KOD

# Kütüphanelerimiz

import numpy as np

import pandas as pd

from sklearn import linear\_model, svm, naive\_bayes, tree, neural\_network

from sklearn.model\_selection import cross\_val\_score

from sklearn.model\_selection import train\_test\_split

from sklearn.model\_selection import GridSearchCV

import matplotlib.pyplot as plt

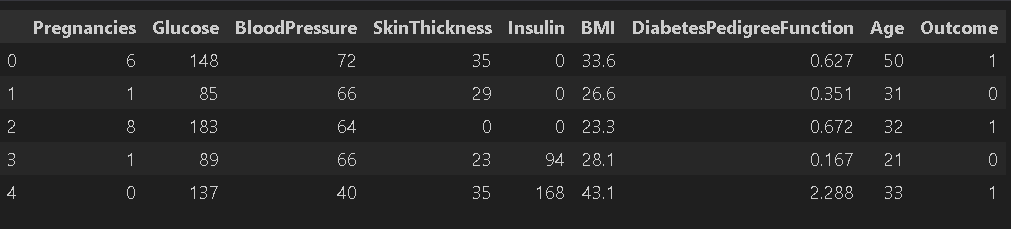
import seaborn as sns

import pickle

# Veri setini yükleme

df = pd.read\_csv("diabetes.csv")

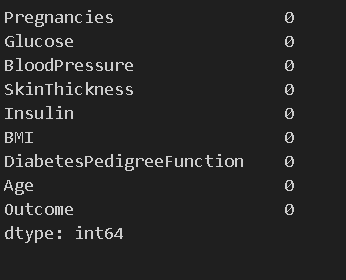
# İlk 5 satırı gösterme

df.head()  
  


# Null değerleri görüntülediğimiz kısım (Bilgi edinme kısmı)

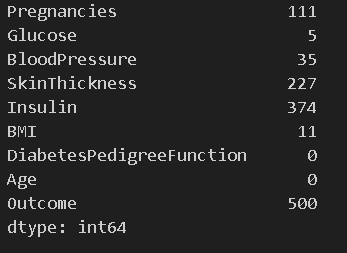
df.info()

# Veri setinde "NULL" kısım var mı? Varsa toplam kaçar adet var?

df.isnull().sum()  
  


# Sıfır değerini içeren verilere bakalım.

df.eq(0).sum()



# Sıfır olan değerler bizleri yanıltacağı için bunları NaN olarak değiştirelim.

# Bir kişi hiç hamile kalmamış olabilir veya bu kişi Erkek olabilir.

# Haricen sonuç (OUTCOME) değerimiz sıfır olabilir. Bu yüzden onu değiştirmeyelim.

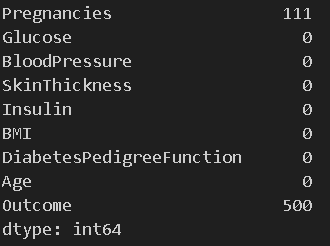
df[['Glucose', 'BloodPressure', 'SkinThickness', 'Insulin', 'BMI', 'DiabetesPedigreeFunction','Age']] = df[['Glucose', 'BloodPressure', 'SkinThickness', 'Insulin', 'BMI', 'DiabetesPedigreeFunction','Age']].replace(0,np.NaN)

# Eksik verileri ortalama ile dolduralım.

df.fillna(df.mean(),inplace=True)

# Şimdi 0 olan değerleri görelim.

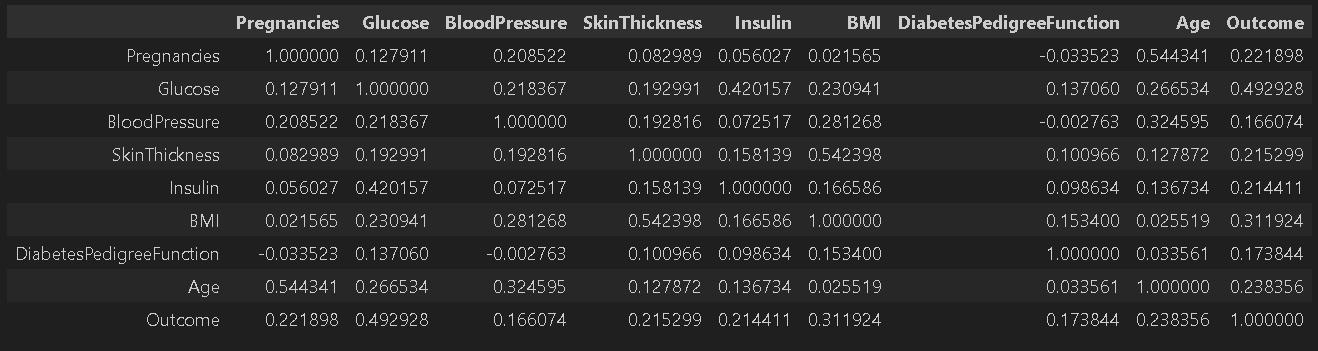
df.eq(0).sum()



# Korelasyon analizi

# Çıktı değişkenine hangi özelliklerin etki ettiğini korelasyon ile görebiliriz.

df.corr()

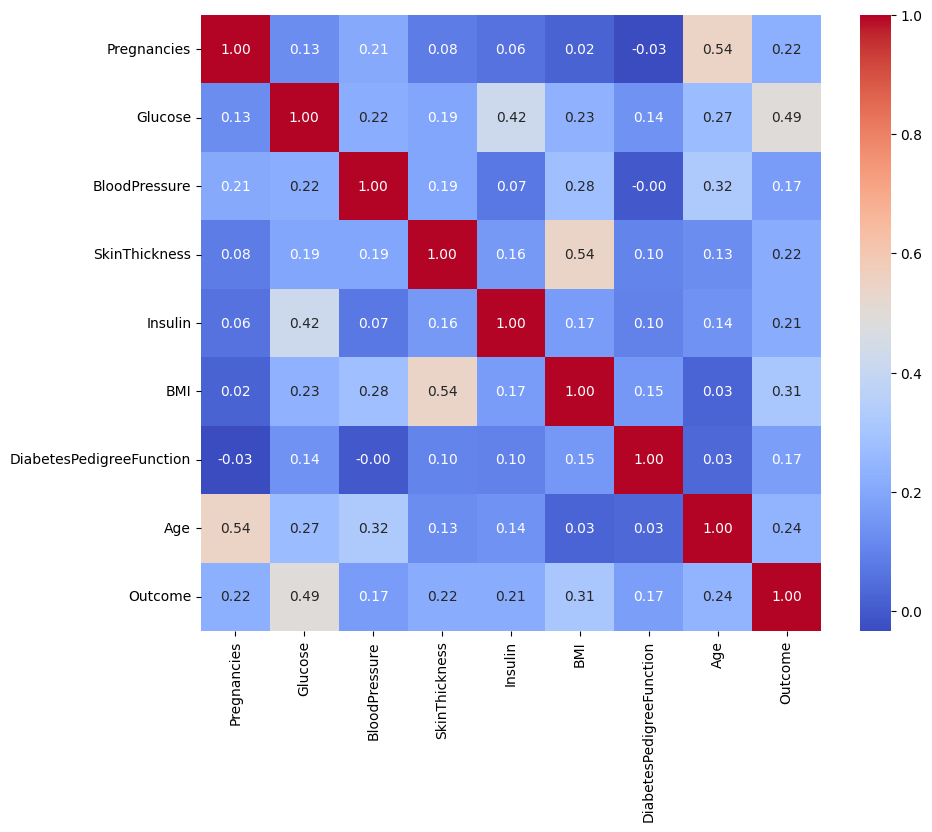


# Korelasyon grafiği

plt.figure(figsize=(10, 8))

sns.heatmap(df.corr(), annot=True, cmap='coolwarm', fmt=".2f")

plt.show()



#Kabarcık Grafiği

X = df[['Glucose', 'BMI', 'Age']].values

y = df['Outcome'].values

# Yaşlarına göre Glukoz ve BMI değerlerini karşılaştıralım.

plt.figure(figsize=(10, 8))

plt.scatter(df[df['Outcome'] == 0]['Age'], df[df['Outcome'] == 0]['Glucose'], color='blue', label='Outcome 0')

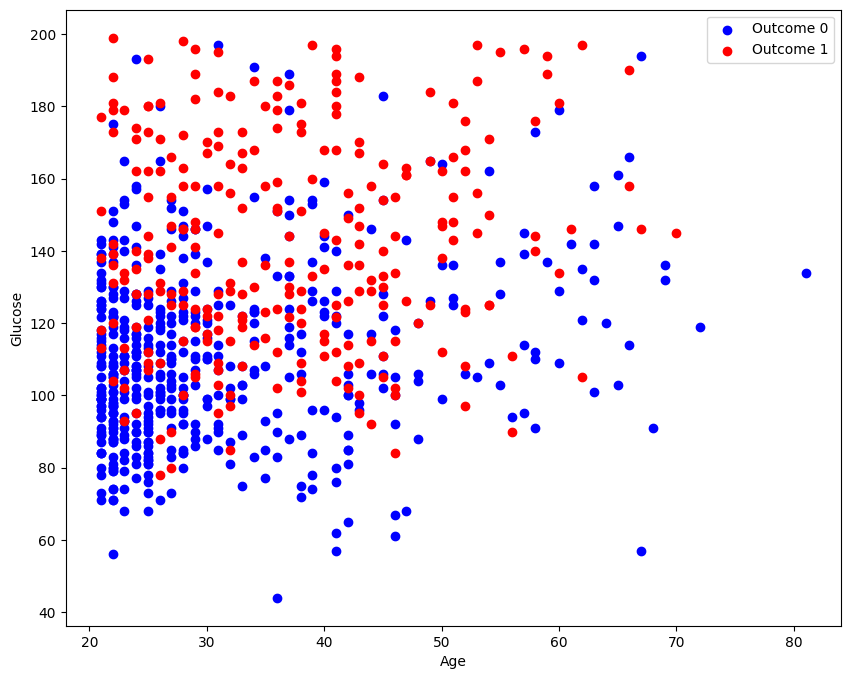
plt.scatter(df[df['Outcome'] == 1]['Age'], df[df['Outcome'] == 1]['Glucose'], color='red', label='Outcome 1')

plt.xlabel('Age')

plt.ylabel('Glucose')

plt.legend()

plt.show()



#Çubuk Grafiği (Maksimum Verileri)  
# Yaş ve Hamilelik durumu için çubuk grafiği çizimi

X = df[['Age']].values

y = df[['Pregnancies']].values

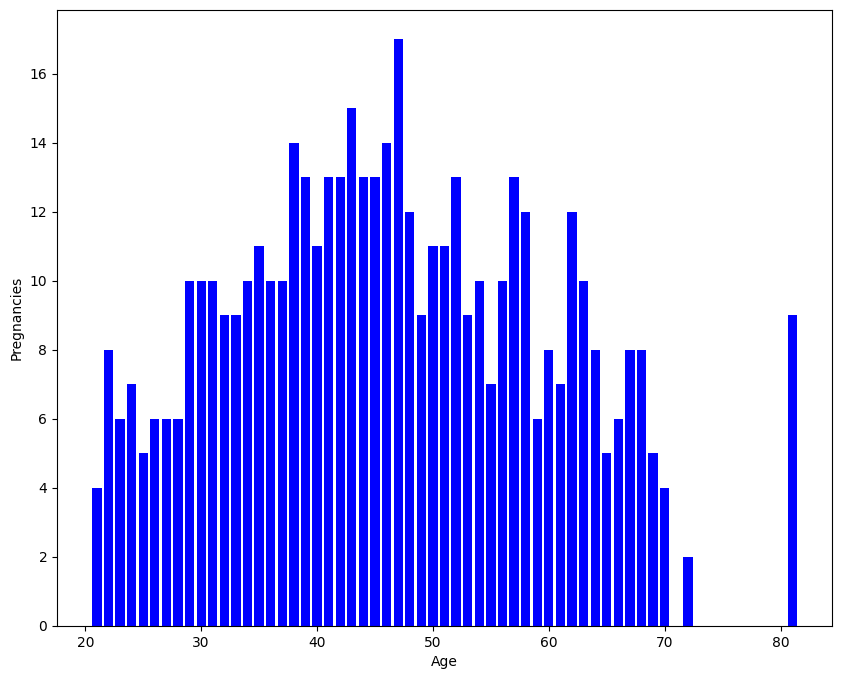
plt.figure(figsize=(10, 8))

plt.bar(X[:, 0], y[:, 0], color='blue') # Use plt.bar for bar chart

plt.xlabel('Age')

plt.ylabel('Pregnancies')

plt.show()



# Outcome değişkenine en yüksek korelasyona sahip 4 özellik

top\_correlated\_features = df.corr().nlargest(4, 'Outcome').index

print(top\_correlated\_features)



# Burada outcome değişkenine en fazla korelasyona sahip 'Glucose', 'BMI', 'Age' özellikleri ile analize devam edelim.

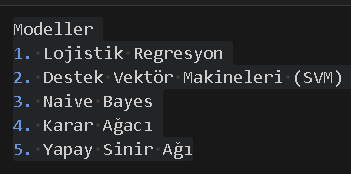
X = df[['Glucose','BMI','Age']].values

y=df.iloc[:,8].values



# Veri setini eğitim ve test kümelerine ayırın (Veri setinin %66'sı eğitim için, %34'ü test için ayrıldı)

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.34, random\_state=42)



# Modelleme ve doğruluk skorlarının hesaplanması

models = {

"Lojistik Regresyon": linear\_model.LogisticRegression(),

"Destek Vektör Makineleri (SVM)": svm.SVC(kernel='linear'),

"Naive Bayes": naive\_bayes.GaussianNB(),

"Karar Ağacı": tree.DecisionTreeClassifier(),

"Yapay Sinir Ağı": neural\_network.MLPClassifier()

}

resultsAll={}

results34 = {}

for name, model in models.items():

score =cross\_val\_score(model,X,y,cv=10,scoring='accuracy').mean()

resultsAll[name] = score

print(f"{name} Doğruluk Skoru: {score}")

**Sonuç:**Lojistik Regresyon Doğruluk Skoru: 0.7669856459330144

Destek Vektör Makineleri (SVM) Doğruluk Skoru: 0.7656527682843473

Naive Bayes Doğruluk Skoru: 0.7734962406015038

Karar Ağacı Doğruluk Skoru: 0.693984962406015

Yapay Sinir Ağı Doğruluk Skoru: 0.7031271360218729

for name, model in models.items():

score = cross\_val\_score(model, X\_train, y\_train, cv=10, scoring='accuracy').mean()

results34[name] = score

print(f"{name} Doğruluk Skoru: {score}")

**Sonuç:**

Lojistik Regresyon Doğruluk Skoru: 0.7823529411764707

Destek Vektör Makineleri (SVM) Doğruluk Skoru: 0.786235294117647

Naive Bayes Doğruluk Skoru: 0.786313725490196

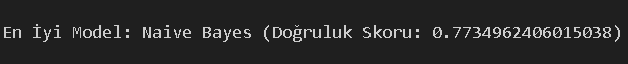
Karar Ağacı Doğruluk Skoru: 0.6974901960784312

Yapay Sinir Ağı Doğruluk Skoru: 0.7094901960784312

# Modelin tamamının(%100) iyi modelin seçilmesi

best\_model = max(resultsAll, key=resultsAll.get)

print(f"\nEn İyi Model: {best\_model} (Doğruluk Skoru: {resultsAll[best\_model]})")



# %34 olan durumda en iyi modelin seçilmesi

best\_model = max(results34, key=results34.get)

print(f"\nEn İyi Model: {best\_model} (Doğruluk Skoru: {results34[best\_model]})")



# Modelin Kaydedileceği Yer

filename='diabets.sav'

# Modelin kaydedilmesi

best\_model\_instance = models[best\_model]

best\_model\_instance.fit(X\_train, y\_train)

pickle.dump(best\_model\_instance, open(filename, 'wb'))

# Modelin yüklenmesi

loaded\_model = pickle.load(open(filename,'rb'))

# prediction

Glucose = 5

BMI = 90

Age = 90

prediction = loaded\_model.predict([[Glucose,BMI,Age]])

if prediction==1:

print("Diyabet olabilir")

else:

print("Diyabet olmayabilir")

