Tugas Besar- Penambangan Data

Nama: Marina Irdayanti
NIM : 1301174319
Kelas: IF-41-GAB03

Dataset: Data Mining Cup 2019 (https://www.data-mining-cup.com/reviews/dmc-2019/)

1. **Data Preparation**

   Menyiapkan dataset sebelum melakukan preprocessing. Data ini memiliki 1879 baris dan 10 kolom.

```
data = pd.read_csv("/content/drive/My Drive/baru/train.csv", sep="|")
data
```

	trustLevel	totalScanTimeInSeconds	grandTotal	lineItemVoids	scansWithoutRegistration	quantityModifications	scannedLineItemsPerSecond	valuePerSecond	lineItem
0	5	1054	54.70	7	0	3	0.027514	0.051898	
1	3	108	27.36	5	2	4	0.129630	0.253333	
2	3	1516	62.16	3	10	5	0.008575	0.041003	
3	6	1791	92.31	8	4	4	0.016192	0.051541	
4	5	430	81.53	3	7	2	0.062791	0.189605	
...	...	...	...	...	...	...	...	...	
1874	1	321	76.03	8	7	2	0.071651	0.236854	
1875	1	397	41.89	5	5	0	0.065491	0.105516	
1876	4	316	41.83	5	8	1	0.094937	0.132373	
1877	2	685	62.68	1	6	2	0.035036	0.091504	
1878	4	1140	38.03	2	2	3	0.016667	0.033360	

1879 rows × 10 columns

2. **Analisa Kualitas Data**

   Berdasarkan metrik yang telah dipilih dan jelaskan sebelumnya, berikut adalah penilaian dataset Data Mining Cup 2019

   2.1 Completeness

   Pada dataset tersebut tidak terdapat missing value dimana, informasi data tentang objek tersebut dapat diketahui. Maka, dataset tersebut memenuhi dari poin completeness dalam penilaian ini. Untuk itu, tidak perlu menggunakan solusi preprocessing karena, dataset telah lengkap dan dibuktikan melalui gambar dibawah ini.

```
trustLevel                    34.018095
totalScanTimeInSeconds      9321.532730
grandTotal                   508.644918
lineItemVoids                 54.699308
scansWithoutRegistration      49.042044
quantityModifications         25.252794
scannedLineItemsPerSecond      0.581375
valuePerSecond                 2.017457
lineItemVoidsPerPosition       7.454044
fraud                          0.553486
dtype: float64
```

```
m.isnull()
```

```
trustLevel                   False
totalScanTimeInSeconds       False
grandTotal                   False
lineItemVoids                False
scansWithoutRegistration     False
quantityModifications        False
scannedLineItemsPerSecond    False
valuePerSecond               False
lineItemVoidsPerPosition     False
fraud                        False
dtype: bool
```

```
print('Percent of missing "fraud" records is %.2f%%' %((data['fraud'].isnull().sum()/data.shape[0])*100))
```

```
Percent of missing "fraud" records is 0.00%
```

2.2 Uniqueness

Ciri dari bagian ini menunjukkan hasil kumpulan data dari dataset tersebut memiliki keunikan dimana, tidak terdapat duplikasi. Dalam dataset ini sesuai dengan pernyataan sebelumnya dimana, tidak terdapat duplikasi antara data satu dengan data lainnya dan dapat disimpulkan bahwa dataset ini memenuhi syarat dari segi uniqueness (keunikan) dalam poin ini. Hasil dari dataset adult.data dapat diperlihatkan melalui gambar dibawah ini.

```
data.duplicated().sum()
```

```
0
```

2.3 Timeliness

Timeliness menampilkan penggunaan waktu yang ada di dalam data tersebut. Untuk dataset ini terdapat atribut atau kolom tabel yang menerangkan kondisi waktu yaitu, valuePerSecond, totalScanTimeInSeconds dan scannedLineItemsPerSecond. Pada kolom tabel ‘valuePerSecond’ menerangkan informasi nilai total rata-rata produk yang dipindai per detik dan kolom ‘totalScanTimeInSeconds’ berisi total waktu dalam detik antara produk pertama dan terakhir dipindai serta, kolom ‘scannedLineItemsPerSecond’ berisi jumlah rata-rata produk yang dipindai per detik. Selanjutnya dataset ini dapat dikatakan sesuai dari segi timeliness karena dapat menerangkan kondisi waktu. Tetapi, kami juga memberikan tambahan seperti menambahkan fitur tambahan dengan cara memfilter isi baris dari kolom yang disebutkan sebelumnya. Karena, untuk melihat seberapa banyak waktu yang digunakan untuk produk tersebut. Berikut ini hasil yang telah dilakukan:

```
data['CountTimeScan'] = data['totalScanTimeInSeconds'] - data['lineItemVoids']
data['CountTimePerSecond'] = data['valuePerSecond'] - data['scannedLineItemsPerSecond']
data
```

CountTimeScan	CountTimePerSecond
1047	0.024383
103	0.123704
1513	0.032427
1783	0.035349
427	0.126814
...	...
313	0.165202
392	0.040025
311	0.037437
684	0.056467
1138	0.016693

2.4 Validity

Pada validity dari dataset ini memiliki hasil yang cukup baik dan tidak diperlukan kembali terkait solusi dalam preprocessing karena telah terbukti

sebelumnya dalam penilaian completness dimana, kondisi data tidak terdapat *missing value*.

2.5 Accuracy

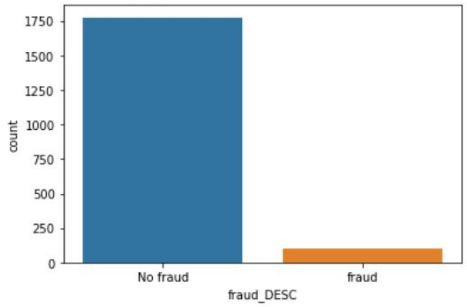
Dataset ini memiliki informasi data yang sesuai dan valid dimana, sebelumnya data ini berasal dari website data mining cup 2019. Untuk itu, dalam akurasi pengukuran jenis dataset yang kita inginkan ini dapat dipastikan sesuai dengan data sebenarnya sehingga, menghasilkan akurasi yang lebih baik.

2.6 Consistency

Konsisten dalam dataset ini sudah bagus pada karena, dapat memahami isi dari informasi data tersebut. Kemudian dalam dataset ini melakukan pengecekan berapa banyak produk yang mengandung fraud atau tidak di kolom 'fraud' dapat diamati, sebagai berikut.

	fraud	fraud_DESC
0	0	No fraud
1	0	No fraud
2	0	No fraud
3	0	No fraud
4	0	No fraud

```
Jumlah produk yang memiliki No fraud :  1775
Jumlah produk yang memiliki fraud    :  104
```



2.7 Interpretability

Interpretabilitas dalam dataset ini menunjukkan data yang berkualitas karena terdapat informasi data yang tidak memiliki duplikasi.

3. Preprocessing

Tahapan selanjutnya yang dilakukan adalah preprocessing data. Tahapan yang kami lakukan yaitu dengan melakukan LabelEncoder() yang berisi kolom 'fraud' yang telah di drop kolomnya dan direduksi dengan tsne. Berikut ini hasil dataset yang telah dilakukan preprocessing sebelumnya beserta hasil grafiknya.

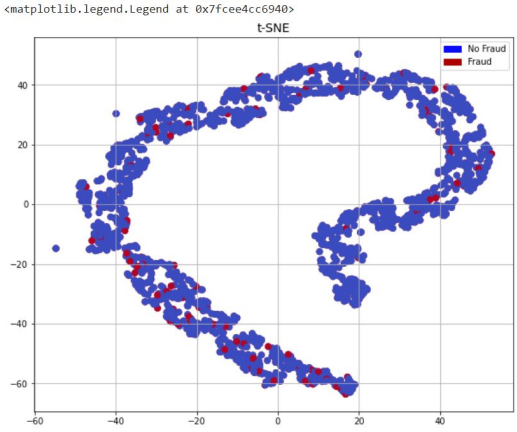
```
x, y = data.drop('fraud', axis=1), data['fraud']
```

```
from sklearn.manifold import TSNE
tsne = TSNE(n_components=2)
X_tsne = tsne.fit_transform(x)
```

```
from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
y = le.fit_transform(data.fraud)
set(y)
```

```
{0, 1}
```

```
<matplotlib.legend.Legend at 0x7fcee4cc6940>
```



4. Metode Klasifikasi

4.1 Logistic Regression

Algoritma Logistic Regression merupakan sebuah model yang digunakan untuk melakukan prediksi apakah sesuatu bernilai benar atau salah (0 atau 1). Metode Logistic Regression sebagai salah satu pendekatan Machine Learning untuk membantu suatu pihak dalam melakukan analisa terkait prediksi diantara beberapa variabel tersebut sehingga bisa dipakai untuk pengambilan keputusan ke depannya.

4.2 Tahap yang dilakukan

- Melakukan prediksi dengan split data menggunakan perbandingan 0.8 dengan 0.2

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(x, y, test_size=0.2)
```

Setelah itu, melakukan klasifikasi dengan algoritma yang digunakan bersamaan dengan feature selection yaitu RFE untuk melihat 13 hasil fitur apa saja yang ditampilkan.

```
from sklearn.linear_model import LogisticRegression
from sklearn.feature_selection import RFE


for n in range(1, len(list(x)) + 1):
    clasiffication = LogisticRegression(solver='lbfgs', multi_class='multinomial', max_iter=1000)
    result_features = RFE(estimator=clasiffication, n_features_to_select=n, step=1).fit(x,y)
    print("Result features: " + str(n) + ":")
    print(x.columns[result_features.get_support()].values)
    print()
```

```
Result features: 1:
['valuePerSecond']

Result features: 2:
['trustLevel' 'valuePerSecond']

Result features: 3:
['trustLevel' 'valuePerSecond' 'lineItemVoidsPerPosition']

Result features: 4:
['trustLevel' 'valuePerSecond' 'lineItemVoidsPerPosition'
 'CountTimePerSecond']

Result features: 5:
```

- Melakukan training data dengan Logistic Regression dan menampilkan confusion matrix serta, menghasilkan akurasi sebanyak 0.98. Sedangkan, untuk klasifikasi menggunakan ensemble Logistic Regression dengan metode AdaBoost mendapatkan akurasi 1.00

```
from sklearn.metrics import classification_report, confusion_matrix

y_pred = result_features.predict(X_test)
print(confusion_matrix(y_test, y_pred))
print(classification_report(y_test, y_pred))
```

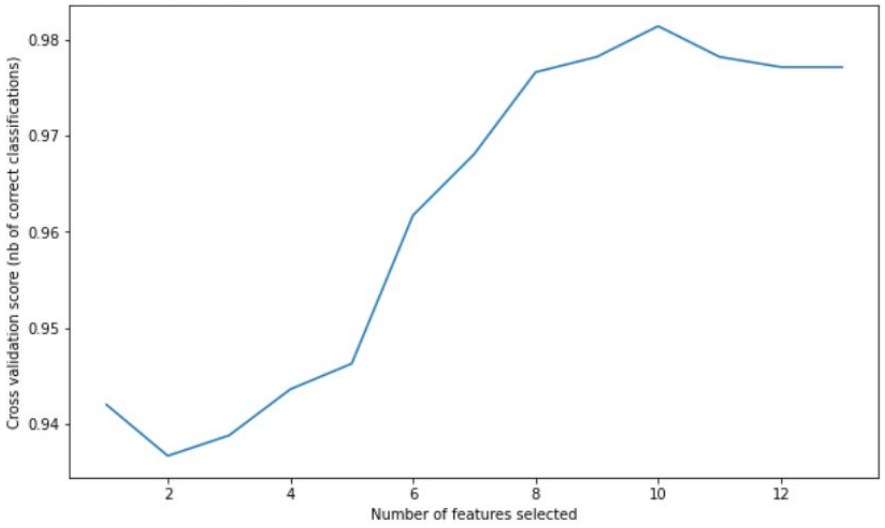
```
[[347   4]
 [  4  21]]
              precision    recall  f1-score   support

           0       0.99      0.99      0.99       351
           1       0.84      0.84      0.84        25

    accuracy                           0.98       376
   macro avg       0.91      0.91      0.91       376
weighted avg       0.98      0.98      0.98       376
```

```
[[354   0]
 [  1  21]]
              precision    recall  f1-score   support

           0       1.00      1.00      1.00       354
           1       1.00      0.95      0.98        22

    accuracy                           1.00       376
   macro avg       1.00      0.98      0.99       376
weighted avg       1.00      1.00      1.00       376
```

- Mencoba mengklasifikasi kembali dengan menggunakan ensemble Logistic Regression yang berfokus pada metode AdaBoost dan feature selection dengan RFECV untuk melihat hasil fitur manakah yang terbaik, yaitu fitur 10. Berikut ini diperoleh hasil klasifikasinya beserta tampilan grafik scorenya.

```
from sklearn.feature_selection import RFECV
from sklearn.ensemble import AdaBoostClassifier


classifier = RFECV(AdaBoostClassifier(), min_features_to_select=1, cv=10, scoring='accuracy')
classifier = classifier.fit(x,y)
print(str(classifier.n_features_) + " features:")
print(np.array(list(x))[np.array(classifier.support_)])
plt.figure(figsize=(10,6))
plt.xlabel("Number of features selected")
plt.ylabel("Cross validation score (nb of correct classifications)")
plt.plot(range(1, len(classifier.grid_scores_) + 1), classifier.grid_scores_)
plt.show()
```

```
10 features:
['trustLevel' 'totalScanTimeInSeconds' 'lineItemVoids'
 'scansWithoutRegistration' 'scannedLineItemsPerSecond' 'valuePerSecond'
 'lineItemVoidsPerPosition' 'ProductScan' 'CountTimeScan'
 'CountTimePerSecond']
```



5. Evaluation

Untuk tahap ini, melakukan perhitungan cost dengan TP, TN, FN, dan FP dimana, (-25)*fp + (-5)*fn + 5*tp + 0*tn. Kemudian, dapat dilihat hasilnya sebagai berikut.

```
TN:  1774
FP:  1
FN:  4
TP:  100
455 € is cost
0.24215007982969664 the average result
```

6. Link Youtube: https://youtu.be/Tj4xC0o-CU0