

Laporan Project Based Machine Learning

Disusun Untuk Memenuhi Tugas Mata Kuliah Pembelajaran Mesin
yang diampu oleh Tjokorda Agung Budi Wirayuda, S.T., M.T.



Disusun Oleh :

Muhammad Ghiyaats Daffa - 1301204068

Muhammad Fachry Gunawan - 1301204504

Fadli Zuhri - 1301202613

Syamaidzar Dwi Novtiar - 1301204273

Program Studi Informatika

Fakultas Informatika

Universitas Telkom

2022/2023

KATA PENGANTAR

Puji syukur atas rahmat Allah SWT, berkat rahmat serta karunia-Nya sehingga laporan tugas dari mata kuliah pembelajaran mesin dengan dapat diselesaikan dengan benar dan tepat waktu.

Laporan ini dibuat dengan tujuan memenuhi tugas dari Bapak Tjokorda Agung Budi Wirayuda, S.T., M.T pada kelas mata kuliah pembelajaran mesin. Selain itu, penyusunan laporan ini bertujuan memenuhi salah satu tugas Pembelajaran Mesin mengenai implementasi kmeans/dbscan/hierarchical.

Penulis menyampaikan ucapan terima kasih kepada Bapak Tjokorda Agung Budi Wirayuda, S.T., M.T.. sebagai dosen pembimbing mata kuliah pembelajaran mesin. Penulis juga mengucapkan terima kasih yang sebesar-besarnya kepada semua pihak yang membantu dalam proses penyusunan laporan ini.

Penulis menyadari bahwa dalam penyusunan dan penulisan masih melakukan banyak kesalahan. Oleh karena itu, penulis memohon maaf atas kesalahan dan ketidaksempurnaan yang pembaca temukan dalam laporan ini. Penulis juga mengharap adanya kritik serta saran dari pembaca apabila menemukan kesalahan dalam laporan ini.

Bandung, 8 Januari 2023

Kelompok

BAB 1

PENDAHULUAN

1. Project Based

Project-based assignment diberikan untuk memberi kesempatan kepada mahasiswa untuk mengeksplor pembelajaran mesin menggunakan metode-metode ensemble (bagging dan boosting) baik untuk keperluan klasifikasi maupun regresi pada datasets dari dunia nyata.

2. Deskripsi & Tujuan Tugas

Tipe tugas yang kami dapatkan adalah tipe 1, karena NIM terkecil dikelompok kami di modulo 4 adalah 1. Tugas tipe 1 merupakan tugas klasifikasi. Tugas klasifikasi adalah tugas menebak kelas/kategori dari calon kreditur (apakah akan menjadi kreditur yang baik atau buruk) berdasarkan profil calon kreditur yang diberikan yang diwakili oleh atribut-atribut-atribut seperti status pekerjaan, status perkawinan, tujuan kredit, usia, jenis kelamin, dll.

BAB II

PEMBAHASAN

1. Formulasi masalah

Masalah yang kami coba selesaikan merupakan menebak kelas/kategori dari calon kreditur (apakah akan menjadi kreditur yang baik atau buruk) berdasarkan profil calon kreditur yang diberikan yang diwakili oleh atribut-atribut seperti status pekerjaan, status perkawinan, tujuan kredit, usia, jenis kelamin, dll.

Kami menggunakan metode ensemble bagging untuk menyelesaikan masalah ini. Metode bagging sendiri adalah sebuah teknik pembelajaran mesin yang bertujuan untuk meningkatkan kecermatan atau akurasi dari suatu model pembelajaran mesin dengan membuat beberapa model yang terdiri dari bagian-bagian dari data training yang dipilih secara acak dan kemudian diagregasikan.

Secara umum, proses bagging melibatkan pembuatan beberapa model yang dibangun dengan menggunakan data training yang diambil secara acak dengan menggunakan sampling dengan replacement. Setelah model-model tersebut dibangun, hasil dari masing-masing model tersebut dapat diagregasikan dengan menggunakan teknik seperti voting untuk menentukan hasil akhir dari model.

Metode bagging dapat digunakan untuk meningkatkan akurasi dari model pembelajaran mesin dengan cara mengurangi variasi dari model yang dihasilkan. Dengan menggunakan beberapa model yang dibangun dengan menggunakan data yang berbeda, metode ini dapat mengurangi overfitting dan memperbaiki generalisasi model terhadap data yang belum pernah dilihat sebelumnya.

2. Eksplorasi dan pra-pemrosesan data

Mengimport library yang dibutuhkan dalam penyelesaian maslaah.

```
1 import pandas as pd
2 import numpy as np
3 from sklearn.tree import DecisionTreeClassifier
4 from sklearn.preprocessing import LabelEncoder
5 from sklearn.model_selection import train_test_split
6 from sklearn.metrics import accuracy_score
7 from scipy.stats import mode
```

Import data ke dalam Google Colab menggunakan pandas.

```
[ ] data = pd.read_excel("https://github.com/SyamDN/project-based-german-credit/blob/main/german_credit.xlsx?raw=true")
data
```

	status	duration	credit_history	purpose	amount	savings	employment_duration	installment_rate	personal_status_sex	other_debtors	...	property
0	no checking account	18	all credits at this bank paid back duly	car (used)	1049	unknown/no savings account	< 1 yr	< 20	female : non-single or male : single	none	...	car or other
1	no checking account	9	all credits at this bank paid back duly	others	2799	unknown/no savings account	1 <= ... < 4 yrs	25 <= ... < 35	male : married/widowed	none	...	unknown / no property
2	... < 0 DM	12	no credits taken/all credits paid back duly	retraining	841	... < 100 DM	4 <= ... < 7 yrs	25 <= ... < 35	female : non-single or male : single	none	...	unknown / no property

Terdapat 1000 baris dan 21 kolom pada dataset German credit.

Pengecekan nilai null.

```
print(data.isnull().sum())
```

status	0
duration	0
credit_history	0
purpose	0
amount	0
savings	0
employment_duration	0
installment_rate	0
personal_status_sex	0
other_debtors	0
present_residence	0
property	0
age	0
other_installment_plans	0
housing	0
number_credits	0
job	0
people_liable	0
telephone	0
foreign_worker	0
credit_risk	0
dtype:	int64

Pengecekan tipe data

```
data.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 21 columns):
#   Column                Non-Null Count  Dtype
---  -
0   status                 1000 non-null   object
1   duration               1000 non-null   int64
2   credit_history         1000 non-null   object
3   purpose                1000 non-null   object
4   amount                1000 non-null   int64
5   savings               1000 non-null   object
6   employment_duration   1000 non-null   object
7   installment_rate      1000 non-null   object
8   personal_status_sex   1000 non-null   object
9   other_debtors         1000 non-null   object
10  present_residence     1000 non-null   object
11  property              1000 non-null   object
12  age                   1000 non-null   int64
13  other_installment_plans 1000 non-null   object
14  housing               1000 non-null   object
15  number_credits        1000 non-null   object
16  job                   1000 non-null   object
17  people_liable         1000 non-null   object
18  telephone             1000 non-null   object
19  foreign_worker        1000 non-null   object
20  credit_risk           1000 non-null   object
dtypes: int64(3), object(18)
memory usage: 164.2+ KB
```

Dari potongan code diatas, dapat dilihat masih banyak tipe data object pada kolom dataset German credit yang harus diubah menjadi integer agar lebih mudah diolah nantinya.

Drop kolom 'number_credits'.

```
data = data.drop('number_credits', axis=1)
```

Kami menghapus kolom 'number_credits' karena kolom tersebut menurut kami tidak memiliki informasi yang berkaitan / bermanfaat yang nantinya dapat menyebabkan model menjadi kurang akurat.

Splitting data

```
data_sc = data.iloc[:, :-1]
data_target = data.iloc[:, -1]
```

Kami melakukan splitting data menjadi 2 yaitu data source (data_sc) dan data target.

```
1 # Membagi data menjadi data training dan data testing dengan proporsi 70% data training dan 30% data testing
2 data_sc_train, data_sc_test, data_target_train, data_target_test = train_test_split(data_sc, data_target, test_size=0.3)
```

Selanjutnya kami membagi menjadi data train dan test dengan pembagian 70% untuk train dan 30% untuk test.

Mengubah data kategorikal menjadi numerik

```
# Buat objek LabelEncoder
le = LabelEncoder()

# Aplikasi label encoding pada kolom yang diinginkan
data_sc['status'] = le.fit_transform(data_sc['status'])
data_sc['credit_history'] = le.fit_transform(data_sc['credit_history'])
data_sc['purpose'] = le.fit_transform(data_sc['purpose'])
data_sc['savings'] = le.fit_transform(data_sc['savings'])
data_sc['employment_duration'] = le.fit_transform(data_sc['employment_duration'])
data_sc['installment_rate'] = le.fit_transform(data_sc['installment_rate'])
data_sc['personal_status_sex'] = le.fit_transform(data_sc['personal_status_sex'])
data_sc['other_debtors'] = le.fit_transform(data_sc['other_debtors'])
data_sc['present_residence'] = le.fit_transform(data_sc['present_residence'])
data_sc['property'] = le.fit_transform(data_sc['property'])
data_sc['other_installment_plans'] = le.fit_transform(data_sc['other_installment_plans'])
data_sc['housing'] = le.fit_transform(data_sc['housing'])
#data_sc['number_credits'] = le.fit_transform(data_sc['number_credits'])
data_sc['job'] = le.fit_transform(data_sc['job'])
data_sc['people_liable'] = le.fit_transform(data_sc['people_liable'])
data_sc['telephone'] = le.fit_transform(data_sc['telephone'])
data_sc['foreign_worker'] = le.fit_transform(data_sc['foreign_worker'])
data_sc
```

	status	duration	credit_history	purpose	amount	savings	employment_duration	installment_rate	personal_status_sex	other_debtors	present_residence	property
0	3	18	0	2	1049	4	2	2	0	2	3	1
1	3	9	0	5	2799	4	0	1	3	2	0	3
2	0	12	4	8	841	0	1	1	0	2	3	3
3	3	12	0	5	2122	4	0	0	3	2	0	3
4	3	12	0	5	2171	4	0	2	3	2	3	1
...
995	3	24	4	4	1987	4	0	1	3	2	3	3
996	3	24	4	5	2303	4	3	2	3	0	2	3
997	1	21	0	5	12680	1	3	2	3	2	3	2
998	0	12	4	4	6468	1	4	1	3	2	2	2
999	3	30	4	2	6350	1	3	2	3	2	3	1

Pada penjelasan sebelumnya, tipe data pada dataset German credit memiliki tipe data object yang masih banyak, sehingga perlu kita ubah menjadi integer / numerik. Kami melakukan ini agar data lebih mudah diolah dan bisa diperiksa variansi datanya.

3. Pemodelan

```
1 n_tree = 100
2 np.random.seed(1301204273)
3 forest = []
4 bootstrap_columns = []
5 for i in range(n_tree):
6     # 1. Bootstrapping
7     # select random rows
8     rows = np.random.randint(len(data_sc_train), size=len(data_sc_train))
9     x_bootstrap = data_sc_train.iloc[rows]
10    y_bootstrap = data_target.iloc[rows]
11
12    # Out of Bag data
13    non_selected_rows = list(set(range(len(data_sc_train))) - set(rows))
14    x_oob = data_sc_train.iloc[non_selected_rows]
15    y_oob = data_target.iloc[non_selected_rows]
16
17    print("Row that are not selected : ",len(x_oob))
18
19    # 2. Select random features (make sure to keep track the selected columns)
20    # Check for any repeated combinations
21    feature_taken = np.random.randint(2, 4)
22    cols = np.random.choice(data_sc_train.columns, feature_taken, replace=False)
23    bootstrap_columns.append(cols)
24
25    # filter based on the selected columns
26    x_bootstrap = x_bootstrap[cols]
27
28    # 3. Build decision tree
29    tree_model = DecisionTreeClassifier()
30    tree_model.fit(x_bootstrap,y_bootstrap)
31
32    # Add it to the forest
33    forest.append(tree_model)
34
35    # 4. Calculate oob score only within its own tree
36    oob_score = tree_model.score(x_oob[cols], y_oob)
37    print("OOB Score : ",oob_score)
```

Selanjutnya, kami membuat model menggunakan random forest yang terdiri dari 100 decision tree. Kami set seed menggunakan salah satu NIM anggota kami dan membuat array kosong yang kami beri nama forest dan bootstrap_columns. Forest ini nantinya akan diisi oleh model-model decision tree yang dihasilkan, sedangkan bootstrap_columns akan diisi oleh kolom yang terpilih secara random dalam pembuatan bootstrap dataset. Lalu kami melakukan perulangan sebanyak jumlah tree yang diinginkan. Di dalam perulangan, kami

membuat bootstrap dataset berulang kali secara random dan memakai dataset tersebut untuk membuat model decision tree menggunakan library “DecisionTreeClassifier”.

Setelah itu, model yang telah dihasilkan kami masukkan ke dalam array forest. Lalu kami cek OOB score nya.

4. Evaluasi

Karena kami mendapatkan tugas untuk melakukan bagging, kami mengumpulkan terlebih dahulu semua hasil prediksi dari tree yang ada, lalu nantinya kami akan melakukan voting terhadap hasil tersebut untuk mendapat hasil akhir sesuai dengan cara kerja metode bagging.

```
1 all_preds = []
2 for i, tree_model in enumerate(forest):
3     # Each tree has different column requirements
4     # Make sure you use the correct columns for each tree
5     data_sc_test_filtered = data_sc_test[bootstrap_columns[i]]
6
7     predictions = tree_model.predict(data_sc_test_filtered)
8
9     # add it to all_preds for voting later
10    all_preds.append(predictions)
11
12 all_preds = np.array(all_preds)
```

Disini kami membuat array kosong all_preds, yang nantinya akan diisi oleh hasil prediksi semua pohon yang ada pada forest. Pada perulangan tersebut, kami melakukan prediksi menggunakan model yang telah dibuat dan dimasukkan ke dalam array all_preds.

```

1 # Insert voting code here
2 voted_predictions = mode(all_preds, axis=0)[0][0]
3
4 # Calculate accuracy of Test data on the Forest
5 acc = accuracy_score(data_target_test, voted_predictions)
6 acc

```

Lalu kami melakukan voting dan mengoutputkan hasil akurasi akhir.

0.7033333333333334

Didapat hasil akurasi akhir adalah 70,34%. Hasil akurasi ini didapatkan berdasarkan hasil voting yang telah kami lakukan.

5. Eksperimen

Untuk eksperimen, karena metode yang kami dapatkan adalah bagging, maka eksperimen yang kami lakukan adalah dengan cara mengubah dataset yang nantinya akan dilakukan bootstrapping.

Eksperimen	Hasil Akurasi
Drop kolom “number_credits”, “telephone”, “age” <pre> 1 data = data.drop('number_credits', axis=1) 2 data = data.drop('telephone', axis=1) 3 data = data.drop('age', axis=1) </pre>	0.7266666666666667
Drop kolom “number_credits”, “people_liable”, “personal_status_sex”, “other_installment_plans” <pre> 1 data = data.drop('number_credits', axis=1) 2 data = data.drop('people_liable', axis=1) 3 data = data.drop('personal_status_sex', axis=1) 4 data = data.drop('other_installment_plans', axis=1) </pre>	0.6933333333333334
Drop kolom “number_credits”, “job”, “present_residence”, “property”	0.7266666666666667

```
data = data.drop('number_credits', axis=1)
data = data.drop('job', axis=1)
data = data.drop('present_residence', axis=1)
data = data.drop('property', axis=1)
```

BAB III

KESIMPULAN

Berdasarkan experiment yang kami lakukan, kami mendapatkan hasil terbesar adalah 0.726667 yang didapat dengan menghapus kolom number_credits, telephone, age dan menghapus kolom number_credits, job, present_residence dan property

Lampiran :

Video Presentasi :

https://telkomuniversityofficial.sharepoint.com/sites/ProjectBased/_layouts/15/stream.aspx?id=%2Fsites%2FProjectBased%2FDokumen%20Berbagi%2FGeneral%2FRecordings%2FBased-20230108_215131-Meeting%20Recording%2Emp4

Link colab :

https://colab.research.google.com/drive/1DmF-IVx91t2fSN0z_lySCvSHoYefBh9w?usp=sharing