LAPORAN TUGAS BESAR

diajukan untuk memenuhi tugas mata kuliah (CII4I3) Penambangan Data

oleh:
Otniel Abiezer (NIM 1301180469)

PROGRAM STUDI S1 INFORMATIKA
FAKULTAS INFORMATIKA
UNIVERSITAS TELKOM
BANDUNG
2022

1. **HUBUNGAN ANTARA FILE**

   Items.csv, category_hierarchy.csv, dan order.csv sebagai train, dan submission.csv sebagai test. Pada items.csv memiliki itemID sebagai PK (Primary Key) dan category sebagai FK (Foreign Key) yang berisi keterangan tentang item, sementara pada category_hierarchy dengan category sebagai PK, dan order.csv dengan menggunakan FK dari items.

2. **PREPROCESSING DAN ANALISIS**

   Semua proses preprocessing yang dilakukan adalah

   1) Imputasi missing value

      Karena hampir semua data tidak memiliki missing value kecuali pada kolom categories di items

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 32776 entries, 0 to 32775
Data columns (total 8 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   itemID      32776 non-null  int64
 1   brand       32776 non-null  int64
 2   feature_1   32776 non-null  int64
 3   feature_2   32776 non-null  int64
 4   feature_3   32776 non-null  int64
 5   feature_4   32776 non-null  int64
 6   feature_5   32776 non-null  int64
 7   categories  25988 non-null  float64
dtypes: float64(1), int64(7)
memory usage: 2.0 MB
```

      Banyaknya data yang missing adalah 20% dan melakukan imputasi missing value dengan nilai -1 karena data bernilai kategori dan -1 untuk menambah kategori baru.

   2) Drop Duplikasi

```
[19] category['category'].nunique()

     4301

[20] category.shape[0]

     4333

[21] category = category.drop_duplicates(subset=['category'], keep='first')
     category.shape[0]

     4301
```

      Pada kolom category di category_hierarchy.csv memiliki duplikasi karena seharusnya seluruh category bernilai unik, tetapi total datanya lebih banyak daripada nilai unik tersebut (4333 > 4301). Untuk itu, dilakukan drop duplikat dengan mengambil nilai pertama saja yang diakui.

   3) Penggabungan ketiga dataframe

	date	userID	itemID	order	brand	feature_1	feature_2	feature_3	feature_4	feature_5	categories	parent_category
0	2020-06-01	38769	3477	1	186	6	0	196	0	45	74.0	3056
1	2020-06-01	42535	30474	1	193	10	3	229	3	132	3459.0	3056
2	2020-06-01	42535	15833	1	1318	4	1	455	0	108	2973.0	566
3	2020-06-01	42535	20131	1	347	4	0	291	3	44	30.0	1682
4	2020-06-01	42535	4325	1	539	6	0	303	0	45	3104.0	1852

Seluruh tiga file csv terpisah disatukan dengan menggunakan Join berdasarkan hubungan PK-FK

4) Membuat Label berdasarkan Date

Karena memerlukan prediksi tentang minggu ke berapa (ada 4 minggu dalam 1 bulan), maka dibuat label baru dengan melihat day pada date lalu melihat kondisinya sebagai berikut

a) Tanggal 1 <= x <= 7 menjadi 1
b) Tanggal 8 <= x <= 14 menjadi 2
c) Tanggal 15 <= x <= 21 menjadi 3
d) Sisanya menjadi 4

	userID	itemID	order	brand	feature_1	feature_2	feature_3	feature_4	feature_5	categories	parent_category	Label
1048570	18379	31073	1	1126	4	0	291	3	129	777.0	3027	4
1048571	18379	11425	1	1445	3	0	-1	-1	-1	1395.0	3233	4
1048572	80	6024	1	361	10	0	503	0	90	2995.0	3189	4
1048573	80	29330	1	406	10	0	485	3	132	3444.0	3898	4
1048574	33107	19304	2	400	10	0	504	3	17	806.0	3027	4

5) Drop kolom tidak perlu

```
[30] hasil_join = hasil_join.drop(columns=['order', 'brand', 'categories', 'userID', 'itemID'])
     hasil_join.tail()
```

	feature_1	feature_2	feature_3	feature_4	feature_5	parent_category	Label
1048570	4	0	291	3	129	3027	4
1048571	3	0	-1	-1	-1	3233	4
1048572	10	0	503	0	90	3189	4
1048573	10	0	485	3	132	3898	4
1048574	10	0	504	3	17	3027	4

6) Melakukan pergeseran nilai +1 dengan kolom bernilai minimal -1

```
[31] hasil_join.min()

feature_1         -1
feature_2          0
feature_3         -1
feature_4         -1
feature_5         -1
parent_category   -1
Label              1
dtype: int64

hasil_join[['feature_1', 'feature_3', 'feature_4', 'feature_5', 'parent_category']] = hasil_join[['feature_1', 'feature_3', 'feature_4', 'feature_5', 'parent_cat
hasil_join[['feature_1', 'feature_3', 'feature_4', 'feature_5', 'parent_category']].head()
```

	feature_1	feature_3	feature_4	feature_5	parent_category
0	7	197	1	46	3057
1	11	230	4	133	3057
2	5	456	1	109	567
3	5	292	4	45	1683
4	7	304	1	46	1853

Hal ini dilakukan agar dapat dilakukan algoritma klasifikasi berikutnya

3. METODE

```
[37] cnb = CategoricalNB()
     cnb.fit(X_train, y_train.values.ravel())

     CategoricalNB()
```

Metode yang digunakan adalah Categorical Naïve Bayes karena semua hasil akhir memiliki nilai kategorikal, sehingga dengan menggunakan categorical naïve bayes merupakan pendekatan yang tepat dan juga merupakan algoritma klasifikasi yang cukup cepat untuk data yang banyak.
Untuk pembagian data test dan data train adalah 75% dan 25%.

4. EVALUASI

Evaluasi dengan menggunakan 2 metrik, yaitu akurasi dan Confusion Matrix seperti yang terlihat

```
[43] print("Accuracy : ", metrics.accuracy_score(y_test, prediksi))

     Accuracy :  0.29571533203125

     metrics.confusion_matrix(y_test, prediksi)

     array([[ 1483,  1439,   882, 55454],
            [ 1548,  1624,   965, 59110],
            [ 1574,  1495,  1049, 57486],
            [ 1688,  1814,  1169, 73364]])
```

Untuk klasifikasi 4 kelas, sudah termasuk baik.

5. VIDEO PRESENTASI

https://youtu.be/TDGDgI0xtLk