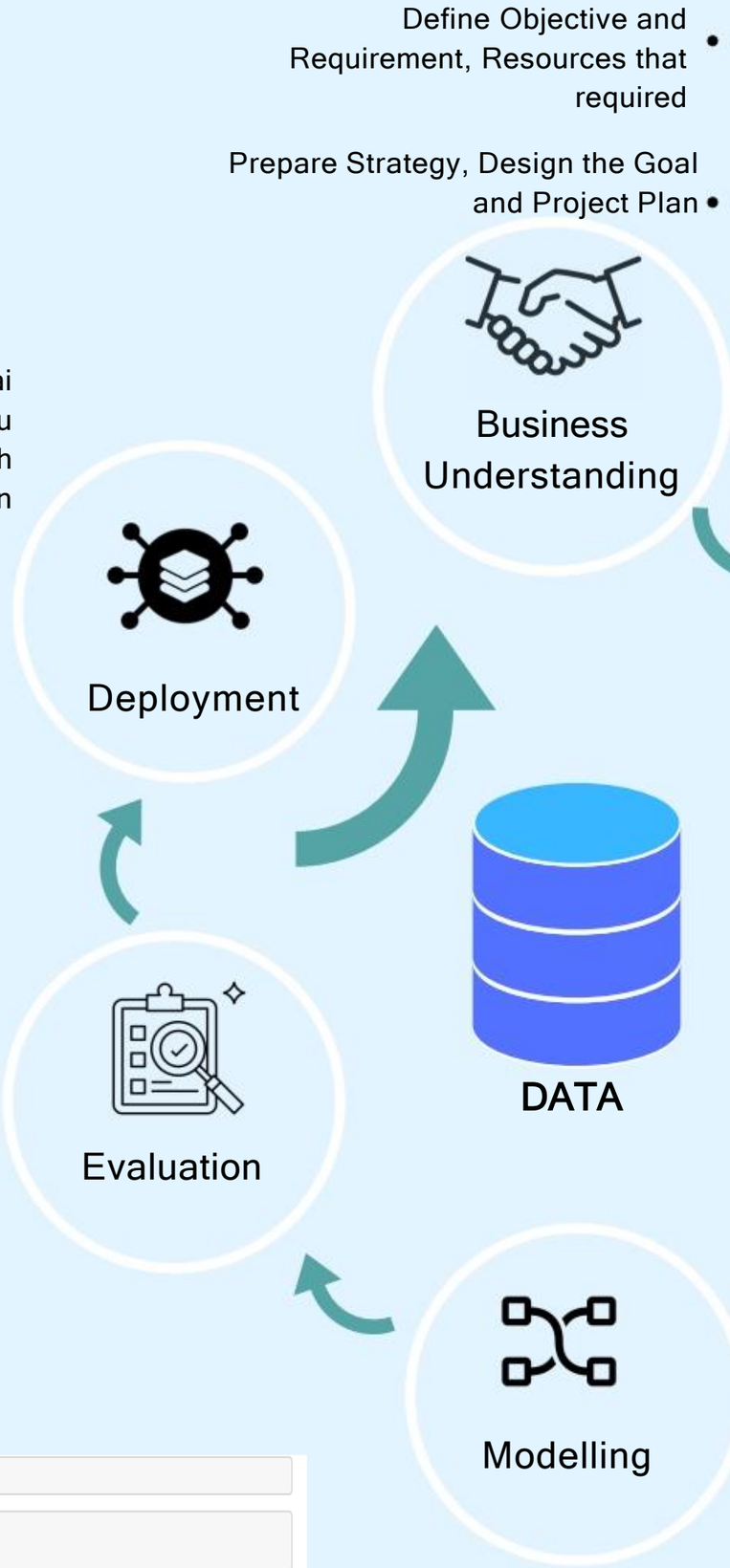
Drugs Recommendation Using K-Nearest Neighbor Algorithm

Institut Teknologi Del
Proyek Penambangan Data
Semester Gasal T.A 2020/2021

Dodi Sanjaya Butarbutar
Jovan Pioma Pakpahan
Rizky Marganda Sitohang
Chatrin

https://github.com/chatrinpandiangan/Drugs-Recommendation-Using-K-Nearest-Neighbor-Algorithm



- Define Objective and Requirement, Resources that required
- Prepare Strategy, Design the Goal and Project Plan

- Tahap terakhir metodologi CRISP-DM
- Pada tahap ini dijelaskan mengenai perencanaan fase penyebaran atau penggunaan model yang sudah dihasilkan, perencanaan pemantauan dan pemeliharaan

- Collect Initial Data : data train & data test
- Describe Data : nominal, numerikal, waktu
- Explore Data : EDA

- Data Cleaning
- Data Reduction
- Data Preprocessing

- Selec Modelling Technique
- K-Nearest Neighbor

- Evaluate the Modelling KNN Result
- Using Confusion Matrix
- Evaluate the result, accuracy, precision, recall
- Fase Interpetasi terhadap hasil Data Mining, dengan tujuan agar hasil pada tahap modelling sesuai dengan yang diharapkan

```
In [31]: from sklearn.neighbors import NearestNeighbors

In [32]: model_knn = NearestNeighbors(metric = 'cosine', algorithm = 'brute')
         model_knn.fit(matrix)
Out[32]: NearestNeighbors(algorithm='brute', metric='cosine')

In [33]: for i in range(len(pivot_data)):
             if pivot_data.iloc[i].name == filtered_drug_rating.iloc[0].name_of_drug:
                 query_index = i
                 break

         distances, indices = model_knn.kneighbors(pivot_data.iloc[query_index, :].values.reshape(1, -1), n_neighbors = 5)

         for i in range(0, len(distances.flatten())):
             if i == 0:
                 print('Recommended drug is {0}'.format(pivot_data.index[query_index]))
                 print('Similar drugs :\n')
             else:
                 print('{0}. {1},  distance:{2}'.format(i, pivot_data.index[indices.flatten()[i]], distances.flatten()[i]))

Recommended drug is Geodon
Similar drugs :

1. Risperidone,  distance:1.0
2. Paliperidone,  distance:1.0
3. Venlafaxine,  distance:1.0
4. Abilify,  distance:1.0
```

KNN Modelling by user rating

```
In [68]: from sklearn.feature_extraction.text import TfidfTransformer

         tfidf_transformer = TfidfTransformer()
         df_train_tfidf = tfidf_transformer.fit_transform(df_train)
         print("Shape of term frequency-inverse document frequency matrix:", df_train_tfidf.shape)

Shape of term frequency-inverse document frequency matrix: (2043, 2074)

In [69]: #  model Multinomial Naïve Bayes

         from sklearn.naive_bayes import MultinomialNB
         from sklearn.metrics import accuracy_score

         mod = MultinomialNB()
         mod.fit(df_train_tfidf, drugs_rating.total_rating_count)
         print(mod)

MultinomialNB(alpha=1.0, class_prior=None, fit_prior=True)

In [71]: from sklearn.model_selection import train_test_split
         predicted = mod.predict(df_train_tfidf)
         print("Accuracy:", accuracy_score(drugs_rating.total_rating_count, predicted))

Accuracy: 0.3499755261869799

In [81]: from sklearn.metrics import classification_report
         print(classification_report(drugs_rating.total_rating_count, predicted))

              precision    recall  f1-score   support

           1       0.34      1.00      0.51       675
           2       0.87      0.14      0.24       275
           3       1.00      0.01      0.01       182
           4       0.00      0.00      0.00       105
           5       0.00      0.00      0.00       104
```

Evaluate using Cofusion Matrix

```
In [38]: filtered_drug_rating
```

Out[38]:

	name_of_drug	average_rating
2	Geodon	10.0
3	Paliperidone	10.0
6	Venlafaxine	10.0
0	Abilify	8.5
1	Aripiprazole	8.5
5	Risperidone	7.5
7	Ziprasidone	7.0
4	Risperdal	6.5

```
In [39]: #get index
         for i in range(len(pivot_data)):
             if pivot_data.iloc[i].name == filtered_drug_rating.iloc[0].name_of_drug:
                 query_index = i
                 break

         distances, indices = model_knn.kneighbors(pivot_data.iloc[query_index, :].values.reshape(1, -1), n_neighbors = 10)

         for i in range(0, len(distances.flatten())):
             if i == 0:
                 print('Recommended drug is {0}'.format(pivot_data.index[query_index]))
                 print('Similar drugs :\n')
             else:
                 print('{0}. {1},  distance:{2}'.format(i, pivot_data.index[indices.flatten()[i]], distances.flatten()[i]))

Recommended drug is Geodon
Similar drugs :

1. Ziprasidone,  distance:0.13503971129809889
2. Paliperidone,  distance:0.15446915993916932
3. Abilify,  distance:0.21597767095967058
4. Invega,  distance:0.2871913267986823
5. Latuda,  distance:0.28782640028508466
6. Lurasidone,  distance:0.3624069808577135
7. Saphris,  distance:0.3655480200582055
8. Aripiprazole,  distance:0.3944000321271519
9. Asenapine,  distance:0.3987872408950962
```

KNN Modelling by use case