

# Banknot Kimlik Doğrulama Analizi

```
column_names = ["variance", "skewness", "curtosis", "entropy", "class"]
dosya_yolu = r"C:\Users\AYSU\OneDrive - İstanbul Medeniyet Üniversitesi\Masaüstü\veri.csv"

df = pd.read_csv(dosya_yolu, header=None, names=column_names)

X = df.drop(labels="class", axis=1)
y = df["class"]
```

--- VERİ SETİ ÖZETİ ---  
 Toplam Örnek Sayısı: 1372  
 Özellik Sayısı: 4

Sınıf Dağılımı:  
 class  
 0 0.555394  
 1 0.444606

Name: proportion, dtype: float64

3.6216,8.6661,-2.8073,-0.44699,0
4.5459,8.1674,-2.4586,-1.4621,0
3.866,-2.6383,1.9242,0.10645,0
3.4566,9.5228,-4.0112,-3.5944,0
0.32924,-4.4552,4.5718,-0.9888,0
4.3684,9.6718,-3.9606,-3.1625,0
3.5912,3.0129,0.72888,0.56421,0
2.0922,-6.81,8.4636,-0.60216,0
3.2032,5.7588,-0.75345,-0.61251,0
1.5356,9.1772,-2.2718,-0.73535,0
1.2247,8.7779,-2.2135,-0.80647,0
3.9899,-2.7066,2.3946,0.86291,0
1.8993,7.6625,0.15394,-3.1108,0
1.5768,10.842,0.5462,0.0262,0

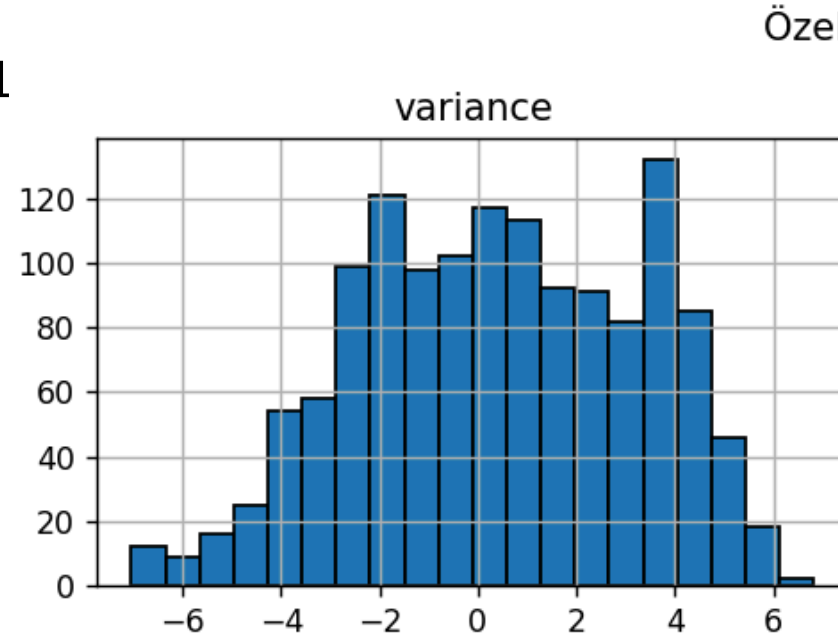
```
print("\n--- VERİ SETİ ÖZETİ ---")
print(f"Toplam Örnek Sayısı: {len(df)}")
print(f"Özellik Sayısı: {X.shape[1]}")
print("\nSınıf Dağılımı:")
print(y.value_counts(normalize=True))

# Histogramlar
X.hist(figsize=(10, 6), bins=20, edgecolor="black")
plt.suptitle("Özellik Dağılımları")
plt.show()
```

# Öznitelikler

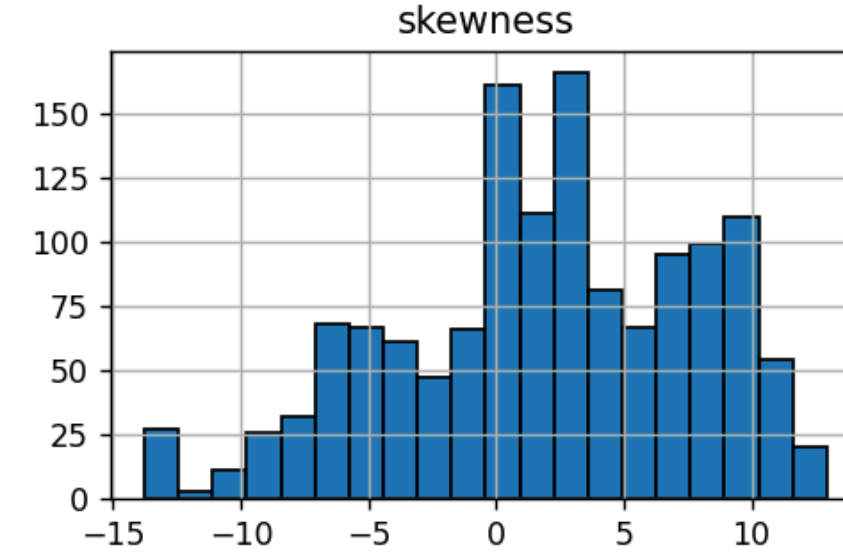
## 1. Variance (Varyans)

Bu grafik, banknotlar üzerindeki doku ve kontrastın (zıtlığın) ne kadar değişken olduğunu gösterir



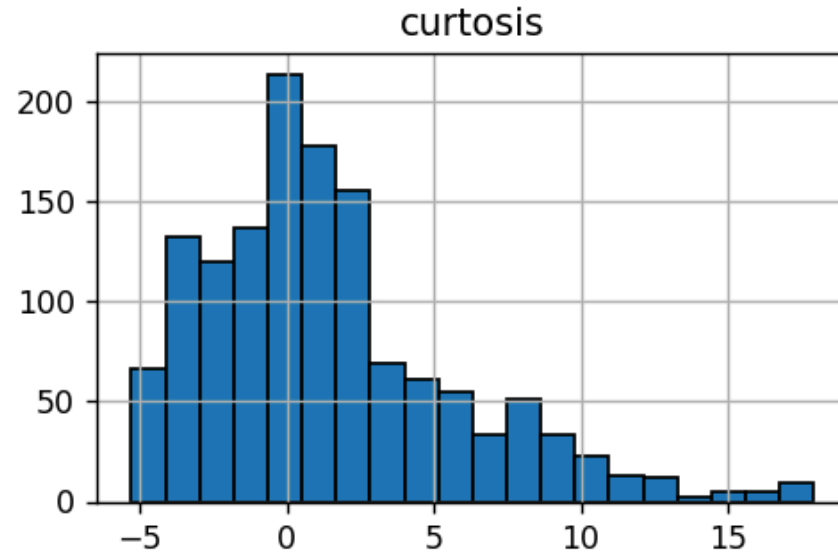
## 2. Skewness (Çarpıklık)

Bu grafik, banknot üzerindeki ışık dağılımının dengesini (açık mı yoksa koyu tonlar mı baskın?) anlatır.



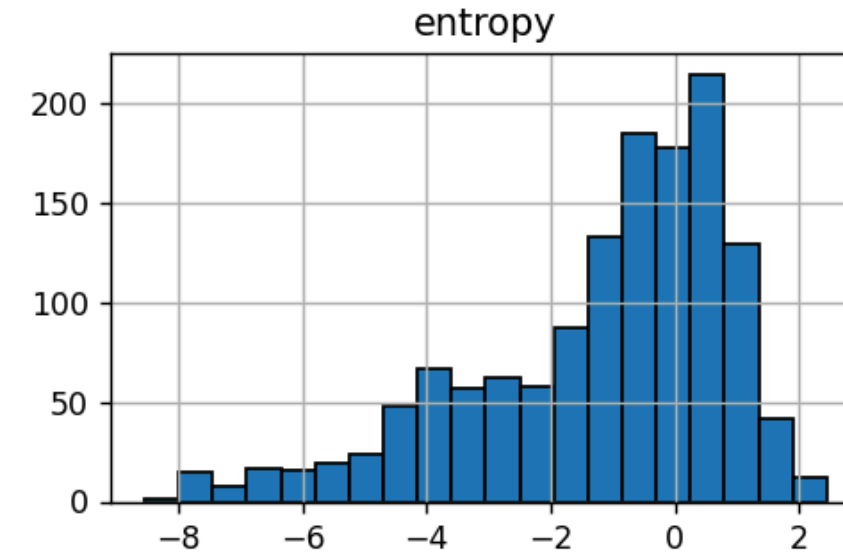
## 3. Curtosis (Basıklık / Sivrilik)

Bu grafik, piksel değerlerindeki ani değişimleri ve uç değerleri (baskıdaki sert geçişleri) gösterir



## 4. Entropy (Entropi / Karmaşıklık)

Bu grafik, banknotun taşıdığı bilgi miktarını ve desen karmaşıklığını ölçer

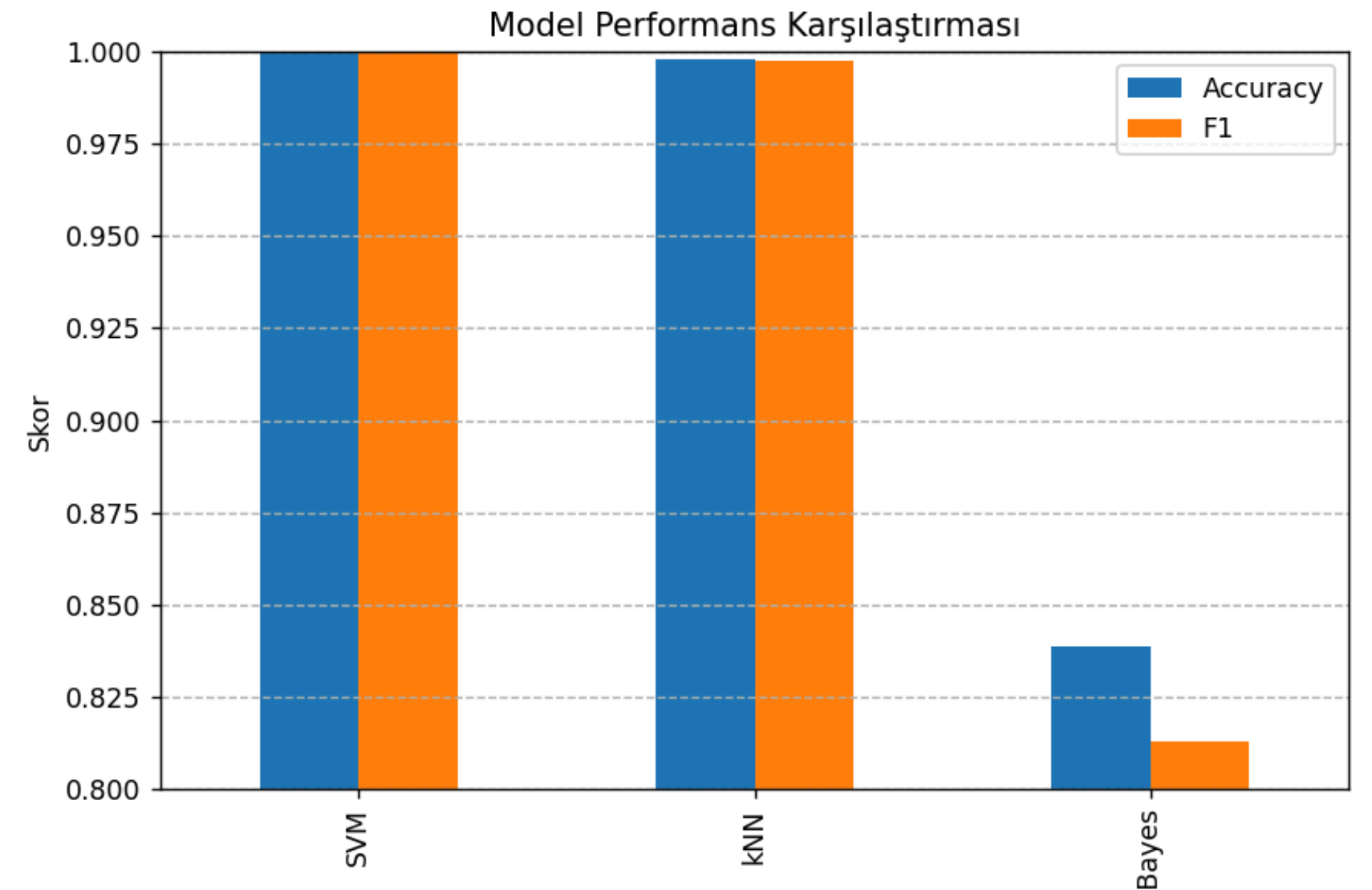


```
models = {
    "SVM": Pipeline([
        ("scaler", StandardScaler()),
        ("model", SVC(kernel="rbf", random_state=42))
    ]),
    "kNN": Pipeline([
        ("scaler", StandardScaler()),
        ("model", KNeighborsClassifier(n_neighbors=5))
    ]),
    "Naive Bayes": Pipeline([
        ("scaler", StandardScaler()),
        ("model", GaussianNB())
    ])
}
```

--- MODEL KARŞILAŞTIRMASI ---

	Accuracy	Precision	Recall	F1
Model				
SVM	1.000000	1.000000	1.000000	1.000000
kNN	0.997812	0.995114	1.000000	0.997547
Naive Bayes	0.838931	0.839373	0.788588	0.812854

Process finished with exit code 0



```

kf = StratifiedKFold(n_splits=3, shuffle=True, random_state=42)
scoring = ["accuracy", "precision", "recall", "f1"]

results = []

print("\n--- MODEL KARŞILAŞTIRMASI ---")
for name, model in models.items():
    cv = cross_validate(model, X, y, cv=kf, scoring=scoring)
    results.append({
        "Model": name,
        "Accuracy": np.mean(cv["test_accuracy"]),
        "Precision": np.mean(cv["test_precision"]),
        "Recall": np.mean(cv["test_recall"]),
        "F1": np.mean(cv["test_f1"])
    })

results_df = pd.DataFrame(results).set_index("Model")
print(results_df)

```

veri setini 2'ye 1 olarak böldüm

```

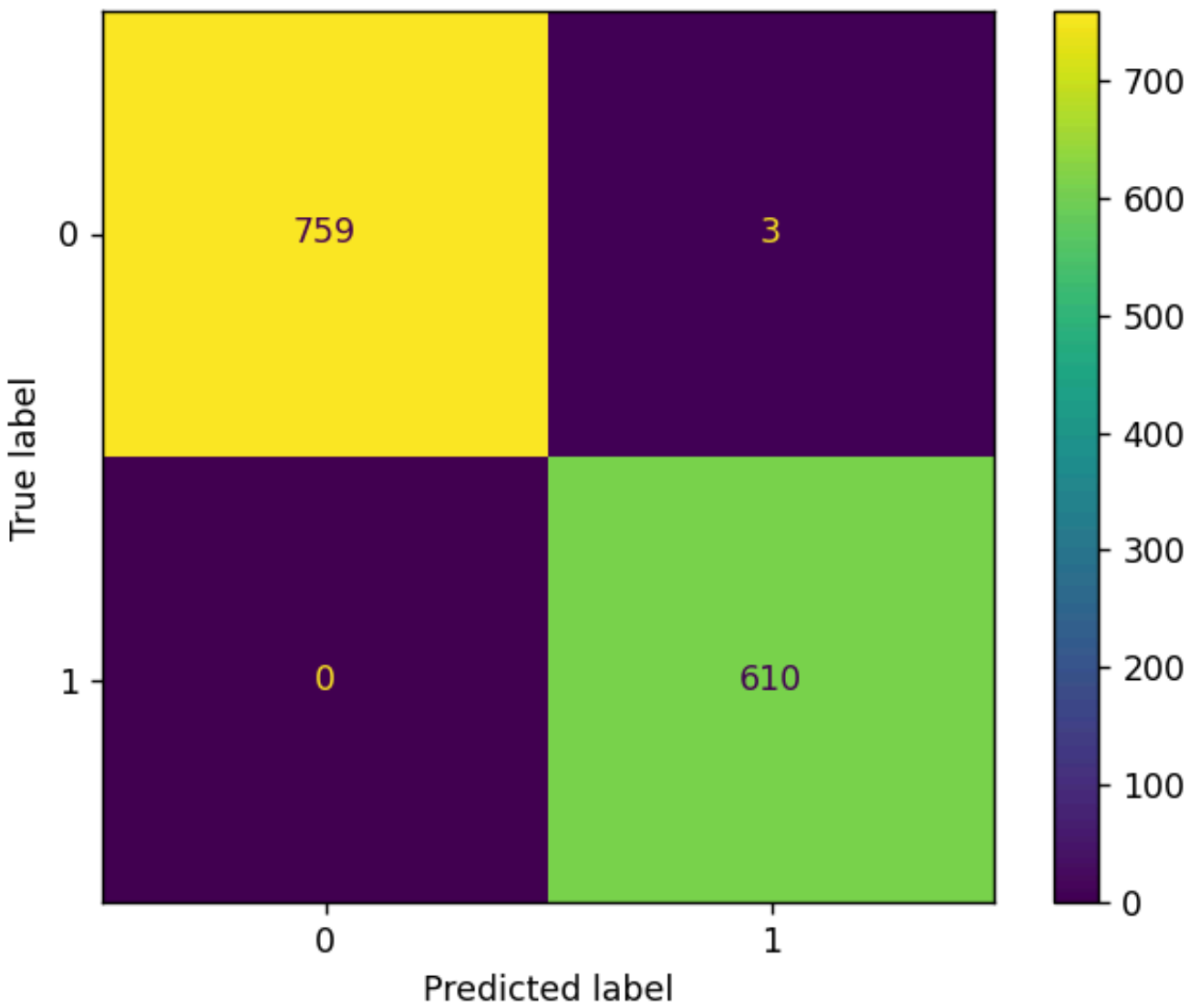
--- MODEL KARŞILAŞTIRMASI ---

```

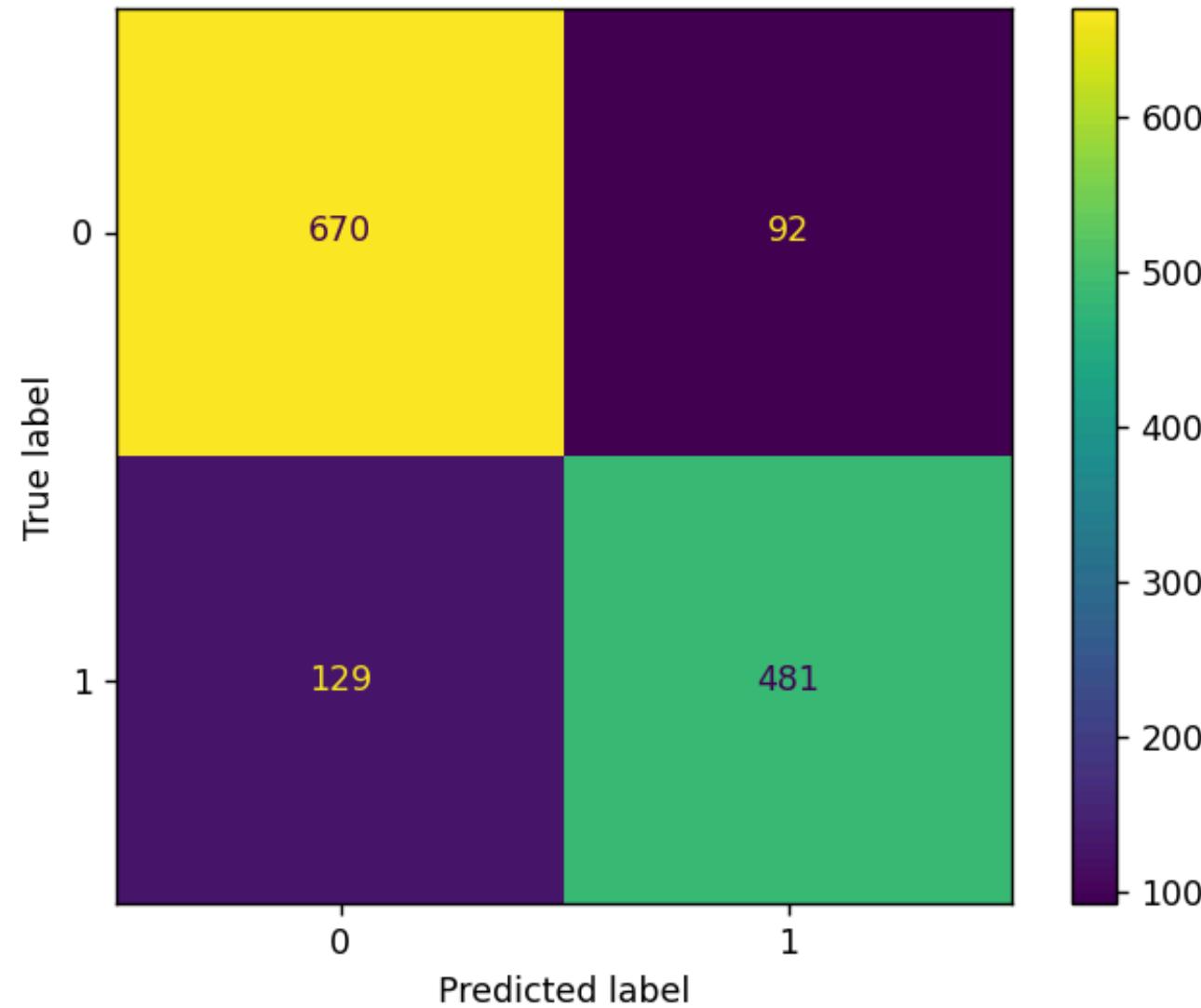
	Accuracy	Precision	Recall	F1
Model				
SVM	1.000000	1.000000	1.000000	1.000000
kNN	0.997812	0.995114	1.000000	0.997547
Naive Bayes	0.838931	0.839373	0.788588	0.812854

# Model Hata Matrisleri

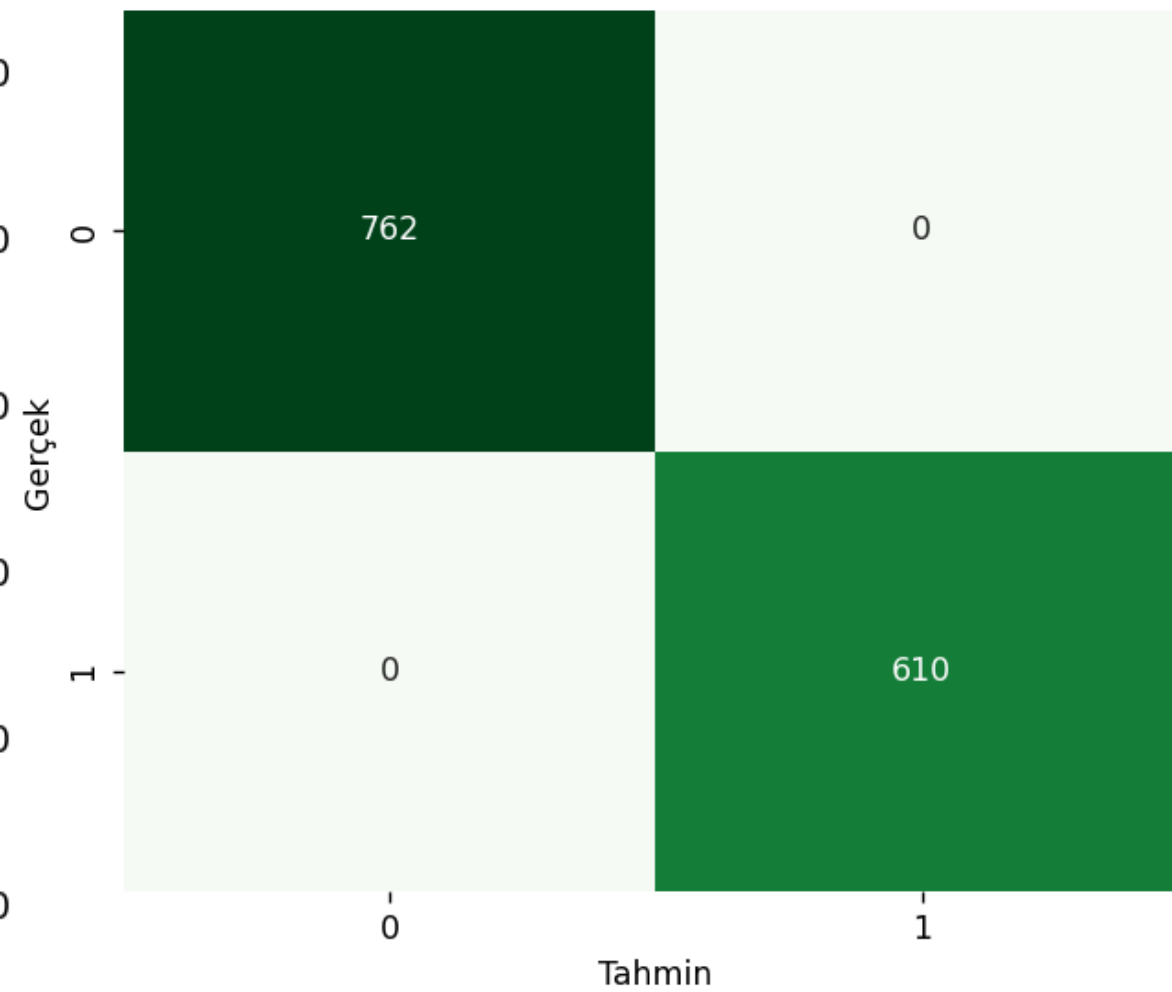
kNN - Confusion Matrix



Naive Bayes - Confusion Matrix

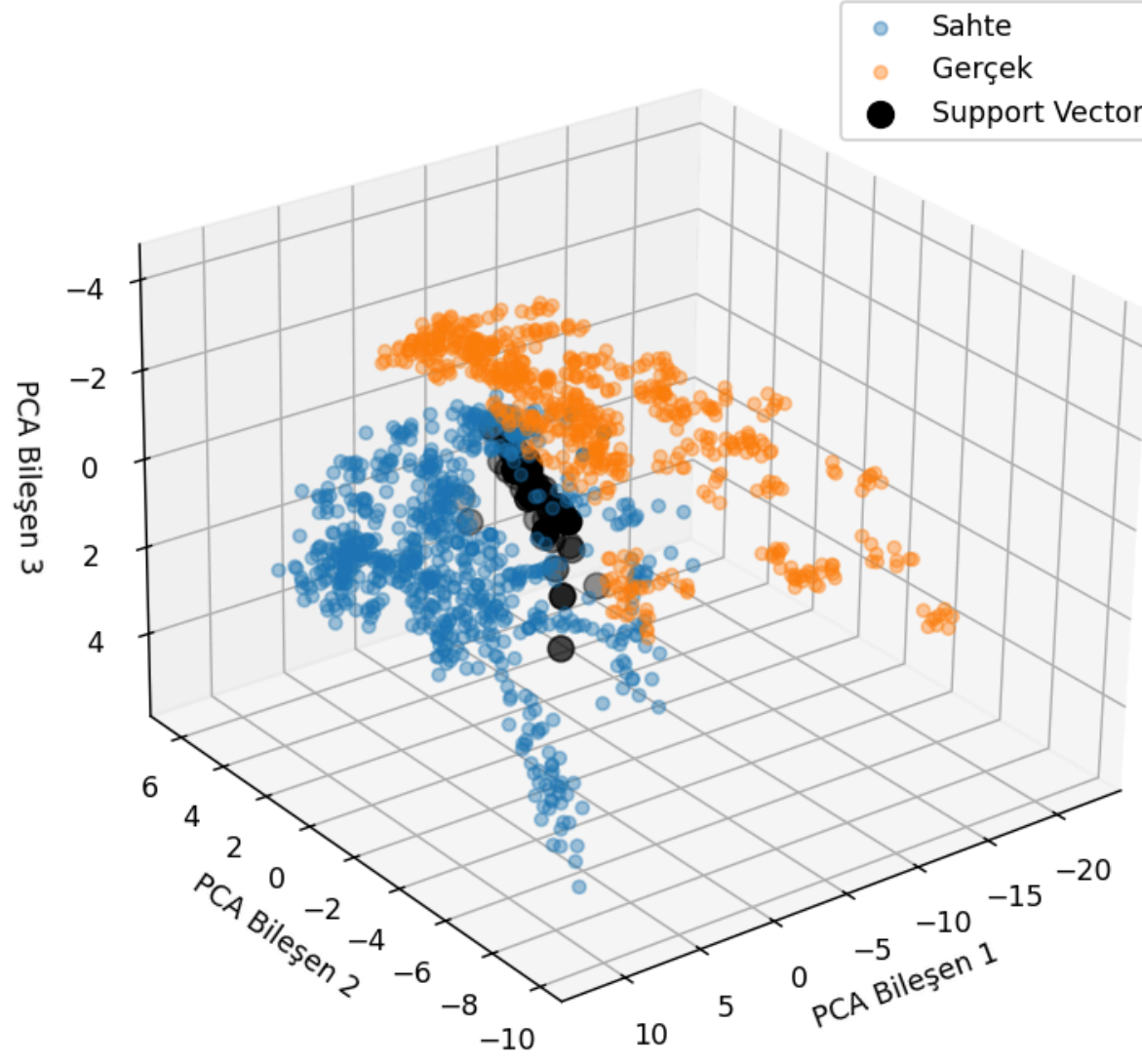


Hata Matrisi - SVM

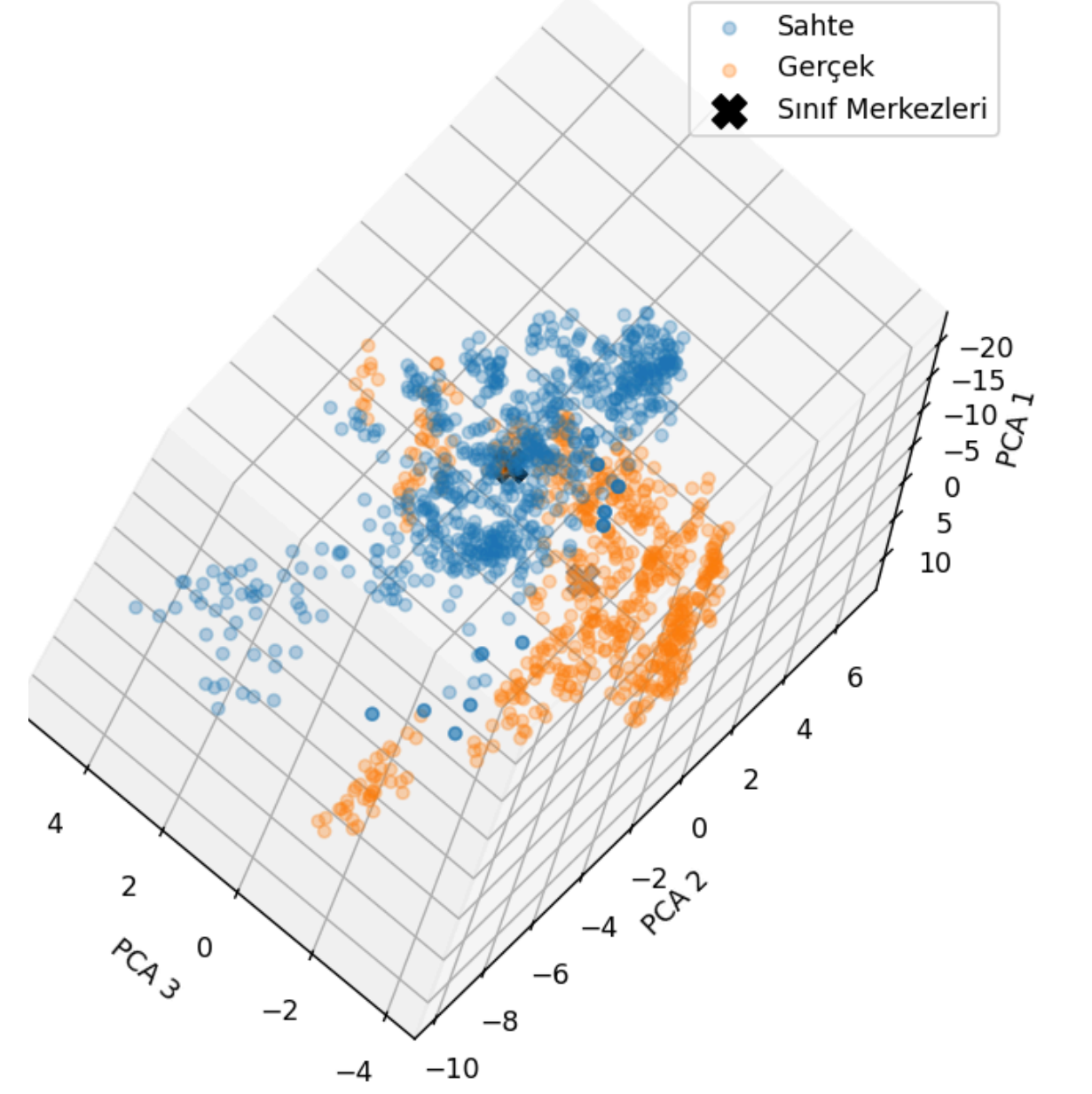




SVM - Support Vector'lar ile 3B Ayrım Mantiğı



Naive Bayes - Olasılık Merkezleri (PCA 3B)



kNN - 3B Komşuluk Mantiğı (PCA)

