

K-means Clustering

```
df = pd.read_csv('./Final_Refined_Encoded_Normalysed.csv')

train, test = train_test_split(df, test_size = 0.3, random_state = 10)

# X = np.array(train.drop(['Price'], 1).astype(float))
X = np.array(train[['SIM', 'CPU', 'GPU', 'memory_card', 'weight_g', 'screen_to_body_ratio', \
    'primary_camera', 'internal_memory', 'Thickness', \
    'display_size', 'OS', 'radio', 'RAM', 'EDGE'\
    ]])
y = np.array(train['Price'])
#train.info()

kmeans = KMeans(algorithm='auto', copy_x=True, init='k-means++', max_iter=600,
    n_clusters=6, n_init=10, n_jobs=1, precompute_distances='auto',
    random_state=None, tol=0.0001, verbose=0)

scaler = MinMaxScaler()
X_scaled = scaler.fit_transform(X)
kmeans.fit(X_scaled)

correct = 0
for i in range(len(X)):
    predict_me = np.array(X[i].astype(float))
    predict_me = predict_me.reshape(-1, len(predict_me))
    prediction = kmeans.predict(predict_me)
    if prediction[0] == y[i]:
        correct += 1

print("Accuracy Rate:", correct/len(X))

# Reference: https://www.datacamp.com/community/tutorials/k-means-clustering-python
```

Output:

→ python3 k-means_clustering.py
Accuracy Rate: 0.39406053683609366

→ python3 k-means_clustering.py
Accuracy Rate: 0.385208452312964

→ python3 k-means_clustering.py
Accuracy Rate: 0.008280982295830954