IN CLASS ACTIVITY 3

Contents

1. Preprocess data:

```
In [3]: df = pd.read_csv(r'C:\Users\win\Documents\Data mining\dataset_Similarity_Prediction\original_training_set\original_training_set.c
```

```
In [4]: df.head()
```

Out[4]:

	id_pair	curated_smiles_molecule_a	curated_smiles_molecule_b	tanimoto_cdk_Extended	TanimotoCombo
0	1	CCN(CC)CC(=O)Nc1c(C)cccc1C	CCCN1CCCC[C@H]1C(=O)Nc1c(C)cccc1C	0.641434	1.623
1	2	Cc1nc2n(c(=O)c1CCN1CCC(c3noc4cc(F)ccc34)CC1)CC...	Cc1nc2n(c(=O)c1CCN1CCC(c3noc4cc(F)ccc34)CC1)CCCC2	0.928846	1.812
2	3	COc1ccccc1OCC(O)CO	COC(=O)CCc1ccc(OCC(O)CNC(C)C)cc1	0.381119	1.064
3	4	CCOc1ccccc1OCCN[C@H](C)Cc1ccc(OC)c(S(N)(=O)=O)c1	CC(C)C(=O)Nc1ccc([N+](=O)[O-])c(C(F)(F)F)c1	0.213429	0.674
4	5	C[C@H](N)Cc1ccccc1	CC(C)(N)Cc1ccccc1	0.905660	1.690

Here, loaded the data which I used in kmeans clustering Assignment. We have 5 columns and 101 rows of chemical lab data set.

```
In [5]: #Drop unwanted columns
        df = df.drop(['id_pair','curated_smiles_molecule_a','curated_smiles_molecule_b','frac_similar'],axis = 1)
        df.head()
```

Out[5]:

	tanimoto_cdk_Extended	TanimotoCombo
0	0.641434	1.623
1	0.928846	1.812
2	0.381119	1.064
3	0.213429	0.674
4	0.905660	1.690

```
In [7]: #Add values in missing rows
        df['tanimoto_cdk_Extended'] = df['tanimoto_cdk_Extended'].fillna(method='ffill')
        df['TanimotoCombo'] = df['TanimotoCombo'].fillna(df['TanimotoCombo'].mean())
```

Selected two columns which helps us in further analysis. And Fill NaN value with ffill method and mean function .

2. Statistics Visualization:

- ### Range:

Range is value is difference of Maximum value and Minimum value .

```
In [42]: #Run for loop for getting range in data set.
         for i in df.columns:
             df_range= df[i].max()-df[i].min()
             print('Range of {}: {}'.format(i,df_range))

         Range of tanimoto_cdk_Extended: 0.894594595
         Range of TanimotoCombo: 1.4260000000000002
```

Here , I ran for loop on both columns and got their range.

Word File Comparison:

I
Range
0.894595
1.426

Here, First range represents value of Tanimoto Cdk Extended. Where second range represents Tanimoto Combo's range

- ### Interquartile range (IQR):

Firstly let's see what does IQR means, IQR is the difference between first quartile and third quartile range.

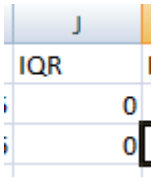
```
In [20]: #Find first quartile
         Q1 = df.tanimoto_cdk_Extended.quantile(0.25)

In [21]: #Find third quartile
         Q3 = df.tanimoto_cdk_Extended.quantile(0.75)

In [22]: #Find Interquartile quartile
         IQR = Q3 - Q1
         print('IQR for tanimoto cdk Extended is: ',IQR)

         IQR for tanimoto cdk Extended is:  0.48233704624999996
```

Here IQR for tanimoto cdk Extended is got 0.48. where taking look on another columns.

```
In [24]: #Find first quartile of TanimotoCombo
         Q1 = df.TanimotoCombo.quantile(0.25)

In [25]: #Find third quartile of TanimotoCombo
         Q3 = df.TanimotoCombo.quantile(0.75)

In [26]: #Find Interquartile quartile
         IQR = Q3 - Q1
         print('IQR for TanimotoCombo is: ',IQR)

         IQR for TanimotoCombo is:  0.756
```

While taking look on another colum TanimotoCombo we got IQR is 0.76.

Word File Comparison:



J
IQR
0
0

Here, We got a questions that why we world file give us 0 in both columns. The reason behind that is our data has lots lof missing and NaN values which we fixed in our preprocessing step.

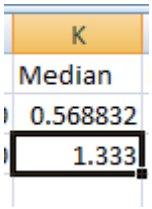
- ## Median:

The median is the middle value in a set of numbers or data.

```
In [32]: #Find median in dataFrame
         median = df.median()
         print('Median of our data frame is: \n',median)

         Median of our data frame is:
          tanimoto_cdk_Extended    0.55814
         TanimotoCombo            1.32200
         dtype: float64
```

Pandas's has a function name median which give median values from given input. Here median of tanimoto cdk Entended is 0.56 and median of TanimotoCombo is 1.32.

Word File Comparison:



K
Median
0.568832
1.333

First value of median give values of tanimoto cdk Entended and second values gives median of TanimotoCombo.

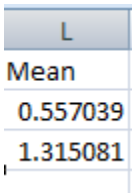
- **Mean:**

The mean value in Python is the arithmetic average of a set of numbers.

```
In [33]: #Find mean in data Frame
         mean = df.mean()
         print('Mean of our data frame is: \n',mean)

         Mean of our data frame is:
          tanimoto_cdk_Extended    0.553941
         TanimotoCombo            1.315081
         dtype: float64
```

Pandas's has a function name mean which give mean values from given input. Here mean of tanimoto cdk Entended is 0.55 and mean of TanimotoCombo is 1.31.

Word File Comparison:



L
Mean
0.557039
1.315081

First value of median give values of tanimoto cdk Entended and second values gives median of TanimotoCombo. We can see that in python and in word both give same output.

- **Mode:**

The mode value is the value that appears most frequently in a given set of data.

```
In [34]: #Find mode in dataframe
         mode= df.mode()
         print('Mode of our data frame is: \n',mode)
```

```
Mode of our data frame is:
    tanimoto_cdk_Extended  TanimotoCombo
0                0.244207       0.674000
1                1.000000       0.729000
2                     NaN       1.315081
```

Mode function can helps us to find mode value in python.

Word File Comparison:

M
Mode
1
0.674

First value of median give values of tanimoto cdk Entended and second values gives median of TanimotoCombo.

- **1st and 3rd quartiles:**

```
In [30]: #First quartile
         first_quartiles = df.quantile(0.25)
         #3rd quartile
         third_quartiles = df.quantile(0.75)
```

```
In [35]: print('1st Quartile of our data Frame is:\n ',first_quartiles)
         print('3rd Quartile of our data Frame is: \n',third_quartiles)
```

```
1st Quartile of our data Frame is:
  tanimoto_cdk_Extended    0.30059
TanimotoCombo            0.90250
Name: 0.25, dtype: float64
3rd Quartile of our data Frame is:
 tanimoto_cdk_Extended    0.782927
TanimotoCombo            1.658500
Name: 0.75, dtype: float64
```

1st quartile means 0.25% of values in data frame where 3rd quartile is 0.75% of values in data Frame. Here I printed both columns 1st and 3rd quartiles values.

Word File Comparison:

N	O
Q1	Q3
0.105405	0.105405
0.574	0.574

Here both Q1 and Q2 values are similar, May be missing values and Null values are giving these output.

- ## Sample variance and sample standard deviation:

Sample variance is a measure of the "average" of the squared deviations from the sample mean, while sample standard deviation is the square root of the sample variance. The variance and standard deviation are both measures of the spread of a data set.

```
In [8]: import statistics
        #variance of dataframe
        variance = statistics.variance(df.TanimotoCombo)
        #Stander deviation
        stdev = statistics.stdev(df.TanimotoCombo)

        print("Sample variance of TanimotoCombo:", variance)
        print("Sample standard deviation of TanimotoCombo:", stdev)
```

```
Sample variance of TanimotoCombo: 0.17135719353535353
Sample standard deviation of TanimotoCombo: 0.41395312963589675
```

For getting variance and standard deviation we imported library which call statistics. which takes data frame as a variance and stdev function input.

```
In [49]: import statistics
         #variance of dataframe
         variance = statistics.variance(df.tanimoto_cdk_Extended)
         #Stander deviation
         stdev = statistics.stdev(df.tanimoto_cdk_Extended)

         print("Sample variance of tanimoto_cdk_Extended:", variance)
         print("Sample standard deviation of tanimoto_cdk_Extended:", stdev)
```

```
Sample variance of tanimoto_cdk_Extended: 0.07358941122658484
Sample standard deviation of tanimoto_cdk_Extended: 0.27127368325472495
```

Word File Comparison:

P	Q
Std	Var
0.270839	0.073354
0.418156	0.174854

In these Snap shot fir raw present value of tanimoto cdk Entended and 2nd raw represents TanimotoCombo.

- ## Find Outliers:

```
In [43]: #Use IQR method for finding outlier
         Q1 = df.tanimoto_cdk_Extended.quantile(0.25)
         Q3 = df.tanimoto_cdk_Extended.quantile(0.75)
         IQR = Q3-Q1
```

```
In [45]: Lower_bound = Q1 - 1.5 * IQR
         Upper_bound = Q3 + 1.5 * IQR
```

```
In [56]: outliers = [x for x in df.tanimoto_cdk_Extended if x < Lower_bound or x > Upper_bound]
```

```
In [58]: outliers
Out[58]: []
```

There are many methods to find outliers where IQR method also one of the method for finding outlier. Here we took Q1 and Q3 quartiles and find Lower and upper bound values. Then for loop to get values below lower bound and above upper bound. Her e, we got empty list which means we don't have any outlier in data.

```
In [59]: Q1 = df.TanimotoCombo.quantile(0.25)
         Q3 = df.TanimotoCombo.quantile(0.75)
         IQR = Q3-Q1
```

```
In [60]: Lower_bound = Q1 - 1.5 * IQR
         Upper_bound = Q3 + 1.5 * IQR
```

```
In [62]: outliers = [x for x in df.TanimotoCombo if x < Lower_bound or x > Upper_bound]
```

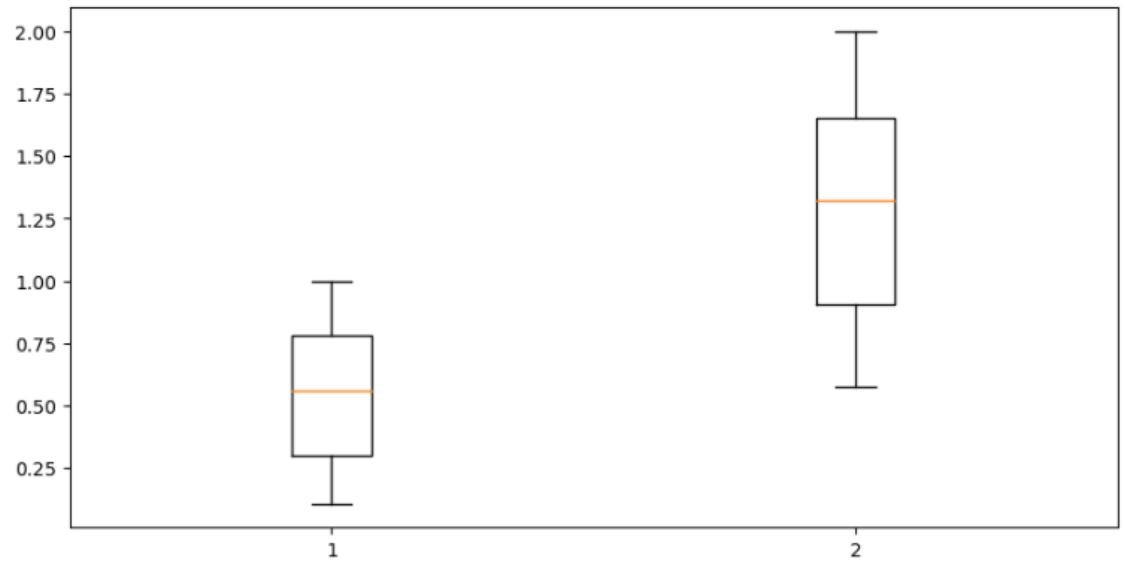
```
In [63]: outliers
Out[63]: []
```

Same method use for another column and got same output. Means our data is good without any outlier.

3. Draw the following diagrams:

- **Box Plot:**

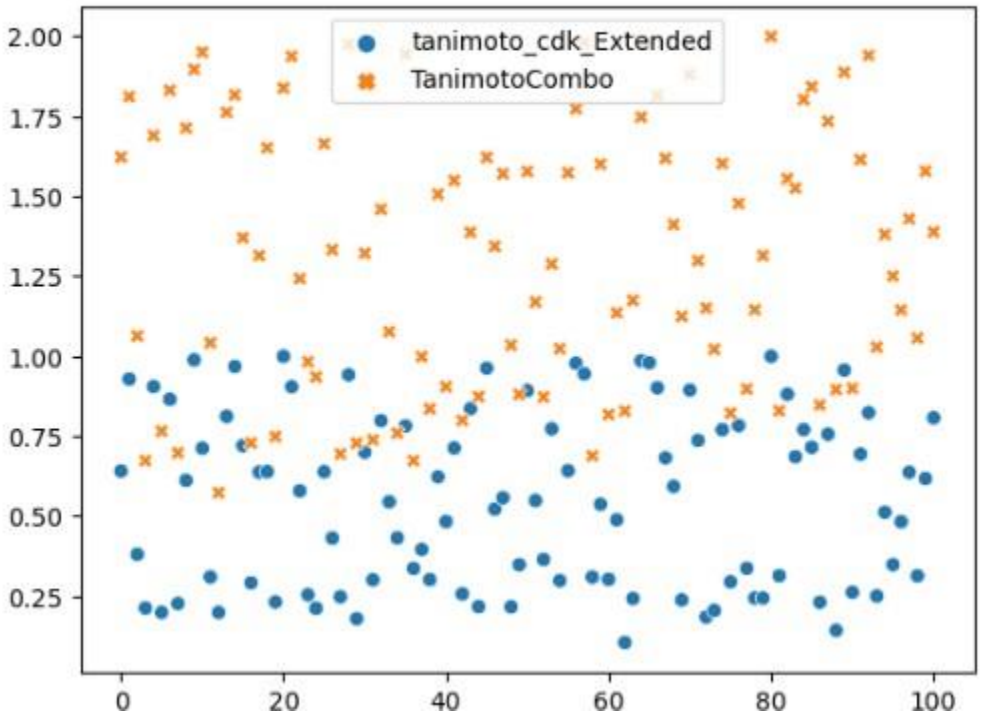
```
In [44]: #box plot fig size
         fig = plt.figure(figsize=(10, 5))
         plt.boxplot(df)
         plt.show()
```



We use box plot mainly for finding outlier. Here, we can see that data don't have any outlier which we show already in IQR outlier finding method.

- **Sea-born Simplot:**

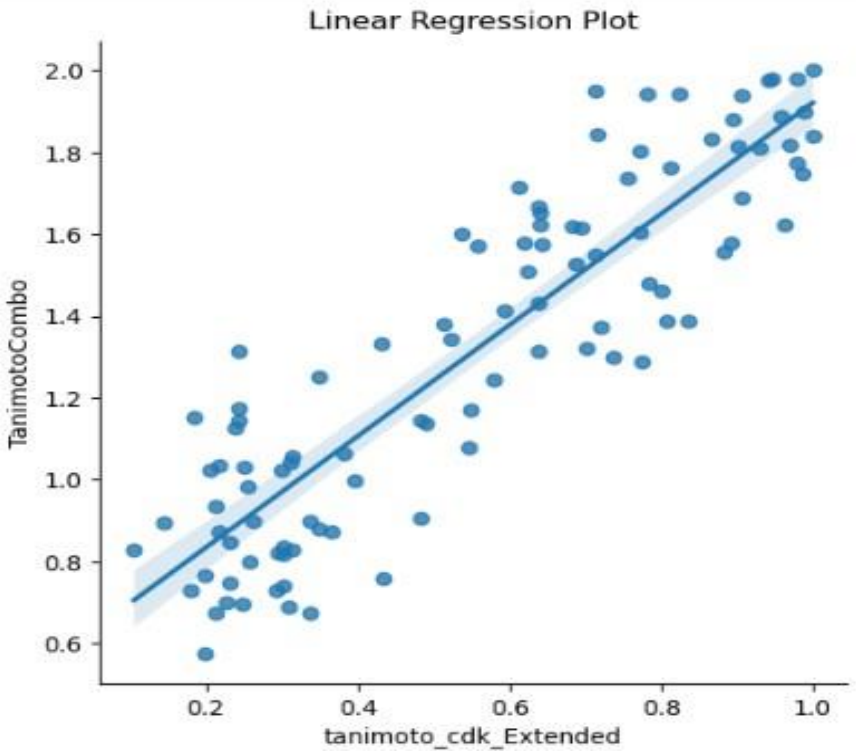
```
In [70]: sns.scatterplot(data=df)
         plt.show()
```



Sea-born Simplot is mainly represented by scatter-plot. Here data mainly present two column where tanimoto cdk Exntended are in blue dot and Tanimoto Combo represented by orange cross.

- **Linear regression:**

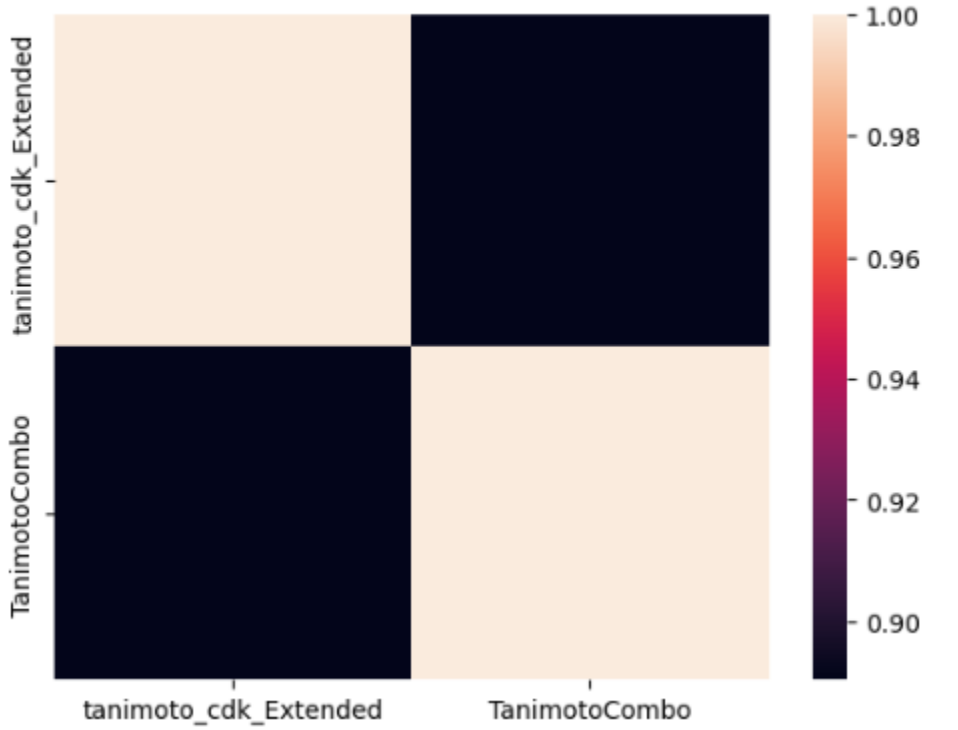
```
In [52]: # Draw a linear regression plot using Seaborn
         sns.lmplot(x='tanimoto_cdk_Extended', y='TanimotoCombo', data=df)
         plt.xlabel('tanimoto_cdk_Extended')
         plt.ylabel('TanimotoCombo')
         plt.title('Linear Regression Plot')
         plt.show()
```



Here we can see the straight line between scatter plots. Which mainly due to Linear Regression. Were X axis is values of Tanimoto_cdk_extended where Y axis represent the value of TanimotoCombo's value.

- **HeatMap:**

```
In [39]: corr = df.corr()
         p1 = sns.heatmap(corr)
```



Heatmap is correlation between two columns. Here we just having two columns hence, Heatmap presented only 4 blogs.

4. 5 Point Summary:

Summary of Tanimoto CDK Extended:

Definition: Tanimoto CDK Extended is a similarity measure used to compare molecular structures based on their fingerprints. It provides a quantitative assessment of the similarity between two molecules.

Range: The Tanimoto CDK Extended similarity measure has a minimum value of 0.105 and a maximum value of 1. This indicates that the similarity score ranges from low to high, with 1 representing identical molecules.

Quartiles: The first quartile (Q1) for Tanimoto CDK Extended is 0.3, the second quartile (Q2) is 0.105, and the third quartile (Q3) is 0.78. These quartiles help to understand the distribution of similarity scores and identify the central tendency of the data.

Application: Tanimoto CDK Extended is used in various fields such as drug discovery, chemoinformatics, and molecular similarity perception based on machine-learning models.

Performance: Models built on the Tanimoto CDK Extended similarity measure have shown promising performance in predicting molecular pairs, with correctly predicted pairs reaching up to 81 out of 100 in certain cases.

Summary of TanimotoCombo:
Definition: TanimotoCombo is a metric used to calculate similarity between molecular structures, incorporating both 2D and 3D similarity assessments. It provides a comprehensive measure of molecular similarity, considering different structural aspects.
Range: The TanimotoCombo metric has a minimum value of 0.57 and a maximum value of 2. This wider range compared to Tanimoto CDK Extended suggests that it captures a broader spectrum of molecular similarities, potentially including more diverse structural features.
Quartiles: The first quartile (Q1) for TanimotoCombo is 0.9, the second quartile (Q2) is 0.57, and the third quartile (Q3) is 1.66. These quartiles provide insights into the distribution of similarity scores and the central tendency of the data, aiding in the interpretation of molecular similarity assessments.
Application: TanimotoCombo is utilized in molecular similarity perception based on machine-learning models, contributing to the development of predictive models for various applications in chemistry and pharmaceutical research.
Performance: Models incorporating the TanimotoCombo metric have shown varying performance, with correctly predicted molecular pairs reaching up to 70 out of 100 in certain cases. This suggests that the TanimotoCombo metric captures a broader range of molecular similarities, potentially leading to more diverse predictive outcomes.

5. Conclusion:

This collaboration combines statistical analysis with Excel and Python visualization, aiming to provide a deeper comprehension of data patterns by combining statistical insights with aesthetically appealing visuals, improving analytical precision and clear communication.

This integration of statistical analysis and visualization tools in Excel with Python provides a powerful method for deciphering data complexities and gaining deeper insights from data patterns

6. Reference:

Dataset from UCI: Similarity Prediction - UCI Machine Learning Repository