K-Nearest Neighbors
~ By Zaid Mallik

Estimated time needed: 30 minutes

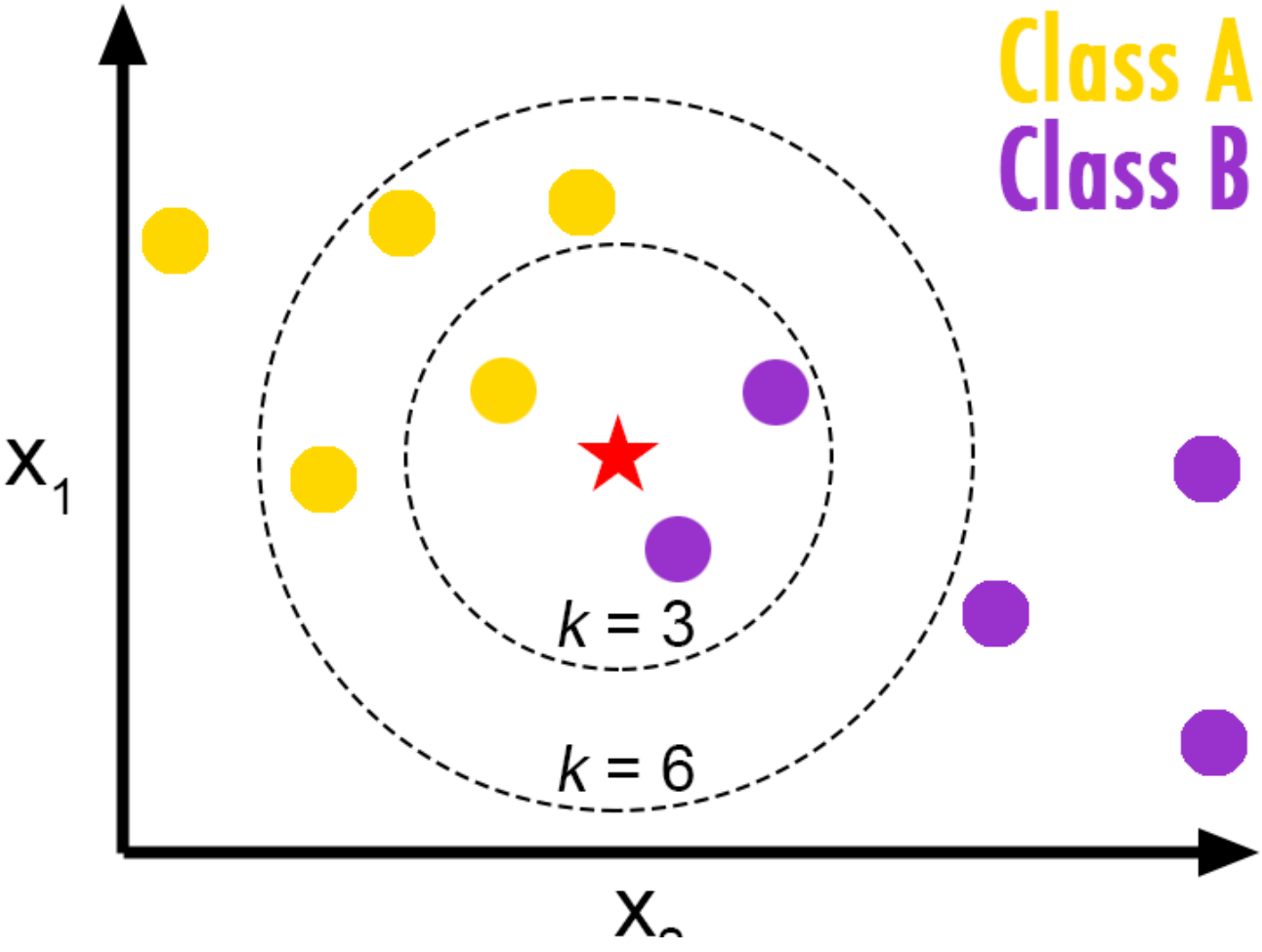
Objectives

After completing this lab you will be able to:

- **Use K Nearest neighbors to classify data**

K-Nearest Neighbors **is an algorithm for supervised learning. Where the data is 'trained' with data points corresponding to their classification. Once a point is to be predicted, it takes into account the 'K' nearest points to it to determine it's classification.**

Here's a visualization of the K-Nearest Neighbors algorithm.



In this case, we have data points of Class A and B. We want to predict what the star (test data point) is. If we consider a k value of 3 (3 nearest data points) we will obtain a prediction of Class B. Yet if we consider a k value of 6, we will obtain a prediction of Class A.

Table of contents

About the dataset

Imagine you are a teacher, teaching a class of 100 students and you have an important projects deadline coming up. You want to know which students will submit on time and which students will come asking for an exntension later.

From past data, do you think you will be able to predict so?

The example focuses on using student data, such as gender, age, and interest in subject, mean test score and if they are an Allen student or not.

The target field, called Assignment Submitted On Time , has two possible values that correspond to the two possibilities, as follows: 0- Not submitted on time 1- submitted on time

Our objective is to build a classifier, to predict the class of unknown cases. We will use a specific type of classification called K nearest neighbour.

Importing Required Libraries

In [1]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn import preprocessing
from sklearn.cluster import KMeans
```

Load Data from CSV File

In [2]:

```
df = pd.read_csv("Downloads/AiSubmissions.csv")
```

Data Visualization and Analysis

In [3]:

```
df.head()
```

Out[3]:

	Roll no	Gender	Partnered	Backlog	Interest	Experience	Allen Student	Mean Test Score	Assignment Submitted On Time
0	**1**	**Male**	**0**	**9**	**10**	**9**	**1**	**40**	**1**
1	**2**	**Male**	**0**	**1**	**7**	**5**	**0**	**32**	**1**
2	**3**	**Female**	**0**	**4**	**5**	**4**	**1**	**25**	**1**
3	**4**	**Female**	**1**	**6**	**4**	**5**	**0**	**26**	**1**
4	**5**	**Male**	**1**	**7**	**8**	**6**	**0**	**35**	**1**

In [4]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 50 entries, 0 to 49
Data columns (total 9 columns):
 #   Column                        Non-Null Count  Dtype
---  ------                        --------------  -----
 0   Roll no                       50 non-null     int64
 1   Gender                        50 non-null     object
 2   Partnered                     50 non-null     int64
 3   Backlog                       50 non-null     int64
 4   Interest                      50 non-null     int64
 5   Experience                    50 non-null     int64
 6   Allen Student                 50 non-null     int64
 7   Mean Test Score               50 non-null     int64
 8   Assignment Submitted On Time  50 non-null     int64
dtypes: int64(8), object(1)
memory usage: 3.6+ KB
```

In [5]:

```
df['Assignment Submitted On Time'].value_counts()
```

Out[5]:

```
0    30
1    20
Name: Assignment Submitted On Time, dtype: int64
```

In [6]:

```
sns.color_palette("Set1")
```

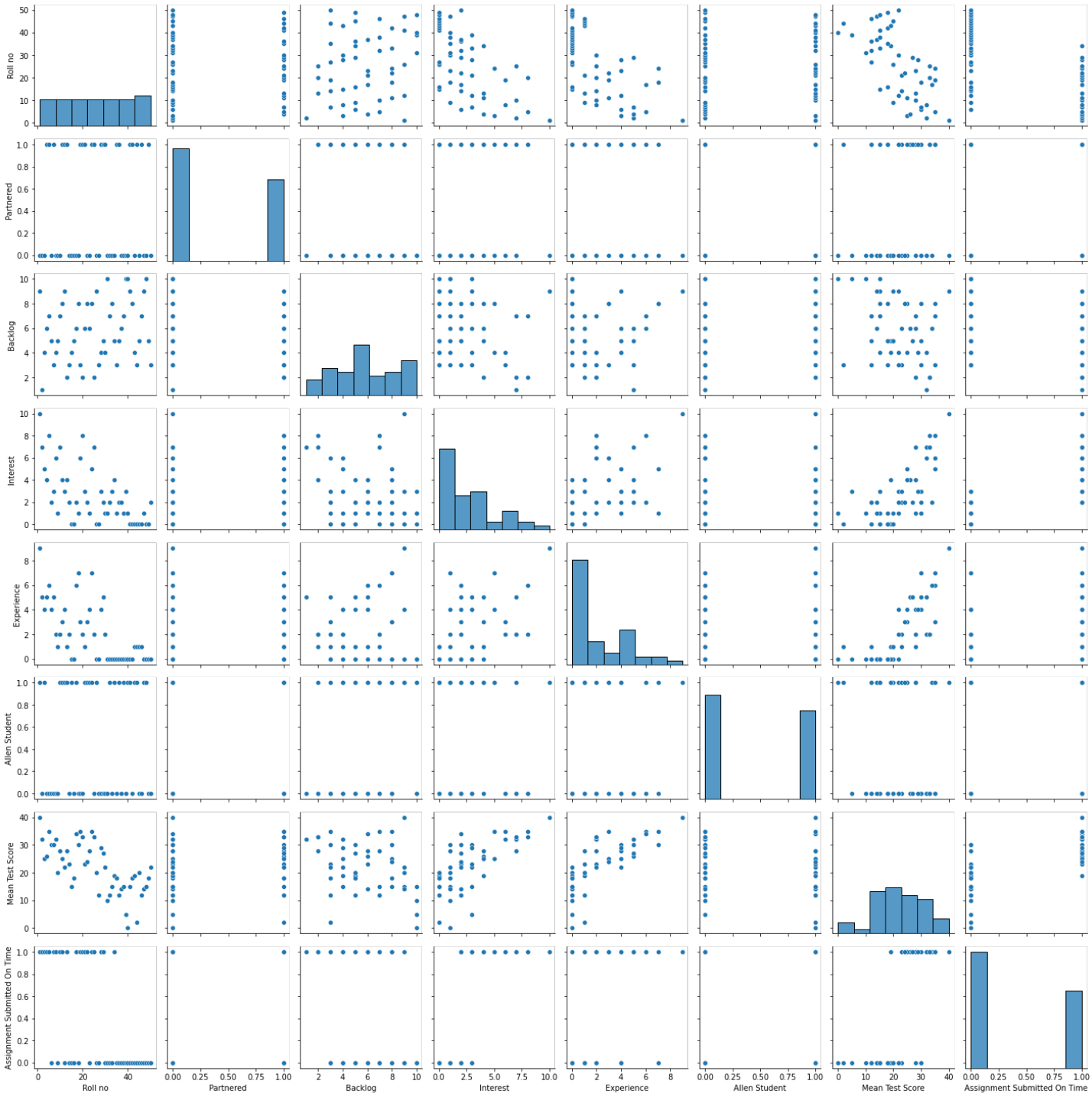
Out[6]:

In [7]:

```
plt.figure(figsize=(12,5), dpi=150)
sns.pairplot(df, palette='Set1')
```

Out[7]:

```
<seaborn.axisgrid.PairGrid at 0x7f78695ab460>

<Figure size 1800x750 with 0 Axes>
```



Understanding the correlation between features of dataset

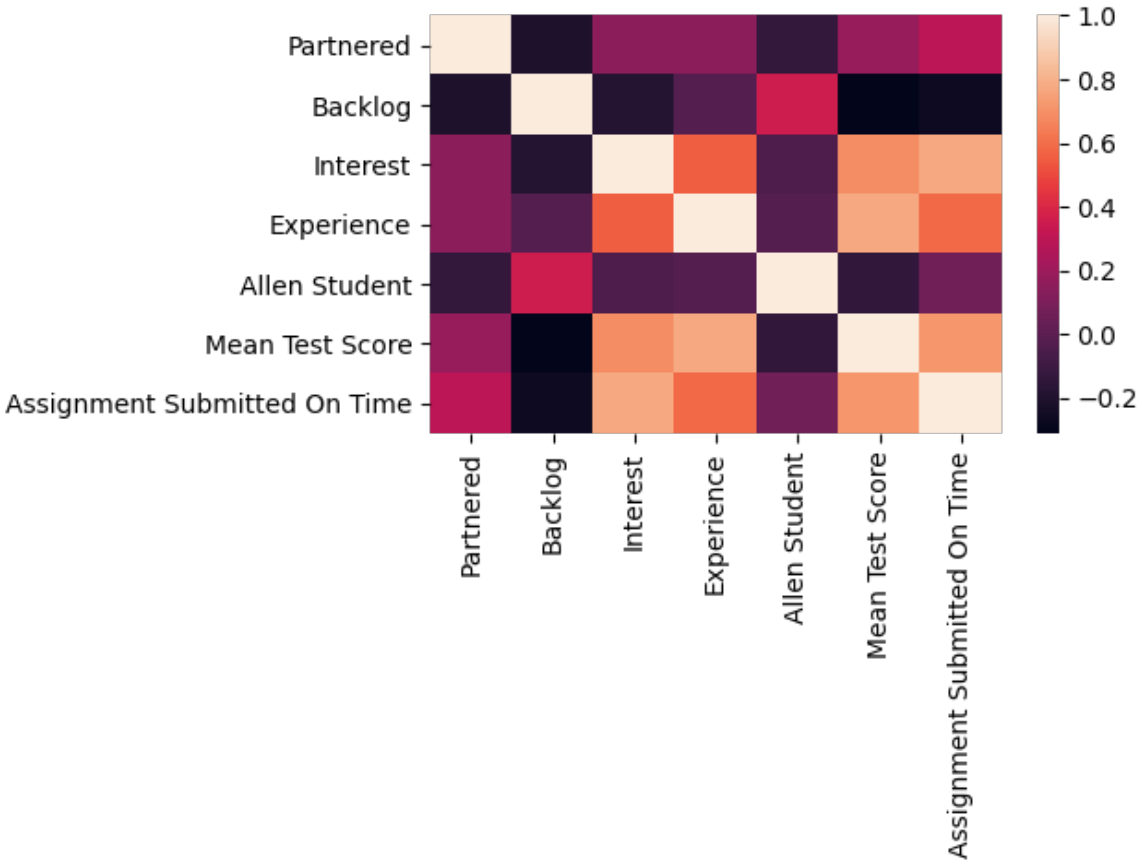
From the given data we can understand that "Interest in the subject". "Mean Test Score" and "Assignment Submitted On Time" are directly proportional

In [8]:

```
plt.figure(figsize=(5,3), dpi=100)

sns.heatmap(df.drop('Roll no', axis=1).corr())
```

Out[8]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f7869d96d90>
```



Defining Feature Set and Label

In [9]:

```
df.columns
```

Out[9]:

```
Index(['Roll no', 'Gender', 'Partnered', 'Backlog', 'Interest', 'Experience',
       'Allen Student', 'Mean Test Score', 'Assignment Submitted On Time'],
      dtype='object')
```

In [10]:

```
y = df['Assignment Submitted On Time'].values
```

In [11]:

```
y[:10]
```

Out[11]:

```
array([1, 1, 1, 1, 1, 0, 1, 1, 0, 1])
```

Encoding "Male" and "Female" into Labels to make our job easier

In [12]:

```
labelEncoder = preprocessing.LabelEncoder()
df['Gender'] = labelEncoder.fit_transform(df['Gender'])
```

In [13]:

```
df.head()
```

Out[13]:

	Roll	Gender	Partnered	Backlog	Interest	Experience	Allen Student	Mean Test Score	Assignment Submitted On Time

	Roll no	Gender	Partnered	Backlog	Interest	Experience	Allen Student	Mean Test Score	Assignment Submitted On Time
0	1	1	0	9	10	9	1	40	1
1	2	1	0	1	7	5	0	32	1
2	3	0	0	4	5	4	1	25	1
3	4	0	1	6	4	5	0	26	1
4	5	1	1	7	8	6	0	35	1

In [14]:

```
X = df.loc[ : , df.columns != 'Assignment Submitted On Time']
```

In [15]:

```
X = X.drop("Roll no", axis=1)
```

In [16]:

```
X.head()
```

Out[16]:

	Gender	Partnered	Backlog	Interest	Experience	Allen Student	Mean Test Score
0	1	0	9	10	9	1	40
1	1	0	1	7	5	0	32
2	0	0	4	5	4	1	25
3	0	1	6	4	5	0	26
4	1	1	7	8	6	0	35

Normalize Data

Data Standardization give data zero mean and unit variance, it is good practice, especially for algorithms such as KNN which is based on distance of cases:

In [17]:

```
X = preprocessing.StandardScaler().fit_transform(X)
X[:10]
```

Out[17]:

```
array([[ 0.92295821, -0.85096294,  1.29777137,  2.9160085 ,  2.94948193,
         1.08347268,  2.03386219],
       [ 0.92295821, -0.85096294, -1.94665705,  1.73384289,  1.24949235,
        -0.92295821,  1.14083818],
       [-1.08347268, -0.85096294, -0.7299964 ,  0.94573249,  0.82449495,
         1.08347268,  0.35944217],
       [-1.08347268,  1.1751393 ,  0.08111071,  0.55167728,  1.24949235,
        -0.92295821,  0.47107017],
       [ 0.92295821,  1.1751393 ,  0.48666426,  2.1278981 ,  1.67448974,
        -0.92295821,  1.47572218],
       [ 0.92295821, -0.85096294, -0.32444284, -0.23643312,  0.82449495,
        -0.92295821,  0.91758217],
       [-1.08347268,  1.1751393 , -1.13554995,  0.15762208,  1.24949235,
        -0.92295821,  0.91758217],
       [-1.08347268, -0.85096294, -0.7299964 ,  1.33978769, -0.02549984,
        -0.92295821,  1.14083818],
       [ 0.92295821, -0.85096294, -0.32444284, -0.63048832, -0.45049724,
        -0.92295821, -0.19869784],
       [-1.08347268, -0.85096294,  0.48666426,  1.73384289, -0.02549984,
         1.08347268,  0.69432617]])
```

Train Test Split

Out of Sample Accuracy is the percentage of correct predictions that the model makes on data that that the model has NOT been trained on. Doing a train and test on the same dataset will most likely have low out-of-sample accuracy, due to the likelihood of being over-fit.

It is important that our models have a high, out-of-sample accuracy, because the purpose of any model, of course, is to make correct predictions on unknown data.

So how can we improve out-of-sample accuracy? One way is to use an evaluation approach called Train/Test Split.

Train/Test Split involves splitting the dataset into training and testing sets respectively, which are mutually exclusive. After which, you train with the training set and test with the testing set.

This will provide a more accurate evaluation on out-of-sample accuracy because the testing dataset is not part of the dataset that have been used to train the data. It is more realistic for real world problems.

In [18]:

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=10
1)
```

In [19]:

```
print ('Train set:', X_train.shape,  y_train.shape)
print ('Test set:', X_test.shape,  y_test.shape)
```

```
Train set: (35, 7) (35,)
Test set: (15, 7) (15,)
```

Classification using KNN

Import library

Classifier implementing the k-nearest neighbors algorithm.

In [20]:

```
from sklearn.neighbors import KNeighborsClassifier
```

Training

Lets start the algorithm with k=3 for now:

In [21]:

```
k = 3
neigh = KNeighborsClassifier(n_neighbors=k,).fit(X_train, y_train)
```

In [22]:

```
neigh
```

Out[22]:

```
KNeighborsClassifier(n_neighbors=3)
```

Predicting

we can use the model to predict the test set:

In [23]:

```
yhat = neigh.predict(X_test)
```

In [24]:

```
yhat
```

Out[24]:

```
array([0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 1, 1, 1, 0, 1])
```

Accuracy evaluation

In multilabel classification, accuracy classification score is a function that computes subset accuracy. This function is equal to the jaccard_similarity_score function. Essentially, it calculates how closely the actual labels and predicted labels are matched in the test set.

In [25]:

```
from sklearn import metrics
print("Train set Accuracy: ", metrics.accuracy_score(y_train, neigh.predict(X_train)))
print("Test set Accuracy: ", metrics.accuracy_score(y_test, yhat))
```

```
Train set Accuracy:  0.9428571428571428
Test set Accuracy:  1.0
```

In [26]:

```
df['kmeans'] = neigh.predict(X)
```

In [27]:

```
z = neigh.predict(X)
print("Accuracy: ", metrics.accuracy_score(y, z))
```

```
Accuracy:  0.96
```

Experimenting with other 'k' values

In [28]:

```
Ks = 10
mean_acc = np.zeros((Ks-1))
std_acc = np.zeros((Ks-1))

for n in range(1,Ks):

    #Train Model and Predict
    neigh = KNeighborsClassifier(n_neighbors = n).fit(X_train,y_train)
    yhat=neigh.predict(X_test)
    mean_acc[n-1] = metrics.accuracy_score(y_test, yhat)


    std_acc[n-1]=np.std(yhat==y_test)/np.sqrt(yhat.shape[0])

mean_acc
```

Out[28]:

```
array([0.73333333, 0.8       , 1.        , 0.93333333, 1.        ,
       0.93333333, 0.93333333, 0.93333333, 0.93333333])
```

In [29]:

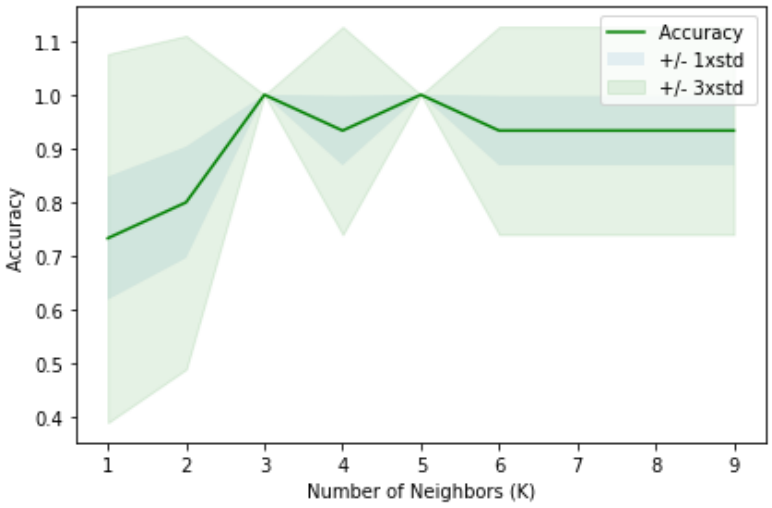
```
print( "The best accuracy was with", mean_acc.max(), "with k=", mean_acc.argmax()+1)
```

```
The best accuracy was with 1.0 with k= 3
```

Visualising Accuracy Vs 'k'

In [30]:

```
plt.plot(range(1,Ks),mean_acc,'g')
plt.fill_between(range(1,Ks),mean_acc - 1 * std_acc,mean_acc + 1 * std_acc, alpha=0.10)
plt.fill_between(range(1,Ks),mean_acc - 3 * std_acc,mean_acc + 3 * std_acc, alpha=0.10,c
olor="green")
plt.legend(('Accuracy ', '+/- 1xstd','+/- 3xstd'))
plt.ylabel('Accuracy ')
plt.xlabel('Number of Neighbors (K)')
plt.tight_layout()
plt.show()
```



'Elbow Point' Method

It might be a smart idea to sweep through the K values within a range and cluster the data points into K different groups every time. After each clustering is completed, we can check some metrics in order to decide whether we should choose the current K or continue evaluating.

One of these metrics is the total distance (it is called as “inertia” in sklearn library) . Inertia shows us the sum of distances to each cluster center. If the total distance is high, it means that the points are far from each other and might be less similar to each other. In this case we can choose to continue evaluating higher K values in order to see if we can reduce the total distance.

However, I should emphasize a really important point here. It’s not always the smartest idea to decrease the distance. Let’s assume that we have 100 data points. If we choose K to be 100, we will end up with a distance value which is equal to 0. But, obviously, it is not something that we wish. We want to have a few number of “good” clusters which contain sufficient information about the data points and do not have any noise or outliars.

Well, the point where we can feel ourselves okay is called “elbow point”. When you plot the Total Distance (Inertia) vs. K-Value graph, you will observe that after a point, total distance will start changing insignificantly compared to previous changes. At this point we can conclude that the data points are clustered sufficiently and further clustering will not contribute more information to our system. Thus, we can choose to stop here and

proceed with the K-value corresponding to elbow point.

In [31]:

```
def cluster_variance(n):
    variances=[]
    kmeans=[]
    outputs=[]
    K=[i for i in range(1,n+1)]
    for i in range(1,n+1):
        variance=0
        model=KMeans(n_clusters=i,random_state=82,verbose=2).fit(X)
        kmeans.append(model)
        variances.append(model.inertia_)

    return variances,K,n
variances,K,n=cluster_variance(10)
plt.plot(K,variances)
plt.ylabel("Inertia ( Total Distance )")
plt.xlabel("K Value")
plt.xticks([i for i in range(1,n+1)])
plt.show()
```

```
Initialization complete
Iteration 0, inertia 625.0391138410141.
Iteration 1, inertia 349.9999999999999.
Converged at iteration 1: strict convergence.
Initialization complete
Iteration 0, inertia 615.4129642726266.
Iteration 1, inertia 349.9999999999999.
Converged at iteration 1: strict convergence.
Initialization complete
Iteration 0, inertia 625.0391138410141.
Iteration 1, inertia 349.9999999999999.
Converged at iteration 1: strict convergence.
Initialization complete
Iteration 0, inertia 598.5874795827135.
Iteration 1, inertia 349.9999999999999.
Converged at iteration 1: strict convergence.
Initialization complete
Iteration 0, inertia 555.7237022224126.
Iteration 1, inertia 349.9999999999999.
Converged at iteration 1: strict convergence.
Initialization complete
Iteration 0, inertia 566.9190443174876.
Iteration 1, inertia 349.9999999999999.
Converged at iteration 1: strict convergence.
Initialization complete
Iteration 0, inertia 565.7939027429976.
Iteration 1, inertia 349.9999999999999.
Converged at iteration 1: strict convergence.
Initialization complete
Iteration 0, inertia 600.0496512095633.
Iteration 1, inertia 349.9999999999999.
Converged at iteration 1: strict convergence.
Initialization complete
Iteration 0, inertia 651.9121198626652.
Iteration 1, inertia 349.9999999999999.
Converged at iteration 1: strict convergence.
Initialization complete
Iteration 0, inertia 683.720693507795.
Iteration 1, inertia 349.9999999999999.
Converged at iteration 1: strict convergence.
Initialization complete
Iteration 0, inertia 418.4506267453929
Iteration 1, inertia 259.6463919575787
Iteration 2, inertia 254.35231486334263
Iteration 3, inertia 253.70065365092526
Converged at iteration 3: strict convergence.
Initialization complete
Iteration 0, inertia 392.40248564026024
Iteration 1, inertia 297.2428798472551
```

```
Iteration 2, inertia 287.83124697147247
Iteration 3, inertia 285.56369138972366
Converged at iteration 3: strict convergence.
Initialization complete
Iteration 0, inertia 474.98036011351127
Iteration 1, inertia 284.0044914534646
Iteration 2, inertia 257.9832974452161
Iteration 3, inertia 254.72374758309974
Iteration 4, inertia 254.22486154253386
Converged at iteration 4: strict convergence.
Initialization complete
Iteration 0, inertia 481.8300194747195
Iteration 1, inertia 285.24341614284896
Converged at iteration 1: strict convergence.
Initialization complete
Iteration 0, inertia 465.8480709757806
Iteration 1, inertia 265.8566522767461
Iteration 2, inertia 253.52037162314969
Converged at iteration 2: strict convergence.
Initialization complete
Iteration 0, inertia 500.38854444598934
Iteration 1, inertia 273.24291920888464
Iteration 2, inertia 267.0951072527633
Iteration 3, inertia 258.7315882272298
Iteration 4, inertia 254.26594550073807
Converged at iteration 4: strict convergence.
Initialization complete
Iteration 0, inertia 414.39960990094477
Iteration 1, inertia 266.4828008312779
Iteration 2, inertia 256.61108721092245
Iteration 3, inertia 254.0135335983399
Iteration 4, inertia 253.52037162314969
Converged at iteration 4: strict convergence.
Initialization complete
Iteration 0, inertia 459.11450799357056
Iteration 1, inertia 264.36453748945297
Iteration 2, inertia 258.28169939132533
Iteration 3, inertia 254.8284159330366
Iteration 4, inertia 254.26594550073807
Converged at iteration 4: strict convergence.
Initialization complete
Iteration 0, inertia 459.08778550735525
Iteration 1, inertia 287.77519242820836
Iteration 2, inertia 259.56077897068644
Iteration 3, inertia 253.8972494306397
Converged at iteration 3: strict convergence.
Initialization complete
Iteration 0, inertia 609.4910506171772
Iteration 1, inertia 296.8524788965992
Iteration 2, inertia 292.8467701340225
Iteration 3, inertia 289.1098604917254
Iteration 4, inertia 286.9236968840029
Iteration 5, inertia 286.0057545019564
Iteration 6, inertia 285.55530591804524
Converged at iteration 6: strict convergence.
Initialization complete
Iteration 0, inertia 333.79422886641134
Iteration 1, inertia 222.96604959573668
Iteration 2, inertia 219.8701250780341
Iteration 3, inertia 219.27621030335538
Converged at iteration 3: strict convergence.
Initialization complete
Iteration 0, inertia 326.798436324039
Iteration 1, inertia 236.50909779175882
Iteration 2, inertia 221.54521388277684
Iteration 3, inertia 218.64081901690992
Converged at iteration 3: strict convergence.
Initialization complete
Iteration 0, inertia 407.5914133492352
Iteration 1, inertia 238.8799442332675
Iteration 2, inertia 228.89593807893024
Iteration 3, inertia 226.066467337046
```

```
Iteration 4, inertia 225.30940498972308
Iteration 5, inertia 224.5218378352823
Iteration 6, inertia 223.10907441105448
Iteration 7, inertia 221.86623742677338
Iteration 8, inertia 216.07330818012477
Iteration 9, inertia 214.02719559908186
Converged at iteration 9: strict convergence.
Initialization complete
Iteration 0, inertia 340.93843259773183
Iteration 1, inertia 233.70149994234325
Iteration 2, inertia 220.53893257310733
Iteration 3, inertia 217.9730580569897
Iteration 4, inertia 216.08692709733958
Iteration 5, inertia 215.4245914380848
Converged at iteration 5: strict convergence.
Initialization complete
Iteration 0, inertia 400.79041641530654
Iteration 1, inertia 258.10174752530907
Iteration 2, inertia 249.80189261076066
Iteration 3, inertia 244.28647329130632
Iteration 4, inertia 236.66816441146116
Iteration 5, inertia 230.0047294573878
Iteration 6, inertia 222.53019889614345
Iteration 7, inertia 219.59042734150677
Iteration 8, inertia 218.84146339541832
Converged at iteration 8: strict convergence.
Initialization complete
Iteration 0, inertia 391.5919045390358
Iteration 1, inertia 244.39669554487952
Iteration 2, inertia 233.01906916716905
Iteration 3, inertia 231.74604610015268
Iteration 4, inertia 230.33068383270935
Converged at iteration 4: strict convergence.
Initialization complete
Iteration 0, inertia 342.84967335986886
Iteration 1, inertia 248.64930589272856
Iteration 2, inertia 235.47621247579116
Iteration 3, inertia 230.81490126922196
Converged at iteration 3: strict convergence.
Initialization complete
Iteration 0, inertia 367.7777608076406
Iteration 1, inertia 254.43094014437094
Iteration 2, inertia 227.62741609863707
Iteration 3, inertia 224.17609998095045
Iteration 4, inertia 223.52989550558715
Iteration 5, inertia 222.79197805719716
Iteration 6, inertia 222.11020593041977
Iteration 7, inertia 221.46589434833388
Iteration 8, inertia 220.66342907288694
Converged at iteration 8: strict convergence.
Initialization complete
Iteration 0, inertia 357.5963107642622
Iteration 1, inertia 247.1021399191044
Iteration 2, inertia 239.68114301901562
Iteration 3, inertia 227.13266583509903
Iteration 4, inertia 220.5473408960513
Iteration 5, inertia 219.59042734150677
Iteration 6, inertia 218.84146339541832
Converged at iteration 6: strict convergence.
Initialization complete
Iteration 0, inertia 438.7299547419944
Iteration 1, inertia 232.98222883709846
Iteration 2, inertia 227.41824003573052
Iteration 3, inertia 226.8100280021204
Converged at iteration 3: strict convergence.
Initialization complete
Iteration 0, inertia 297.2882131769322
Iteration 1, inertia 201.65390143894052
Iteration 2, inertia 196.10650264832384
Iteration 3, inertia 192.51668307589256
Iteration 4, inertia 189.3539677224021
Iteration 5, inertia 188.0049410824471
```

```
Converged at iteration 5: strict convergence.
Initialization complete
Iteration 0, inertia 321.7091331825229
Iteration 1, inertia 197.79997519616185
Iteration 2, inertia 188.0074692092698
Iteration 3, inertia 186.96169727219097
Converged at iteration 3: strict convergence.
Initialization complete
Iteration 0, inertia 325.9382654878208
Iteration 1, inertia 196.27984100590598
Iteration 2, inertia 194.78409508816873
Converged at iteration 2: strict convergence.
Initialization complete
Iteration 0, inertia 279.7537816674906
Iteration 1, inertia 184.811686068227
Iteration 2, inertia 183.1784952535614
Converged at iteration 2: strict convergence.
Initialization complete
Iteration 0, inertia 277.9145845792228
Iteration 1, inertia 212.63125549439914
Iteration 2, inertia 207.5219673891437
Iteration 3, inertia 205.1666974452497
Iteration 4, inertia 203.11653763647914
Converged at iteration 4: strict convergence.
Initialization complete
Iteration 0, inertia 293.20966052017445
Iteration 1, inertia 211.76139566047596
Iteration 2, inertia 206.27869596945092
Converged at iteration 2: strict convergence.
Initialization complete
Iteration 0, inertia 335.49080501438465
Iteration 1, inertia 231.71807904159473
Iteration 2, inertia 201.06237636313668
Iteration 3, inertia 195.63447542547348
Iteration 4, inertia 194.01285133791717
Iteration 5, inertia 193.0116423933738
Converged at iteration 5: strict convergence.
Initialization complete
Iteration 0, inertia 329.48604552764704
Iteration 1, inertia 204.70459165183385
Iteration 2, inertia 198.98344608312755
Iteration 3, inertia 197.72586388565782
Iteration 4, inertia 196.8452470267709
Converged at iteration 4: strict convergence.
Initialization complete
Iteration 0, inertia 336.23769634958074
Iteration 1, inertia 204.80587262503792
Iteration 2, inertia 196.38304594114237
Iteration 3, inertia 193.3079135558365
Iteration 4, inertia 191.37043028707404
Iteration 5, inertia 190.66290512994073
Converged at iteration 5: strict convergence.
Initialization complete
Iteration 0, inertia 301.1720046770292
Iteration 1, inertia 196.65633459455844
Iteration 2, inertia 190.04311350708608
Iteration 3, inertia 188.3221637981468
Converged at iteration 3: strict convergence.
Initialization complete
Iteration 0, inertia 267.1553746363496
Iteration 1, inertia 185.70125878001815
Iteration 2, inertia 177.2537059057514
Iteration 3, inertia 175.19619286994225
Iteration 4, inertia 171.95899468760848
Iteration 5, inertia 170.90162606424948
Converged at iteration 5: strict convergence.
Initialization complete
Iteration 0, inertia 262.3818816141696
Iteration 1, inertia 192.7928552538736
Iteration 2, inertia 190.8447921763318
Iteration 3, inertia 190.00551815851844
Iteration 4, inertia 188.4882619775712
```

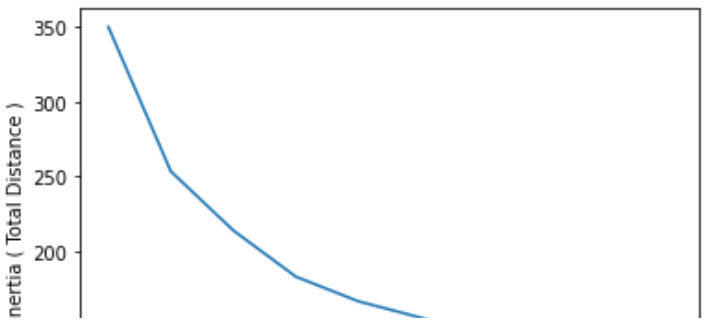
```
Iteration 5, inertia 184.37320925456885
Iteration 6, inertia 182.34865327025767
Converged at iteration 6: strict convergence.
Initialization complete
Iteration 0, inertia 266.4190207558643
Iteration 1, inertia 186.45089583341152
Iteration 2, inertia 180.55384554745177
Iteration 3, inertia 173.92040546957338
Converged at iteration 3: strict convergence.
Initialization complete
Iteration 0, inertia 287.21503206752425
Iteration 1, inertia 203.29218990569814
Iteration 2, inertia 196.39339585812004
Iteration 3, inertia 195.47200488244206
Converged at iteration 3: strict convergence.
Initialization complete
Iteration 0, inertia 261.8835964430159
Iteration 1, inertia 182.8767930715748
Iteration 2, inertia 171.9124415258616
Iteration 3, inertia 170.04815520382007
Iteration 4, inertia 168.68703780739256
Iteration 5, inertia 167.88623752168377
Converged at iteration 5: strict convergence.
Initialization complete
Iteration 0, inertia 256.5429437988528
Iteration 1, inertia 181.70256964403353
Iteration 2, inertia 173.41269959304014
Iteration 3, inertia 168.2539961275117
Iteration 4, inertia 166.70928605026316
Converged at iteration 4: strict convergence.
Initialization complete
Iteration 0, inertia 281.5810695108498
Iteration 1, inertia 196.25851571485205
Iteration 2, inertia 193.1068081394385
Iteration 3, inertia 187.73700141380732
Iteration 4, inertia 181.75834100929066
Iteration 5, inertia 179.9547471527436
Iteration 6, inertia 178.763352483852
Converged at iteration 6: strict convergence.
Initialization complete
Iteration 0, inertia 259.5498814847192
Iteration 1, inertia 189.29680704172875
Iteration 2, inertia 174.77301870271327
Iteration 3, inertia 169.12142628961058
Converged at iteration 3: strict convergence.
Initialization complete
Iteration 0, inertia 273.45347839097576
Iteration 1, inertia 184.23038879115293
Iteration 2, inertia 179.30470039651794
Iteration 3, inertia 177.68794524767387
Iteration 4, inertia 176.2307871138918
Converged at iteration 4: strict convergence.
Initialization complete
Iteration 0, inertia 270.1121738679323
Iteration 1, inertia 193.11201371378635
Iteration 2, inertia 177.12140394670467
Iteration 3, inertia 169.23528927092428
Iteration 4, inertia 167.96459391212707
Converged at iteration 4: strict convergence.
Initialization complete
Iteration 0, inertia 246.25350719096036
Iteration 1, inertia 177.24852357886843
Iteration 2, inertia 164.84736504621708
Iteration 3, inertia 161.95372259102848
Iteration 4, inertia 160.61372622470796
Iteration 5, inertia 159.38826454619036
Converged at iteration 5: strict convergence.
Initialization complete
Iteration 0, inertia 249.58951135166052
Iteration 1, inertia 176.97052721327188
Iteration 2, inertia 168.6283368267229
Iteration 3, inertia 163.72798853339035
```

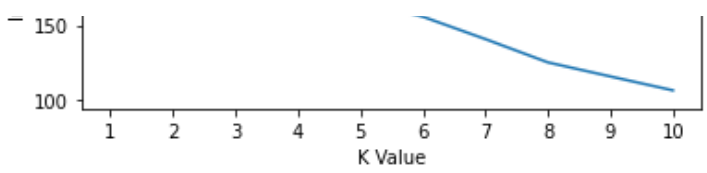
```
Iteration 4, inertia 161.5967192056607
Iteration 5, inertia 160.4979334817324
Iteration 6, inertia 158.79712738284218
Converged at iteration 6: strict convergence.
Initialization complete
Iteration 0, inertia 251.39384514581582
Iteration 1, inertia 175.29651992616462
Iteration 2, inertia 164.5168281610261
Iteration 3, inertia 157.43444559246987
Converged at iteration 3: strict convergence.
Initialization complete
Iteration 0, inertia 247.40661191705766
Iteration 1, inertia 174.1290614303776
Iteration 2, inertia 166.52269351095995
Iteration 3, inertia 165.5608570944946
Converged at iteration 3: strict convergence.
Initialization complete
Iteration 0, inertia 256.2931462985183
Iteration 1, inertia 174.26820547804405
Iteration 2, inertia 158.30139338067184
Iteration 3, inertia 157.28355630042373
Converged at iteration 3: strict convergence.
Initialization complete
Iteration 0, inertia 216.32961296748752
Iteration 1, inertia 165.05658189643788
Iteration 2, inertia 161.24670290729176
Converged at iteration 2: strict convergence.
Initialization complete
Iteration 0, inertia 227.96073719670076
Iteration 1, inertia 164.40067282077607
Iteration 2, inertia 162.58705279702605
Iteration 3, inertia 160.45995641587234
Iteration 4, inertia 158.80403267871876
Converged at iteration 4: strict convergence.
Initialization complete
Iteration 0, inertia 259.8687457948806
Iteration 1, inertia 172.75723619920282
Iteration 2, inertia 160.02558160684237
Iteration 3, inertia 155.77258576102764
Converged at iteration 3: strict convergence.
Initialization complete
Iteration 0, inertia 226.8756910109856
Iteration 1, inertia 168.01343817964278
Iteration 2, inertia 162.83010084157362
Converged at iteration 2: strict convergence.
Initialization complete
Iteration 0, inertia 236.1453760888487
Iteration 1, inertia 168.18746371061704
Iteration 2, inertia 162.52325196092286
Iteration 3, inertia 161.4471490836097
Converged at iteration 3: strict convergence.
Initialization complete
Iteration 0, inertia 222.43299251526236
Iteration 1, inertia 159.09899000550175
Iteration 2, inertia 154.4499000824817
Iteration 3, inertia 153.2683937410967
Converged at iteration 3: strict convergence.
Initialization complete
Iteration 0, inertia 204.99558918083278
Iteration 1, inertia 149.6085548812723
Iteration 2, inertia 146.06249918885044
Converged at iteration 2: strict convergence.
Initialization complete
Iteration 0, inertia 188.2097225474321
Iteration 1, inertia 148.78981214662744
Iteration 2, inertia 144.60226139855715
Converged at iteration 2: strict convergence.
Initialization complete
Iteration 0, inertia 225.24774773550124
Iteration 1, inertia 159.98919531145833
Iteration 2, inertia 153.4113281532158
Iteration 3, inertia 151.41743920791748
```

```
Iteration 4, inertia 145.82592288574435
Converged at iteration 4: strict convergence.
Initialization complete
Iteration 0, inertia 222.28295247602867
Iteration 1, inertia 149.41433911149727
Iteration 2, inertia 143.254689104873
Iteration 3, inertia 142.05492924217785
Converged at iteration 3: strict convergence.
Initialization complete
Iteration 0, inertia 217.2984113336426
Iteration 1, inertia 151.3198994724002
Iteration 2, inertia 149.68258141366255
Iteration 3, inertia 147.26716859742993
Iteration 4, inertia 145.95501639089673
Converged at iteration 4: strict convergence.
Initialization complete
Iteration 0, inertia 227.6849392180198
Iteration 1, inertia 159.6604514274703
Iteration 2, inertia 149.94651200156127
Iteration 3, inertia 148.5994023010137
Iteration 4, inertia 147.18855969373757
Iteration 5, inertia 145.68334682470856
Iteration 6, inertia 141.9930202001154
Iteration 7, inertia 140.80207778324385
Converged at iteration 7: strict convergence.
Initialization complete
Iteration 0, inertia 214.08110261555885
Iteration 1, inertia 151.67753994725444
Iteration 2, inertia 148.35426401947305
Iteration 3, inertia 146.54346708313938
Converged at iteration 3: strict convergence.
Initialization complete
Iteration 0, inertia 216.13394820792652
Iteration 1, inertia 156.76498828658075
Iteration 2, inertia 151.16766457375488
Iteration 3, inertia 149.34700535220057
Iteration 4, inertia 146.70627132412855
Converged at iteration 4: strict convergence.
Initialization complete
Iteration 0, inertia 226.6168006749088
Iteration 1, inertia 159.4560018097475
Iteration 2, inertia 154.67385445599902
Iteration 3, inertia 149.8038322006136
Iteration 4, inertia 148.17832494186797
Converged at iteration 4: strict convergence.
Initialization complete
Iteration 0, inertia 206.82603712215823
Iteration 1, inertia 139.4881852347726
Iteration 2, inertia 135.724971875088
Iteration 3, inertia 134.93576766782655
Converged at iteration 3: strict convergence.
Initialization complete
Iteration 0, inertia 207.34406857923085
Iteration 1, inertia 149.3209401926673
Iteration 2, inertia 145.27661136726095
Converged at iteration 2: strict convergence.
Initialization complete
Iteration 0, inertia 189.18552591838016
Iteration 1, inertia 133.807937638494
Iteration 2, inertia 130.96937665986022
Converged at iteration 2: strict convergence.
Initialization complete
Iteration 0, inertia 197.44624571312028
Iteration 1, inertia 141.31162700190217
Iteration 2, inertia 138.46163076508356
Converged at iteration 2: strict convergence.
Initialization complete
Iteration 0, inertia 203.1851696114035
Iteration 1, inertia 137.15904280168283
Iteration 2, inertia 134.40840892948344
Converged at iteration 2: strict convergence.
Initialization complete
```

```
Iteration 0, inertia 190.4335657260305
Iteration 1, inertia 139.36360954408278
Iteration 2, inertia 135.88282930255232
Iteration 3, inertia 134.47878768808255
Converged at iteration 3: strict convergence.
Initialization complete
Iteration 0, inertia 175.17062394904937
Iteration 1, inertia 138.22273321190454
Iteration 2, inertia 134.7664345407111
Iteration 3, inertia 133.46885505531182
Converged at iteration 3: strict convergence.
Initialization complete
Iteration 0, inertia 186.08975720490716
Iteration 1, inertia 139.28918370254527
Iteration 2, inertia 133.47279608640443
Converged at iteration 2: strict convergence.
Initialization complete
Iteration 0, inertia 188.2319502272804
Iteration 1, inertia 147.4281019349446
Iteration 2, inertia 142.91506331276017
Iteration 3, inertia 139.36724772547376
Converged at iteration 3: strict convergence.
Initialization complete
Iteration 0, inertia 198.3862455363162
Iteration 1, inertia 143.33906550866416
Iteration 2, inertia 131.86358362366468
Iteration 3, inertia 125.24614521142001
Converged at iteration 3: strict convergence.
Initialization complete
Iteration 0, inertia 178.87535190705506
Iteration 1, inertia 126.1495591115203
Iteration 2, inertia 122.25180773293962
Iteration 3, inertia 121.16263601772995
Converged at iteration 3: strict convergence.
Initialization complete
Iteration 0, inertia 184.33521937491528
Iteration 1, inertia 129.68290575219166
Iteration 2, inertia 126.21705798518326
Converged at iteration 2: strict convergence.
Initialization complete
Iteration 0, inertia 177.0472884913679
Iteration 1, inertia 134.26701475216998
Iteration 2, inertia 129.32109494217653
Iteration 3, inertia 127.11381161328669
Converged at iteration 3: strict convergence.
Initialization complete
Iteration 0, inertia 173.11598834067084
Iteration 1, inertia 122.00906352476522
Iteration 2, inertia 115.84693608759675
Converged at iteration 2: strict convergence.
Initialization complete
Iteration 0, inertia 177.86507699495795
Iteration 1, inertia 125.06707156892531
Iteration 2, inertia 119.95007931044444
Converged at iteration 2: strict convergence.
Initialization complete
Iteration 0, inertia 175.66110145644302
Iteration 1, inertia 126.19650810574714
Iteration 2, inertia 120.51668740687202
Converged at iteration 2: strict convergence.
Initialization complete
Iteration 0, inertia 184.06362347686772
Iteration 1, inertia 131.51137158992148
Converged at iteration 1: strict convergence.
Initialization complete
Iteration 0, inertia 185.83394332915606
Iteration 1, inertia 125.92715687899396
Iteration 2, inertia 123.24813824985726
Iteration 3, inertia 122.24065385025663
Converged at iteration 3: strict convergence.
Initialization complete
Iteration 0, inertia 188.74813072729683
```

```
Iteration 1, inertia 141.24478956506056
Iteration 2, inertia 140.46037631925117
Converged at iteration 2: strict convergence.
Initialization complete
Iteration 0, inertia 165.2627094667331
Iteration 1, inertia 118.80343335261017
Converged at iteration 1: strict convergence.
Initialization complete
Iteration 0, inertia 170.51275630336147
Iteration 1, inertia 120.76733574766358
Iteration 2, inertia 116.86958436908287
Iteration 3, inertia 115.7804126538732
Converged at iteration 3: strict convergence.
Initialization complete
Iteration 0, inertia 159.84639270361876
Iteration 1, inertia 118.20361595215073
Iteration 2, inertia 117.21041096964689
Converged at iteration 2: strict convergence.
Initialization complete
Iteration 0, inertia 167.08614852431822
Iteration 1, inertia 122.52129438128205
Iteration 2, inertia 120.54455901877738
Iteration 3, inertia 118.50109904466633
Converged at iteration 3: strict convergence.
Initialization complete
Iteration 0, inertia 151.8203038997543
Iteration 1, inertia 121.18708263720936
Iteration 2, inertia 116.9765598471621
Converged at iteration 2: strict convergence.
Initialization complete
Iteration 0, inertia 179.97393325905415
Iteration 1, inertia 119.87775887193898
Iteration 2, inertia 110.01608925281887
Iteration 3, inertia 109.13331715459262
Converged at iteration 3: strict convergence.
Initialization complete
Iteration 0, inertia 179.68513884664085
Iteration 1, inertia 120.11270276446781
Iteration 2, inertia 114.12683693140806
Converged at iteration 2: strict convergence.
Initialization complete
Iteration 0, inertia 157.8105187548865
Iteration 1, inertia 113.63653750993582
Iteration 2, inertia 112.30472885220388
Converged at iteration 2: strict convergence.
Initialization complete
Iteration 0, inertia 159.73749604443893
Iteration 1, inertia 115.13708447231178
Iteration 2, inertia 110.06950384800852
Iteration 3, inertia 106.56844759764452
Converged at iteration 3: strict convergence.
Initialization complete
Iteration 0, inertia 161.96603380537658
Iteration 1, inertia 120.54504090574532
Iteration 2, inertia 114.47206624373821
Converged at iteration 2: strict convergence.
Initialization complete
Iteration 0, inertia 162.2649673047038
Iteration 1, inertia 112.198015854853
Iteration 2, inertia 109.21685341873946
Converged at iteration 2: strict convergence.
```





As you can see in the figure above, after K=3 , reduction in total distance changes more slowly and the slope of the line significantly decreased. From this figure, we can infer that going beyond this K value(3) will not contribute much to our clustering algorithm and will only make our clusters more complicated.

Predicting Submission Based on your Values :)

In [32]:

```
features = {'Gender':0, 'partnered':0, 'backlog':0, 'interest':0, 'experience':0, 'allen
student':0, 'meantestscore':0}

print("Welcome! If you'd like to find out if your student will submit your assignment on
time, please use me.")
print("Keep in mind, the values are like this")
print("Gender ---> 0:Female, 1:Male \n"
      "Partnered ---> 0:No, 1;Yes \n"
      "Backlog ----> Any value from 0-10 \n"
      "Interest ----> Any value from 0-10 \n"
      "Experience ----> Any value from 0-10 \n"
      "Allen Student ----> 0:No, 1;Yes \n"
      "Mean Test Score ---> Any value from 0-40")

for feature, value in features.items():
    features.update({feature: int(input(f"Enter Value for {feature}: "))})
_df = pd.DataFrame(features, index=range(0, 1))

submit = neigh.predict(_df)
_df['Assignment Submmitted'] = submit

if(submit == 1):
    print("Worry not! The given student will submit project on time :D")
else:
    print("I'm sorry! The given student will NOT submit project on time :/")
```

```
Welcome! If you'd like to find out if your student will submit your assignment on time, p
lease use me.
Keep in mind, the values are like this
Gender ---> 0:Female, 1:Male
Partnered ---> 0:No, 1;Yes
Backlog ----> Any value from 0-10
Interest ----> Any value from 0-10
Experience ----> Any value from 0-10
Allen Student ----> 0:No, 1;Yes
Mean Test Score ---> Any value from 0-40
Enter Value for Gender: 1
Enter Value for partnered: 0
Enter Value for backlog: 7
Enter Value for interest: 7
Enter Value for experience: 2
Enter Value for allenstudent: 1
Enter Value for meantestscore: 36
Worry not! The given student will submit project on time :D
```

In [33]:

```
_df
```

Out[33]:

	Gender	partnered	backlog	interest	experience	allenstudent	meantestscore	Assignment Submmitted
0	1	0	7	7	2	1	36	1

Repeating the same process but with generating random data under same features

In [34]:

```
mydf = pd.read_csv("Downloads/ai1.csv")
```

In [35]:

```
labelEncoder = preprocessing.LabelEncoder()
mydf['Gender'] = labelEncoder.fit_transform(mydf['Gender'])
```

In [36]:

```
for i in mydf.columns:
    mydf[i]
```

In [37]:

```
def simulate_df(df=df, size_of_simulated_df=2000):
    return df.sample(size_of_simulated_df, replace=True).reset_index(drop=True)

dd = simulate_df(mydf)
```

In [38]:

```
dd.head()
```

Out[38]:

	Roll no	Gender	Partnered	Backlog	Interest	Experience	Allen Student	Mean Test Score	Assignment Submitted On Time
0	90	1	1	8	3	4	0	90	1
1	34	1	0	3	1	3	0	34	0
2	11	0	1	8	4	3	1	11	1
3	71	0	0	8	6	1	0	71	1
4	71	0	0	8	6	1	0	71	1

In [39]:

```
dd = dd.drop("Roll no", axis=1)
```

In [40]:

```
dd.shape
```

Out[40]:

```
(2000, 8)
```

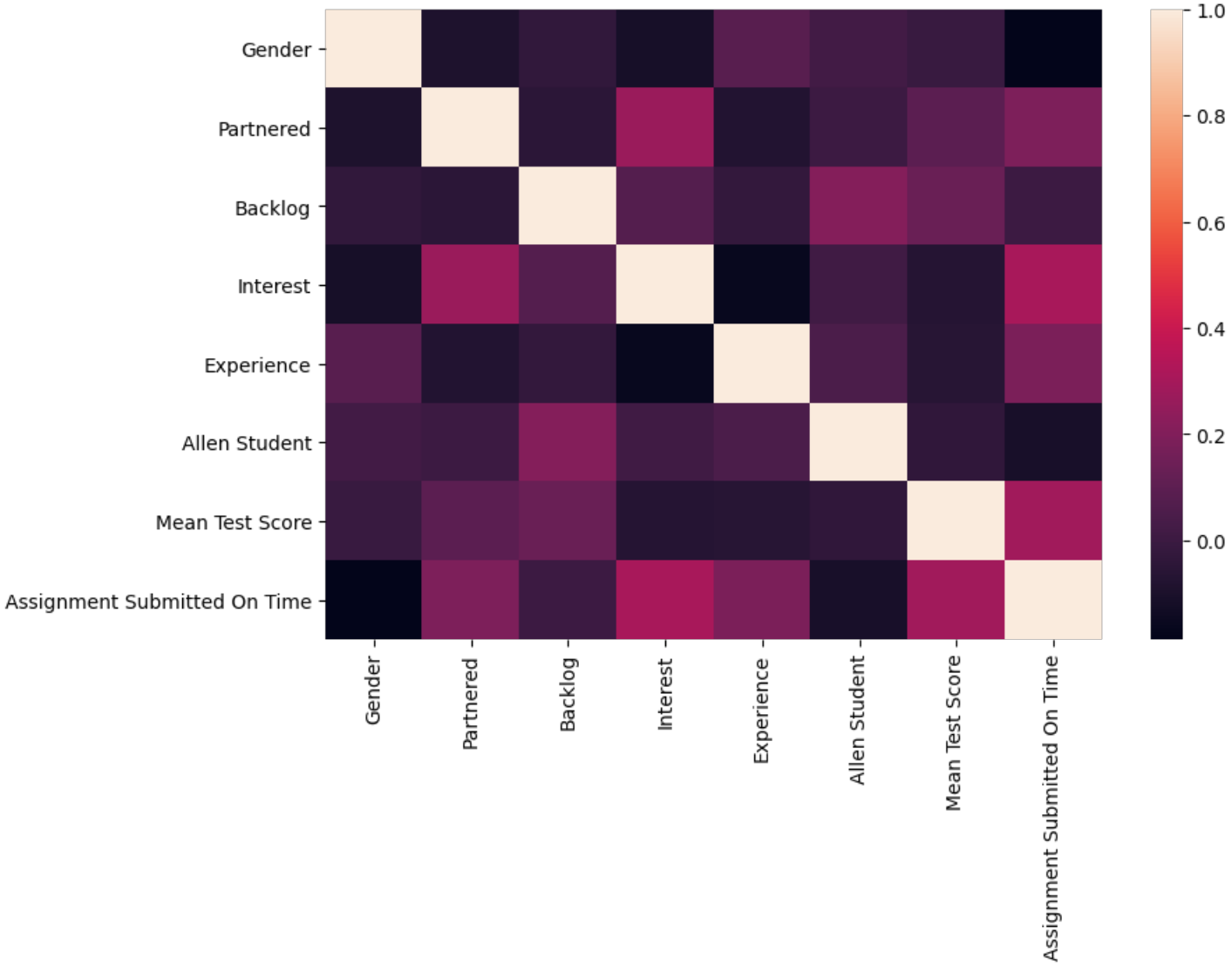
In [41]:

```
plt.figure(figsize=(9,6), dpi=100)
```

```
sns.heatmap(dd.corr())
```

Out[41]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f786a6794f0>
```



In [42]:

```
X1 = dd.loc[ : , dd.columns != 'Assignment Submitted On Time']
```

In [43]:

```
X1 = preprocessing.StandardScaler().fit_transform(X1)
y1 = dd['Assignment Submitted On Time'].values
```

In [44]:

```
X1.shape
```

Out[44]:

```
(2000, 7)
```

In [45]:

```
from sklearn.model_selection import train_test_split
X1_train, X1_test, y1_train, y1_test = train_test_split(X1, y1, test_size=0.3, random_st
ate=101)
```

In [46]:

```
print ('Train set:', X1_train.shape,  y1_train.shape)
print ('Test set:', X1_test.shape,  y1_test.shape)
```

```
Train set: (1400, 7) (1400,)
Test set: (600, 7) (600,)
```

In [47]:

```
k = 3
neigh1 = KNeighborsClassifier(n_neighbors=k,).fit(X1_train, y1_train)
```

In [48]:

```
yhat1 = neigh1.predict(X1_test)
```

In [49]:

```
from sklearn import metrics
print("Train set Accuracy: ", metrics.accuracy_score(y1_train, neigh1.predict(X1_train)))
print("Test set Accuracy: ", metrics.accuracy_score(y1_test, yhat1))
```

```
Train set Accuracy:  1.0
Test set Accuracy:  1.0
```

In [50]:

```
dd['kmeans'] = neigh1.predict(X1)
```

In [51]:

```
dd.head(20)
```

Out[51]:

	Gender	Partnered	Backlog	Interest	Experience	Allen Student	Mean Test Score	Assignment Submitted On Time	kmeans
0	1	1	8	3	4	0	90	1	1
1	1	0	3	1	3	0	34	0	0
2	0	1	8	4	3	1	11	1	1
3	0	0	8	6	1	0	71	1	1
4	0	0	8	6	1	0	71	1	1
5	1	0	4	2	7	1	84	1	1
6	1	0	3	1	3	0	34	0	0
7	0	0	9	5	3	1	26	1	1
8	0	0	4	4	1	1	42	0	0
9	1	0	3	4	1	0	27	1	1
10	1	0	4	3	4	0	28	0	0
11	0	1	6	4	5	0	4	1	1
12	1	0	4	3	4	0	28	0	0
13	0	1	9	3	7	1	39	1	1
14	0	1	3	3	3	0	83	1	1
15	0	1	8	8	1	1	52	0	0
16	0	0	9	5	1	0	88	1	1
17	0	1	9	3	7	1	39	1	1
18	1	0	9	10	9	1	1	1	1
19	1	1	8	3	4	0	90	1	1

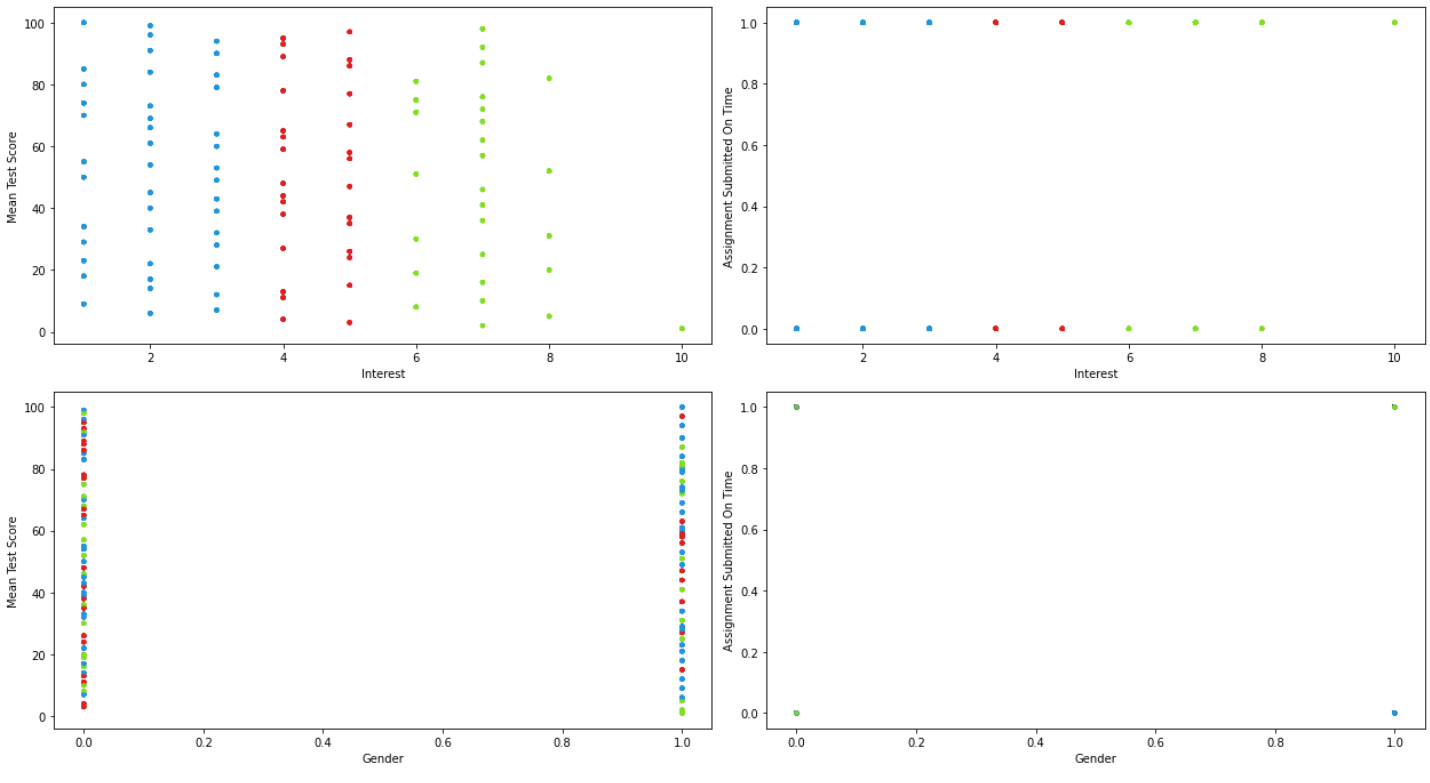
In [52]:

```
from sklearn.cluster import KMeans
```

```
# k means
kmeans = KMeans(n_clusters=3, random_state=0)
dd['cluster'] = kmeans.fit_predict(dd[['Interest', 'Assignment Submitted On Time']])

# get centroids
centroids = kmeans.cluster_centers_
cen_x = [i[0] for i in centroids]
cen_y = [i[1] for i in centroids]
## add to df
dd['cen_x'] = dd.cluster.map({0:cen_x[0], 1:cen_x[1], 2:cen_x[2]})
dd['cen_y'] = dd.cluster.map({0:cen_y[0], 1:cen_y[1], 2:cen_y[2]})
# define and map colors
colors = ['#DF2020', '#81DF20', '#2095DF']
dd['c'] = dd.cluster.map({0:colors[0], 1:colors[1], 2:colors[2]})
```

In [54]:

```
#plt.scatter(dd['Interest'], dd['Mean Test Score'], c=dd.c, alpha = 0.6, s=10)

fig, ((ax1, ax2), (ax3, ax4)) = plt.subplots(2, 2)

plt.rcParams["figure.figsize"] = [17.50, 9.50]
plt.rcParams["figure.autolayout"] = True

#Interest vs Mean Test Score
ax1.scatter(dd['Interest'], dd['Mean Test Score'], c=dd.c, alpha = 0.6, s=10)
ax1.set_xlabel("Interest")
ax1.set_ylabel("Mean Test Score")

#Interest vs Assignment Submitted On Time
ax2.scatter(dd['Interest'], dd['Assignment Submitted On Time'], c=dd.c, alpha = 0.6, s=1
0)
ax2.set_xlabel("Interest")
ax2.set_ylabel("Assignment Submitted On Time")

#Gender vs Mean Test Score
ax3.scatter(dd['Gender'], dd['Mean Test Score'], c=dd.c, alpha = 0.6, s=10)
ax3.set_xlabel("Gender")
ax3.set_ylabel("Mean Test Score")

#Gender vs Assignment Submitted On Time
ax4.scatter(dd['Gender'], dd['Assignment Submitted On Time'], c=dd.c, alpha = 0.6, s=10)
ax4.set_xlabel("Gender")
ax4.set_ylabel("Assignment Submitted On Time")

plt.show()
```



In [ ]: