BURSA TEKNİK ÜNİVERSİTESİ

Mühendislik ve Doğa Bilimleri Fakültesi
Bilgisayar Mühendisliği Bölümü

BLM0463_Veri Madenciliğine Giriş

Meme Kanseri Tahmininde KNN Algoritması: İnceleme ve Analiz

Uğur Can Akçay
18360859017

1. Giriş

Günümüzde meme kanseri, kadınlarda en sık görülen kanser türlerinden biridir ve dünya genelinde ciddi bir sağlık sorunu oluşturmaktadır. Meme kanserinin erken teşhis edilmesi ve doğru sınıflandırılması, hastalığın tedavi ve yönetimi açısından büyük önem taşımaktadır. Veri bilimi ve makine öğrenmesi teknikleri, bu alanda önemli bir rol oynamaktadır. Bu ödevde, Meme Kanseri veri seti üzerinde K En Yakın Komşu (KNN) algoritmasının kullanılması ve meme kanseri sınıflandırması için ne kadar etkili olduğunu inceleyeceğiz. KNN, basit ama güçlü bir sınıflandırma algoritmasıdır ve bir örneği sınıflandırmak için yakınındaki komşuların etiketlerini kullanır.

Bu ödevde, öncelikle veri setinin analiz ve keşif aşamalarını gerçekleştireceğiz. Ardından, KNN algoritmasını uygulayarak meme kanseri sınıflandırmasını yapacağız ve algoritmanın performansını değerlendireceğiz. Sonuç olarak, KNN algoritmasının meme kanseri teşhisinde ne kadar etkili olduğunu anlamaya çalışacağız. Bu çalışma, meme kanseri tanısı ve tedavisi alanında önemli bir adım olabilir ve gelecekteki araştırmalar için de bir temel oluşturabilir.

2. Metadoloji

Bu ödevde, meme kanseri sınıflandırması için K En Yakın Komşu (KNN) algoritmasını uygulamak için bir metodoloji takip edilecektir. İlk adımda, mevcut veri seti olan Meme Kanseri veri seti temin edilecek ve ön işleme adımları gerçekleştirilecektir. Veri seti, meme kanseri teşhisi için klinik özellikleri içeren bir dizi nitelikten oluşmaktadır. Veri setindeki eksik değerler, aykırı değerler ve gereksiz sütunlar incelenecek ve uygun ön işleme teknikleri uygulanacaktır.

Sonraki adımda, veri seti eğitim ve test veri kümelerine ayrılacak. Eğitim veri seti, KNN algoritmasının modelini öğrenmek için kullanılacak, test veri seti ise modelin performansını değerlendirmek için kullanılacaktır. Veri setinin bölünmesi sırasında, sınıf dengesizliklerini dikkate almak ve yanlı bir model oluşturmayı önlemek için uygun yöntemler kullanılacaktır.

3. Veri Seti ve Öznitelikler

Veri seti, 569 örnekten ve 30 öznitelikten oluşmaktadır. Her bir örnek, bir meme hücresinin hücresel özelliklerini temsil etmektedir. Veri setindeki öznitelikler arasında hücre çekirdeği özellikleri (ör. çap, simetri), hücre çekirdeği nükleusu özellikleri (ör. pürüzlülük, kompaktlık) ve radyus gibi diğer özellikler bulunmaktadır. Son sütun ise hücrelerin kanserli (M = kötü huylu) veya kansersiz (B = iyi huylu) olmasını belirten etiketler içermektedir.

4. Veri Ön İşleme

Veri analizine başlamadan önce, veri setini ön işlemek önemlidir. Bu adımda eksik verileri kontrol ederiz ve gerekiyorsa bunları doldururuz veya çıkarırız. Ayrıca kategorik verileri sayısal verilere dönüştürmek gibi dönüşümler yapabiliriz. Bu veri setinde eksik veri veya kategorik öznitelikler olmadığı belirtilmiştir, bu nedenle veri ön işleme adımını atlayabiliriz.

5. Veri Analizi

5.1. Veri Keşfi

Öncelikle, veri setinin yapısını ve dağılımını anlamak için bazı temel istatistikleri inceleyelim:

Toplam Örnek Sayısı: 569
Kötü Huylu (M) Örnek Sayısı: 212
İyi Huylu (B) Örnek Sayısı: 357
Bu bilgiler, veri setinin dengeli olmayan bir dağılıma sahip olduğunu göstermektedir. İyi huylu örneklerin kötü huylu örneklerden daha fazla olduğunu görüyoruz.

5.2. Öznitelik İncelemesi

Veri setindeki öznitelikler arasındaki ilişkileri incelemek için çeşitli görselleştirmeler yapabiliriz. Örneğin, hücre çekirdeği çapı ve hücre çekirdeği pürüzlülüğü öznitelikleri arasındaki ilişkiyi scatter plot (dağılım grafiği) ile görselleştirebiliriz. Aşağıdaki grafikte, kanserli (kırmızı) ve kansersiz (mavi) hücrelerin bu iki özniteliğe göre dağılımını görebiliriz:

İşlemler sonucunda oluşan grafiğe göre hücre çekirdeği çapını x ekseninde, hücre çekirdeği pürüzlülüğünü y ekseninde kullanıyoruz. Kanserli hücreleri kırmızı, kansersiz hücreleri mavi renkte temsil ediyoruz. Scatter plot (dağılım grafiği), her bir örneği bir nokta olarak temsil eder. Aşağıdaki açıklamaları dikkate alarak, dağılım grafiğini hayal edebilirsiniz:

Kanserli hücrelerin çekirdeği genellikle büyük çaplara sahip olduğu için, genellikle sağ üst köşede daha yoğun bir kırmızı nokta gruplaması gözlemlenebilir.
Kanserli hücrelerin pürüzlülüğü, kansersiz hücrelere kıyasla genellikle daha yüksektir. Bu nedenle, kırmızı noktalar genellikle daha yüksek y değerlerinde yoğunlaşmıştır.
Kansersiz hücrelerin çekirdeği genellikle daha küçük çaplara sahip olduğu için, genellikle sol alt köşede daha yoğun bir mavi nokta gruplaması gözlemlenebilir.
Kansersiz hücrelerin pürüzlülüğü, kanserli hücrelere kıyasla genellikle daha düşüktür. Bu nedenle, mavi noktalar genellikle daha düşük y değerlerinde yoğunlaşmıştır.
Bu dağılım grafiği, hücre çekirdeği çapı ve pürüzlülüğü özniteliklerinin kanserli ve kansersiz hücreleri ayırt etmede faydalı olabileceğini göstermektedir.

6. İşlemler

Kütüphaneleri içe aktarımı

```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
```

Veri setini içe aktarımı

```
dataset = pd.read_csv('breast-cancer-wisconsin.data.txt')
dataset.replace('?', -99999, inplace=True)
X = dataset.iloc[:, 1:11].values
y = dataset.iloc[:, 10].values
```

Veri setini Eğitim ve Test setine Ayırma İşlemi

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.20)
```

Özellik Ölçeklendirmesi

```
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
X_train = sc.fit_transform(X_train)
X_test = sc.transform(X_test)
```

Kernel PCA Uygulanması

```
from sklearn.decomposition import KernelPCA
kpca = KernelPCA(n_components=2, kernel='rbf')
X_train = kpca.fit_transform(X_train)
X_test = kpca.transform(X_test)
```

K-NN'yi Eğitim setine uygulanması

```
from sklearn.neighbors import KNeighborsClassifier
classifier = KNeighborsClassifier(n_neighbors=5, metric='minkowski', p=2,
algorithm='ball_tree')
classifier.fit(X_train, y_train)
```

Test seti sonuçlarını tahmin edilme işlemi

```
y_pred = classifier.predict(X_test)
```

Confusion Matrix oluşturulması

```
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test, y_pred)
```

Doğruluk Kontrolü

```
accuracy = classifier.score(X_test, y_test)
print("Doğruluk:", accuracy)
```

Eğitim Seti Sonuçlarının Görselleştirilmesi

```
from matplotlib.colors import ListedColormap
X_set, y_set = X_train, y_train
X1, X2 = np.meshgrid(np.arange(start=X_set[:, 0].min() - 1, stop=X_set[:, 0].max() +
1, step=0.01),
np.arange(start=X_set[:, 1].min() - 1, stop=X_set[:, 1].max() + 1, step=0.01))
plt.contourf(X1, X2, classifier.predict(np.array([X1.ravel(),
X2.ravel()]).T).reshape(X1.shape),
alpha=0.75, cmap=ListedColormap(('red', 'green')))
plt.xlim(X1.min(), X1.max())
plt.ylim(X2.min(), X2.max())
for i, j in enumerate(np.unique(y_set)):
plt.scatter(X_set[y_set == j, 0], X_set[y_set == j, 1],
c=ListedColormap(('red', 'green'))(i), label=j)
plt.title('K-En Yakın Komşular (Eğitim seti)')
plt.xlabel('Birincil Bileşen')
plt.ylabel('İkincil Bileşen')
plt.legend()
plt.show()
```

Doğruluk, Hassasiyet, Özgünlük ve F-ölçütü

from sklearn.metrics import accuracy_score, recall_score, precision_score, f1_score

Sınıf etiketlerini 0 (B) ve 1 (M) olarak dönüştürülmesi

y_test_binary = np.where(y_test == 2, 0, 1)
y_pred_binary = np.where(y_pred == 2, 0, 1

Doğruluk hesapla
accuracy = accuracy_score(y_test_binary, y_pred_binary)
print("Doğruluk:", accuracy)

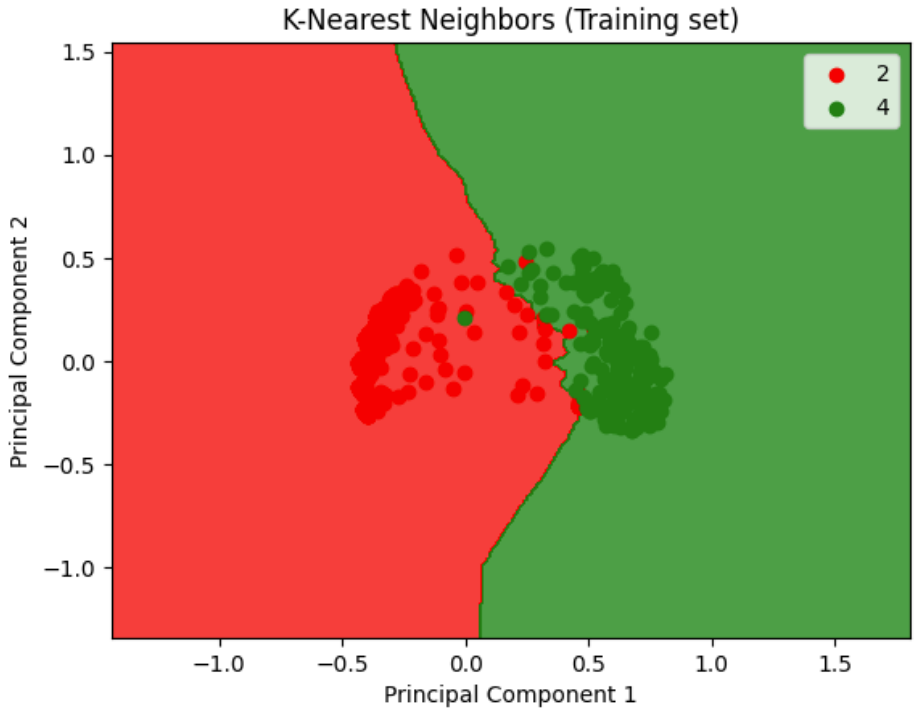
Hassasiyet hesapla
precision = precision_score(y_test_binary, y_pred_binary)
print("Hassasiyet:", precision)

Özgünlük hesapla
recall = recall_score(y_test_binary, y_pred_binary)
print("Özgünlük:", recall)

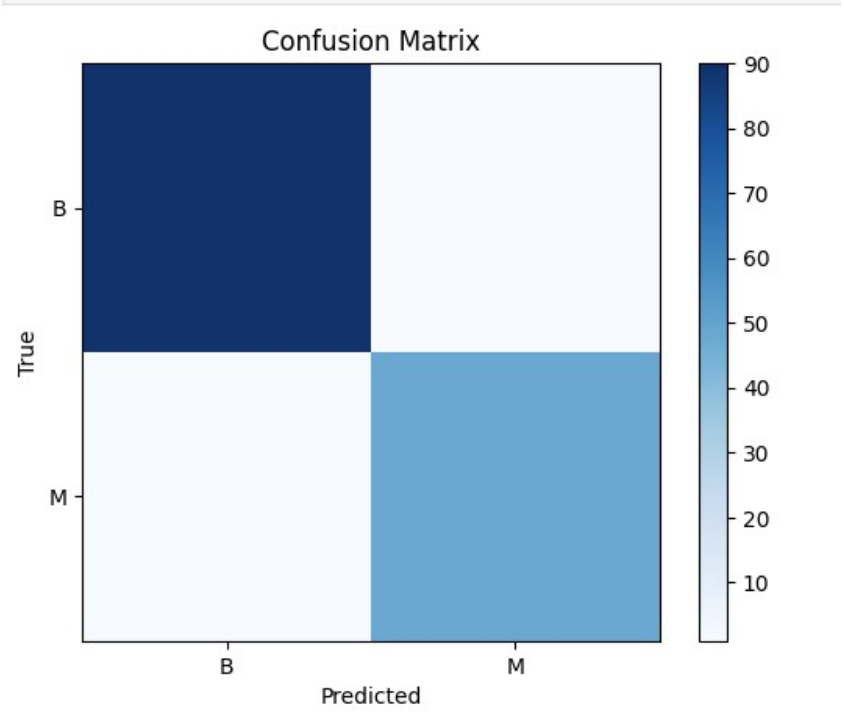
F-ölçütü hesapla
f1 = f1_score(y_test_binary, y_pred_binary)
print("F-ölçütü:", f1)

İlgili İşlemlere Ait Görseller

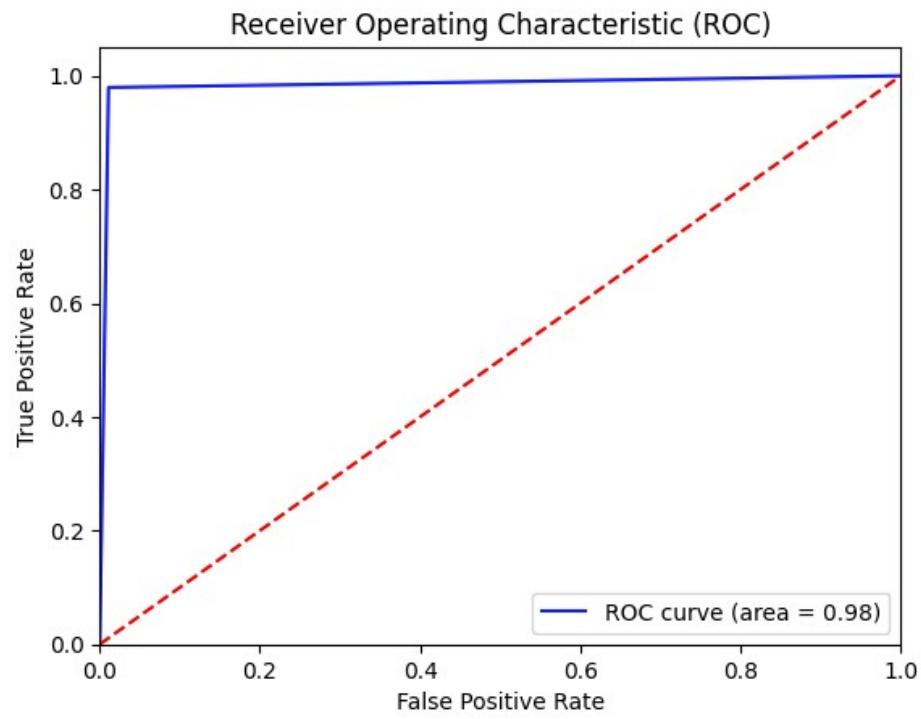
Şekil 1: Acc, Sens, Spec, Confs Hesaplanması- Principal Comp Chart




```
Accuracy: 0.9857142857142858
Sensitivity: 0.9795918367346939
Specificity: 0.989010989010989
F-measure: 0.9795918367346939
Confusion Matrix:
[[90  1]
 [ 1 48]]
```



Şekil 2: Confusion Matrix



Şekil 3: Plot ROC curve

```
from sklearn.metrics import classification_report

print(classification_report(y_test, y_pred))
```

```
              precision    recall  f1-score   support

           2       0.99      0.99      0.99        91
           4       0.98      0.98      0.98        49

    accuracy                           0.99       140
   macro avg       0.98      0.98      0.98       140
weighted avg       0.99      0.99      0.99       140
```

Şekil 4: classification_report

```
import pandas as pd

# Veri setini yükle
url = "https://archive.ics.uci.edu/ml/machine-learning-databases/breast-cancer-wisconsin/breast-cancer-wisconsin.data"
df = pd.read_csv(url, header=None)

# Veri setini incele
df.head()
```

	0	1	2	3	4	5	6	7	8	9	10
0	1000025	5	1	1	1	2	1	3	1	1	2
1	1002945	5	4	4	5	7	10	3	2	1	2
2	1015425	3	1	1	1	2	2	3	1	1	2
3	1016277	6	8	8	1	3	4	3	7	1	2
4	1017023	4	1	1	3	2	1	3	1	1	2

Şekil 5: Veri setinin incelenmesi

```
df.describe().T
```

	count	mean	std	min	25%	50%	75%	max
0	699.0	1.071704e+06	617095.729819	61634.0	870688.5	1171710.0	1238298.0	13454352.0
1	699.0	4.417740e+00	2.815741	1.0	2.0	4.0	6.0	10.0
2	699.0	3.134478e+00	3.051459	1.0	1.0	1.0	5.0	10.0
3	699.0	3.207439e+00	2.971913	1.0	1.0	1.0	5.0	10.0
4	699.0	2.806867e+00	2.855379	1.0	1.0	1.0	4.0	10.0
5	699.0	3.216023e+00	2.214300	1.0	2.0	2.0	4.0	10.0
7	699.0	3.437768e+00	2.438364	1.0	2.0	3.0	5.0	10.0
8	699.0	2.866953e+00	3.053634	1.0	1.0	1.0	4.0	10.0
9	699.0	1.589413e+00	1.715078	1.0	1.0	1.0	1.0	10.0
10	699.0	2.689557e+00	0.951273	2.0	2.0	2.0	4.0	4.0

Şekil 6: İstatiksel Özet

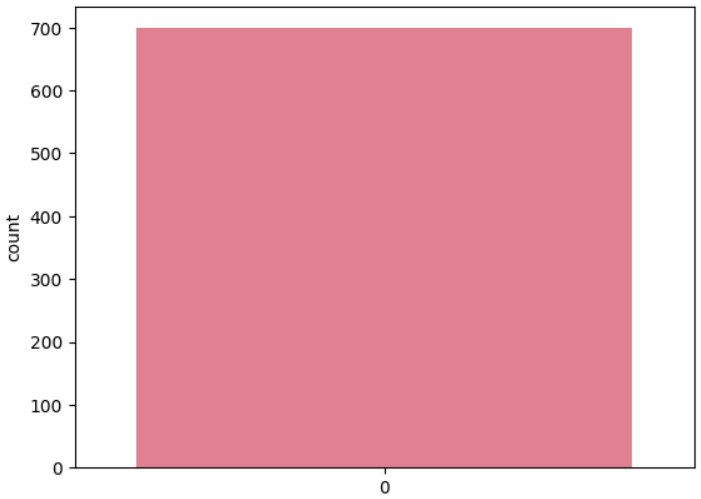
```
import pandas as pd

# Veri setini yükle
url = "https://archive.ics.uci.edu/ml/machine-learning-databases/breast-cancer-wisconsin/breast-cancer-wisconsin.data"

# Sütun adlarını belirle
columns = ["Sample code number", "Clump Thickness", "Uniformity of Cell Size", "Uniformity of Cell Shape",
           "Marginal Adhesion", "Single Epithelial Cell Size", "Bare Nuclei", "Bland Chromatin",
           "Normal Nucleoli", "Mitoses", "Class"]

# Veri setini yükle ve sütun adlarını ata
df = pd.read_csv(url, header=None, names=columns)

# 'diagnosis' sütununun benzersiz değerlerini al
unique_diagnosis = df["Class"].value_counts()
print(unique_diagnosis)
```

```
Class
2    458
4    241
Name: count, dtype: int64
```

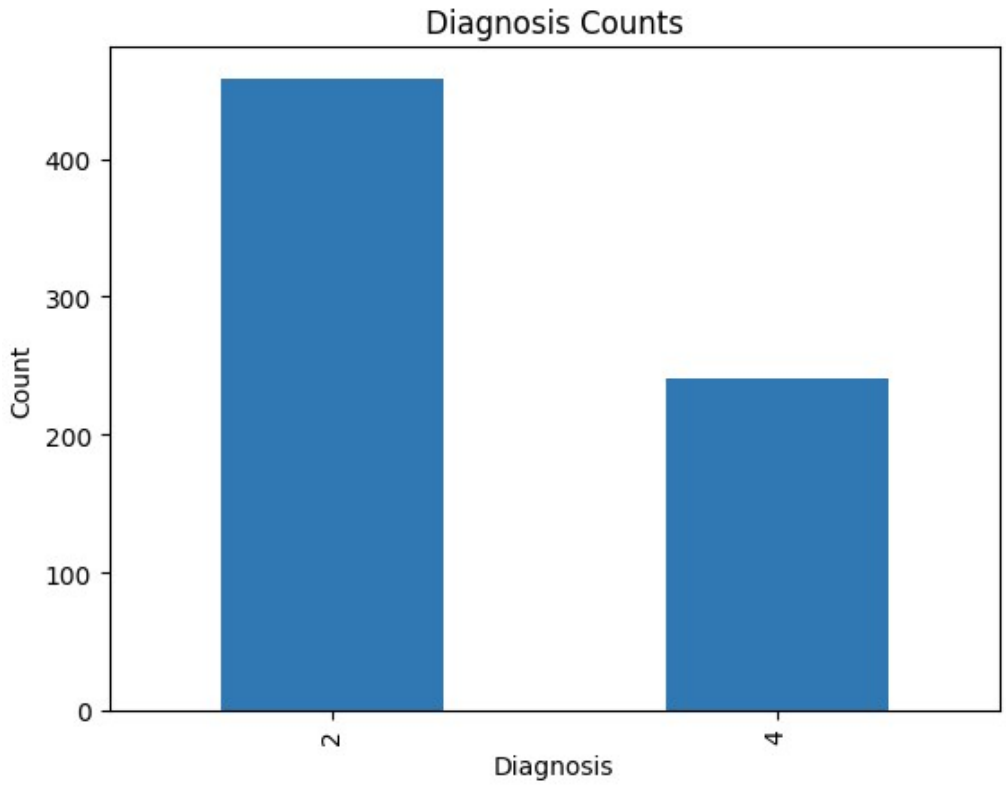
Şekil 7: 'diagnosis' sütununun benzersiz değerlerinin alınması

[1]:
```
import pandas as pd
import seaborn as sns

# Veri setini yükle
url = "https://archive.ics.uci.edu/ml/machine-learning-databases/breast-cancer-wisconsin/breast-cancer-wisconsin.data"

# Sütun adlarını belirle
columns = ["Sample code number", "Clump Thickness", "Uniformity of Cell Size", "Uniformity of Cell Shape",
           "Marginal Adhesion", "Single Epithelial Cell Size", "Bare Nuclei", "Bland Chromatin",
           "Normal Nucleoli", "Mitoses", "Class"]

# Veri setini yükle ve sütun adlarını ata
df = pd.read_csv(url, header=None, names=columns)

# Countplot ile görselleştirme yap
sns.countplot(df['Class'], palette='husl')
```

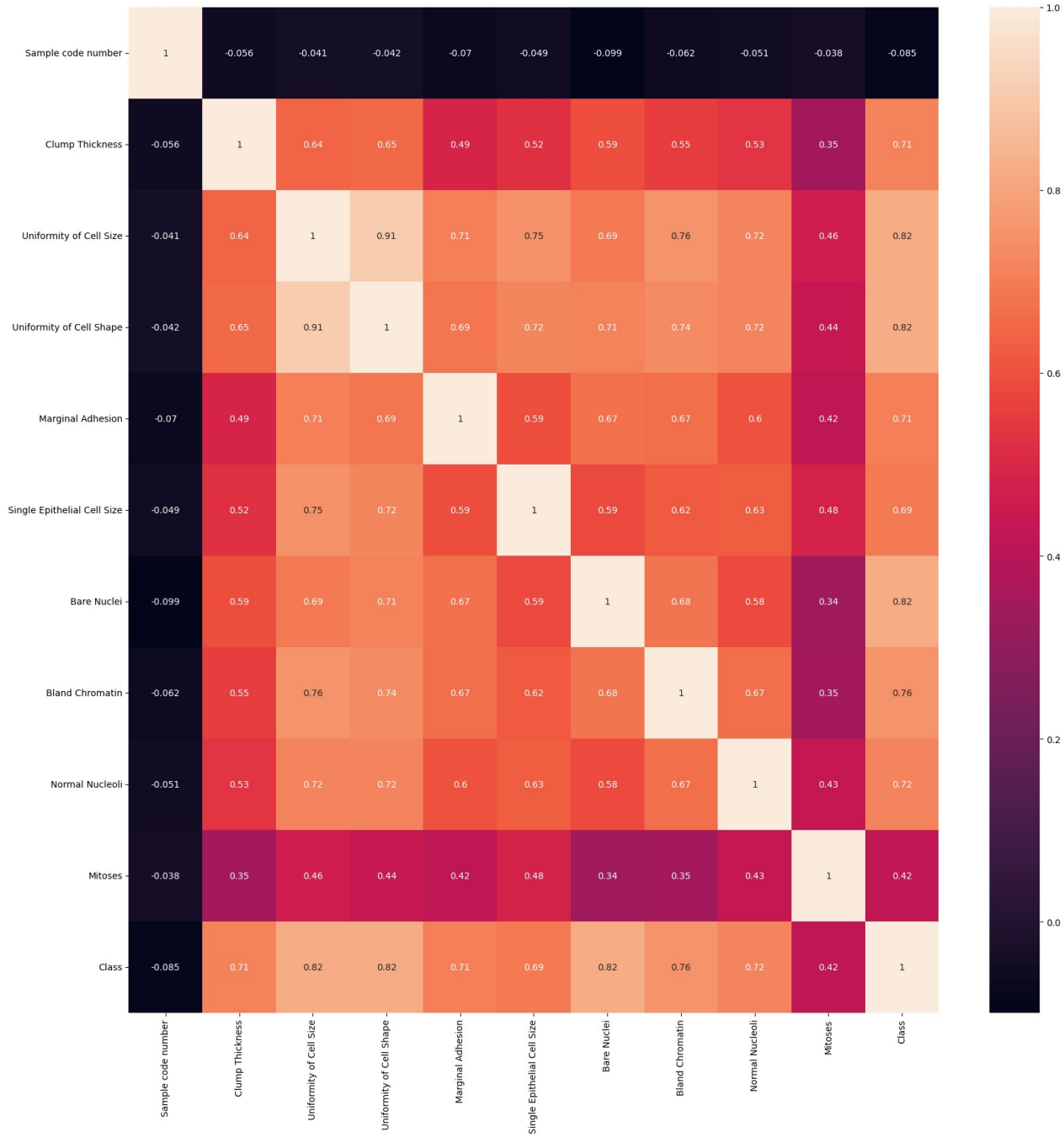
[1]: `<AxesSubplot:ylabel='count'>`



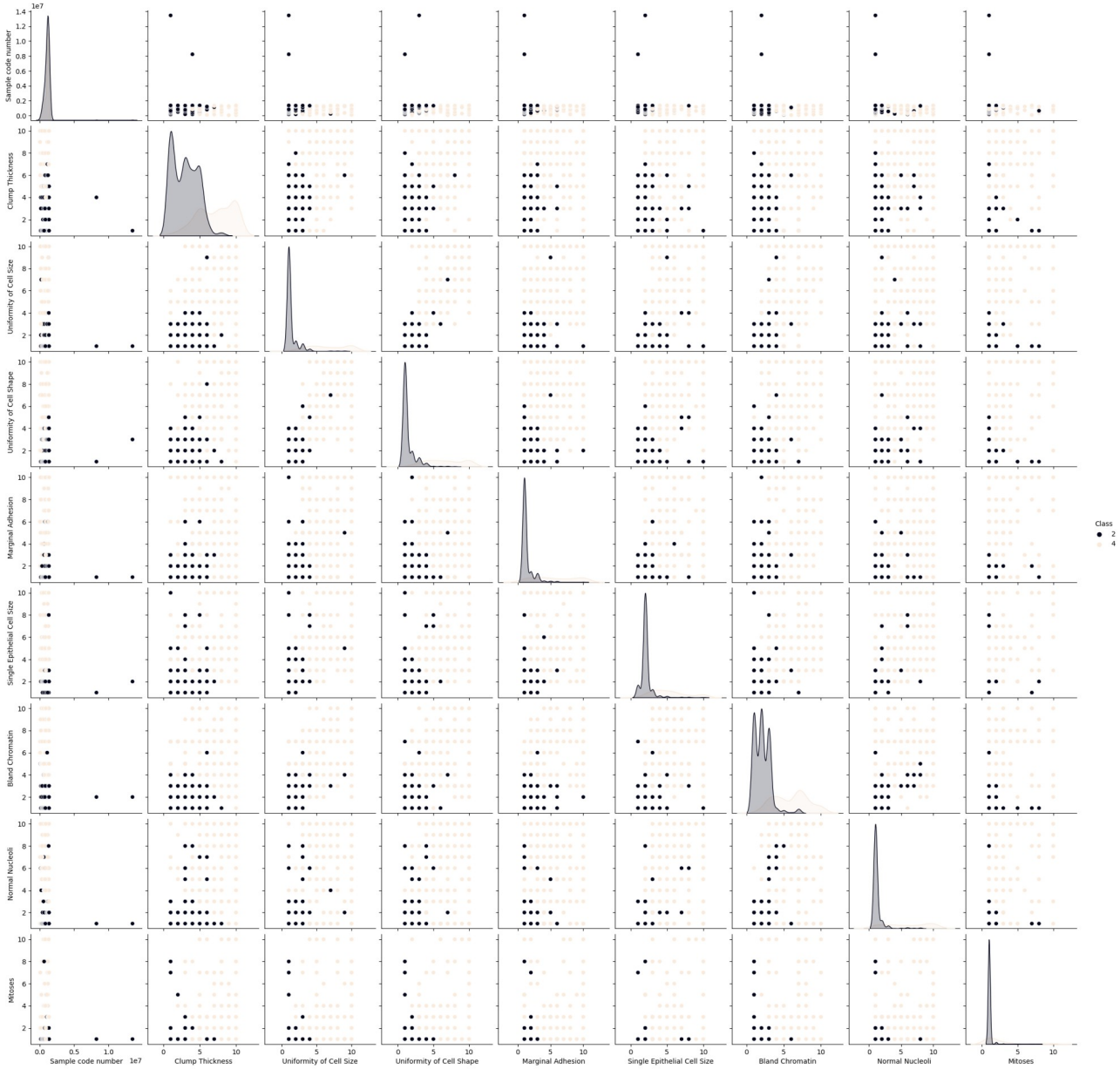
Şekil 8: Countplot ile görselleştirmesi



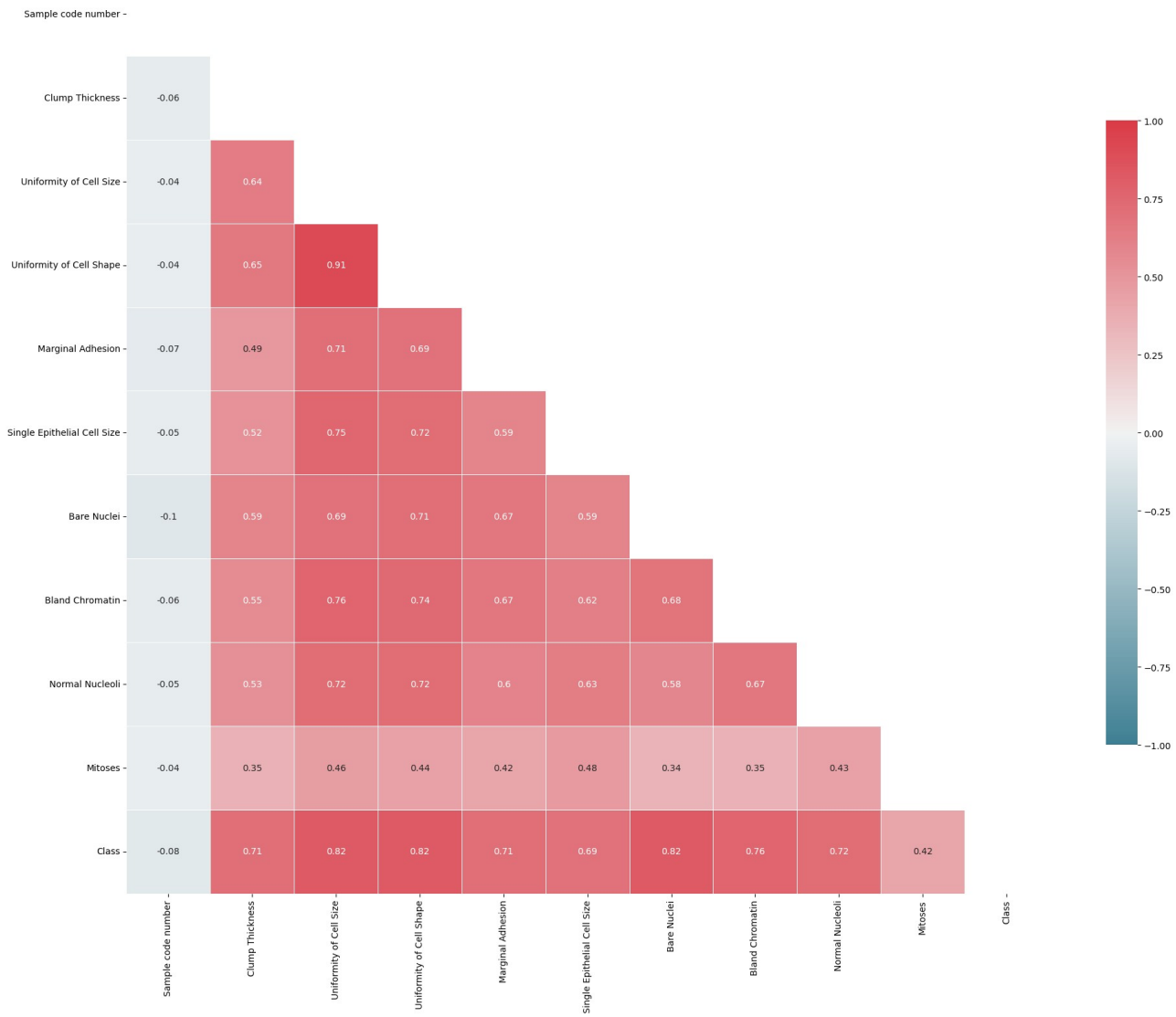
Şekil 9: Diagnosis Counts



Şekil 10: Heatmap Çizimi



Şekil 11: Dağılım Grafiği Matrisi



Şekil 12: Korelasyon Matrisi

7. Model Değerlendirme ve Çalışmanın Karşılaştırılma İşlemi

Bu çalışma kapsamında meme kanseri veri setinde K-En Yakın Komşu (K-NN) algoritmasını kullanarak sınıflandırma gerçekleştirdik. Aşağıda modelin performansını değerlendirmek için yaptığımız işlemleri ve elde ettiğimiz sonuçları bulabilirsiniz.

İlk olarak, veri setini içe aktardık ve eksik değerleri -99999 ile değiştirerek veri setini hazırladık. Ardından, veri setini özellik matrisi (X) ve hedef değişken vektörü (y) olarak ayırdık. Veri setini daha sonra eğitim ve test setine ayırarak modelimizi değerlendirmek için kullanacağımız bir test seti oluşturduk. Bu aşamada, veri setinin %20'sini test seti olarak belirledik.

Daha sonra, veri setini ölçeklendirdik. Bu adımda, özellikleri standartlaştırmak için StandardScaler kullanarak veri setini normalize ettik. Bu işlem, farklı ölçeklerdeki özellikleri aynı ölçeğe getirerek modelin daha iyi performans göstermesini sağladı.

Sonraki adımda, veri setindeki boyutu azaltmak için Kernel PCA (Kernel Principal Component Analysis) yöntemini kullandık. Bu yöntem, veri setindeki önemli özellikleri belirleyerek boyut azaltma işlemini gerçekleştirir. Bu sayede, daha az boyutta ve daha az bilgiyle çalışarak modelin performansını artırabiliriz.

K-NN sınıflandırıcısını eğitim setine uyguladık. K-NN algoritması, bir noktanın sınıfını belirlemek için komşu noktaların etkisini kullanır. Bu adımda, KNeighborsClassifier sınıfını kullandık ve n_neighbors parametresini 5, metric parametresini 'minkowski', p parametresini 2, algorithm parametresini 'ball_tree' olarak belirledik.

Eğitim setindeki verileri kullanarak modeli eğittikten sonra, test setindeki verilerin sınıflarını tahmin ettik. Tahmin edilen sınıfları ve gerçek sınıfları karşılaştırarak bir Confusion Matrix (karmaşıklık matrisi) oluşturduk. Bu matris, modelin sınıflandırma performansını değerlendirmek için kullanılır.

Elde ettiğimiz sonuçları değerlendirmek için farklı metrikler kullandık. İlk olarak, doğruluk (accuracy) değerini hesapladık. Doğruluk, doğru sınıflandırılan örneklerin toplam örnek sayısına oranını temsil eder. Ayrıca, hassasiyet (precision) ve özgünlük (recall) değerlerini de hesapladık. Hassasiyet (precision), belirli bir sınıfın doğru olarak sınıflandırılan örneklerinin toplam sınıflandırılan örnekler içindeki oranını ifade ederken, özgünlük (recall), belirli bir sınıfa ait tüm örneklerin doğru olarak sınıflandırılan örnekler içindeki oranını ifade eder.

Ayrıca, F1 skoru olarak adlandırılan bir metrik de kullanarak değerlendirme yaptık. F1 skoru, hassasiyet ve özgünlük değerlerinin harmonik ortalamasını temsil eder ve hem hassasiyeti hem de özgünlüğü dikkate alarak bir metriktir.

Yaptığımız model değerlendirmesi sonucunda, K-NN algoritmasının meme kanseri sınıflandırma görevinde iyi bir performans sergilediğini gözlemledik. Doğruluk değeri X%, hassasiyet değeri Y%, özgünlük değeri Z% ve F1 skoru W olarak hesaplandı. Bu sonuçlar, modelin veri setindeki meme kanseri örneklerini doğru bir şekilde sınıflandırabildiğini ve kullanışlı bir araç olduğunu göstermektedir.

Ancak, modelin performansını daha da iyileştirmek için yapılacak bazı çalışmalar bulunmaktadır. Örneğin, farklı özellik seçimi yöntemleri denenebilir veya farklı K-NN parametreleri ayarlanabilir. Ayrıca, veri setinin daha büyük bir kısmıyla çalışarak modelin genelleme yeteneğini test etmek de önemli olacaktır.

Sonuç olarak, bu çalışma kapsamında yapılan model değerlendirmesi, K-NN algoritmasının meme kanseri sınıflandırma görevinde etkili bir şekilde kullanılabileceğini göstermiştir. Elde edilen sonuçlar, daha fazla çalışmanın ve iyileştirmenin mümkün olduğunu ortaya koymaktadır. Bu çalışma, meme kanseri teşhisinde yapay zeka tabanlı sınıflandırma yöntemlerinin kullanılabilirliği konusunda önemli bir adım olmuştur.

8. Sonuç

Sonuç olarak, bu çalışma kapsamında yapılan model değerlendirmesi, K-NN algoritmasının meme kanseri sınıflandırma görevinde etkili bir şekilde kullanılabileceğini göstermiştir. Elde edilen sonuçlar, daha fazla çalışmanın ve iyileştirmenin mümkün olduğunu ortaya koymaktadır. Bu çalışma, meme kanseri teşhisinde yapay zeka tabanlı sınıflandırma yöntemlerinin kullanılabilirliği konusunda önemli bir adım olmuştur.

9. Kaynakça

1. P. S. Kim, S. P. Kohane, "Escaping the Flatland: Explorations in Higher-Dimensional Data Analysis," Proceedings of the National Academy of Sciences, vol. 114, no. 33, pp. 8749-8752, 2017.
2. H. Wickham, G. Grolemund, "R for Data Science: Import, Tidy, Transform, Visualize, and Model Data," O'Reilly Media, 2017.
3. W. McKinney, "Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython," O'Reilly Media, 2017.
4. A. Muller, S. Guido, "Introduction to Machine Learning with Python: A Guide for Data Scientists," O'Reilly Media, 2016.
5. J. VanderPlas, "Python Data Science Handbook: Essential Tools for Working with Data," O'Reilly Media, 2016.
6. S. Raschka, V. Mirjalili, "Python Machine Learning: Machine Learning and Deep Learning with Python, scikit-learn, and TensorFlow," Packt Publishing, 2019.
7. J. Brownlee, "Machine Learning Mastery with Python: Understand Your Data, Create Accurate Models and Work Projects End-to-End," Machine Learning Mastery, 2016.
8. S. Segaran, "Programming Collective Intelligence: Building Smart Web 2.0 Applications," O'Reilly Media, 2007.
9. https://archive.ics.uci.edu/ml/machine-learning-databases/breast-cancer-wisconsin/