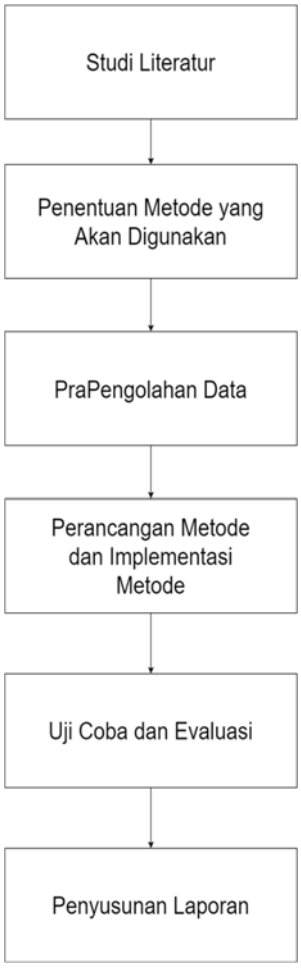
PENERAPAN DATA MINING CLASSIFICATION UNTUK DATA BLOGGER MENGGUNAKAN KNN DAN EVALUASI MODEL DENGAN HOLD OUT ESTIMATION

Hanan Nadia - 1810511098
Irza Ramira Putra - 18105111100
Deo Haganta Depari - 1810511104
Nadhifa Zhafira - 1810511111
Quina Alifa – 1810511115

Program ini bertujuan untuk mengklasifikasikan data blogger, apakah blogger tersebut merupakan blogger professional atau bukan (PB) berdasarkan tingkat edukasi (degree), tingkah politik (caprice), topik (topics), pergantian media local (LMT) dan ruang local, politik dan sosial (LPSS). Metode klasifikasi yang digunakan adalah algoritma KNN (K-Nearest Neighbor)

Metodologi Penelitian

Secara umum, penelitian ini dilakukan dalam beberapa tahapan yang diawali dari studi literatur, penentuan metode yang akan digunakan, pra pengolahan data, perancangan metode dan implementasi metode, uji coba dan evaluasi dan penyusunan laporan.



Studi Literatur

Mempelajari Jurnal Penelitian terdahulu dan Jurnal Penelitian yang menghasilkan *dataset* yang digunakan.

Penentuan Metode yang Akan Digunakan

Dari informasi yang telah didapatkan dari studi literatur, penulis akan menentukan metode apa yang akan digunakan selain dari metode yang telah digunakan pada jurnal pada penelitian terdahulu, metode yang digunakan adalah algoritma KNN (*K-Nearest Neighbor*) untuk mengklasifikasikan dataset *blogger* menjadi dua kelompok. Kemudian untuk evaluasi model, penulis menggunakan metode *Hold Out Estimation*.

Pra Pengolahan Data

Jumlah responden dalam penelitian ini sebesar 100 responden, dengan dataset blogger ini, peneliti akan mengklasifikasikan jenis blogger kedalam 2 kelompok yaitu *Blogger Professional (BP)* dan *Blogger Musiman(BM)*.

Berikut merupakan data training dari dataset blogger.

Degree	**Caprice**	**Topics**	**LMT**	**LPSS**	**PB**
high	left	impression	yes	yes	yes
high	left	political	yes	yes	yes
medium	middle	tourism	yes	yes	yes
. . .	. . .	. . .	. . .	. . .	. . .
medium	right	news	yes	yes	no
medium	left	impression	yes	yes	yes

Tabel 2 merupakan dataset blogger yang dipakai dalam penelitian ini. Data tersebut berjumlah 100 data yang direpresentasikan dalam bentuk tabel.

Dataset yang digunakan pada penelitian terdahulu, yang juga terdapat pada situs *"UCI Machine Learning Repository"* merupakan *spreadsheet* bertipe *excel,* maka dari itu *dataset* perlu diubah dulu menjadi tipe *csv*, sehingga bisa digunakan pada proses Implementasi Metode.

Perancangan Metode dan Implementasi Metode

Berdasarkan metode yang telah ditentukan, maka *library* yang digunakan adalah:

- *Pandas*
- *Numpy*
- *Preprocessing* dari *sklearn*
- *Train_test_split* dari *sklearn.model_selection*
- *StratifiedKFold* dari *sklearn.model_selection*

```
 8    import pandas as pd
 9    import numpy as np
10    from sklearn import preprocessing
11    from sklearn.model_selection import train_test_split
12    from sklearn.neighbors import KNeighborsClassifier
```

Langkah implementasi metodenya adalah sebagai berikut:

- Membaca *dataset csv* yang telah diolah sebelumnya sesuai dengan kolom yang tepat

```
15    names = ['degree','caprice','topic','lmt','lpss','pb']
16    dataset = pd.read_csv("kohkiloyeh.csv", names=names)
```

- Konversi kategori dari tipe *string* menjadi *tipe numeric* untuk setiap kolom, dengan menggunakan *preprocessing label encoder*

```
19    degrees = dataset['degree']
20    caprices= dataset['caprice']
21    topics = dataset['topic']
22    lmts = dataset['lmt']
23    lpsss = dataset['lpss']
24    pbs = dataset['pb']
25
26    le = preprocessing.LabelEncoder()
27    dg_en = le.fit_transform(degrees)
28    #high=0;low=1;med=2
29    cp_en = le.fit_transform(caprices)
30    #left=0;middle=1;rigth=2
31    tp_en = le.fit_transform(topics)
32    #impression=0;news=1;political=2;scientific=3;tourism=4
33    lmt_en = le.fit_transform(lmts)
34    #no=0;yes=1
35    lpss_en = le.fit_transform(lpsss)
36    #no=0;yes=1
37    pb_en = le.fit_transform(pbs)
38    #no=0;yes=1
```

  - Untuk Degrees, *string* berubah menjadi *numeric* dengan ketentuan : high=0 ; low=1 ; med=2
  - Untuk Caprices, *string* berubah menjadi *numeric* dengan ketentuan : left=0 ; middle=1 ; right=2
  - Untuk Topics, *string* berubah menjadi *numeric* dengan ketentuan : impression=0 ; news=1 ; political=2 ; scientific=3 ; tourism=4
  - Untuk lmt, *string* berubah menjadi *numeric* dengan ketentuan : no=0 ; yes=1
  - Untuk lpss, *string* berubah menjadi *numeric* dengan ketentuan : no=0 ; yes=1
  - Untuk pb, *string* berubah menjadi *numeric* dengan ketentuan :

no=0 ; yes=1

- Pisahkan *dataset array* menjadi dua kategori, satu *array* fitur dan satu lagi *array* untuk kelas

```
41    fitur_gabung = np.array([dg_en,cp_en,tp_en,lmt_en,lpss_en])
42    X_data = np.ndarray.transpose(fitur_gabung)
43    y_data = pb_en
```

- Pisahkan data fitur dan data kelas menjadi 2 kategori, satu data untuk *training* dan satu untuk *testing*, menggunakan *train test split* dengan parameter data dan ukuran data *testing*

```
46    X_train, X_test, y_train, y_test = train_test_split(X_data, y_data, test_size=0.2, random_state=0)
```

- Melakukan klasifikasi dengan menggunakan algoritma KNN

```
48    model = KNeighborsClassifier(n_neighbors=7, weights="distance")
49    model.fit(X_train, y_train)
```

- Menyimpan dan mencetak hasil prediksi dari data *testing*, lalu mencetak hasil kelas yang sebenarnya

```
51    # getting prediction from data testing
52    predicted = model.predict(X_test)
53
54    # print prediction
55    print('- Classification using KNN -')
56    print("Hasil Klasifikasi (KNN) dengan Data Testing          : \n", predicted)
57    print()
```

- Membuat perhitungan besarnya nilai prediksi yang salah (error) dan mencetak nilai nya
- Membuat perhitungan nilai prediksi yang akurat dengan cara mengurangi 100% dengan nilai prediksi yang salah dan mencetak nilai nya

```
55    print('- Classification using KNN -')
56    print("Hasil Klasifikasi (KNN) dengan Data Testing          : \n", predicted)
57    print()
58
59    # print category(class) from real data
60    print("Hasil Klasifikasi yang benar dengan Data Training    : \n", y_test)
61    print()
62
63    #print presentation of prediciton error
64    error = ((y_test != predicted).sum()/len(predicted))*100
65    print("Error Prediction = %.2f" %error,"%")
66
67    #print presentation of accuracy
68    accuracy = 100-error
69    print("Accuracy         = %.2f" %accuracy,"%")
70    print()
```

- Menerapkan algoritma evaluasi Hold Out Estimation pada model data yang digunakan dan mencetak hasil akurasi, *sensitivity dan specificity* dari model

```
def Conf_matrix(y_actual, y_pred):
    TP = 0
    FP = 0
    TN = 0
    FN = 0

    for i in range(len(y_pred)):
        if y_actual[i]==y_pred[i]==1:
            TP += 1
        if y_pred[i]==1 and y_actual[i] !=y_pred[i]:
            FP += 1
        if y_actual[i]==y_pred[i]==0:
            TN += 1
        if y_pred[i]==0 and y_actual[i]!= y_pred[i]:
            FN += 1
    return (TP, FN, TN, FP)

#hold out estimation evaluation
TP, FN, TN, FP = Conf_matrix(y_test, predicted)

print('- Model Evaluation Hold Out Estimation -')
print('Accuracy     = ', (TP+TN)/(TP+TN+FP+FN))
print('Sensitivity  = ', TP/(TP+FN))
print('Specificity  = ', TN/(TN+FP))
```

Uji Coba dan Evaluasi

Pengujian dilakukan untuk mendapatkan pengaturan *n neighbors, weights(uniform/distance)*, dan *test size* yang terbaik berdasarkan nilai error prediksi yang paling kecil dan akurasi, nilai sensitivity dan specificity yang paling besar.

Untuk memudahkan dalam mendapatkan pengaturan yang terbaik, penulis menggunakan perulangan dan menyimpan setiap *output* ke dalam *file notepad*, dari evaluasi *output* tersebut didapatkan 2 pengaturan terbaik, keduanya memiliki *n neighbors* bernilai 7 dan menggunakan *weights distance* dengan :

- pengaturan pertama menggunakan *test size* bernilai 0.1 (Hanya menggunakan 10 persen data dari *dataset* sebagai data testing), menghasilkan output :

```
- Classification using KNN -
Hasil Klasifikasi (KNN) dengan Data Testing         :
 [1 1 1 1 1 0 1 0 1 1]

Hasil Klasifikasi yang benar dengan Data Training   :
 [1 1 1 1 1 0 1 0 1 1]

Error Prediction = 0.00 %
Accuracy         = 100.00 %

- Model Evaluation Hold Out Estimation -
Accuracy    =  1.0
Sensitivity  =  1.0
Specificity  =  1.0
```

Analisa :

		Data									
		1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
HASIL	**Klasifikasi**	1	1	1	1	1	0	1	0	1	1
	Sebenarnya	1	1	1	1	1	0	1	0	1	1

Hasil nilai evaluasi klasifikasi:

- Error Prediction = 0.00 %
- Accuracy = 100.00 %

Hasil nilai evaluasi model:

- Accuracy = 1.0
- Sensitivity = 1.0
- Specificity = 1.0

- pengaturan kedua menggunakan *test size* bernilai 0.2 (Hanya menggunakan 20 persen data dari *dataset* sebagai data testing), menghasilkan output :

```
- Classification using KNN -
Hasil Klasifikasi (KNN) dengan Data Testing         :
 [1 1 1 1 1 0 1 0 1 1 1 1 1 0 0 1 1 1 1 0]

Hasil Klasifikasi yang benar dengan Data Training    :
 [1 1 1 1 1 0 1 0 1 1 1 1 1 0 1 1 0 1 1 0]

Error Prediction = 10.00 %
Accuracy         = 90.00 %

- Model Evaluation Hold Out Estimation -
Accuracy     =  0.9
Sensitivity  =  0.9333333333333333
Specificity  =  0.8
```

Analisa :

		Data																			
		1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**
HASIL	**KLASIFIKASI**	1	1	1	1	1	0	1	0	1	1	1	1	1	0	0	1	1	1	1	0
	SEBENARNYA	1	1	1	1	1	0	1	0	1	1	1	1	1	0	1	1	0	1	1	0

Hasil nilai evaluasi klasifikasi:

- Error Prediction = 10.00 %
- Accuracy = 90.00 %

Hasil nilai evaluasi model:

- Accuracy = 0.9
- Sensitivity = 0.9333333333333333
- Specificity = 0.8