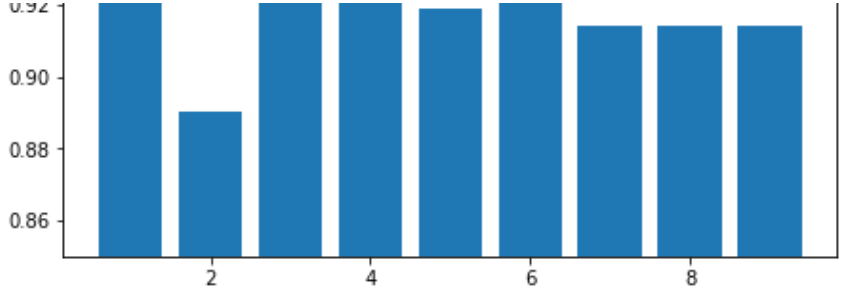
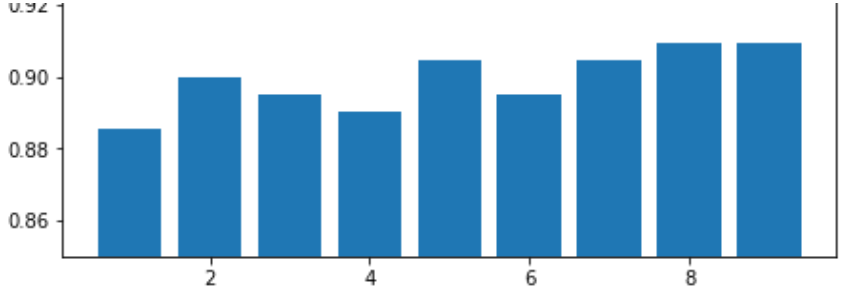
```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd

from sklearn.model_selection import KFold, train_test_split
from sklearn.neighbors import KNeighborsClassifier
from sklearn.decomposition import PCA
from sklearn.tree import DecisionTreeClassifier, plot_tree


df = pd.read_csv('seeds_dataset.txt', sep='\s+', header=None)
y = np.array(df[7])
df = (df-df.mean())/df.std()
X = np.array(df.drop(columns=7))


seed = 42
k_cross = 5
k_neighbors = np.arange(1, 10)
metrics = ['euclidean', 'manhattan', 'chebyshev', 'minkowski']


pca = PCA(n_components=2)
X_trans = pca.fit_transform(X)

kf = KFold(n_splits=k_cross, shuffle=True, random_state=42)


acc = np.zeros((len(k_neighbors), len(metrics)))

for ik, k in enumerate(k_neighbors):
  for im, metric in enumerate(metrics):
    temp_acc = np.zeros(k_cross)
    #this gonna store he accuracies o average for (metric, k_neighbors)

    clf = KNeighborsClassifier(n_neighbors=k, metric=metric)
    it = 0  #Bc I'm way too lazy to effectively use enumerate() on KFoldClassifier.split() lol
    for train_index, test_index in kf.split(X):
```

0s completed at 9:04 AM

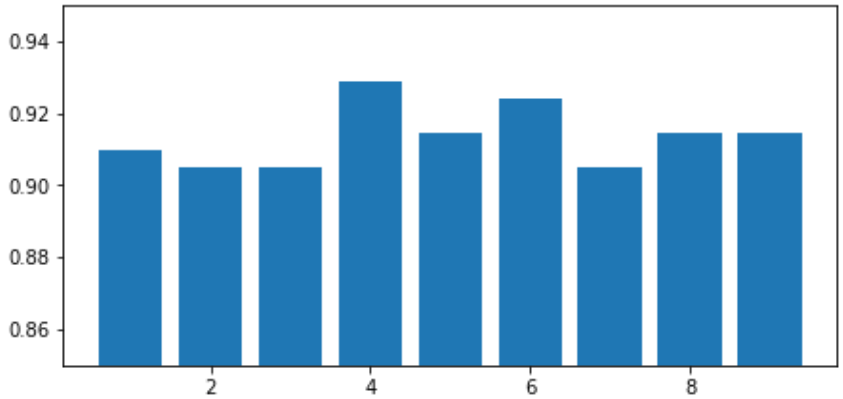
```
      if it >= 5: #Probably not necessary
        break

    acc[ik, im] = np.mean(temp_acc)


acc_trans = np.zeros((len(k_neighbors), len(metrics)))

for ik, k in enumerate(k_neighbors):
  for im, metric in enumerate(metrics):
    temp_acc = np.zeros(k_cross)

    clf = KNeighborsClassifier(n_neighbors=k, metric=metric)
    it = 0
    for train_index, test_index in kf.split(X_trans):

      clf.fit(X_trans[train_index], y[train_index])
      labels_pred = clf.predict(X_trans[test_index])
      temp_acc[it] = clf.score(X_trans[test_index], y[test_index])

      it += 1
      if it >= 5:
        break

    acc_trans[ik, im] = np.mean(temp_acc)


fig, axs = plt.subplots(len(metrics), 2, figsize=(16,16))

for im, metric in enumerate(metrics):
  axs[im,0].bar(k_neighbors, np.transpose(acc)[im])
  axs[im,1].bar(k_neighbors, np.transpose(acc_trans)[im])
```



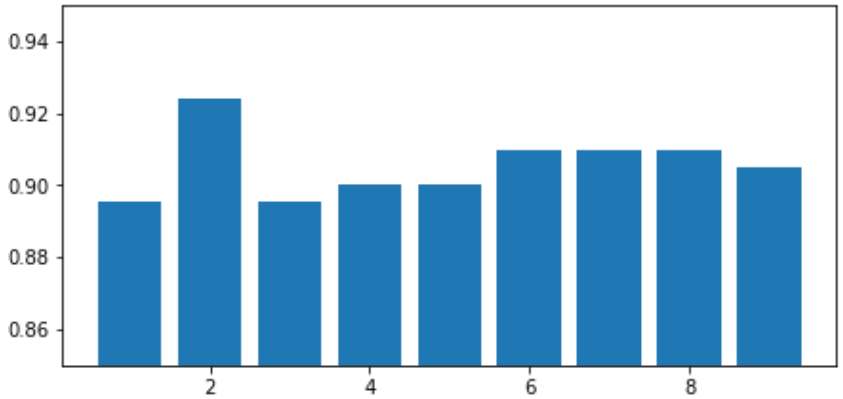
manhattan



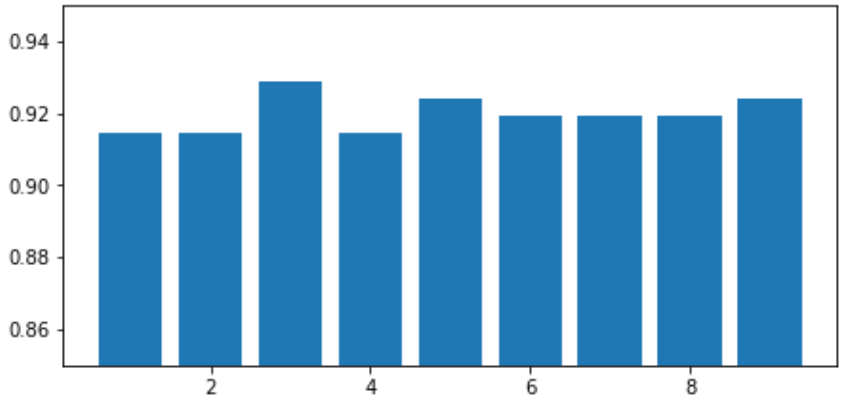
manhattan after PCA



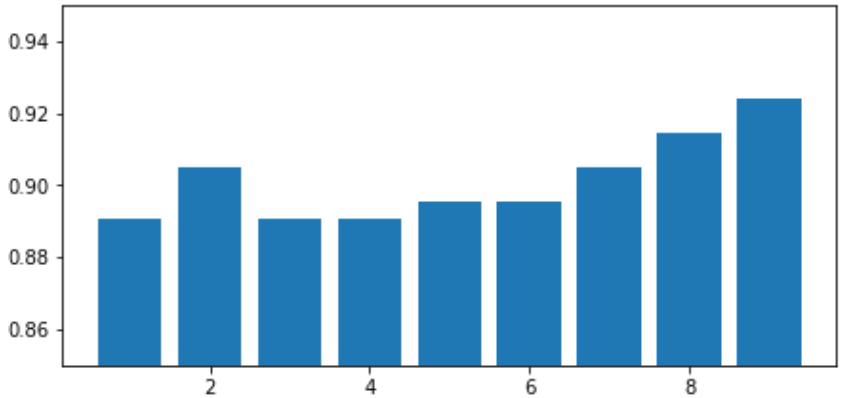
chebyshev



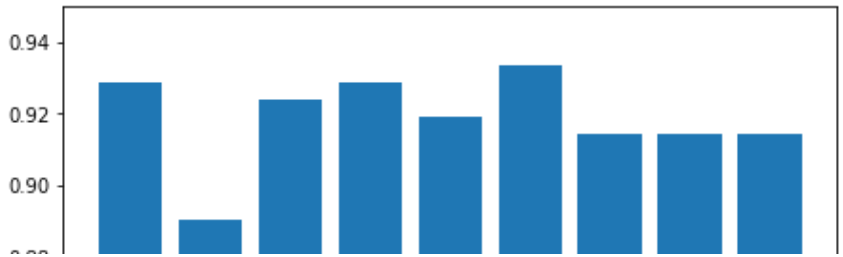
chebyshev after PCA

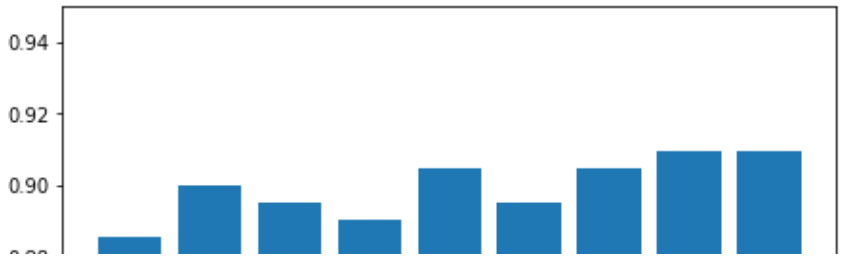


minkowski



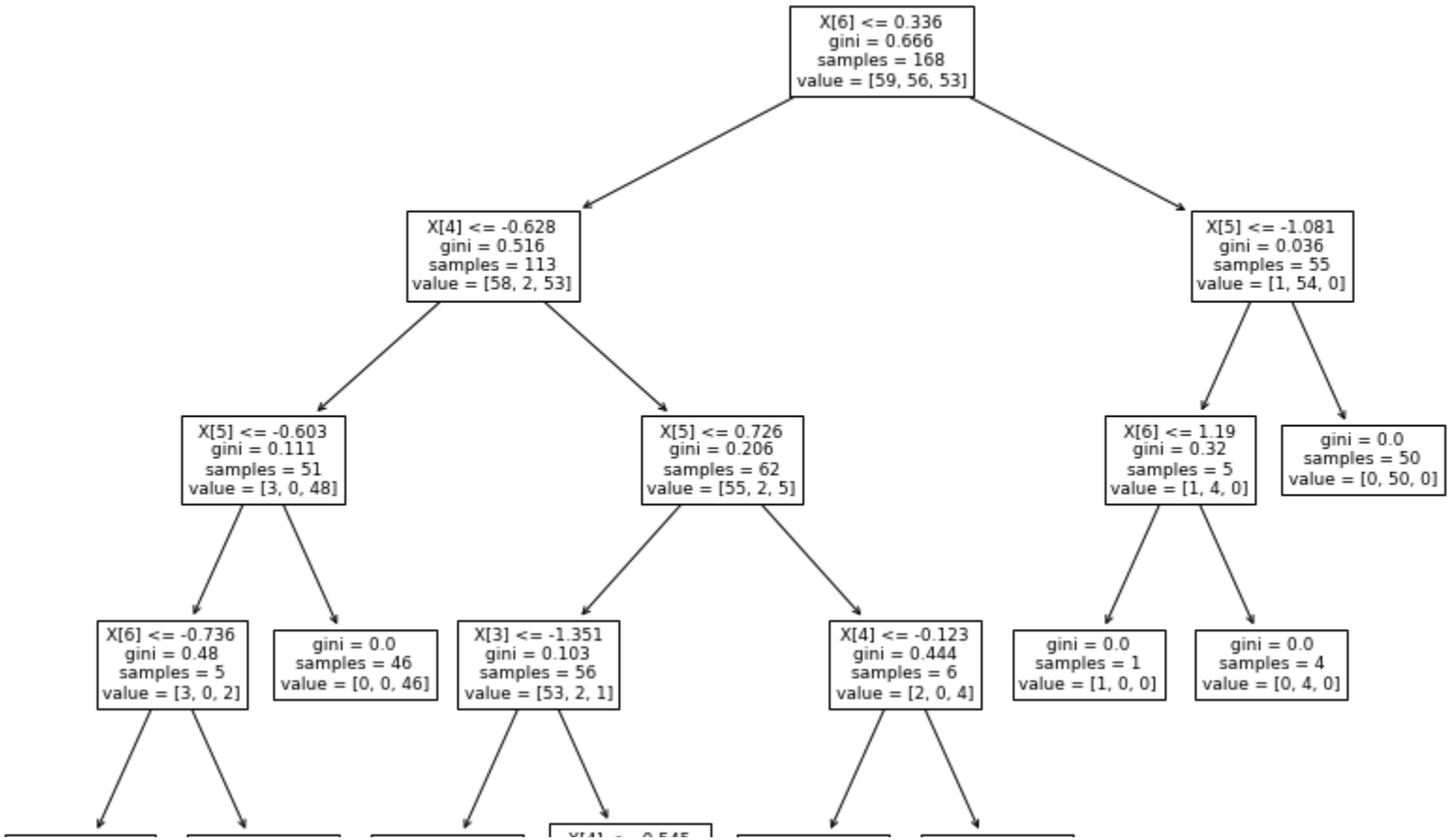
minkowski after PCA

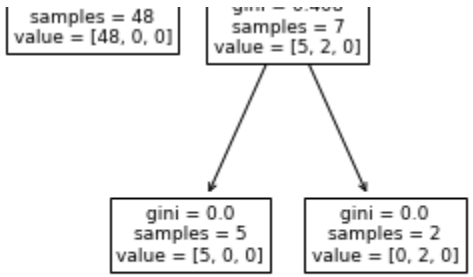




```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.20, random_state=seed)
tree_clf = DecisionTreeClassifier(random_state=seed, criterion='gini')


plt.figure(figsize=(14,14))
tree_clf.fit(X_train, y_train)
print(tree_clf.score(X_test, y_test))
plot_tree(tree_clf)
plt.show()
```

```
0.8333333333333334
```




samples = 48
value = [48, 0, 0]
samples = 7
value = [5, 2, 0]
gini = 0.0
samples = 5
value = [5, 0, 0]
gini = 0.0
samples = 2
value = [0, 2, 0]