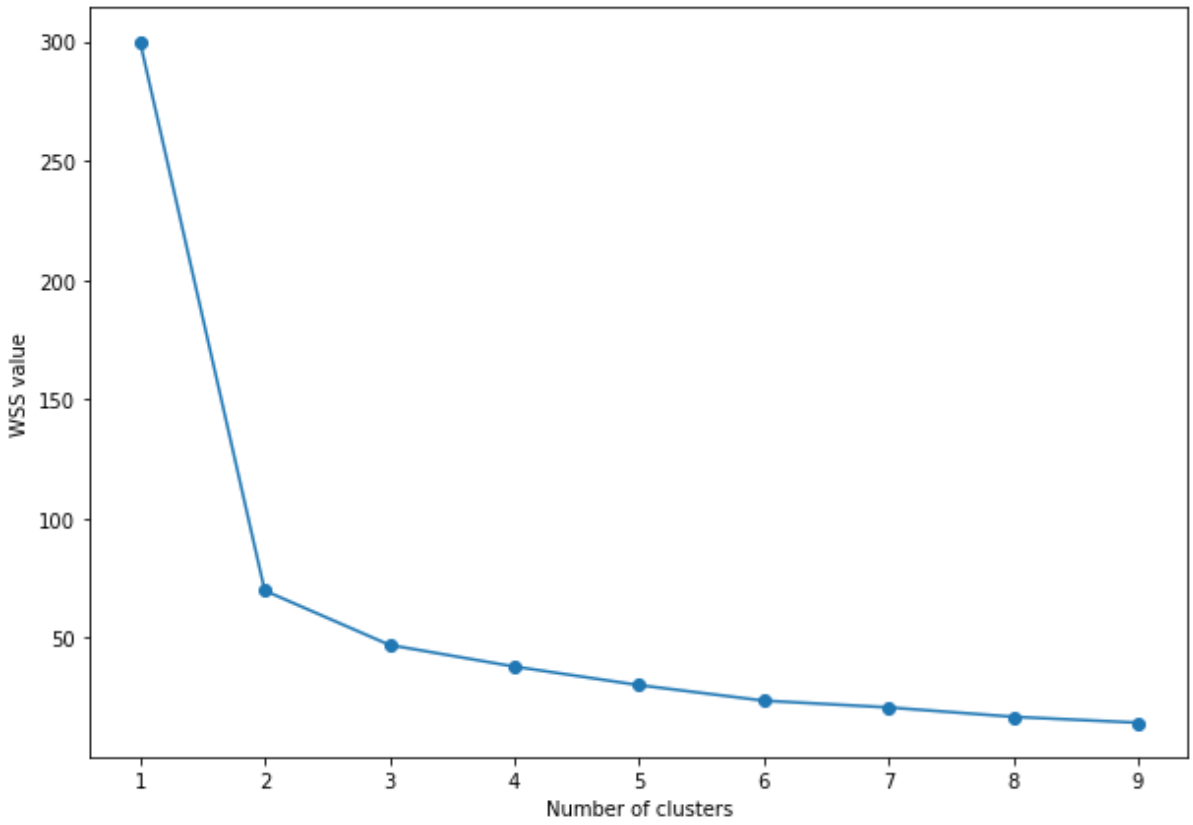
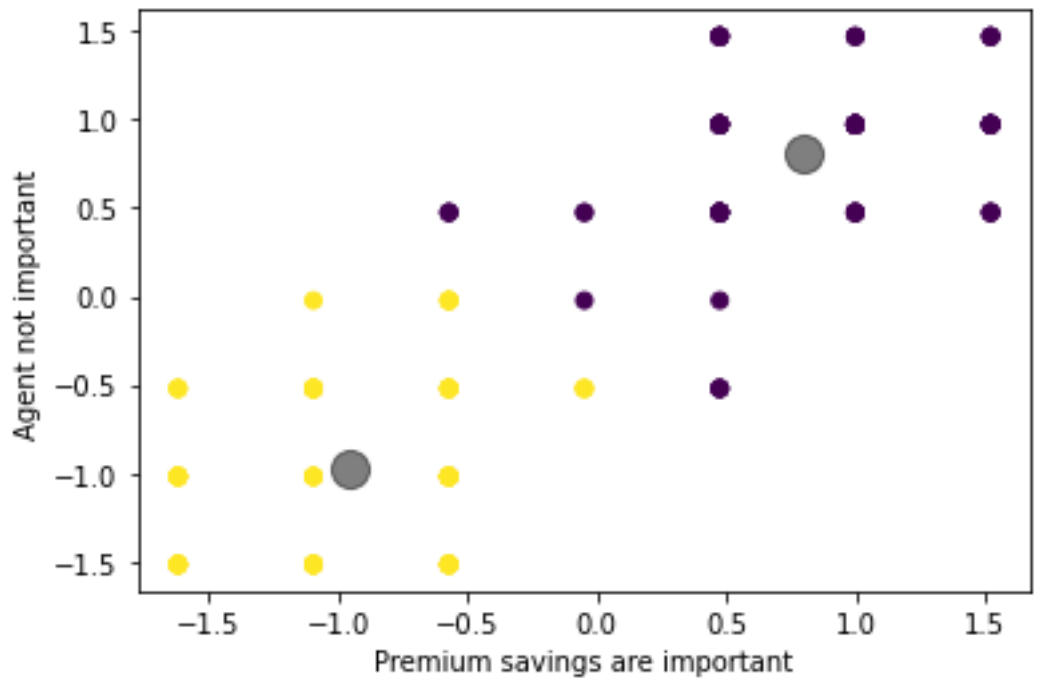
```
# -*- coding: utf-8 -*-
"""
Created on Sat Aug 15 11:10:14 2020

K-means using euclidean distance for

@author: Walter
"""

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sn

%matplotlib inline

data = pd.read_csv('sample.csv')

from sklearn.cluster import KMeans
from sklearn.preprocessing import StandardScaler

#data.head(10)
#data.info

X = data
X =X.values

#X

scaler = StandardScaler()
X_scaled = scaler.fit_transform( X )
```

```
#scale the values so one doesn’t dominate

#X

cluster_range = range( 1, 10)

# searching for optimal K value across the range 1 - 10

cluster_errors = []

for num_clusters in cluster_range:
  clusters = KMeans( num_clusters )
  clusters.fit( X_scaled )
  cluster_errors.append( clusters.inertia_ )

  clusters_df = pd.DataFrame( { "num_clusters":cluster_range, "cluster_errors": cluster_errors } )



#clusters_df[0:10]
#

#plt.figure(figsize=(10,7))
#plt.plot( clusters_df.num_clusters, clusters_df.cluster_errors, marker = "o" )
#plt.xlabel('Number of clusters')
#plt.ylabel('WSS value')
```




```
#the 'elbow' is at k == 2


kmeans = KMeans(n_clusters=2)

kmeans.fit(X_scaled)

y_kmeans = kmeans.predict(X_scaled)


#y_kmeans


centers = kmeans.cluster_centers_

#centers


plt.scatter(X_scaled[:, 0], X_scaled[:, 1], c=y_kmeans)

plt.scatter(centers[:, 0], centers[:, 1], c='black', s=200, alpha=0.5)

plt.xlabel('Premium savings are important')

plt.ylabel('Agent not important')
```



#Comparing it with k = 3

#kmeans = KMeans(n_clusters=3)

#kmeans.fit(X_scaled)

#y_kmeans = kmeans.predict(X_scaled)

#y_kmeans

#centers = kmeans.cluster_centers_

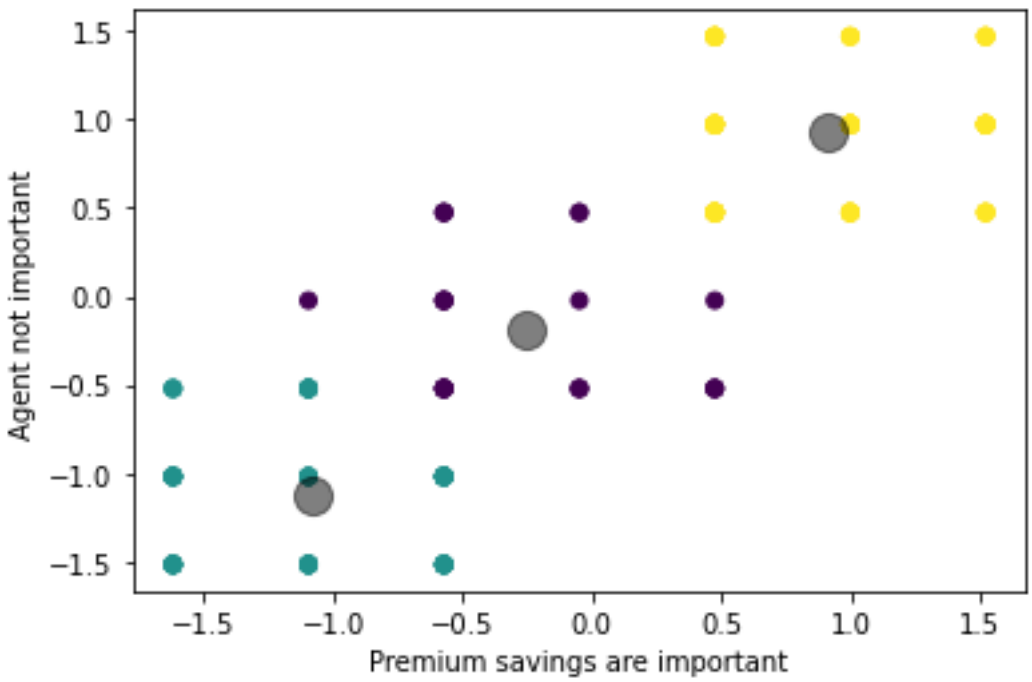
#centers

#plt.scatter(X_scaled[:, 0], X_scaled[:, 1], c=y_kmeans)

#plt.scatter(centers[:, 0], centers[:, 1], c='black', s=200, alpha=0.5)

#plt.xlabel('Premium savings are important')

#plt.ylabel('Agent not important')


1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
Agent not important
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
Premium savings are important