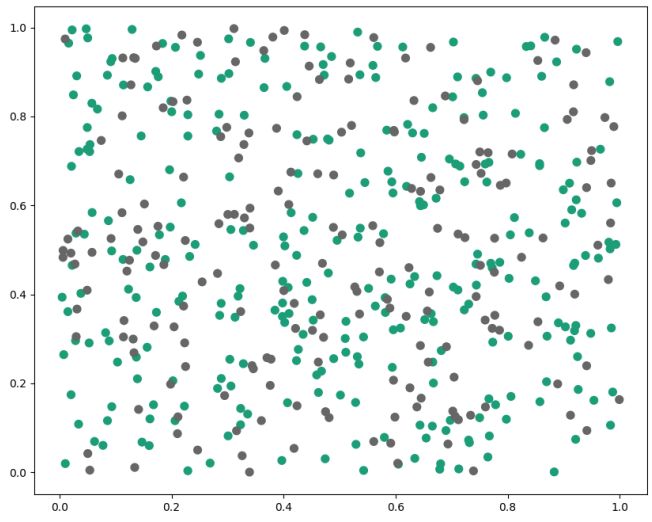
Ceyda Uymaz 200301503

500,5 lik data oluşturuyorum ilk 300ünü 1. Sınıf kalan son 200 ünü 2. Sınıf yapıyorum. Data 300'e 200 yani dengesiz olduğundan 300/500 = 0,6 sonucunu elde etmeyi bekliyorum.

```
import pandas as pd
import numpy as np
from sklearn.svm import SVC
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
from sklearn import datasets
# IRIS Data Set
X = np.random.rand(500,5)
y = []
for i in range(500):
    if i<300:
        y.append(1)
    else:
        y.append(2)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=1, stratify = y)
# Feature Scaling
sc = StandardScaler()
sc.fit(X_train)
X_train_std = sc.transform(X_train)
X_test_std = sc.transform(X_test)
# Training a SVM classifier using SVC class
svm = SVC(kernel= 'linear', random_state=1, C=0.1)
svm.fit(X_train_std, y_train)
# Mode performance

y_pred = svm.predict(X_test_std)
print('Accuracy: %.3f' % accuracy_score(y_test, y_pred))
plt.scatter(X[:, 3], X[:, 4], c=y, s=50, cmap='Dark2')
#plt.plot([0, 1.01],[1, 1.01])
plt.show()
```

```
Accuracy: 0.600
```

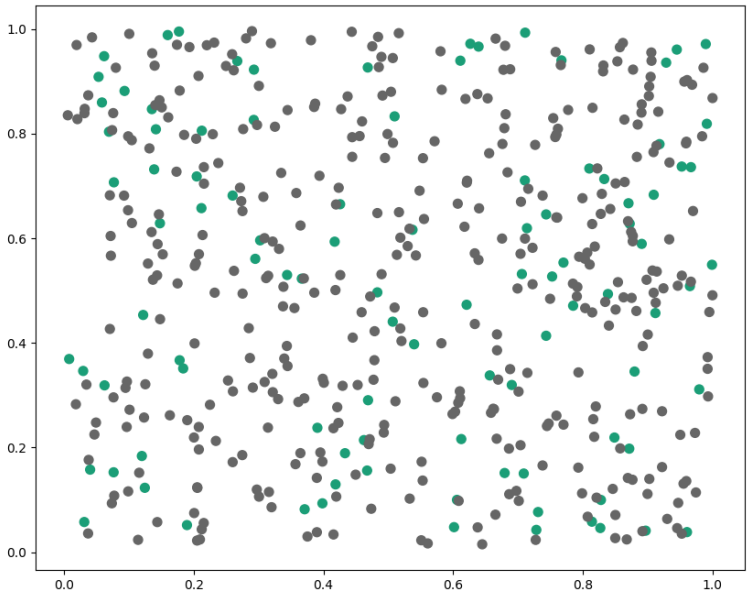


500,5 lik datamı 300’e 200 değilde 400’e 100 ayırıyorum 400/500=0.80 başarı oranı bekliyorum. Burda elde ettiğim başarı 2.sınıfımdaki datadamın az olduğundan yani datamın dengesiz olduğundan kaynaklı.

```
import pandas as pd
import numpy as np
from sklearn.svm import SVC
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
from sklearn import datasets
# IRIS Data Set
X = np.random.rand(500,5)
y = []
for i in range(500):
    if i<100:
        y.append(1)
    else:
        y.append(2)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=1, stratify = y)
# Feature Scaling
sc = StandardScaler()
sc.fit(X_train)
X_train_std = sc.transform(X_train)
X_test_std = sc.transform(X_test)
# Training a SVM classifier using SVC class
svm = SVC(kernel= 'linear', random_state=1, C=0.1)
svm.fit(X_train_std, y_train)
# Mode performance

y_pred = svm.predict(X_test_std)
print('Accuracy: %.3f' % accuracy_score(y_test, y_pred))
plt.scatter(X[:, 3], X[:, 4], c=y, s=50, cmap='Dark2')
#plt.plot([0, 1.01],[1, 1.01])
plt.show()
```

```
Accuracy: 0.800
```

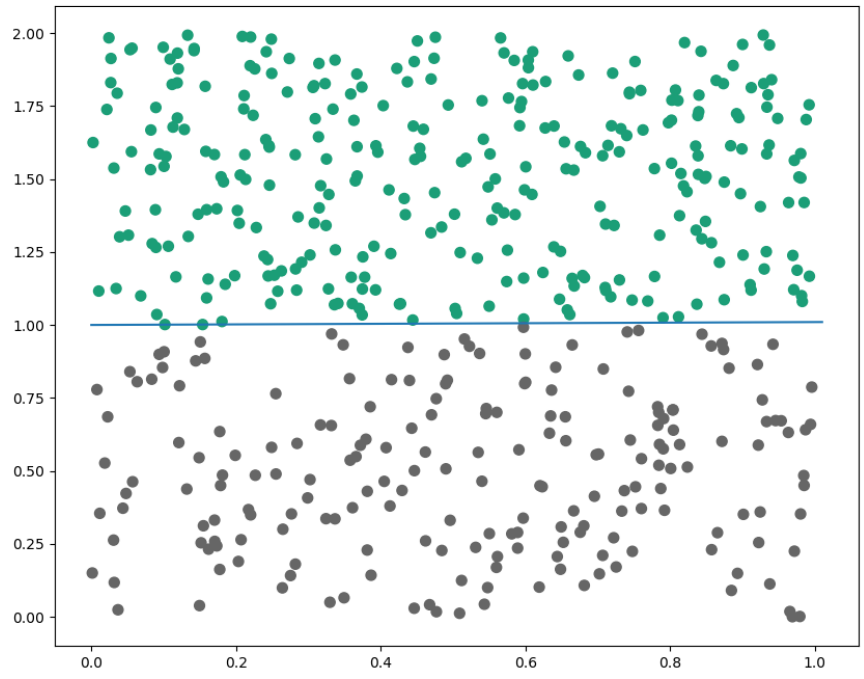


500,5 lik 300'e 200 olarak ayırdığım datama yanlılık ekliycem. Bunun için 1.sınıftaki dataların 5.satırındaki random sayılara 1 ekliyorum.

```
import pandas as pd
import numpy as np
from sklearn.svm import SVC
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
from sklearn import datasets
# IRIS Data Set
X = np.random.rand(500,5)
y = []
for i in range(500):
    if i<300:
        y.append(1)
    else:
        y.append(2)
for i in range(300):
    X[i][4] =X[i][4]+1
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=1, stratify = y)
# Feature Scaling
sc = StandardScaler()
sc.fit(X_train)
X_train_std = sc.transform(X_train)
X_test_std = sc.transform(X_test)
# Training a SVM classifier using SVC class
svm = SVC(kernel= 'linear', random_state=1, C=0.1)
svm.fit(X_train_std, y_train)
# Mode performance

y_pred = svm.predict(X_test_std)
print('Accuracy: %.3f' % accuracy_score(y_test, y_pred))
plt.scatter(X[:, 3], X[:, 4], c=y, s=50, cmap='Dark2')
plt.plot([0, 1.01],[1, 1.01])
plt.show()
```

```
Accuracy: 0.987
```

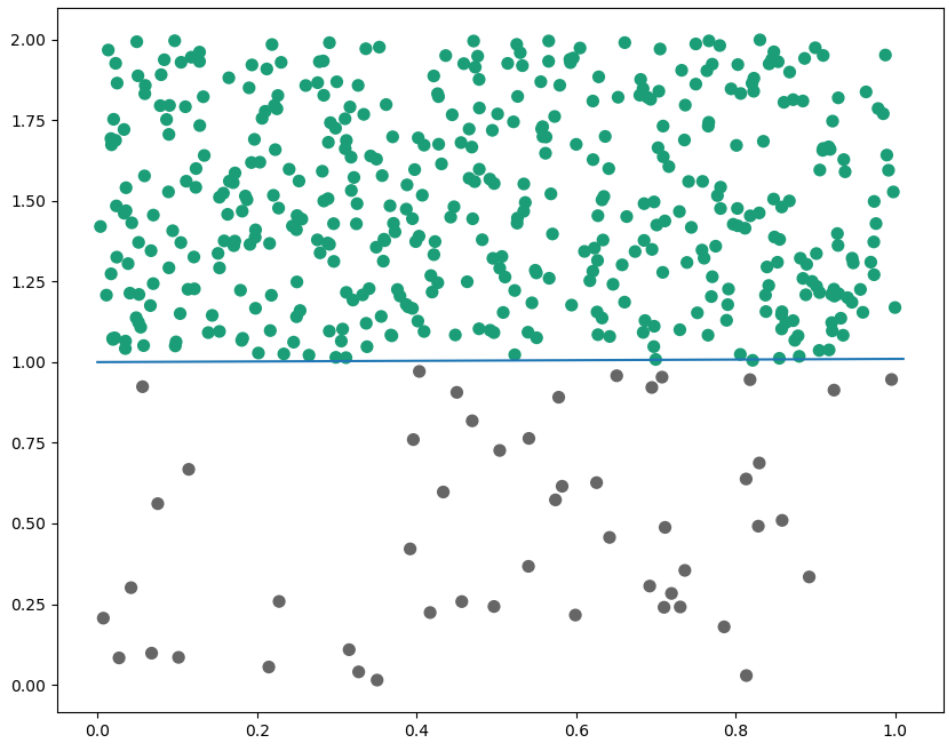


Şimdi datamı 450'e 50 olarak ayırıyorum

```
import pandas as pd
import numpy as np
from sklearn.svm import SVC
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
from sklearn import datasets
# IRIS Data Set
X = np.random.rand(500,5)
y = []
for i in range(500):
    if i<450:
        y.append(1)
    else:
        y.append(2)
for i in range(450):
    X[i][4] =X[i][4]+1
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=1, stratify = y)
# Feature Scaling
sc = StandardScaler()
sc.fit(X_train)
X_train_std = sc.transform(X_train)
X_test_std = sc.transform(X_test)
# Training a SVM classifier using SVC class
svm = SVC(kernel= 'linear', random_state=1, C=0.1)
svm.fit(X_train_std, y_train)
# Mode performance

y_pred = svm.predict(X_test_std)
print('Accuracy: %.3f' % accuracy_score(y_test, y_pred))
plt.scatter(X[:, 3], X[:, 4], c=y, s=50, cmap='Dark2')
plt.plot([0, 1.01],[1, 1.01])
plt.show()
```

```
Accuracy: 0.960
```

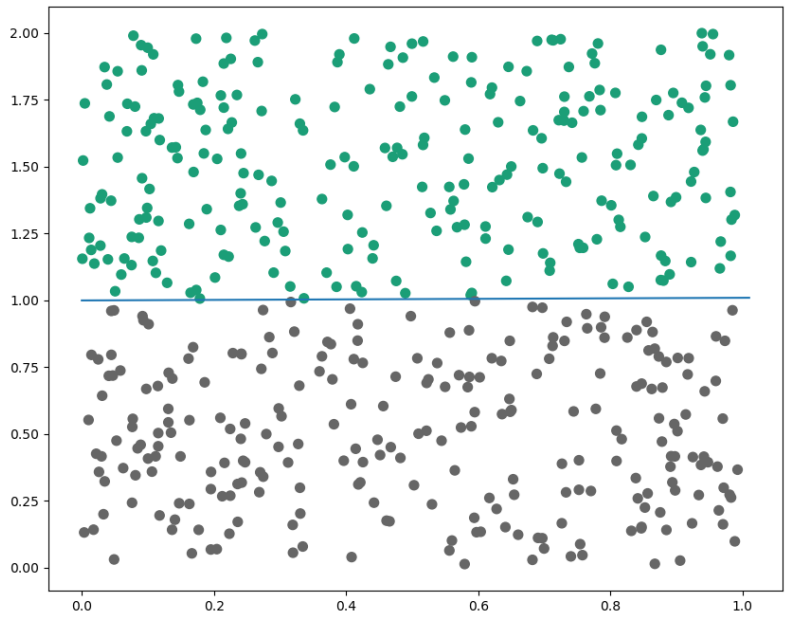


Şimdi datamı 250'e 250 olarak ayırıyorum

```
import pandas as pd
import numpy as np
from sklearn.svm import SVC
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
from sklearn import datasets
# IRIS Data Set
X = np.random.rand(500,5)
y = []
for i in range(500):
    if i<250:
        y.append(1)
    else:
        y.append(2)
for i in range(250):
    X[i][4] =X[i][4]+1
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=1, stratify = y)
# Feature Scaling
sc = StandardScaler()
sc.fit(X_train)
X_train_std = sc.transform(X_train)
X_test_std = sc.transform(X_test)
# Training a SVM classifier using SVC class
svm = SVC(kernel= 'linear', random_state=1, C=0.1)
svm.fit(X_train_std, y_train)
# Mode performance

y_pred = svm.predict(X_test_std)
print('Accuracy: %.3f' % accuracy_score(y_test, y_pred))
plt.scatter(X[:, 3], X[:, 4], c=y, s=50, cmap='Dark2')
plt.plot([0, 1.01],[1, 1.01])
plt.show()
```

```
Accuracy: 0.993
```



Şunu fark ediyorum ki eğitim setimde yanlılık yokken 1. Sınıf sayısı 2. Sınıf sayısından fazla olunca başarı oranı artıyor ama bişey kazanmamış oluyoruz çünkü bu başarı oranı data dengesizliği yüzünden oluşan başarı.

Fakat yanlılık eklediğimde sınıf sayısı ne kadar dengeliyse başarı oranımda o kadar artıyor. (veri setimi 250-250 ayırdığımda yani eşit ayırdığımda daha yüksek başarı oranı elde ettim)