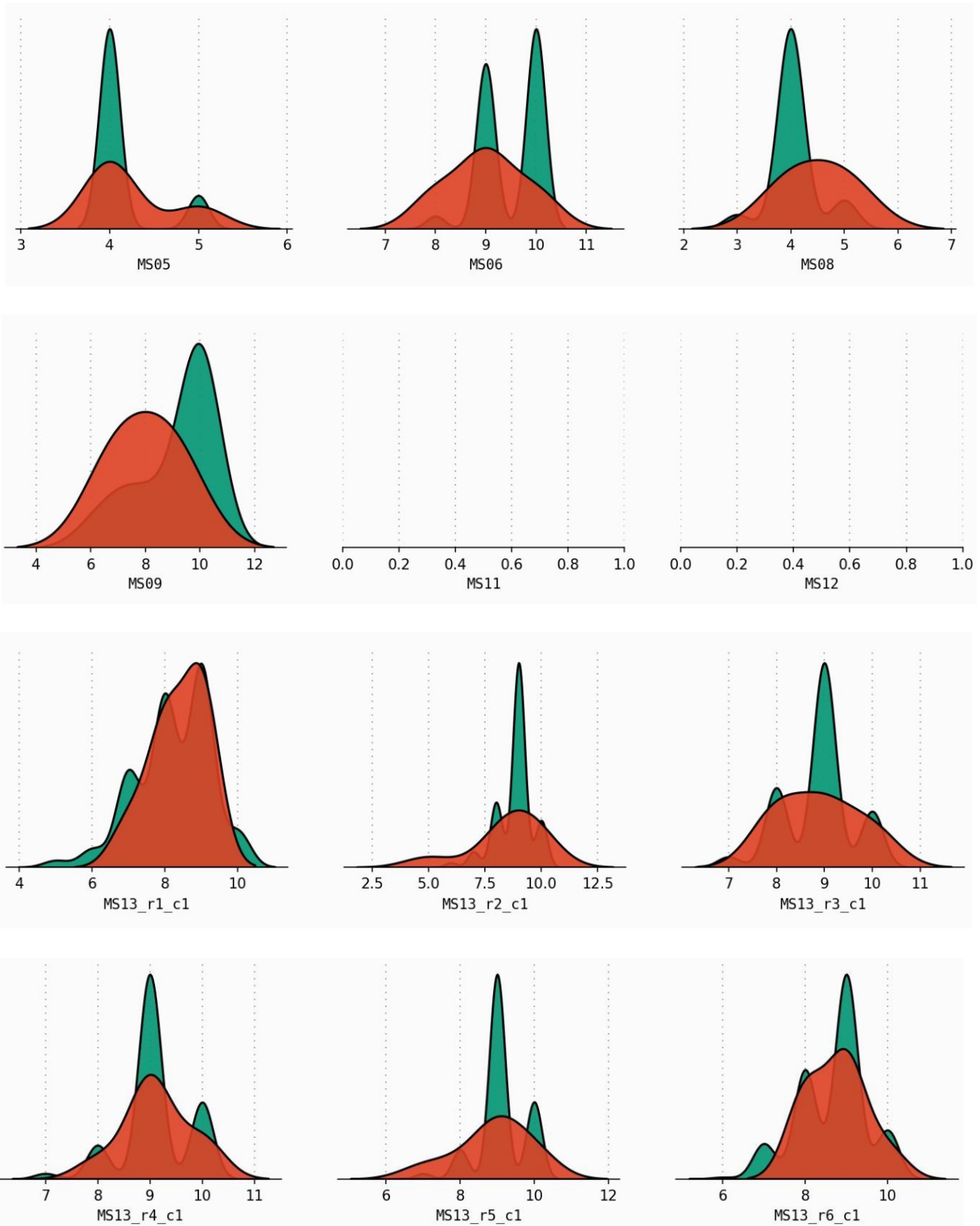
            plot += 1
        else:
            fig.delaxes(axes[row, col])
plt.show()
```

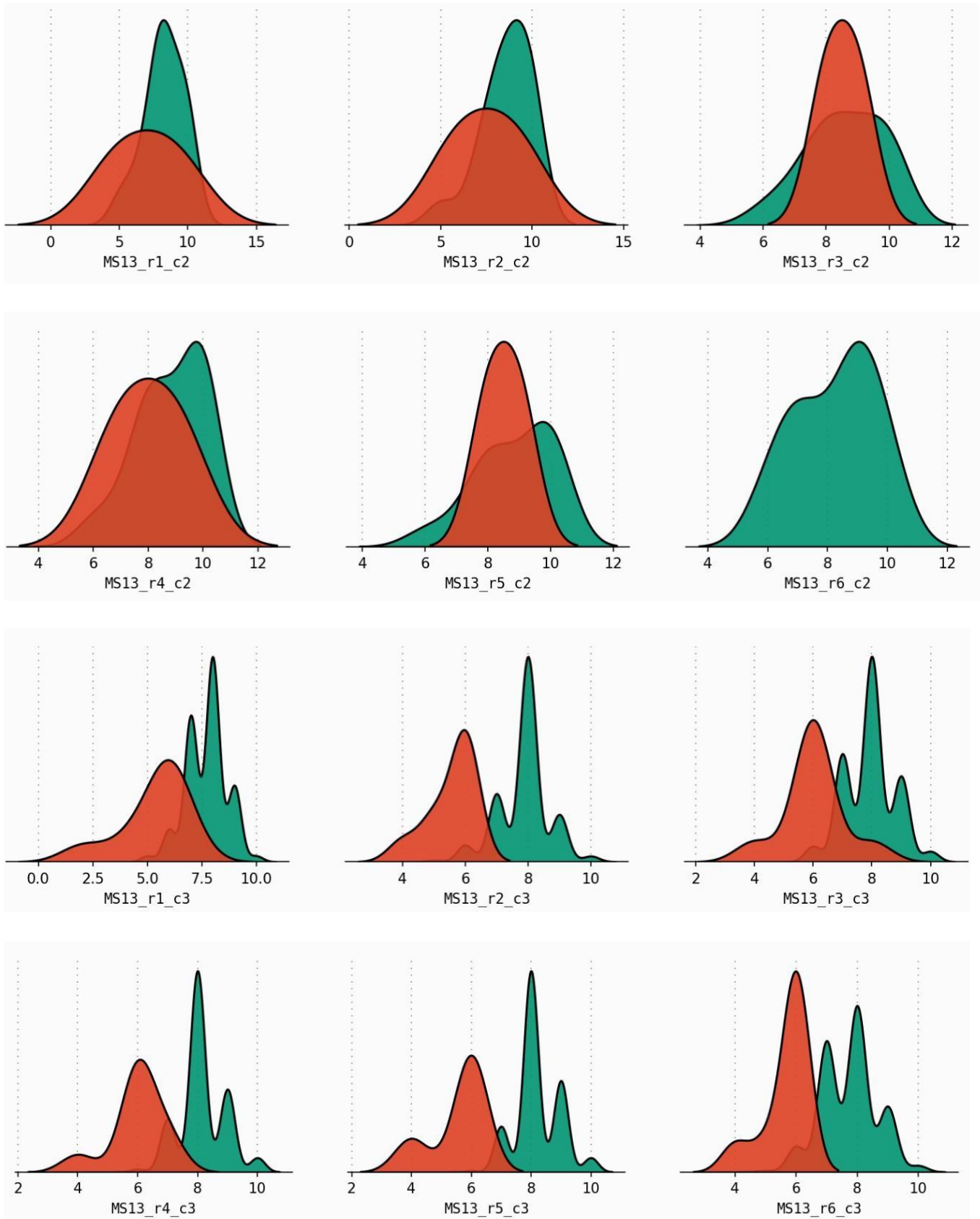


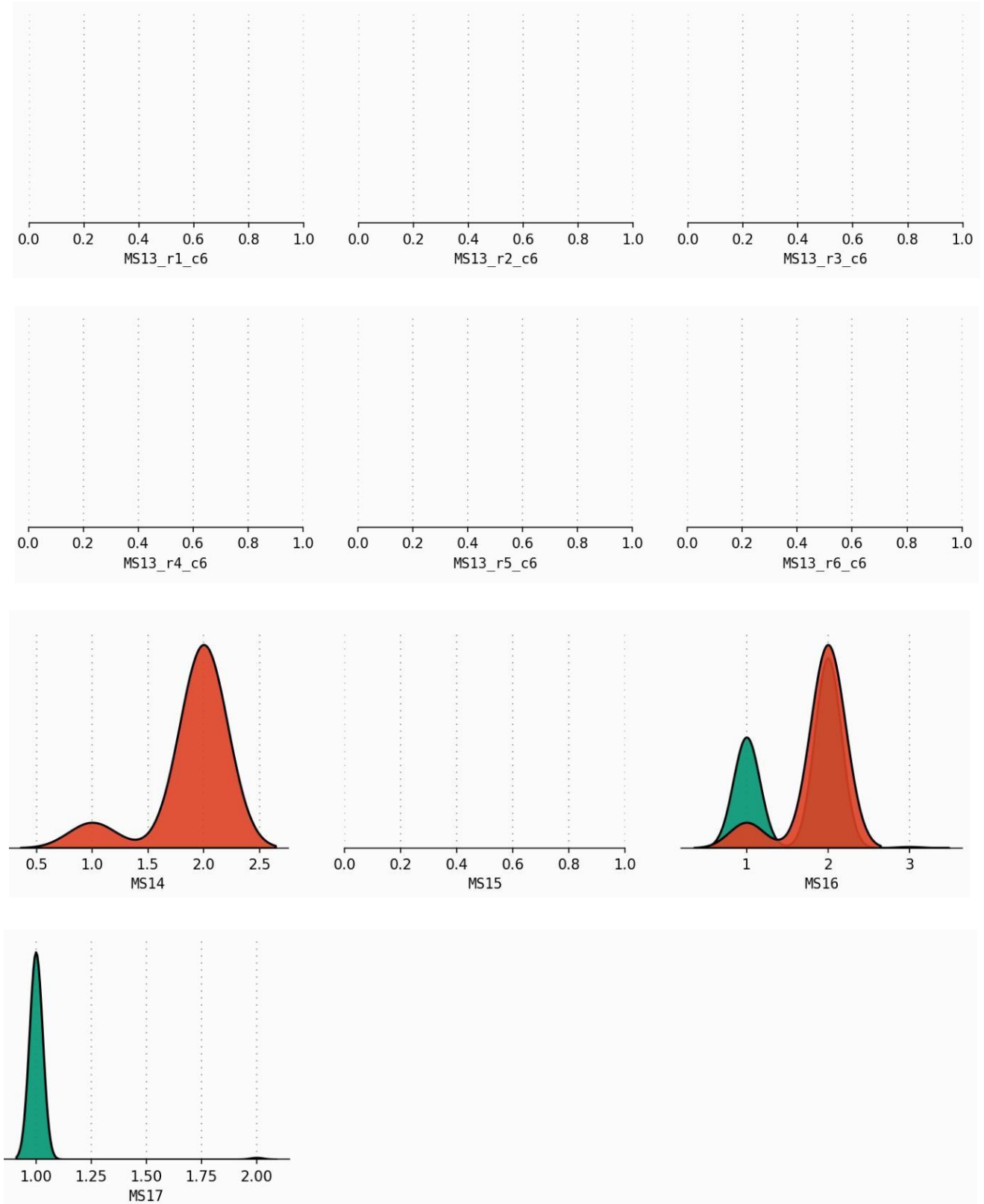












Grafik-12: A Markası'na Güven Konusunda KDE (Kernel Density Estimation) Grafikleri

Kırmızı alan, düşük güven olarak belirlenen grubu gösterir. Bu, güven seviyesi 3 veya daha düşük olan bireylerin dağılımını temsil eder. Bu bireylerin çeşitli değişkenlere (yaş, ilk kota vb.) göre yoğunluğunu görebilirsin.

Yeşil alan, yüksek güven olarak belirlenen grubu temsil eder. Güven seviyesi 4 ve üzeri olan bireylerin dağılımını gösterir. Yani, bu gruptaki bireyler bu değişkenlerde nasıl dağılmış, onu anlamak için kullanıyoruz.

```
low_c = '#dd4124'
high_c = '#009473'
background_color = '#fbfbfb'
newdata2 = data.iloc[:, 6:]
continuous2 = [var for var in newdata2.columns if newdata2[var].dtype != 'O']

n_vars2 = len(continuous2)
n_cols = 3
n_rows2 = math.ceil(n_vars2 / n_cols)

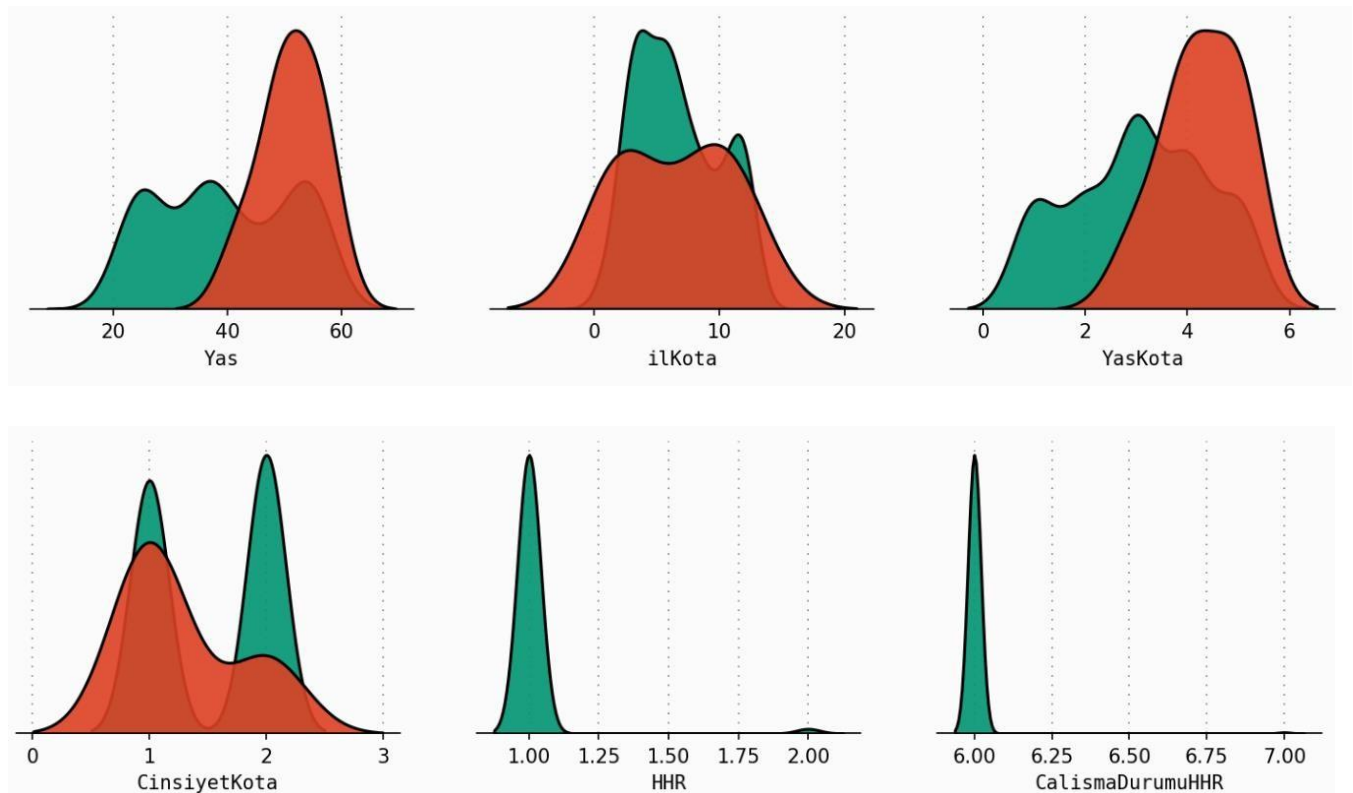
fig2, axes2 = plt.subplots(n_rows2, n_cols, figsize=(
    12, n_rows2 * 4), dpi=150, facecolor=background_color)
fig2.subplots_adjust(wspace=0.2, hspace=0.5)
y2 = data.iloc[:, 5].values
Yes = data[y2 >= 7]
No = data[y2 <= 6]
plot = 0
for row in range(n_rows2):
    for col in range(n_cols):
        if plot < n_vars2:
            ax2 = axes2[row, col]
            ax2.set_facecolor(background_color)
            ax2.tick_params(axis='y', left=False)
            ax2.get_yaxis().set_visible(False)
            ax2.set_axisbelow(True)
            for s in ["top", "right", "left"]:
                ax2.spines[s].set_visible(False)
```

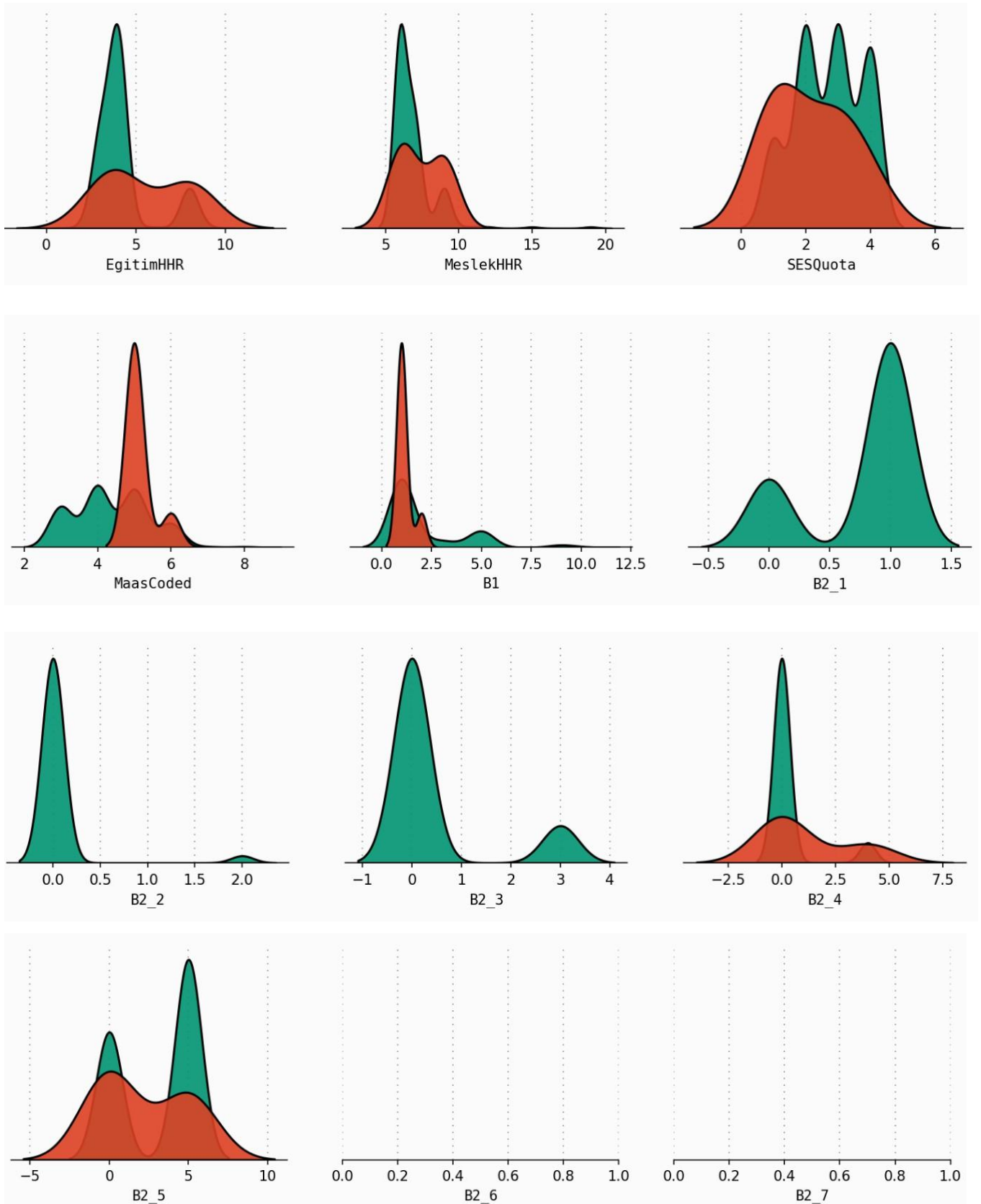
```

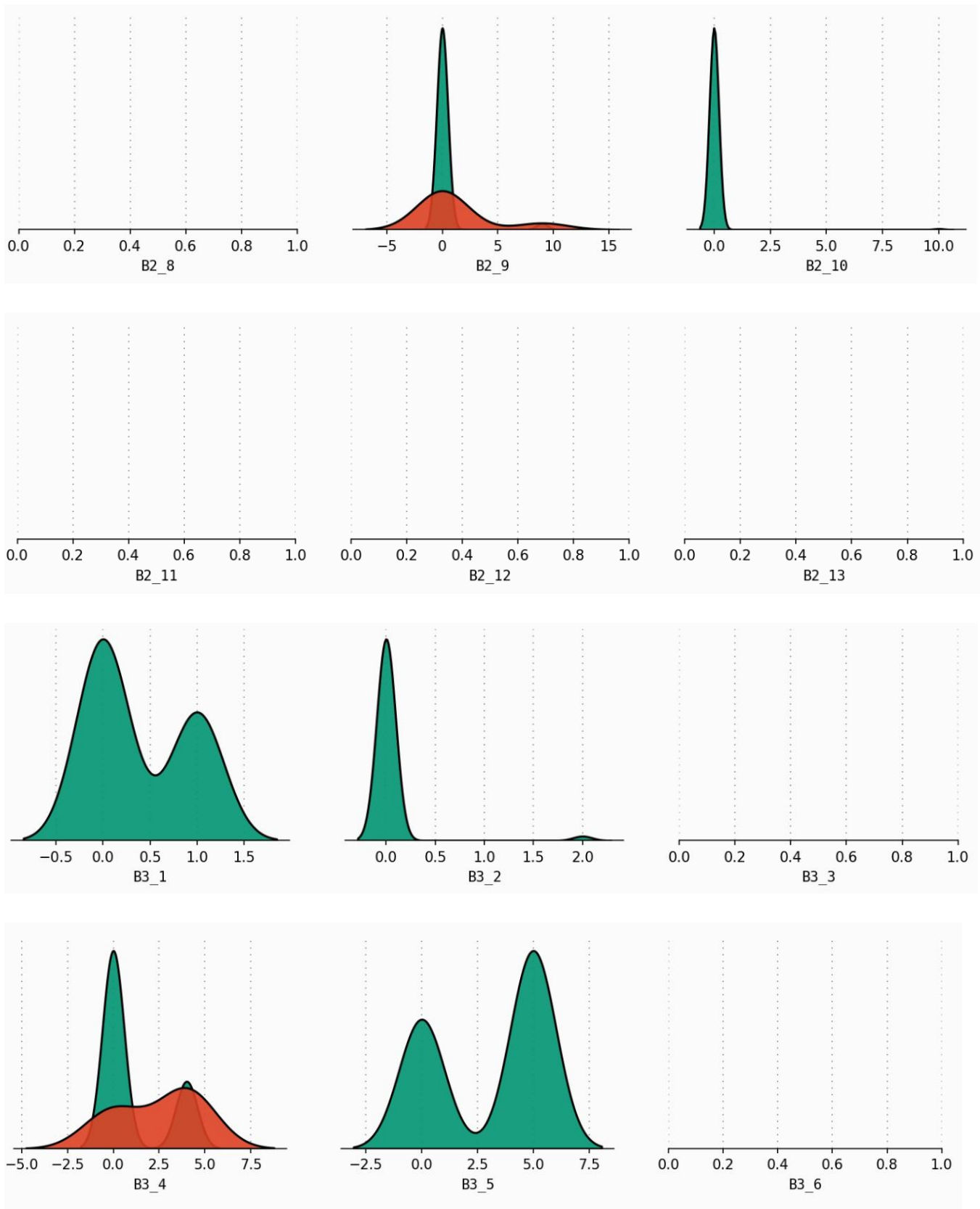
sns.kdeplot(Yes[continuous2[plot]], ax=ax2, color=high_c,
            ec='black', fill=True, linewidth=1.5, alpha=0.9, zorder=3,
            legend=False, warn_singular=False)
sns.kdeplot(No[continuous2[plot]], ax=ax2, color=low_c,
            fill=True, ec='black', linewidth=1.5, alpha=0.9, zorder=3,
            legend=False, warn_singular=False)
ax2.grid(which='major', axis='x', zorder=0,
        color='gray', linestyle=':', dashes=(1, 5))
ax2.set_xlabel(continuous2[plot], fontfamily='monospace')
plot += 1
else:
    fig2.delaxes(axes2[row, col])

plt.show()

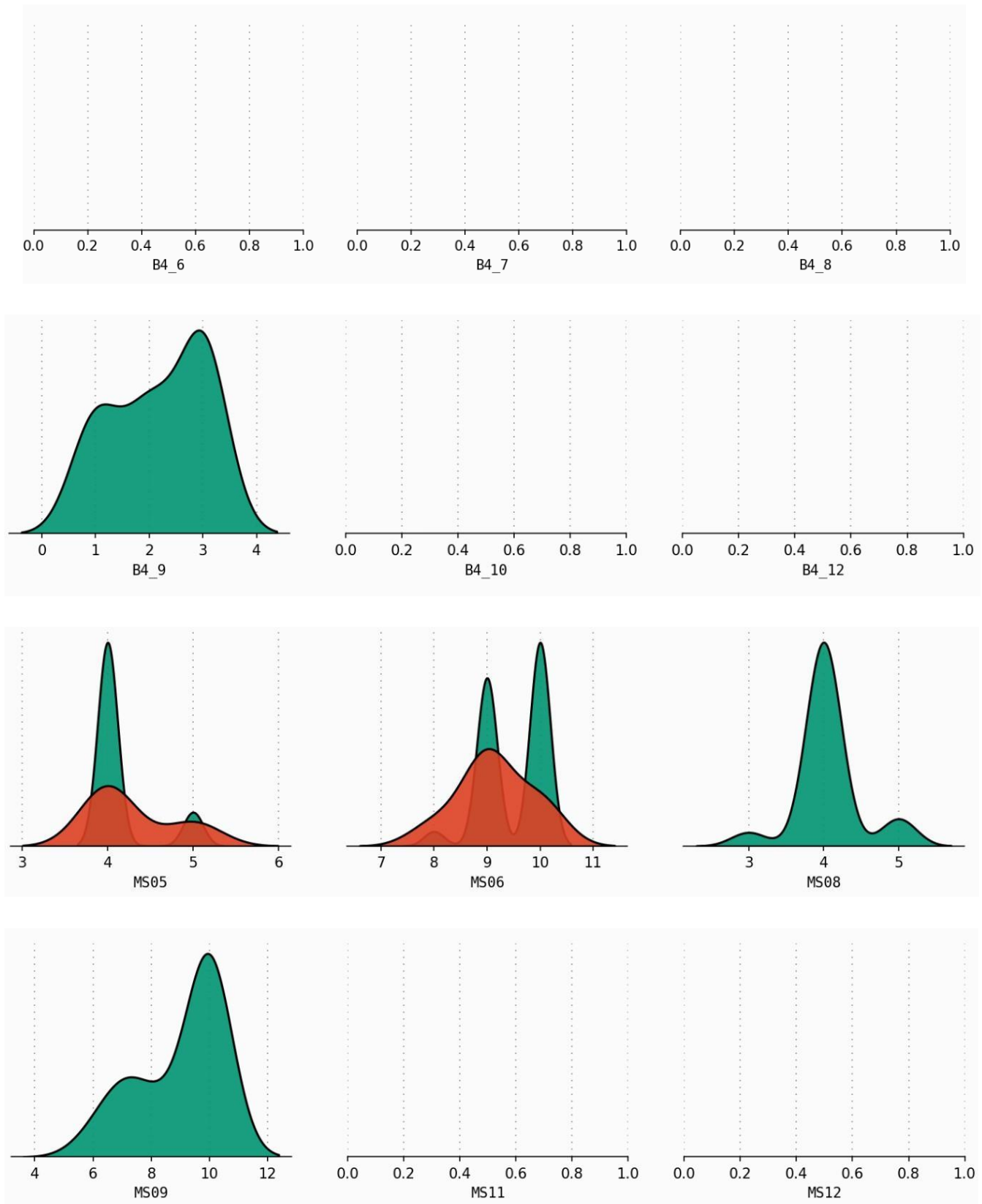
```

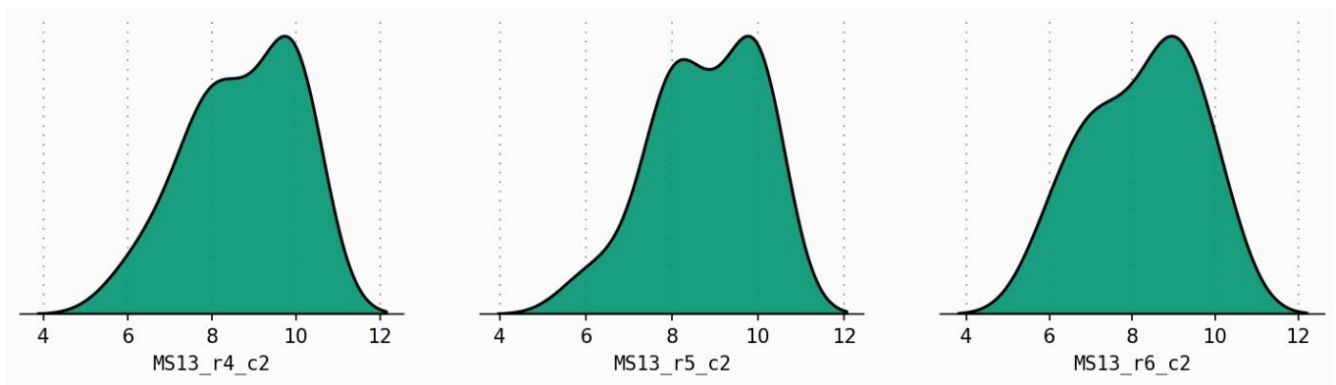
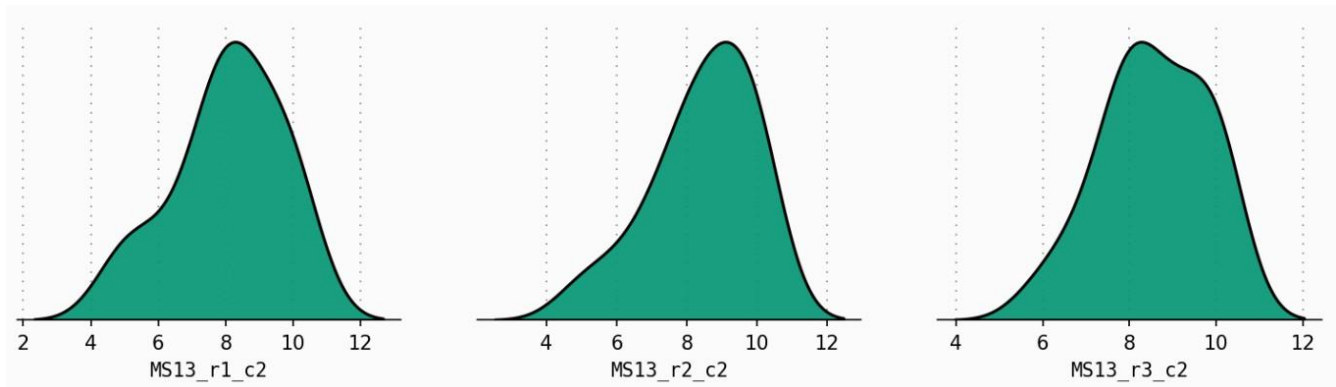
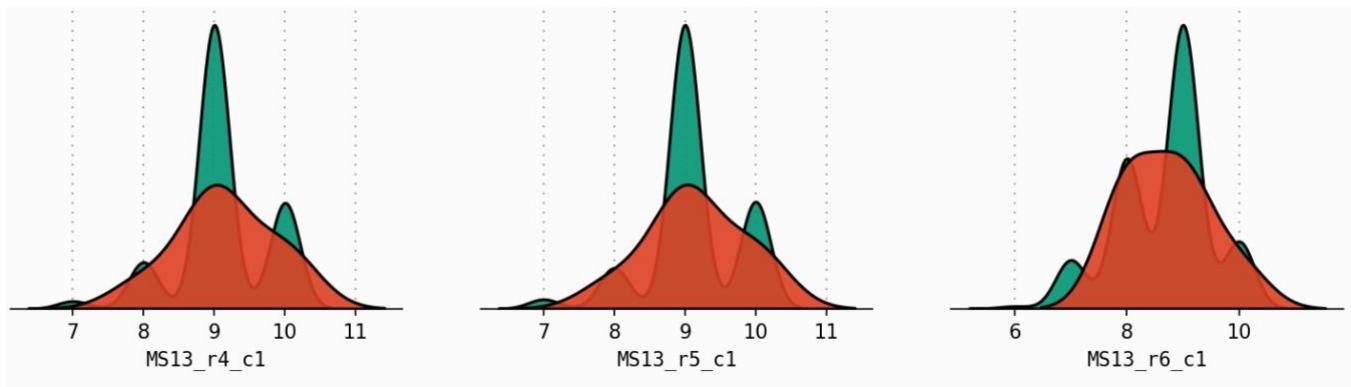
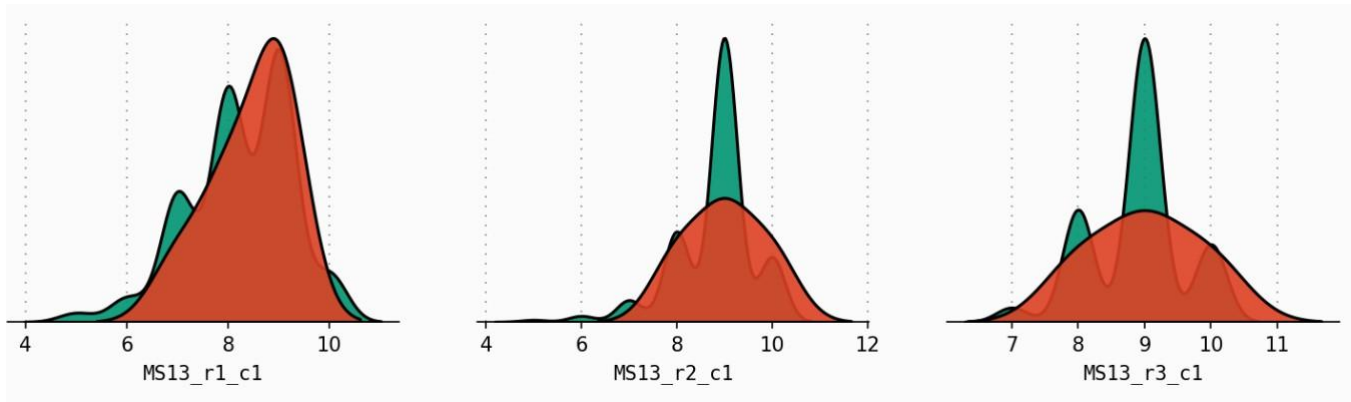


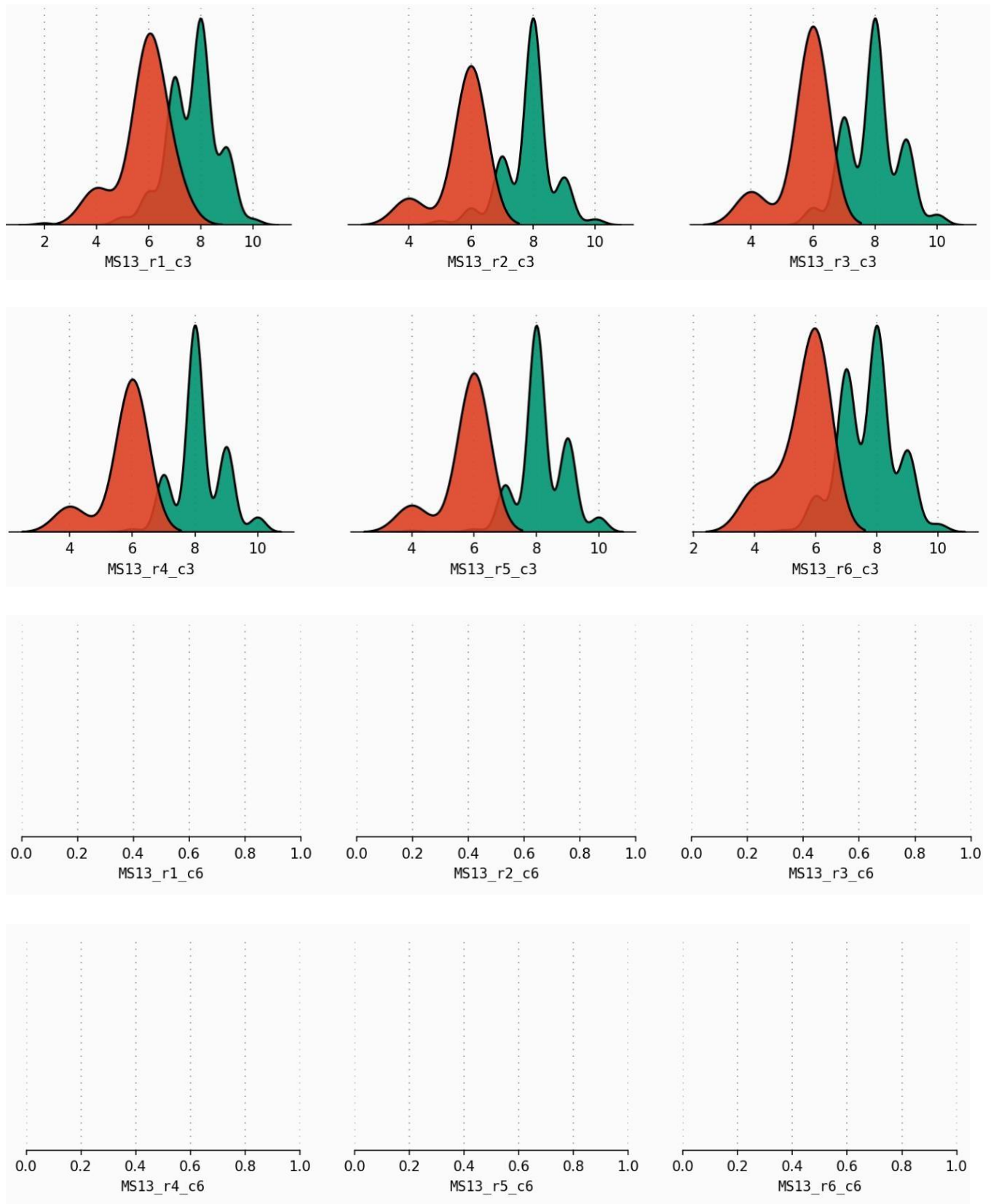


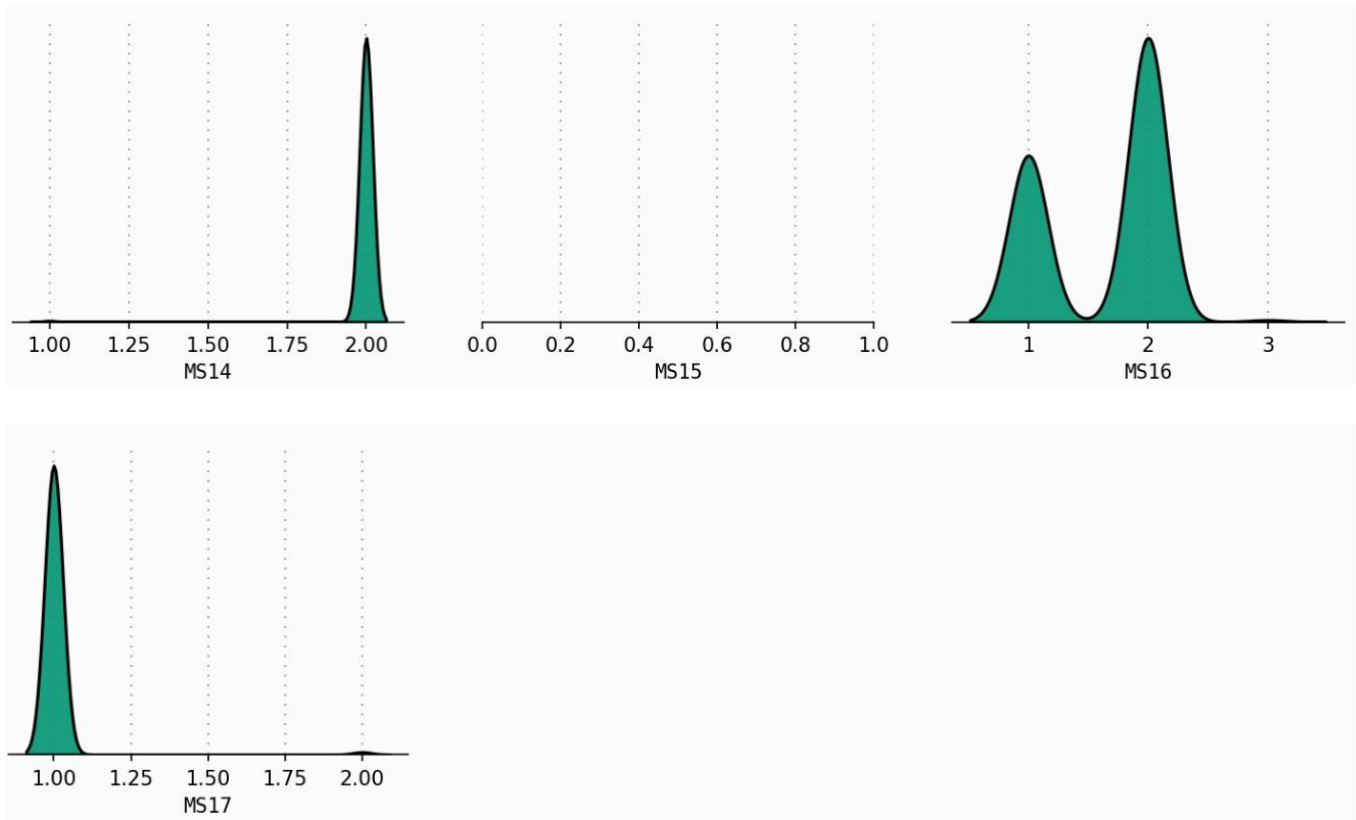












Grafik-13: A Markası'nı Tavsiye Etme Konusunda KDE (Kernel Density Estimation) Grafikleri

Kırmızı alan, düşük tavsiye edilen olarak belirlenen grubu gösterir. Bu, güven seviyesi 6 veya daha düşük olan bireylerin dağılımını temsil eder. Bu bireylerin çeşitli değişkenlere (yaş, ilk kota vb.) göre yoğunluğunu görebilirsin.

Yeşil alan, yüksek tavsiye edilen olarak belirlenen grubu temsil eder. Güven seviyesi 7 ve üzeri olan bireylerin dağılımını gösterir. Yani, bu gruptaki bireyler bu değişkenlerde nasıl dağılmış, onu anlamak için kullanıyoruz.

Bu grafikler, iki grup arasındaki dağılım farklarını görsel olarak anlamak için kullanılıyor. KDE grafiği, bir değişkenin dağılımını pürüzsüz bir şekilde gösterir. Yani, bu grafiklere bakarak verilerin nasıl bir dağılım gösterdiğini ve yoğunlaşma noktalarını (yani pik yapma noktalarını) görebilirsiniz. Her değişkenin ortalaması, standart sapması ve veri yayılımı farklıdır. Bu da her bir değişken için farklı şekillerde kırmızı grafiklerin ortaya çıkmasına neden olur. Örneğin, bir değişkende düşük güven grubundaki kişilerin verileri dar bir aralıkta toplanmış olabilirken, başka bir değişkende daha geniş bir yayılım gösterebilir.

Kırmızı (Düşük güven) grubu daha geniş bir yaş aralığına yayılmış ve 20-40 yaş aralığında yoğunlaşıyor. Bu durum, yaş faktörünün güven düzeyini etkileyen bir değişken olabileceğini gösterir. Özellikle, yaş arttıkça güvenin de arttığına dair bir eğilim var gibi görünüyor.

```
brands = ['B Markası', 'C Markası', 'A Markası', 'F Markası', 'D Markası',
          'E Markası', 'G Markası', 'H Markası', 'I Markası', 'J Markası', 'K Markası', 'L
          Markası']

brand_counts = {brand: {'first_mention': 0,
                        'spontaneous': 0, 'aided': 0} for brand in brands}

total_participants = len(data)

for index, row in data.iterrows():
    if not pd.isna(row['B1']) and int(row['B1']) <= len(brands):
        brand_counts[brands[int(row['B1']) - 1]]['first_mention'] += 1

    for i in range(1, len(brands) + 1):
        b2_col = f'B2_{i}'
        b3_col = f'B3_{i}'

        if not pd.isna(row[b2_col]) and row[b2_col] == i:
            brand_counts[brands[i - 1]]['spontaneous'] += 1

        if not pd.isna(row[b3_col]) and row[b3_col] == i:
            brand_counts[brands[i - 1]]['aided'] += 1

normalized_data = {brand: {'first_mention': 0, 'spontaneous': 0,
                           'aided': 0, 'never_heard': 0} for brand in brands}

for brand in brands:
    first_mention_percentage = (
        brand_counts[brand]['first_mention'] / total_participants) * 100
    spontaneous_percentage = (
        brand_counts[brand]['spontaneous'] / total_participants) * 100
```

```

aided_percentage = (brand_counts[brand]
                    ['aided'] / total_participants) * 100
total_percentage = first_mention_percentage + \
    spontaneous_percentage + aided_percentage

never_heard_percentage = max(0, 100 - total_percentage)

normalized_data[brand]['first_mention'] = first_mention_percentage
normalized_data[brand]['spontaneous'] = spontaneous_percentage
normalized_data[brand]['aided'] = aided_percentage
normalized_data[brand]['never_heard'] = never_heard_percentage

first_mentions = [normalized_data[brand]['first_mention'] for brand in brands]
spontaneous_recalls = [normalized_data[brand]['spontaneous']
                      for brand in brands]
aided_recalls = [normalized_data[brand]['aided'] for brand in brands]
never_heard = [normalized_data[brand]['never_heard'] for brand in brands]

plt.figure(figsize=(10, 7))
bar_width = 0.6

p1 = plt.bar(brands, first_mentions, bar_width,
             label='İlk Akla Gelen', color='#2A7F62')
p2 = plt.bar(brands, spontaneous_recalls, bar_width, bottom=first_mentions,
             label='Spontan Hatırlanan', color='#54BAB9')
p3 = plt.bar(brands, aided_recalls, bar_width, bottom=[i+j for i, j in zip(
    first_mentions, spontaneous_recalls)], label='Yardımlı Hatırlanan',
             color='#F6D7A7')
p4 = plt.bar(brands, never_heard, bar_width, bottom=[i+j+k for i, j, k in zip(
    first_mentions, spontaneous_recalls, aided_recalls)], label='Hiç Duyulmayan',
             color='grey')

plt.ylabel('Yüzde (%)')
plt.title('Marka Bilinirliği (%)', fontsize=14)
plt.xticks(rotation=45, ha='right')

```

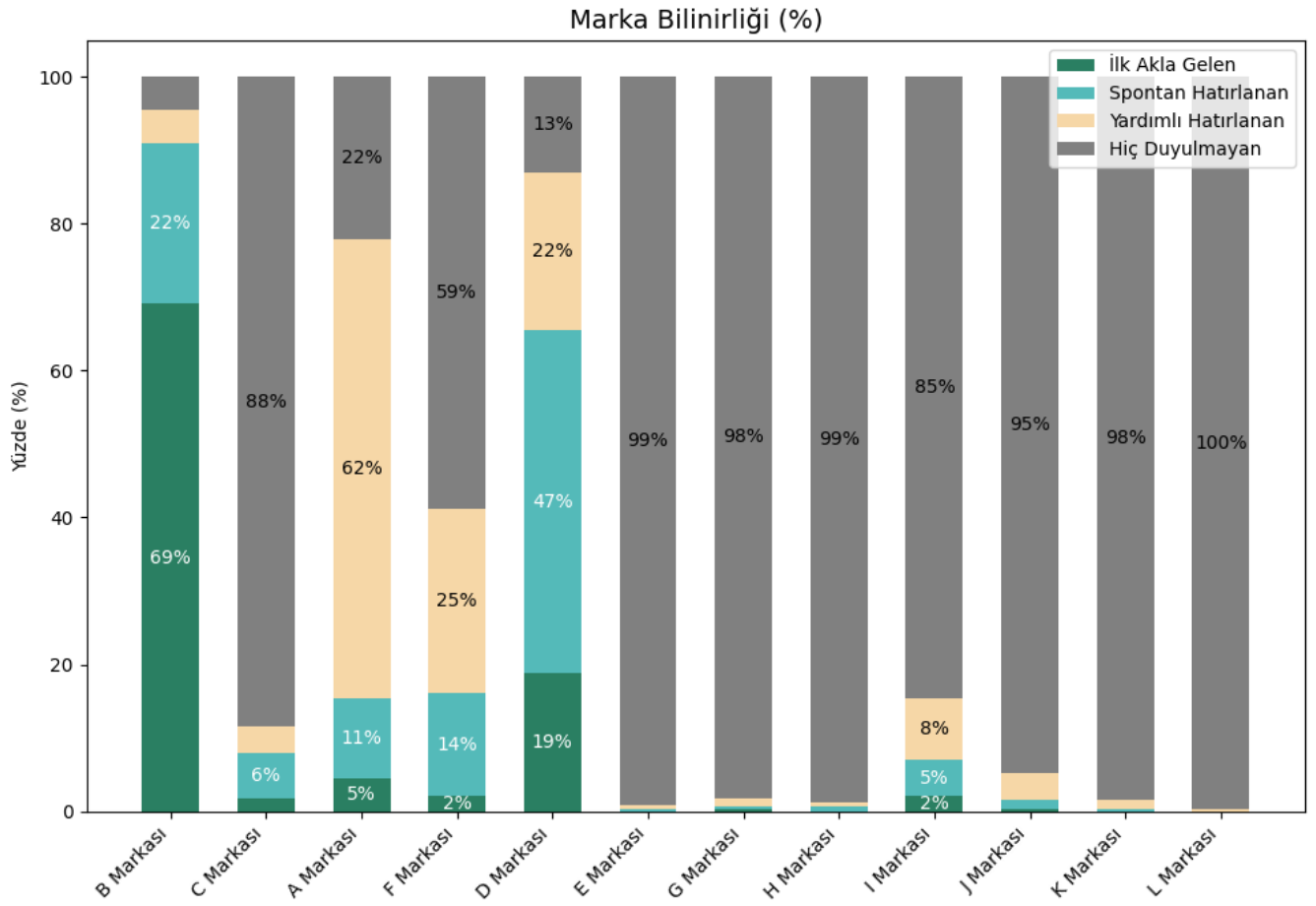
```
plt.legend()

for i, brand in enumerate(brands):
    if first_mentions[i] > 2:
        plt.text(i, first_mentions[i] / 2, f'{first_mentions[i]:.0f}%',
                  ha='center', va='center', color='white', fontsize=10)

    if spontaneous_recalls[i] > 2:
        plt.text(i, first_mentions[i] + spontaneous_recalls[i] / 2,
                  f'{spontaneous_recalls[i]:.0f}%', ha='center', va='center',
color='white', fontsize=10)

    if aided_recalls[i] > 5:
        plt.text(i, first_mentions[i] + spontaneous_recalls[i] + aided_recalls[i] /
2,
                  f'{aided_recalls[i]:.0f}%', ha='center', va='center',
color='black', fontsize=10)

    if never_heard[i] > 5:
        plt.text(i, 100 - never_heard[i] / 2, f'{never_heard[i]:.0f}%',
                  ha='center', va='center', color='black', fontsize=10)
plt.tight_layout()
plt.show()
```

**Grafik-14:** Marka Bilinirliği Bar Grafiği

Katılımcıların aklına ilk gelen mantolama sistemi markası B Markası (%69) olmuştur. B Markası'nı sırasıyla D Markası (%19) ve A Markası (%5) izlemektedir. Toplamda en çok bilinen marka olarak ilk üç; B Markası (%95), D Markası (%88) ve A Markası'dır. (%78)

```
brands = {1: 'B Markası', 2: 'C Markası', 3: 'A Markası', 4: 'F Markası', 5: 'D
Markası',
6: 'E Markası', 7: 'G Markası', 8: 'H Markası', 9: 'I Markası', 10: 'J Markası',
11: 'K Markası', 12: 'L Markası'}
```

```
data['B1map'] = data['B1'].map(brands)
```

```
def calculate_percentage_by_group(df, group_column, b1_column):
    group_counts = df.groupby(
        [group_column, b1_column]).size().unstack(fill_value=0)
```

```
group_percentages = group_counts.div(
    group_counts.sum(axis=1), axis=0) * 100
return group_percentages, group_counts.sum(axis=1)
age_group_percentages, age_group_counts = calculate_percentage_by_group(
    data, 'YasKota', 'B1map')
gender_percentages, gender_counts = calculate_percentage_by_group(
    data, 'CinsiyetKota', 'B1map')
ses_group_percentages, ses_group_counts = calculate_percentage_by_group(
    data, 'SESQuota', 'B1map')

for df_percentages in [age_group_percentages, gender_percentages,
ses_group_percentages]:

    for brand in brands.values():
        if brand not in df_percentages.columns:
            df_percentages[brand] = 0

age_group_percentages['Katılımcı Sayısı'] = age_group_counts.astype(int)
gender_percentages['Katılımcı Sayısı'] = gender_counts.astype(int)
ses_group_percentages['Katılımcı Sayısı'] = ses_group_counts.astype(int)

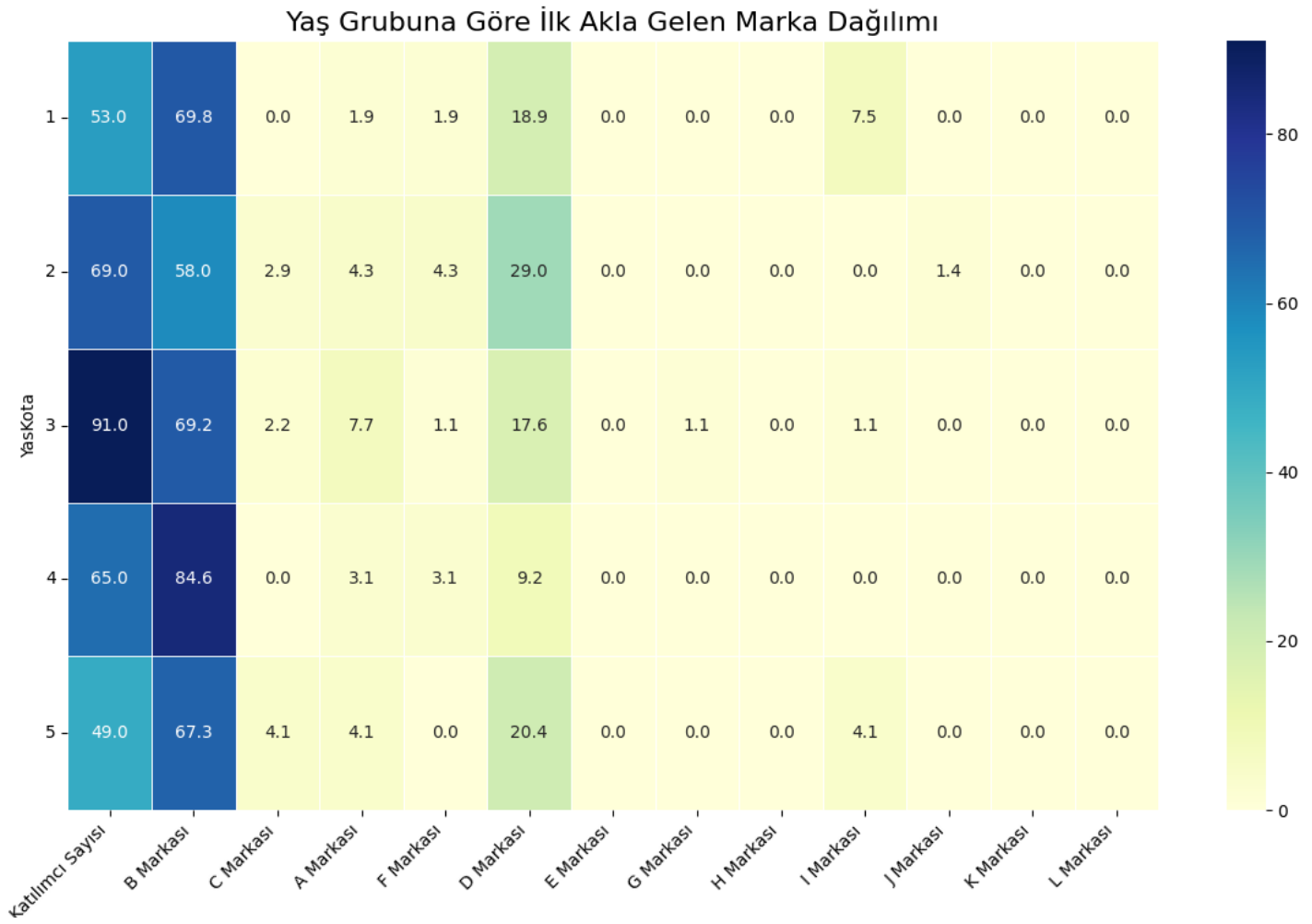
age_group_percentages = age_group_percentages[[
    'Katılımcı Sayısı'] + list(brands.values())]
gender_percentages = gender_percentages[[
    'Katılımcı Sayısı'] + list(brands.values())]
ses_group_percentages = ses_group_percentages[[
    'Katılımcı Sayısı'] + list(brands.values())]

age_group_percentages['Katılımcı Sayısı'] = age_group_percentages['Katılımcı
Sayısı'].astype(
    int)
gender_percentages['Katılımcı Sayısı'] = gender_percentages['Katılımcı
Sayısı'].astype(
    int)
```

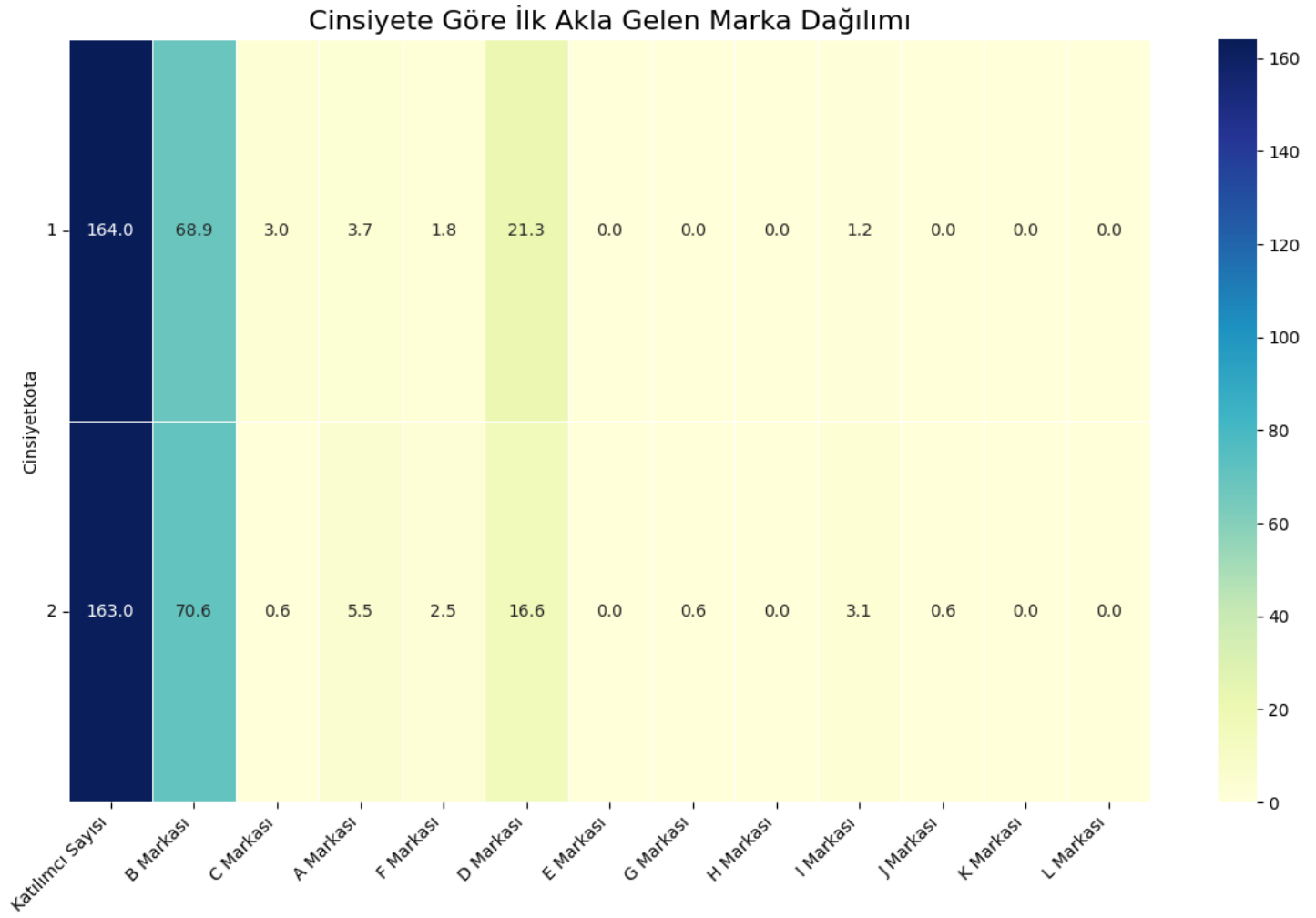
```
ses_group_percentages['Katılımcı Sayısı'] = ses_group_percentages['Katılımcı
Sayısı'].astype(
    int)

def plot_heatmap(df, title):
    plt.figure(figsize=(12, 8))
    sns.heatmap(df.astype(float), annot=True, fmt=".1f",
                cmap="YlGnBu", cbar=True, linewidths=0.5)
    plt.title(title, fontsize=16)
    plt.yticks(rotation=0)
    plt.xticks(rotation=45, ha='right')
    plt.tight_layout()
    plt.show()

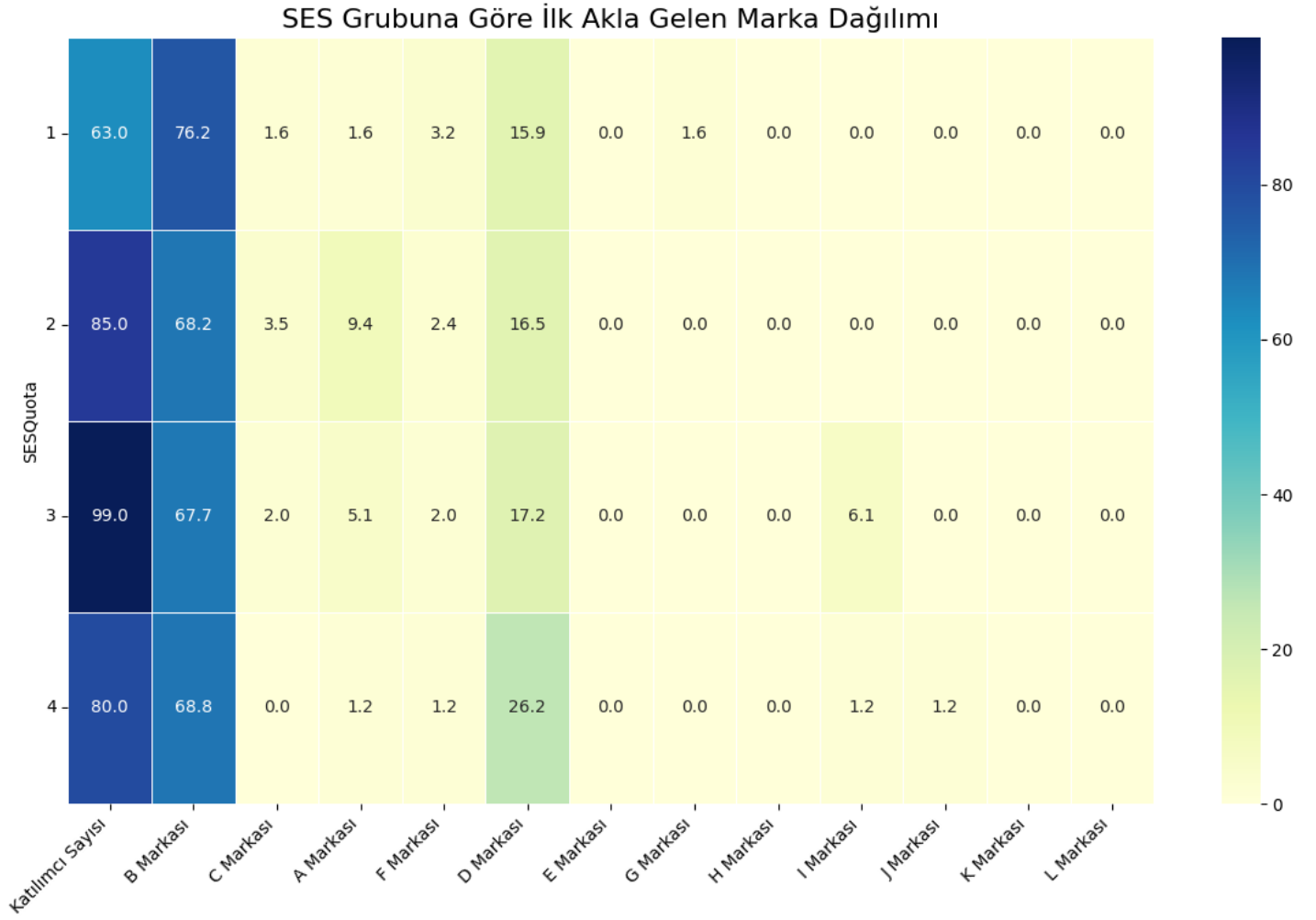
plot_heatmap(age_group_percentages,
             'Yaş Grubuna Göre İlk Akla Gelen Marka Dağılımı')
plot_heatmap(gender_percentages,
             'Cinsiyete Göre İlk Akla Gelen Marka Dağılımı')
plot_heatmap(ses_group_percentages,
             'SES Grubuna Göre İlk Akla Gelen Marka Dağılımı')
```



Grafik-15: Yaş Grubuna Göre İlk Akla Gelen Marka Dağılımı Grafiği



Grafik-16: Cinsiyete Göre İlk Akla Gelen Marka Dağılımı Grafiği



Grafik-17: SES Grubuna Göre İlk Akla Gelen Marka Dağılım Grafiği

Tüm yaş gruplarında akla ilk gelen marka, B Markası'dır. 164 kadın katılımcının %3.7'si, 163 erkek katılımcının %5.5'i «Akla İlk Gelen Marka» sorulduğunda A Markası cevabını vermiştir. A Markası markası C1 SES grubunda %9.4'luk oranla akla ilk gelen marka olmuştur.

```
brands = ['B Markası', 'C Markası', 'A Markası', 'F Markası', 'D Markası',
          'E Markası', 'G Markası', 'H Markası', 'I Markası', 'J Markası', 'K Markası', 'L
          Markası']

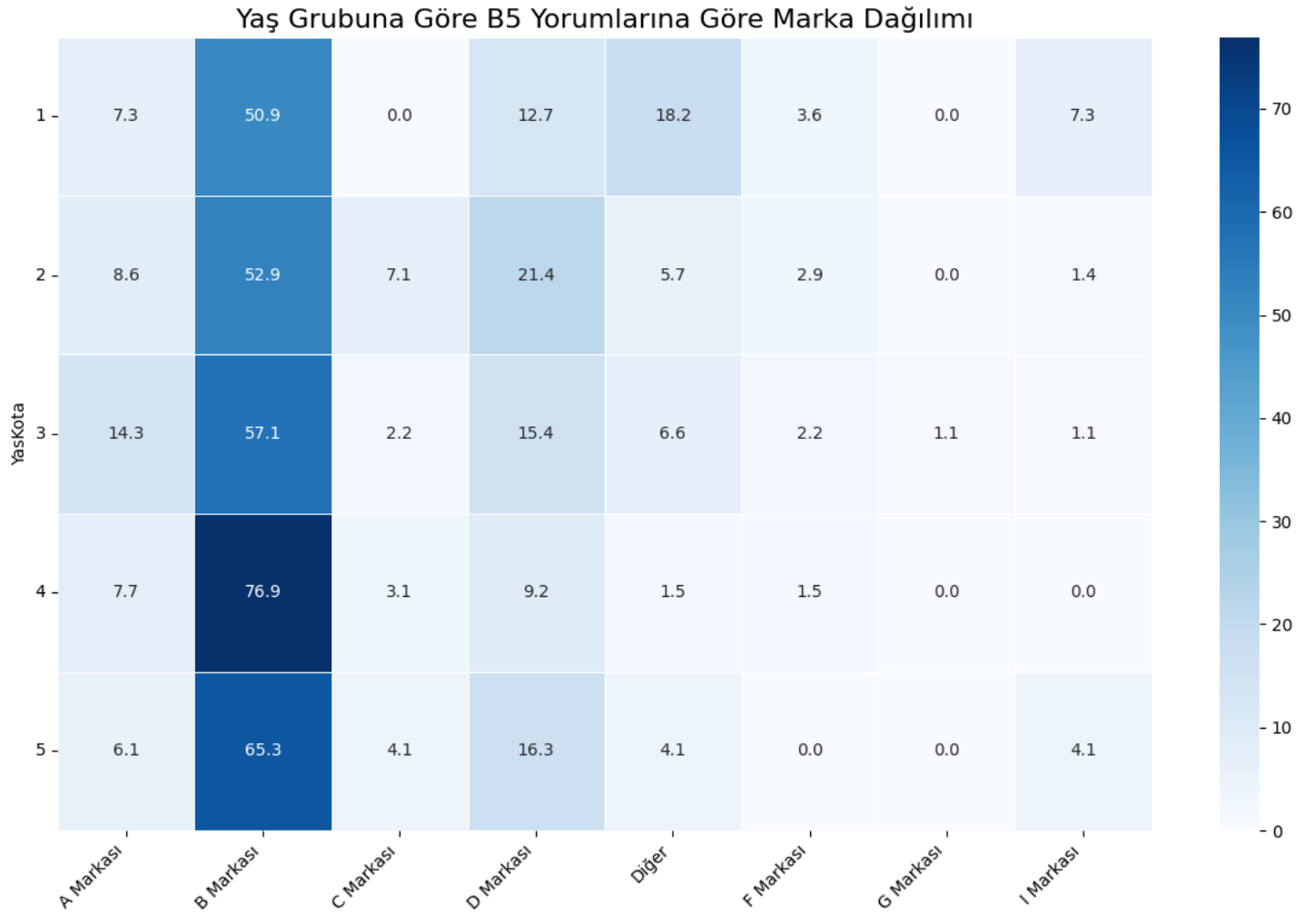
def identify_brand_in_comment(comment, brands):
    for brand in brands:
        if pd.notna(comment) and brand.lower() in comment.lower():
            return brand
    return 'Diğer'
```

```
data['B5map'] = data['B5'].apply(
    lambda x: identify_brand_in_comment(x, brands))
age_group_analysis = data.groupby(
    ['YasKota', 'B5map']).size().unstack(fill_value=0)
gender_analysis = data.groupby(
    ['CinsiyetKota', 'B5map']).size().unstack(fill_value=0)
ses_quota_analysis = data.groupby(
    ['SESQuota', 'B5map']).size().unstack(fill_value=0)

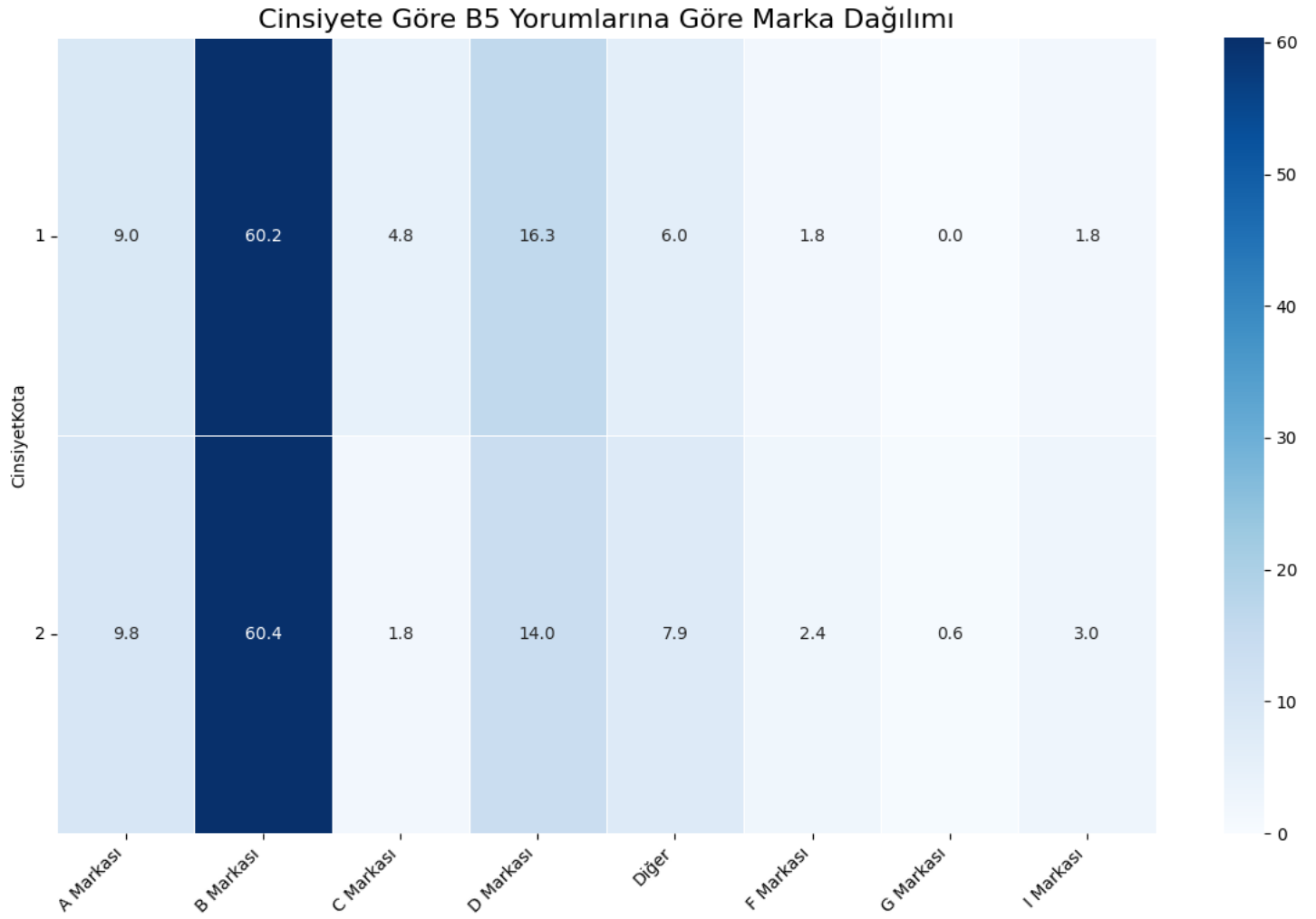
age_group_percentages = age_group_analysis.div(
    age_group_analysis.sum(axis=1), axis=0) * 100
gender_percentages = gender_analysis.div(
    gender_analysis.sum(axis=1), axis=0) * 100
ses_group_percentages = ses_quota_analysis.div(
    ses_quota_analysis.sum(axis=1), axis=0) * 100

def plot_heatmap(df, title):
    plt.figure(figsize=(12, 8))
    sns.heatmap(df, annot=True, fmt=".1f",
                cmap="Blues", cbar=True, linewidths=0.5)
    plt.title(title, fontsize=16)
    plt.xticks(rotation=45, ha='right')
    plt.yticks(rotation=0)
    plt.tight_layout()
    plt.show()

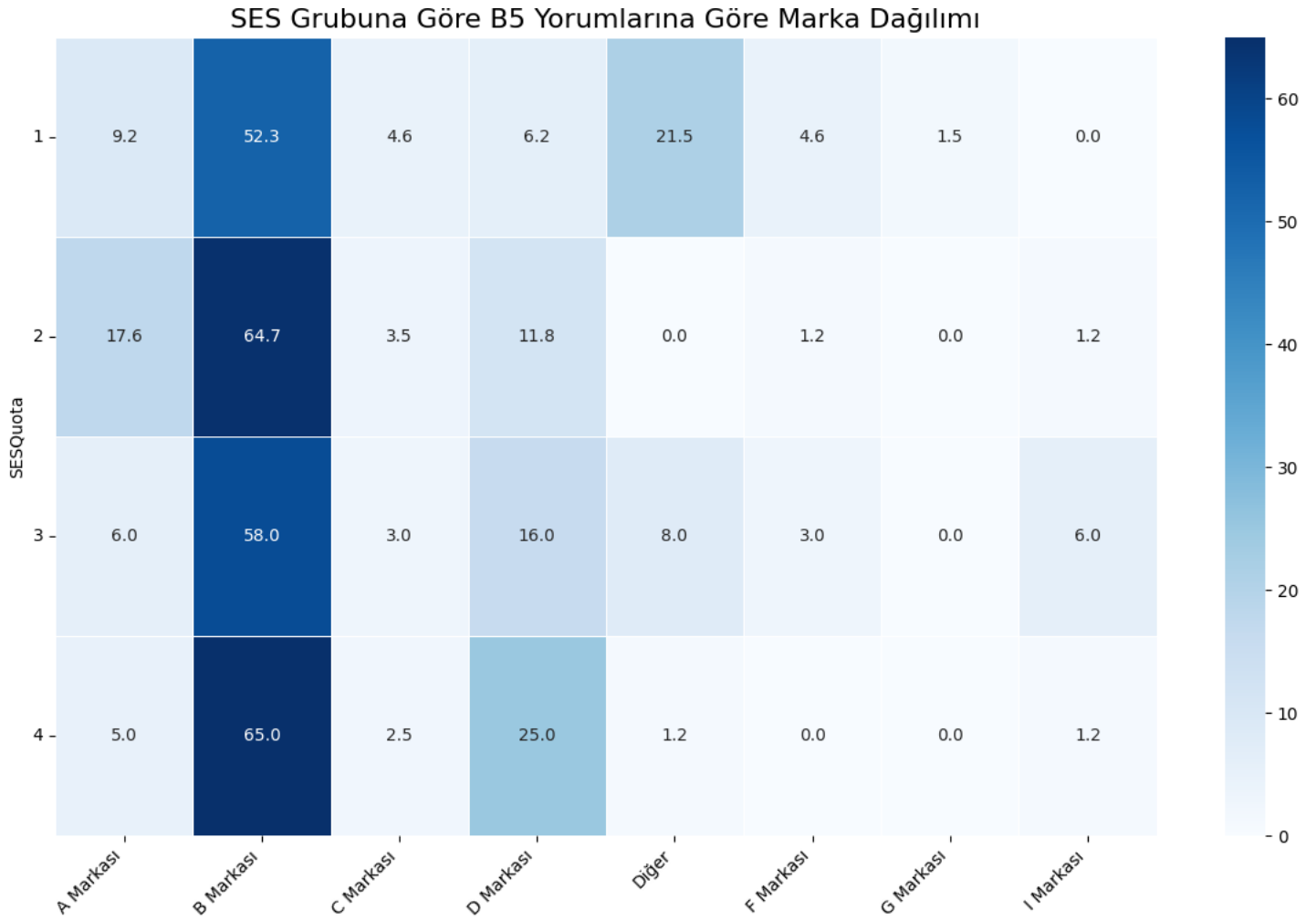
plot_heatmap(age_group_percentages,
              'Yaş Grubuna Göre B5 Yorumlarına Göre Marka Dağılımı')
plot_heatmap(gender_percentages,
              'Cinsiyete Göre B5 Yorumlarına Göre Marka Dağılımı')
plot_heatmap(ses_group_percentages,
              'SES Grubuna Göre B5 Yorumlarına Göre Marka Dağılımı')
```



Grafik-18: Yaş Grubuna Göre B5 Yorumlarına Göre Marka Dağılım Grafiği



Grafik-19: Cinsiyete Göre B5 Yorumlarına Göre Marka Dağılımı Grafiği



Grafik-20: SES Grubuna Göre B5 Yorumlarına Göre Marka Dağılımı Grafiği

Katılımcıların %61'i, Avrupa'nın mantolama devi ve pazar lideri mantolama sistemleri üreticisinin «B Markası» olduğunu belirtmiştir. 45-54 yaş aralığındaki katılımcıların %76.9'u, kadın katılımcıların %57.2'si B Markası cevabını vermiştir. A Markası cevabını verenler %14.3'ü «35- 44 yaş» aralığında, %1.5'i «25-34 yaş»; %7.7'si «45-54 yaş» aralığında, %18.2'si «55 yaş ve üzeri ve %4.1'i «18-24 yaş» aralığındadır. Yine bu katılımcıların %9'u erkek, %9.8'i kadındır.

```
comments = data['MS01'].dropna()
def clean_text(text):
    text = text.lower()
    text = re.sub(r'[\^\w\s]', '', text)
    text = text.strip()
    return text
```

```

categories = {
    "Lider ve Başarılı Mantolama Markası": ["lider", "başarılı", "en iyi", "dev",
"avrupa", "öncü", "tanınan", "popüler", "kaliteli", "bilinen", "güçlü", "en
bilindik"],
    "Kaliteli ve Güvenilir Mantolama": ["kaliteli", "güvenilir", "güven", "kalite",
"yüksek performans", "üstün", "seçilen", "tercih edilen", "uzun ömürlü"],
    "Ekonomik ve Tasarruflu Mantolama": ["ekonomik", "tasarruf", "fiyat", "ucuz",
"maliyet", "ekonomi"],
    "Çevre Dostu ve Dayanıklı": ["çevre dostu", "dayanıklı", "uzun ömür", "iklim",
"çürümeye karşı", "sürdürülebilir", "doğa", "dayanıklılık"],
    "Isı Yalıtımı ve Konfor Sağlayan": ["ısı", "yalıtım", "soğuk", "sıcak",
"konfor", "ısı tutucu", "ısı kaybını önleme", "ev", "duvar"],
    "Sevilen ve Popüler Marka": ["sevilen", "güzel", "popüler", "beğenilen", "takdir
edilen"],
    "Haşere ve Bakteri Önleyici": ["haşere", "böcek", "mikrop", "bakteri", "küf",
"nem", "koku"],
    "Orta Seviye ve Sıradan Mantolama Markası": ["orta", "sıradan", "normal",
"bilinen", "herhangi bir", "tanınmayan"],
    "Yanmazlık ve Alev Geçirmezlik Özelliği": ["yanmaz", "alev", "geçirmez", "ısıya
dayanıklı"],
    "İnsan Sağlığına Duyarlı": ["sağlık", "insan sağlığı", "duyarlı", "önem veren"],
    "Kullanım Kolaylığı": ["kullanım kolaylığı", "kolay kullanım", "erişilebilir",
"kolay ulaşılan", "kullanımı rahat"],
    "En İyi Termal Yalıtkanlar": ["termal yalıtkan", "ısı iletkenliği", "termal"],
}

```

```

def categorize_comment(comment):
    for category, keywords in categories.items():
        for keyword in keywords:
            if keyword in comment:
                return category
    return "Diğer"

cleaned_comments = comments.apply(clean_text)
categorized_comments = cleaned_comments.apply(categorize_comment)

```

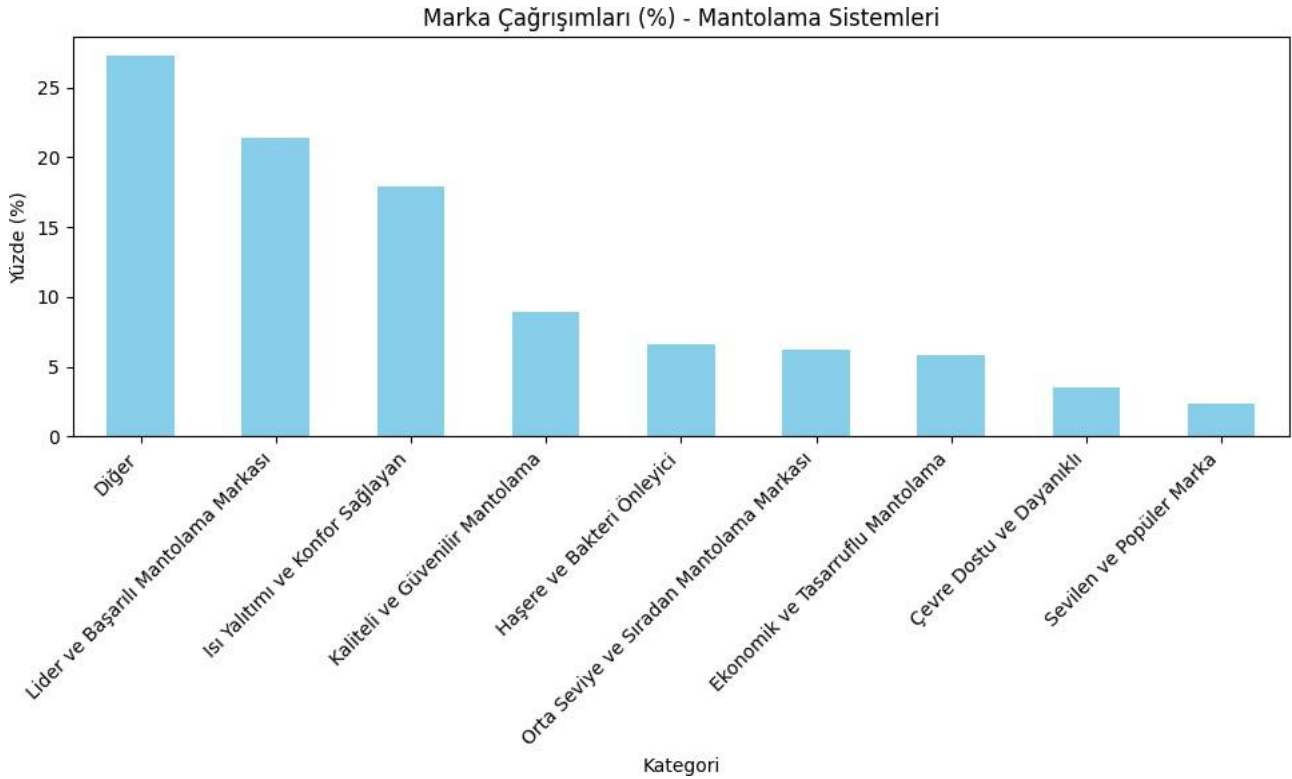
```

category_counts = categorized_comments.value_counts(normalize=True) * 100

plt.figure(figsize=(10, 6))
category_counts.plot(kind='bar', color='skyblue')
plt.title('Marka Çağrışımları (%) - Mantolama Sistemleri')
plt.ylabel('Yüzde (%)')
plt.xlabel('Kategori')
plt.xticks(rotation=45, ha='right')
plt.tight_layout()
plt.show()

result_table = pd.DataFrame(category_counts).reset_index()
result_table.columns = ['Kategori', 'Yüzde (%)']
print(result_table)

```



Grafik-21: A Markası Marka Çağrışımları (%) - Mantolama Sistemleri Bar Grafiği

A Markası dendiğinde katılımcıların aklına gelen ilk 3 şey; «Lider ve Başarılı Mantolama Markası» (%21.400778), «Isı Yalıtımı ve Konfor Sağlayan» (%17.898833), «Kaliteli ve Güvenilir Mantolama» (%8.949416) olduğudur.

```

comments = data['MS04'].dropna()

def clean_text(text):
    text = text.lower()
    text = re.sub(r'^\w\s|', '', text)
    text = text.strip()
    return text

categories_ms04 = {
    "Lider ve Başarılı Mantolama Markası": ["lider", "başarılı", "öncü",
"tanınan", "popüler", "kaliteli", "bilinen", "güçlü", "en bilindik"],
    "Kaliteli ve Şık Mantolama": ["kaliteli", "şık", "güzel", "estetik"],
    "Ekonomik ve Tasarruflu Mantolama": ["ekonomik", "tasarruf", "ucuz",
"fiyat", "maliyet"],
    "Güvenilir ve Sevilen Marka": ["güvenilir", "sevilen", "beğenilen",
"takdir edilen"],
    "Isı Yalıtımı ve Dayanıklılık": ["ısı", "yalıtım", "konfor",
"dayanıklı", "nem", "rutubet", "çürümeye karşı", "duvar", "soğuk", "sıcak"],
    "Çevre Dostu ve Enerji Tasarrufu": ["çevre dostu", "iklim", "enerji
tasarrufu", "çevreye duyarlı", "doğa"],
    "Haşere ve Böcek Yapmayan": ["haşere", "böcek", "küf", "mikrop",
"bakteri"],
    "Pahalı Yalıtım": ["pahalı", "yüksek fiyat"],
    "İnsan Sağlığına Duyarlı": ["insan sağlığı", "sağlık", "duyarlı", "önem
veren"],
    "Kullanım Kolaylığı ve Renovasyon": ["kullanım kolaylığı", "kolay
kullanım", "renovasyon", "kolay çözümler", "ilgili", "ulaşılabilir"],
    "En İyi İç ve Dış Cephe Yalıtımı": ["iç cephe", "dış cephe", "termal
yalıtkan", "ısı iletkenliği", "en iyi"],
    "Sıradan Yalıtım Markası": ["sıradan", "orta düzey", "normal"],
    "Duvar Boyası": ["duvar boyası"],
}

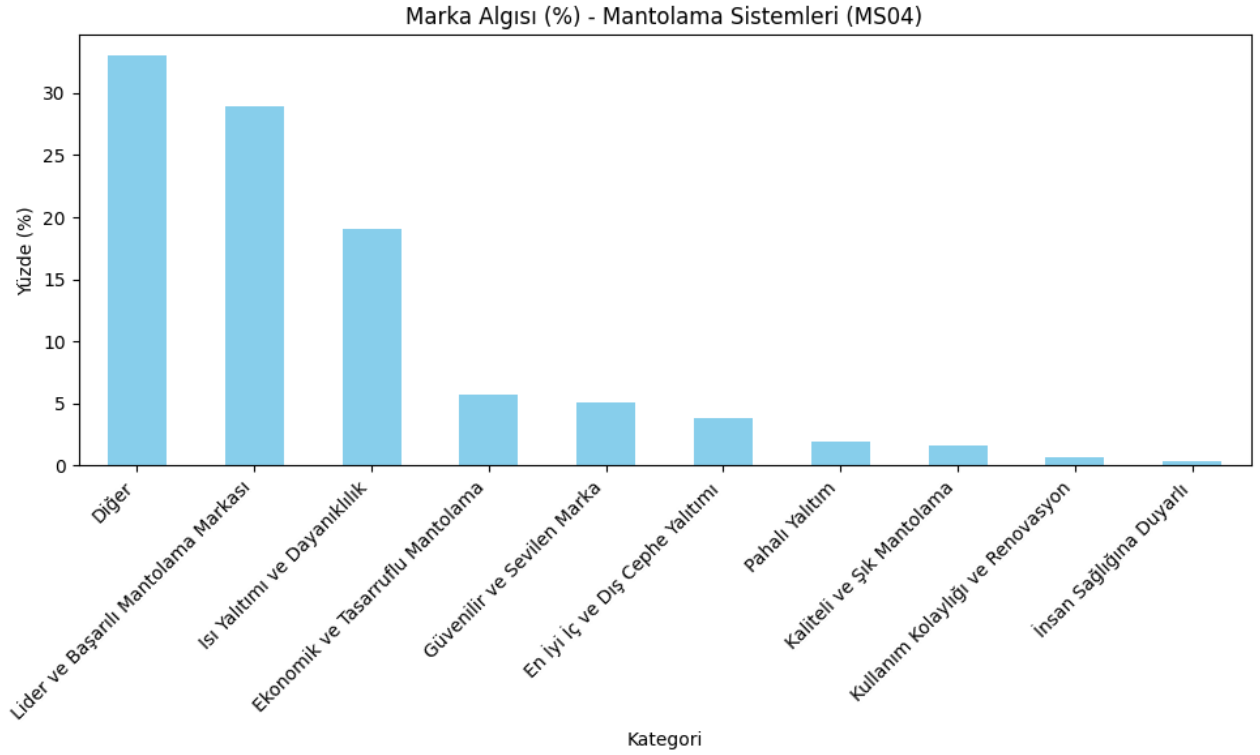
```



```
def categorize_comment(comment, categories):
    for category, keywords in categories.items():
        for keyword in keywords:
            if keyword in comment:
                return category
    return "Diğer"

cleaned_comments = comments.apply(clean_text)
categorized_comments = cleaned_comments.apply(
    categorize_comment, categories=categories_ms04)
category_counts = categorized_comments.value_counts(normalize=True) * 100
plt.figure(figsize=(10, 6))
category_counts.plot(kind='bar', color='skyblue')
plt.title('Marka Algısı (%) - Mantolama Sistemleri (MS04)')
plt.ylabel('Yüzde (%)')
plt.xlabel('Kategori')
plt.xticks(rotation=45, ha='right')
plt.tight_layout()
plt.show()

result_table = pd.DataFrame(category_counts).reset_index()
result_table.columns = ['Kategori', 'Yüzde (%)']
print(result_table)
```



Grafik-22: B Markası Marka Algısı (%) -Mantolama Sistemleri Bar Grafiği

B Markası dendiğinde katılımcıların aklına gelen ilk 3 şey; «Lider ve Başarılı Mantolama Markası» (%28.888889), «Isı Yalıtımı ve Dayanıklılık» (%19.047619), «Ekonomik ve Tasarruflu Mantolama» (%5.714286) olduğudur.

```
comments = data['MS07'].dropna()

def clean_text(text):
    text = text.lower()
    text = re.sub(r'^\w\s', '', text)
    text = text.strip()
    return text

categories_ms07 = {
    "Lider ve Başarılı Mantolama Markası": ["lider", "başarılı", "öncü", "tanınan",
    "popüler", "kaliteli", "bilinen", "güçlü", "en bilindik"],
    "Alman Mantolama Sistemi": ["alman", "almanya", "alman mantolama sistemi"],
    "Kaliteli ve Şık Mantolama": ["kaliteli", "şık", "güzel", "estetik"],
```

```

    "Ekonomik ve Tasarruflu Mantolama": ["ekonomik", "tasarruf", "ucuz", "fiyat",
    "maliyet", "dış cephe"],
    "Güvenilir ve Dayanıklı Mantolama": ["güvenilir", "dayanıklı", "uzun ömürlü",
    "ömür", "sağlam"],
    "Isı Yalıtımı ve Dayanıklılık": ["ısı", "yalıtım", "konfor", "dayanıklı", "nem",
    "rutubet", "çürümeye karşı", "duvar", "soğuk", "sıcak"],
    "Çevre Dostu ve Enerji Tasarrufu": ["çevre dostu", "iklim", "enerji tasarrufu",
    "çevreye duyarlı", "doğa"],
    "Haşere ve Böcek Yapmayan": ["haşere", "böcek", "küf", "mikrop", "bakteri"],
    "Boya ve Yapıştırıcı": ["boya", "yapıştırıcı", "cephe uygulamaları"],
    "Kullanım Kolaylığı ve Renovasyon": ["kullanım kolaylığı", "kolay kullanım",
    "renovasyon", "kolay çözümler", "ilgili", "ulaşılabilir"],
    "Sıradan Yalıtım Markası": ["sıradan", "orta düzey", "normal"],
    "Diğer": ["diğer"]
}

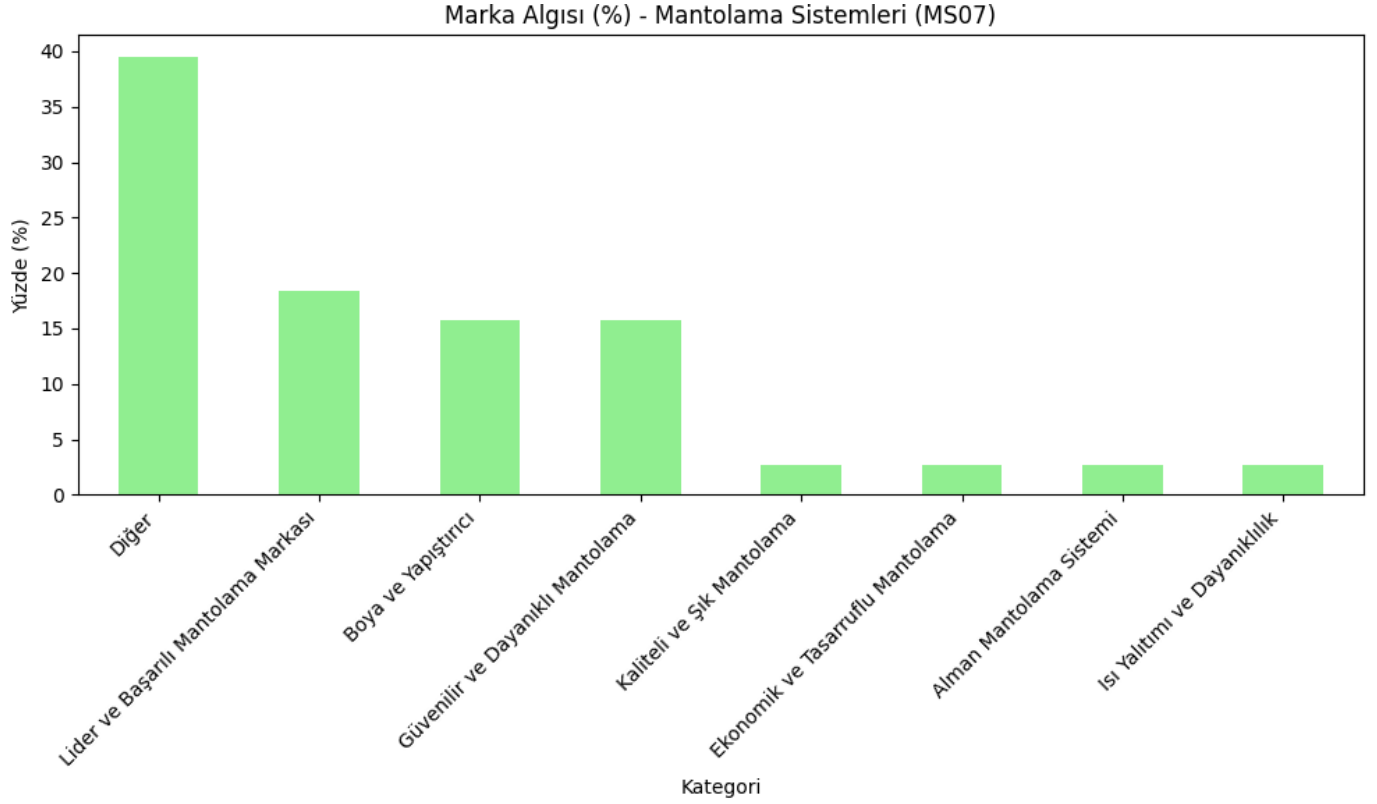
def categorize_comment(comment, categories):
    for category, keywords in categories.items():
        for keyword in keywords:
            if keyword in comment:
                return category
    return "Diğer"

cleaned_comments = comments.apply(clean_text)
categorized_comments = cleaned_comments.apply(
    categorize_comment, categories=categories_ms07)
category_counts = categorized_comments.value_counts(normalize=True) * 100
plt.figure(figsize=(10, 6))
category_counts.plot(kind='bar', color='lightgreen')
plt.title('Marka Algısı (%) - Mantolama Sistemleri (MS07)')
plt.ylabel('Yüzde (%)')
plt.xlabel('Kategori')
plt.xticks(rotation=45, ha='right')
plt.tight_layout()
plt.show()

result_table = pd.DataFrame(category_counts).reset_index()
result_table.columns = ['Kategori', 'Yüzde (%)']

```

```
print(result_table)
```



Grafik-23: C Markası Marka Algısı (%) -Mantolama Sistemleri Bar Grafiği

C Markası dendiğinde katılımcıların aklına gelen ilk 3 şey; «Lider ve Başarılı Mantolama Markası» (%18.421053), «Boya ve Yapıştırıcı» (%15.789474), «Güvenilir ve Dayanıklı Mantolama» (%15.789474) olduğudur.

```
AMarkası_scores =
data['MS02'].dropna()
BMarkası_scores =
data['MS05'].dropna()
CMarkası_scores =
data['MS08'].dropna()
EMarkası_scores =
data['MS11'].dropna()
AMarkası_avg = AMarkası_scores.mean()
BMarkası_avg = BMarkası_scores.mean()
CMarkası_avg = CMarkası_scores.mean()
```

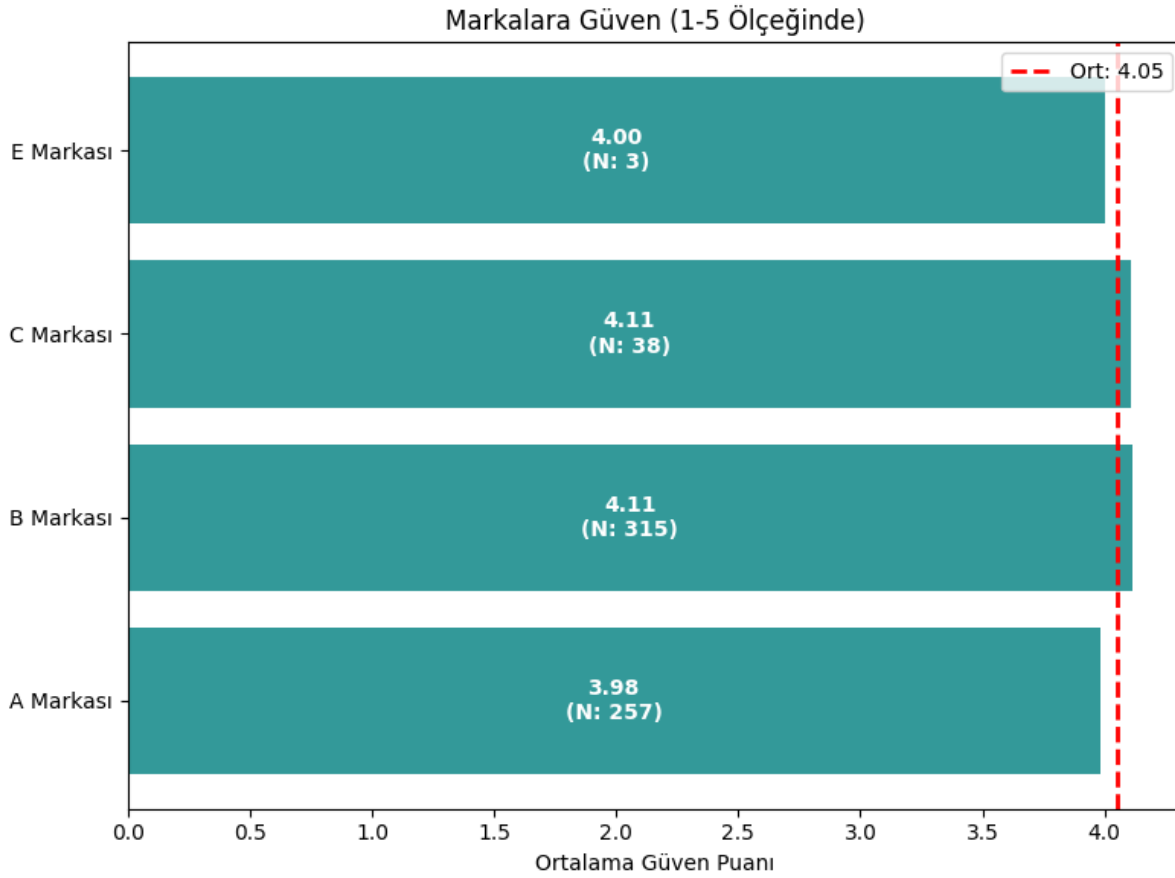
```

EMarkası_avg = EMarkası_scores.mean()

AMarkası_count = len(AMarkası_scores)
BMarkası_count = len(BMarkası_scores)
CMarkası_count = len(CMarkası_scores)
EMarkası_count = len(EMarkası_scores)
general_avg = (AMarkası_avg + BMarkası_avg + CMarkası_avg +
EMarkası_avg) / 4
scores_df = pd.DataFrame({
    'Marka': ['A Markası', 'B Markası', 'C Markası', 'E Markası'],
    'Ortalama Güven Puanı': [AMarkası_avg, BMarkası_avg, CMarkası_avg, EMarkası_avg],
    'Katılımcı Sayısı': [AMarkası_count, BMarkası_count, CMarkası_count, EMarkası_count]
})
plt.figure(figsize=(8, 6))
bars = plt.barh(
    scores_df['Marka'], scores_df['Ortalama Güven Puanı'], color='teal', alpha=0.8)
plt.axvline(x=general_avg, color='red', linestyle='--',
            label=f'Ort: {general_avg:.2f}', lw=2)
plt.xlabel('Ortalama Güven Puanı')
plt.title('Markalara Güven (1-5 Ölçeğinde)')
plt.legend(loc='upper right')

for i, bar in enumerate(bars):
    width = bar.get_width()
    participants = scores_df['Katılımcı Sayısı'][i]
    plt.text(width / 2, bar.get_y() + bar.get_height() / 2,
            f'{width:.2f}\n(N: {participants})', ha='center', va='center',
            color='white', fontweight='bold')
plt.tight_layout()
plt.show()

```



Grafik-24: Markalara Güven Bar Grafiği

Katılımcılara markalara olan güven düzeyleri sorulduğunda, en çok güvendikleri markanın 4.13/5 ortalama ile B Markası ve C Markası olduğunu belirtmişlerdir. Genel güven ortalaması 4.05/5'dir. E Markası ve A Markası genel ortalamasının altında kalan markalar olmuşlardır.

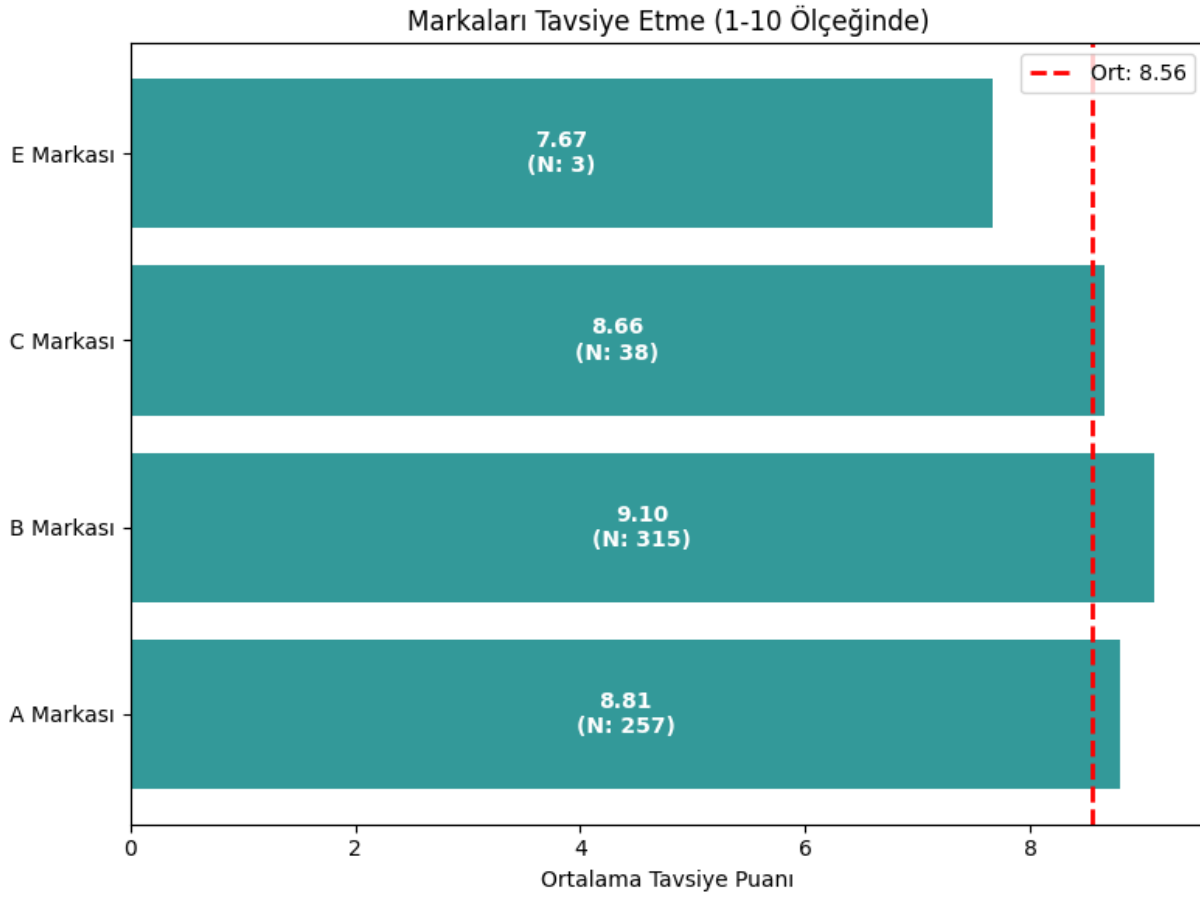
```
AMarkası_scores = data['MS03'].dropna()
BMarkası_scores = data['MS06'].dropna()
CMarkası_scores = data['MS09'].dropna()
EMarkası_scores = data['MS12'].dropna()
AMarkası_avg = AMarkası_scores.mean()
BMarkası_avg = BMarkası_scores.mean()
CMarkası_avg = CMarkası_scores.mean()
EMarkası_avg = EMarkası_scores.mean()

AMarkası_count = len(AMarkası_scores)
```

```
BMarkası_count = len(BMarkası_scores)
CMarkası_count = len(CMarkası_scores)
EMarkası_count = len(EMarkası_scores)

general_avg = (AMarkası_avg + BMarkası_avg + CMarkası_avg + EMarkası_avg) / 4

scores_df = pd.DataFrame({
    'Marka': ['A Markası', 'B Markası', 'C Markası', 'E Markası'],
    'Ortalama Tavsiye Puanı': [AMarkası_avg, BMarkası_avg, CMarkası_avg, EMarkası_avg],
    'Katılımcı Sayısı': [AMarkası_count, BMarkası_count, CMarkası_count,
EMarkası_count]}) plt.figure(figsize=(8, 6))
bars = plt.barh(
    scores_df['Marka'], scores_df['Ortalama Tavsiye Puanı'], color='teal',
alpha=0.8)
plt.axvline(x=general_avg, color='red', linestyle='--',
            label=f'Ort: {general_avg:.2f}', lw=2)
plt.xlabel('Ortalama Tavsiye Puanı')
plt.title('Markaları Tavsiye Etme (1-10 Ölçeğinde)')
plt.legend(loc='upper right')
for i, bar in enumerate(bars):
    width = bar.get_width()
    participants = scores_df['Katılımcı Sayısı'][i]
    plt.text(width / 2, bar.get_y() + bar.get_height() / 2,
            f'{width:.2f}\n(N: {participants})', ha='center', va='center',
color='white', fontweight='bold')
plt.tight_layout()
plt.show()
```



Grafik-25: Markaları Tavsiye Etme Bar Grafiği

Katılımcılara markaları tavsiye etme düzeyleri sorulduğunda, en çok tavsiye edilen marka 9.10/10 ortalama ile B Markası'dır. İkinci sırada 8.81/10 ortalama ile A Markası, üçüncü sırada ise 8.66/10 ortalama ile C Markası bulunmaktadır. Genel tavsiye etme ortalaması 8.56/10'dur. Sadece E Markası genel ortalamasının altında kalmıştır.

```
columns = ['MS13_r1_c1', 'MS13_r1_c2', 'MS13_r1_c3', 'MS13_r1_c6']

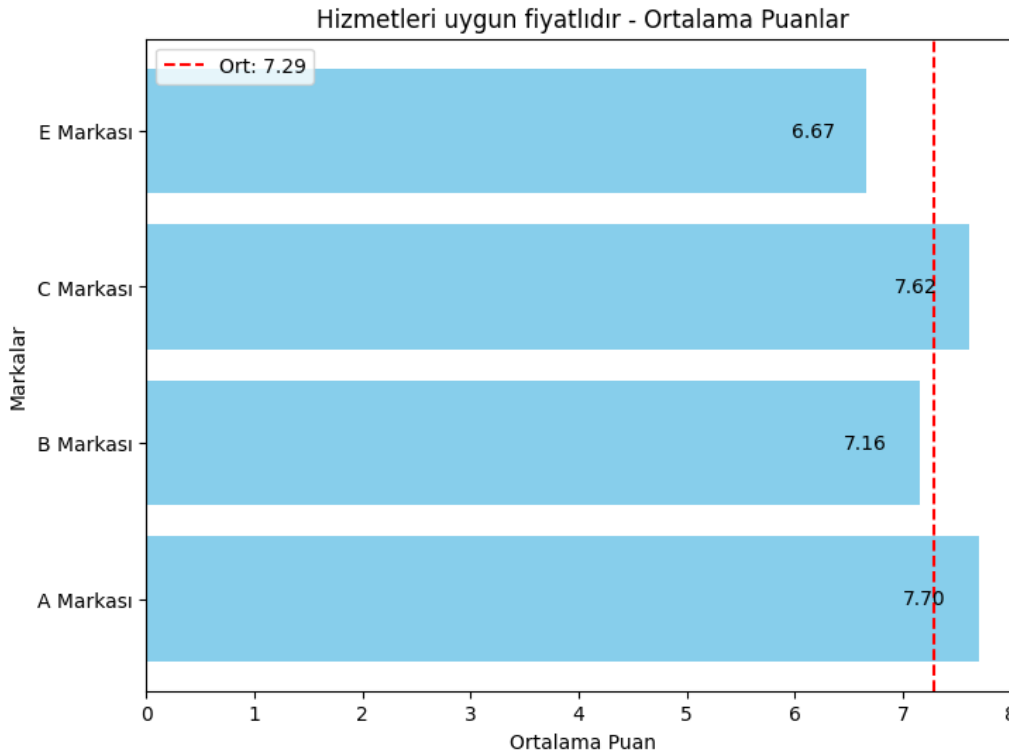
scores = data[columns].mean()
scores_df = pd.DataFrame({
    'Markalar': ['A Markası', 'B Markası', 'C Markası', 'E Markası'],
    'Ortalama': scores
})

general_average = scores.mean()
```



```
plt.figure(figsize=(8, 6))
bars = plt.barh(scores_df['Markalar'], scores_df['Ortalama'], color='skyblue')

for bar in bars:
    plt.text(bar.get_width() - 0.3, bar.get_y() + bar.get_height()/2,
             f'{bar.get_width():.2f}', va='center', ha='right', color='black')
plt.axvline(general_average, color='red', linestyle='--',
            label=f'Ort: {general_average:.2f}')
plt.title('Hizmetleri uygun fiyatlıdır - Ortalama Puanlar')
plt.xlabel('Ortalama Puan')
plt.ylabel('Markalar')
plt.legend()
plt.show()
```



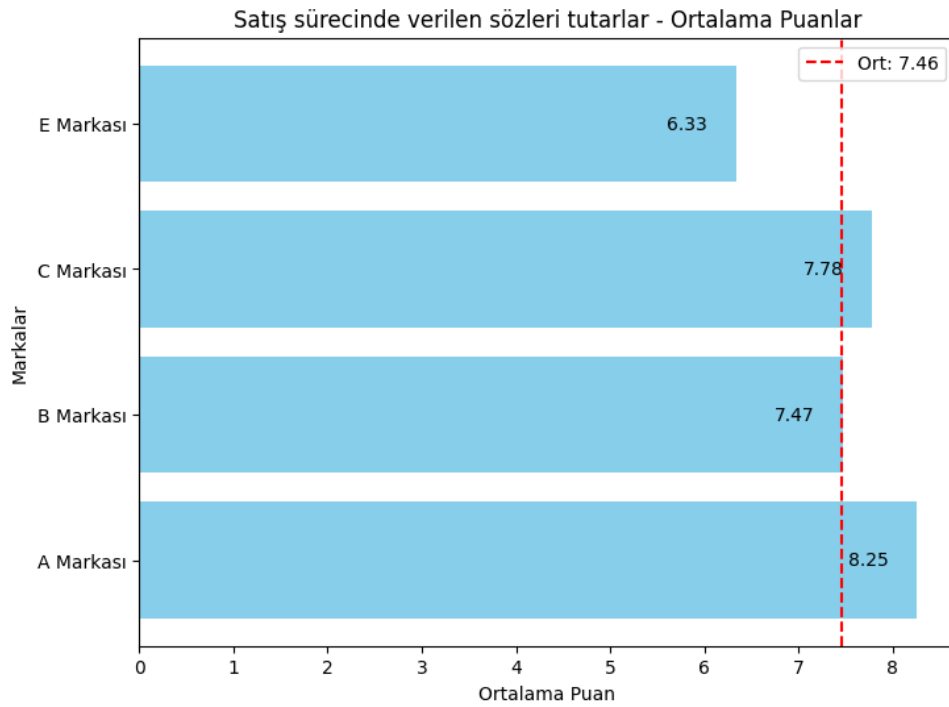
Grafik-26: Markaların Hizmetleri Uygun Fiyatlıdır Yorumu için Bar Grafiği

```
columns = ['MS13_r2_c1', 'MS13_r2_c2', 'MS13_r2_c3', 'MS13_r2_c6']
scores = data[columns].mean()
scores_df = pd.DataFrame({
    'Markalar': ['A Markası', 'B Markası', 'C Markası', 'E Markası'],
```

```

    'Ortalama': scores}}
general_average = scores.mean()
plt.figure(figsize=(8, 6))
bars = plt.barh(scores_df['Markalar'], scores_df['Ortalama'], color='skyblue')
for bar in bars:
    plt.text(bar.get_width() - 0.3, bar.get_y() + bar.get_height()/2,
             f'{bar.get_width():.2f}', va='center', ha='right', color='black')
plt.axvline(general_average, color='red', linestyle='--',
            label=f'Ort: {general_average:.2f}')
plt.title('Satış sürecinde verilen sözleri tutarlar - Ortalama Puanlar')
plt.xlabel('Ortalama Puan')
plt.ylabel('Markalar')
plt.legend()
plt.show()

```



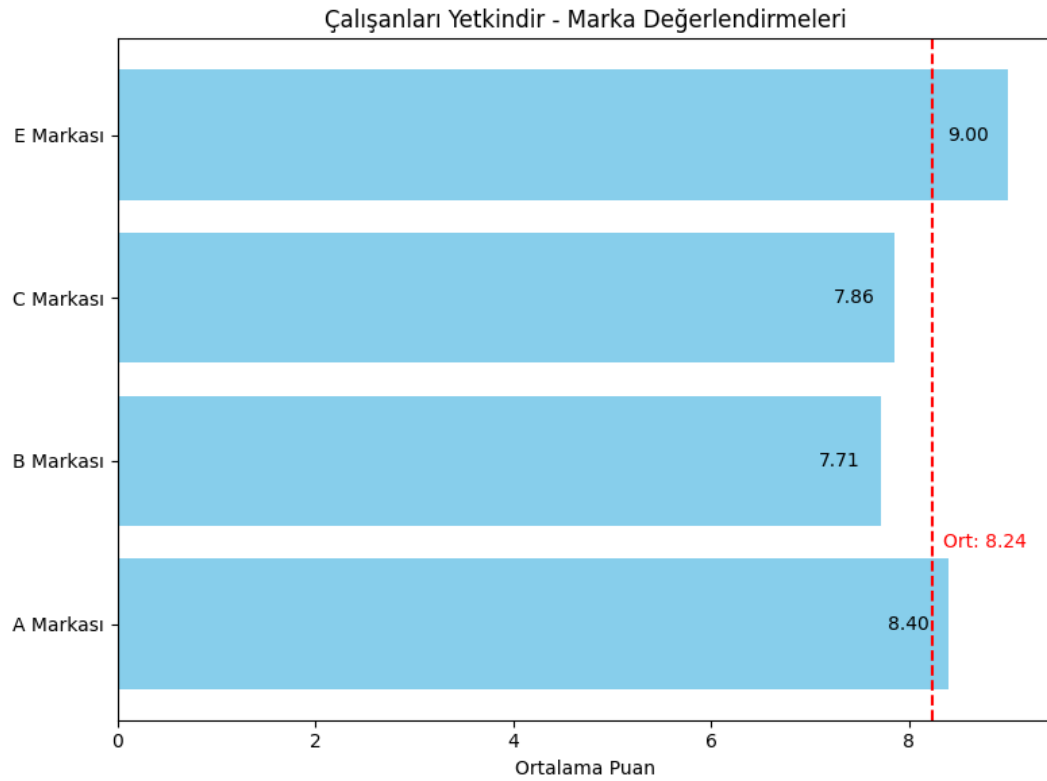
Grafik-27: Markaların Satış Sürecinde Verilen Sözleri Tutarlar Yorumu için Bar Grafiği

```

columns = ['MS13_r3_c1', 'MS13_r3_c2', 'MS13_r3_c3', 'MS13_r3_c6']
filtered_data = data[columns].dropna(how='all')
averages_yetkindir = filtered_data.mean()
plt.figure(figsize=(8, 6))
brands = ['A Markası', 'B Markası', 'C Markası', 'E Markası']

```

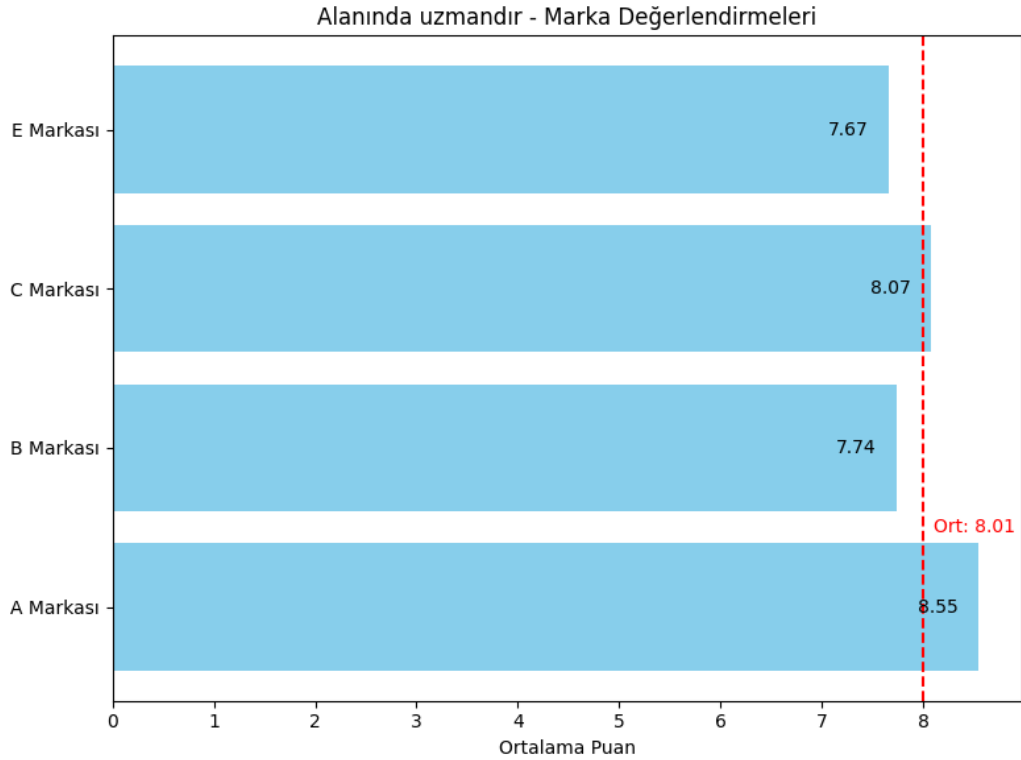
```
plt.barh(brands, averages_yetkindir, color='skyblue')
for i, avg in enumerate(averages_yetkindir):
    plt.text(avg - 0.2, i, f'{avg:.2f}',
             va='center', ha='right', color='black')
overall_average = averages_yetkindir.mean()
plt.axvline(overall_average, color='red', linestyle='--')
plt.text(overall_average + 0.1, 0.5,
        f'Ort: {overall_average:.2f}', color='red', va='center')
plt.title('Çalışanları Yetkindir - Marka Değerlendirmeleri')
plt.xlabel('Ortalama Puan')
plt.tight_layout()
plt.show()
```



Grafik-28: Markaların Çalışanları Yetkindir Yorumu için Bar Grafiği

```
columns = ['MS13_r4_c1', 'MS13_r4_c2', 'MS13_r4_c3', 'MS13_r4_c6']
filtered_data = data[columns].dropna(how='all')
averages_yetkindir = filtered_data.mean()
plt.figure(figsize=(8, 6))
brands = ['A Markası', 'B Markası', 'C Markası', 'E Markası']
```

```
plt.barh(brands, averages_yetkindir, color='skyblue')
for i, avg in enumerate(averages_yetkindir):
    plt.text(avg - 0.2, i, f'{avg:.2f}',
             va='center', ha='right', color='black')
overall_average = averages_yetkindir.mean()
plt.axvline(overall_average, color='red', linestyle='--')
plt.text(overall_average + 0.1, 0.5,
        f'Ort: {overall_average:.2f}', color='red', va='center')
plt.title('Alanında uzmandır - Marka Değerlendirmeleri')
plt.xlabel('Ortalama Puan')
plt.tight_layout()
plt.show()
```



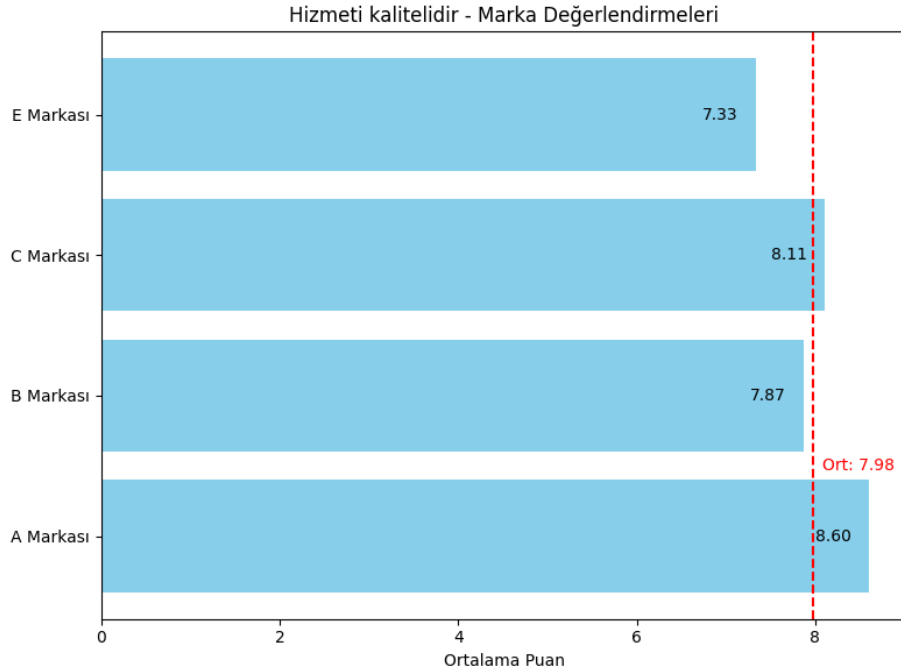
Grafik-29: Markalar Alanında Uzmandır Yorumu için Bar Grafiği

```
columns = ['MS13_r5_c1', 'MS13_r5_c2', 'MS13_r5_c3', 'MS13_r5_c6']
filtered_data = data[columns].dropna(how='all')
averages_yetkindir = filtered_data.mean()
plt.figure(figsize=(8, 6))
brands = ['A Markası', 'B Markası', 'C Markası', 'E Markası']
```

```

plt.barh(brands, averages_yetkindir, color='skyblue')
for i, avg in enumerate(averages_yetkindir):
    plt.text(avg - 0.2, i, f'{avg:.2f}',
             va='center', ha='right', color='black')
overall_average = averages_yetkindir.mean()
plt.axvline(overall_average, color='red', linestyle='--')
plt.text(overall_average + 0.1, 0.5,
        f'Ort: {overall_average:.2f}', color='red', va='center')
plt.title('Hizmeti kalitelidir - Marka Değerlendirmeleri')
plt.xlabel('Ortalama Puan')
plt.tight_layout()
plt.show()

```



Grafik-30: Markaların Hizmetleri Kalitelidir Yorumu için Bar Grafiği

```

columns = ['MS13_r6_c1', 'MS13_r6_c2', 'MS13_r6_c3', 'MS13_r6_c6']
filtered_data = data[columns].dropna(how='all')
averages_yetkindir = filtered_data.mean()
plt.figure(figsize=(8, 6))

```

```

brands = ['A Markası', 'B Markası', 'C Markası', 'E Markası']
plt.barh(brands, averages_yetkindir, color='skyblue')
for i, avg in enumerate(averages_yetkindir):
    plt.text(avg - 0.2, i, f'{avg:.2f}',
             va='center', ha='right', color='black')
overall_average = averages_yetkindir.mean()
plt.axvline(overall_average, color='red', linestyle='--')
plt.text(overall_average + 0.1, 0.5,
        f'Ort: {overall_average:.2f}', color='red', va='center')
plt.title('Bir sorun olduğunda muhatap bulabilmek kolaydır - Marka
Değerlendirmeleri')
plt.xlabel('Ortalama Puan')
plt.tight_layout()
plt.show()

```



Grafik-31: Markalar için Bir Sorunla Karşılaşıldığında Muhattap Bulabilmek Kolaydır Yorumu için Bar Grafiği

```

ms14_counts = data['MS14'].value_counts(normalize=True) * 100
ms15_counts = data[data['MS14'] == 1]['MS15'].value_counts(
    normalize=True) * 100
labels_MS14 = ['Evet, haberim var/duydum', 'Hayır, haberim yok/hiç duymadım']

```

```
percentages_MS14 = [ms14_counts.get(1, 0), ms14_counts.get(2, 0)]
participants_MS14 = len(data['MS14'].dropna())

labels_MS15 = ['Evet, kullanmayı düşünüyorum',
               'Hayır, kullanmayı düşünmüyorum']
percentages_MS15 = [ms15_counts.get(1, 0), ms15_counts.get(2, 0)]
participants_MS15 = len(data[data['MS14'] == 1]['MS15'].dropna())

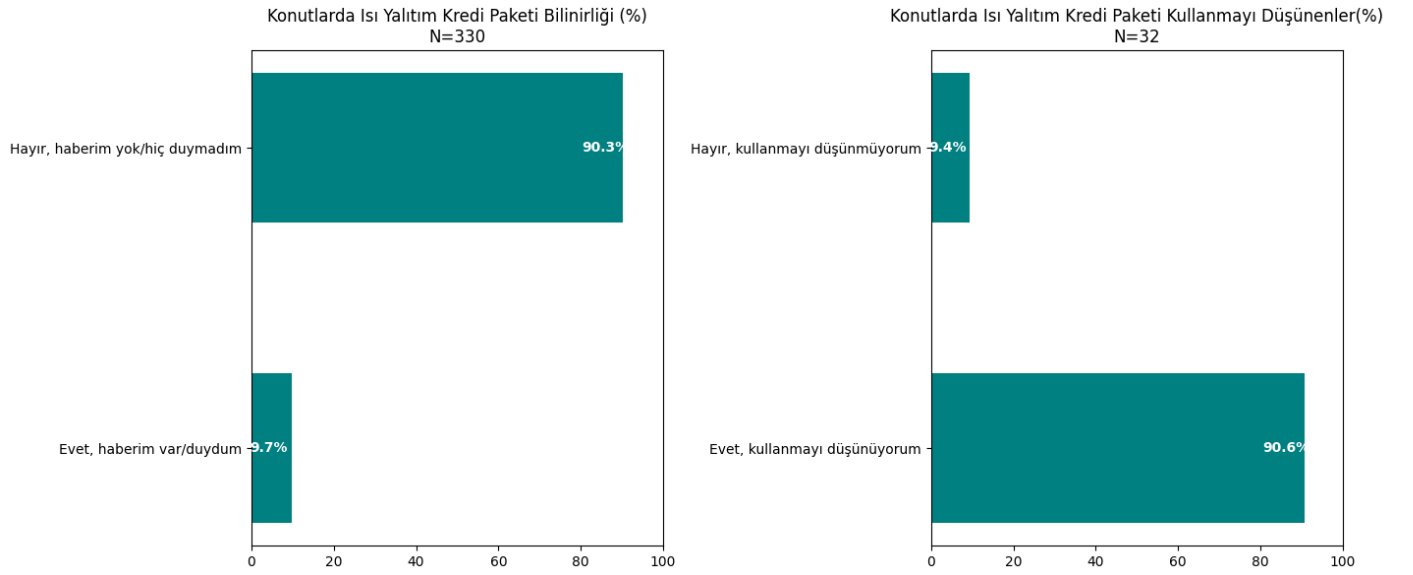
fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(14, 6))

ax1.barh(labels_MS14, percentages_MS14, color='teal', height=0.5)
for i, v in enumerate(percentages_MS14):
    ax1.text(v - 10, i, f'{v:.1f}%', color='white',
            va='center', fontweight='bold')

ax1.set_title(
    f'Konutlarda Isı Yalıtım Kredi Paketi Bilinirliği (%) \nN={participants_MS14}')
ax1.set_xlim(0, 100)

ax2.barh(labels_MS15, percentages_MS15, color='teal', height=0.5)
for i, v in enumerate(percentages_MS15):
    ax2.text(v - 10, i, f'{v:.1f}%', color='white',
            va='center', fontweight='bold')

ax2.set_title(
    f'Konutlarda Isı Yalıtım Kredi Paketi Kullanmayı  
Düşünenler (%) \nN={participants_MS15}')
ax2.set_xlim(0, 100)
plt.tight_layout()
plt.show()
```



Grafik-32: Konutlarda Isı Yalıtım Kredi Paket Bilinirliği ve Katılımcılardan Kullanmayı Düşünenler Bar Grafiği

Katılımcıların %9.7'si, «Konutlarda ısı yalıtım paketi»nden haberdar olduğunu belirtmiştir. Haberdar olan katılımcıların %90.6'sı bu paketi kullanmayı düşünmektedir.

```
ms16_data = data['MS16'].value_counts(normalize=True) * 100

responses = ['Evet, mantolama yaptırmayı düşünüyoruz',
             'Evet, mantolamayı yeniletmeyi düşünüyoruz',
             'Hayır']

percentages = ms16_data.reindex(
    [1, 2, 3], fill_value=0).values

fig, ax = plt.subplots(figsize=(8, 6))
bars = ax.barh(responses, percentages, color='teal')
for bar in bars:
    width = bar.get_width()
    label_x_pos = width + 1 if width < 10 else width - 5
    ax.text(label_x_pos, bar.get_y() + bar.get_height()/2, f'{width:.1f}%',
            va='center', ha='center', color='white' if width >= 10 else 'black',
            fontsize=12)
```

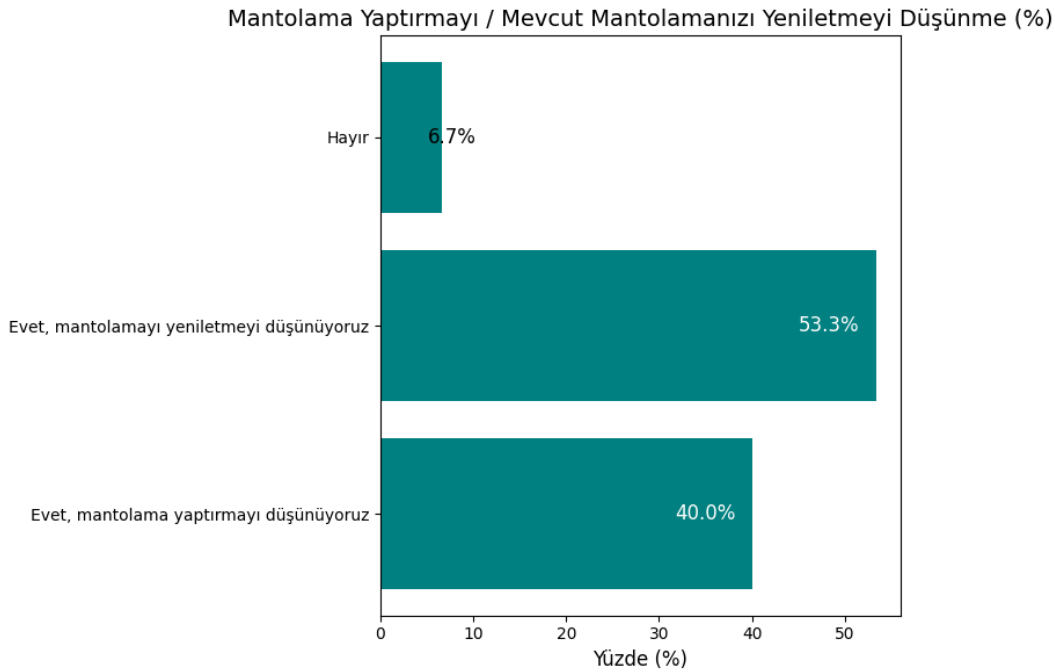


```

ax.set_title(
    'Mantolama Yaptırmayı / Mevcut Mantolamanızı Yeniletmeyi Düşünme (%)',
    fontsize=14)
ax.set_xlabel('Yüzde (%)', fontsize=12)

plt.tight_layout()
plt.show()

```



Grafik-33: Mantolama Yaptırmayı / Yeniletmeyi Düşünenler Bar Grafiği

Katılımcıların %93.3'ü, yakın gelecekte mantolama yaptırmayı/mantolama yeniletmeyi düşünmektedir. Bu katılımcıların %10'u A Markası'nın kendileri ile iletişime geçmesine izin vermiştir.

```

comments = data['MS01'].dropna()
text = ' '.join(comments)
wordcloud = WordCloud(width=800, height=400,
    background_color='white').generate(text)

```

```
plt.figure(figsize=(10, 5))
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis('off')
plt.show()
```



Grafik-34: A Markası için Kelime Bulutu

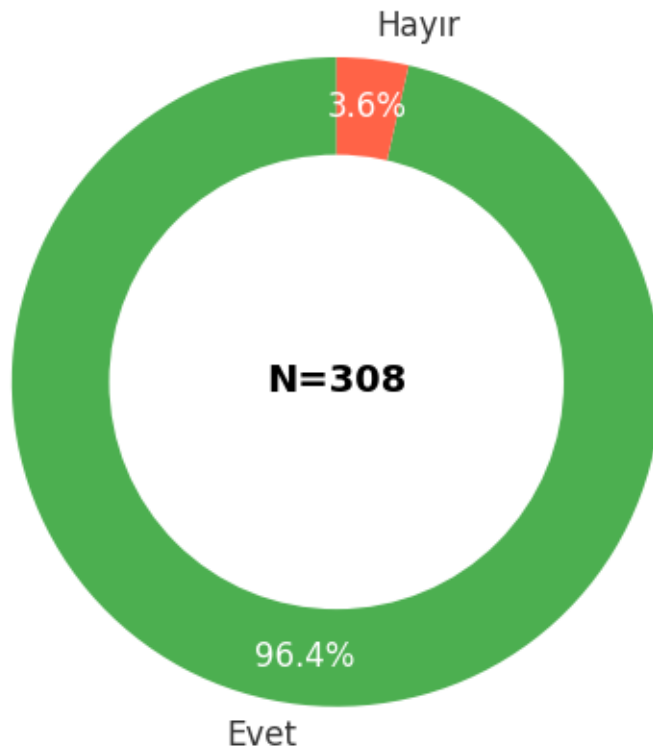
Katılımcıların A Markası için yorumlarında Kelime Bulutu çalışması yapılmıştır. Fazla tekrarlanan kelimeler daha büyük, az tekrarlananlar ve tekrarlanmayanlar ise küçük gösterilmiştir.

```
msl7_data = data['MSI7']
permission_counts = msl7_data.value_counts()
total_participants = permission_counts.sum()
permission_percentages = (permission_counts / total_participants) * 100
labels = ['Evet', 'Hayır']
sizes = permission_percentages
colors = ['#4CAF50', '#FF6347']

fig, ax = plt.subplots(figsize=(7, 5))
wedges, texts, autotexts = ax.pie(sizes, labels=labels, colors=colors,
    autopct='%1.1f%%', startangle=90, wedgeprops=dict(width=0.4), pctdistance=0.85)
```

```
for text in texts:
    text.set_fontsize(12)
    text.set_color('#333')
for autotext in autotexts:
    autotext.set_color('white')
    autotext.set_fontsize(12)
centre_circle = plt.Circle((0, 0), 0.70, fc='white')
fig.gca().add_artist(centre_circle)
ax.text(0, 0, f'N={total_participants}', ha='center',
        va='center', fontsize=14, fontweight='bold')
plt.title("A Markası'na İletişim İzni Verme Durumu (%)", fontsize=16, pad=20)
plt.tight_layout()
plt.show()
```

A Markası'na İletişim İzni Verme Durumu (%)



Grafik-35: A Markası'na İletişim İzni Verme Durum Grafiği

6. SONUÇ

Eğlenceli bir çalışmanın ardından bu sonuçlar çıkmıştır. Umarım raporu beğenirsiniz.

7. KAYNAKLAR

- https://www.academia.edu/62899108/Makine_%C3%96%C4%9Frenmesine_Yolculuk

Berkant Aslan, Makine Öğrenmesine Yolculuk e-Kitabı, 2021