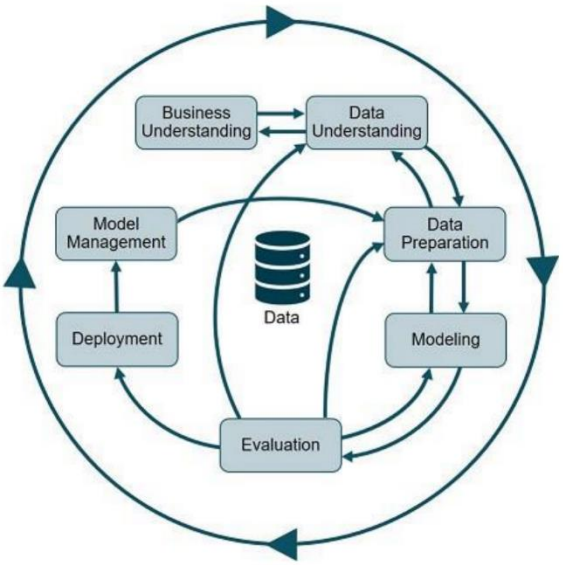
PENILAIN KREDIT BISNIS PADA BANK

oleh Julio Andarestu

Pada zaman yang semakin modern ini semakin banyak orang menggunakan jasa bank untuk menyimpan uang yang mereka hasilkan agar tersimpan dengan aman. Bank bukan hanya tempat untuk menyimpan uang tapi para nasabah juga bisa untuk melakukan pinjaman uang dengan beberapa persyaratan dan itu harus disetujui oleh kedua belah pihak. Apabila terdapat nasabah yang ingin mengajukan pinjaman maka harus melampirkan beberapa persyaratan dokumen yang harus diberikan kepada pihak bank. Persyaratan yang telah dilampirkan tadi tentunya menjadi pertimbangan pihak bank untuk menyetujui pinjaman yang diajukan oleh nasabah dengan cara melihat jaminan apa yang bisa diberi oleh nasabah yang mengajukan pinjaman. Jaminan tersebut bisa dilihat dengan cara menganalisis dan menentukan penilain kredit pada nasabah. Terdapat beberapa faktor yang akan mempengaruhi penilain kredit tersebut hingga nasabah tersebut bisa dikatakan layak atau tidak layak untuk melakukan pinjaman. Pihak bank pada umumnya melakukan penilaian kredit ini dengan menggunakan suatu metode dalam ilmu komputer yaitu Data Mining. Metode Data Mining adalah proses analisa yang dilakukan secara otomatis pada data yang kompleks dan berjumlah besar untuk memperoleh sebuah pola atau kecenderungan yang umumnya tidak disadari (Pramudiono)[1].

Terdapat standar apabila ingin menganalisis penilain kredit nasabah yaitu biasa disebut Cross-Industry Standard Process for Data Mining (CRISP-DM) merupakan salah satu model proses datamining (datamining framework) yang awalnya (1996) dibangun oleh 5 perusahaan yaitu Integral Solutions Ltd (ISL), Teradata, Daimler AG, NCR Corporation dan OHRA. Framework ini kemudian dikembangan oleh ratusan organisasi dan perusahaan di Eropa untuk dijadikan methodology standard non-proprietary bagi data mining[2]. CRIPS-DM itu sendiri memiliki fase dalam suatu siklus hidup proyek data mining yang dapat dilihat pada gambar berikut ini.



Gambar Proses Data Mining CRISP-DM

CRISP-DM terdapat 6 fase yang dimulai dengan Business Understanding, Data Understanding, Data Preparation, Modeling, Evaluation, dan Deployment. Apabila diimplementasikan dalam penilaian kredit pada bank maka dimulai dengan :

1. **Business Understanding**

   Pada fase ini menentukan produk yang ingin diteliti yaitu memberi pinjaman pada *customer*, menentukan kebutuhan yaitu butuh *data science* agar mengurangi risiko untuk memberikan pinjaman ke *customer* yang tidak bisa bayar dan bisa tahu customer yang layak menerima pinjaman, menentukan *objective* yaitu memprediksi *score* seseorang apakah layak untuk diberikan pinjaman dan menentukan tujuannya yaitu meminimalisir *cost* yang dikeluarkan.

2. **Data Understanding**

   Setelah melalui fase *Business* understanding maka pihak bank harus memahami data yang telah didapatkan seperti mengidentifikasi pengetahuan awal dan menilai data tersebut apakah sudah sesuai dengan apa yang dibutuhkan. Data didapatkan dari *Kaggle* sebanyak 10.000.

```
[2]: data = pd.read_csv('Churn_Modelling.csv')
```

```
[63]: data
```

[63]:

	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSalary	Exited
0	1	15634602	Hargrave	619	France	Female	42	2	0.00	1	1	1	101348.88	1
1	2	15647311	Hill	608	Spain	Female	41	1	83807.86	1	0	1	112542.58	0
2	3	15619304	Onio	502	France	Female	42	8	159660.80	3	1	0	113931.57	1
3	4	15701354	Boni	699	France	Female	39	1	0.00	2	0	0	93826.63	0
4	5	15737888	Mitchell	850	Spain	Female	43	2	125510.82	1	1	1	79084.10	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
9995	9996	15606229	Obijiaku	771	France	Male	39	5	0.00	2	1	0	96270.64	0
9996	9997	15569892	Johnstone	516	France	Male	35	10	57369.61	1	1	1	101699.77	0
9997	9998	15584532	Liu	709	France	Female	36	7	0.00	1	0	1	42085.58	1
9998	9999	15682355	Sabbatini	772	Germany	Male	42	3	75075.31	2	1	0	92888.52	1
9999	10000	15628319	Walker	792	France	Female	28	4	130142.79	1	1	0	38190.78	0

10000 rows × 14 columns

Atribut Data

```
[6]: data.columns
```

```
[6]: Index(['RowNumber', 'CustomerId', 'Surname', 'CreditScore', 'Geography',
       'Gender', 'Age', 'Tenure', 'Balance', 'NumOfProducts', 'HasCrCard',
       'IsActiveMember', 'EstimatedSalary', 'Exited'],
      dtype='object')
```

Pada gambar dapat diketahui terdapat atribut RowNumber, CustomerID, Surname, CreditScore, Geography, Gender, Age, Tenure, Balance, NumOfProducts, HasCrCard, IsActiveMember, EstimatedSalary, Exited dari dataset *Churn_Modelling*.

3. Data Preparation

Pada tahap ini pihak bank harus membersihkan data yang telah dikumpulkan pada tahap sebelumnya. Melakukan pembersihan apabila terdapat data yang kosong atau data yang tidak digunakan. Hal ini dapat memudahkan pihak bank untuk melakukan modeling pada tahap berikutnya apabila datanya sudah bersih dan sesuai dengan apa yang dibutuhkan. Pada tahap ini, mungkin ada atau tidak ada masalah dengan data margin Anda. Catatan yang digunakan dan duplikat. Untuk alasan ini, teknik pretreatment berikut diperlukan.

Data Cleaning

Membantu membersihkan nilai kosong, tupel yang tidak konsisten atau berpotensi kosong (nilai dan noise yang hilang).

Mencari Missing Value

```
[5]: data.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10000 entries, 0 to 9999
Data columns (total 14 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   RowNumber        10000 non-null  int64
 1   CustomerId       10000 non-null  int64
 2   Surname          10000 non-null  object
 3   CreditScore      10000 non-null  int64
 4   Geography        10000 non-null  object
 5   Gender           10000 non-null  object
 6   Age              10000 non-null  int64
 7   Tenure           10000 non-null  int64
 8   Balance          10000 non-null  float64
 9   NumOfProducts    10000 non-null  int64
 10  HasCrCard        10000 non-null  int64
 11  IsActiveMember   10000 non-null  int64
 12  EstimatedSalary  10000 non-null  float64
 13  Exited           10000 non-null  int64
dtypes: float64(2), int64(9), object(3)
memory usage: 1.1+ MB
```

Pada gambar dapat dijelaskan bahwa semua atribut lengkap berjumlah 10.000 tanpa adanya yang bernilai Null. Adapun untuk tipe data juga dapat dijelaskan bahwa sudah benar semuanya.

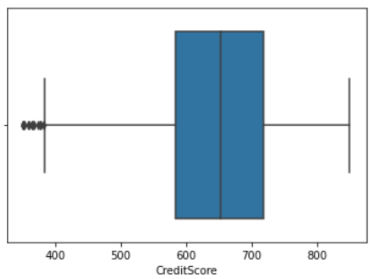
```
[7]: data.isna().sum()

[7]: RowNumber          0
     CustomerId         0
     Surname            0
     CreditScore        0
     Geography          0
     Gender             0
     Age                0
     Tenure             0
     Balance            0
     NumOfProducts      0
     HasCrCard          0
     IsActiveMember     0
     EstimatedSalary    0
     Exited             0
     dtype: int64
```

Dapat dilihat bahwa tidak ada atribut yang datanya tidak ada nilai atau kosong.

Mencari Outliers

Atribut Credit Score

```
[10]: fig = plt.figure()
      ax = sns.boxplot(data = data, x = 'CreditScore')
      plt.show()
```



Pada gambar dapat dilihat bahwa terdapat outliers pada atribut Credit Score

Atribut Credit Score Menggunakan Z Score

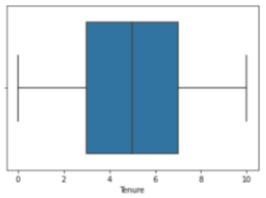
```
[15]: q1 = data["CreditScore"].quantile(0.25)
      q3 = data["CreditScore"].quantile(0.75)

      iqr = q3-q1
      fence_low  = q1-1.5*iqr
      fence_high = q3+1.5*iqr

      data.loc[(data["CreditScore"] < fence_low) | (data["CreditScore"] > fence_high)]
```

	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSalary	Exited
7	8	15656148	Obinna	376	Germany	Female	29	4	115046.74	4	1	0	119346.88	1
942	943	15804586	Lin	376	France	Female	46	6	0.00	1	1	0	157333.69	1
1193	1194	15779947	Thomas	363	Spain	Female	28	6	146098.43	3	1	0	100615.14	1
1405	1406	15612494	Panicucci	359	France	Female	44	6	128747.69	1	1	0	146955.71	1
1631	1632	15685372	Azubuike	350	Spain	Male	54	1	152677.48	1	1	1	191973.49	1
1838	1839	15758813	Campbell	350	Germany	Male	39	0	109733.20	2	0	0	123602.11	1
1962	1963	15692416	Aikenhead	358	Spain	Female	52	8	143542.36	3	1	0	141959.11	1
2473	2474	15679249	Chou	351	Germany	Female	57	4	163146.46	1	1	0	169621.69	1
2579	2580	15597896	Ozoemena	365	Germany	Male	30	0	127760.07	1	1	0	81537.85	1
8154	8155	15791533	Ch'ien	367	Spain	Male	42	6	93608.28	1	1	0	168816.73	1
8723	8724	15803202	Onyekachi	350	France	Male	51	10	0.00	1	1	1	125823.79	1
8762	8763	15765173	Lin	350	France	Female	60	3	0.00	1	0	0	113796.15	1
9210	9211	15792650	Watts	382	Spain	Male	36	0	0.00	1	1	1	179540.73	1
9356	9357	15734711	Loggia	373	France	Male	42	7	0.00	1	1	0	77786.37	1
9624	9625	15668309	Maslow	350	France	Female	40	0	111098.85	1	1	1	172321.21	1

Dapat dilihat bahwa pada test Z Score terhadap atribut Credit Score didapatkan hasil terdapat beberapa outliers tetapi tidak perlu dilakukan Tindakan karena merupakan data yang sah.

Atribut Tenure

```
[9]: fig = plt.figure()
     ax = sns.boxplot(data = data, x = 'Tenure')
     plt.show()
```



Dapat dilihat bahwa tidak terdapat outliers pada atribut Tenure.

Atribut Tenure Menggunakan Z Score

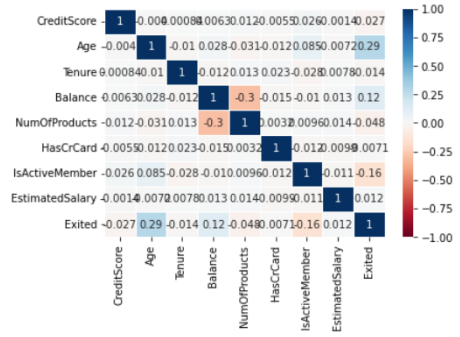
```
[14]: q1 = data["Tenure"].quantile(0.25)
      q3 = data["Tenure"].quantile(0.75)

      iqr = q3-q1
      fence_low  = q1-1.5*iqr
      fence_high = q3+1.5*iqr

      data.loc[(data["Tenure"] < fence_low) | (data["Tenure"] > fence_high)]
```

	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSalary	Exited

Dapat dilihat bahwa tidak terdapat outliers pada atribut Tenure saat dicoba menggunakan Z Score.

Melakukan Uji Korelasi

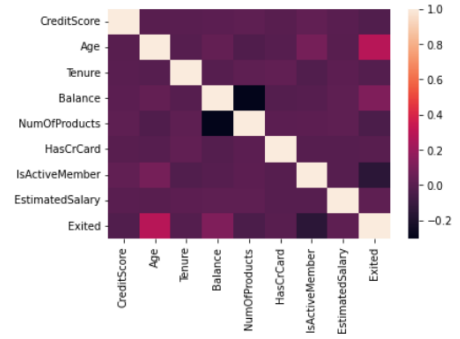
```
[17]: df = data[['CreditScore', 'Age', 'Tenure', 'Balance', 'NumOfProducts', 'HasCrCard', 'IsActiveMember',
                 'EstimatedSalary', 'Exited']]
      corr = df.corr()
      sns.heatmap(corr, cmap='RdBu', linecolor='white', linewidths=0.5, vmin=-1, vmax=1, annot=True)

[17]: <AxesSubplot:>
```



Dapat dilihat dari angka setiap korelasi antar atribut tidak ada yang mendekati satu yang berarti korelasi antar atribut rendah. Nilai tertinggi yaitu 0.3 yang mana berarti korelasinya rendah.

Mencari Korelasi Menggunakan Heat Map

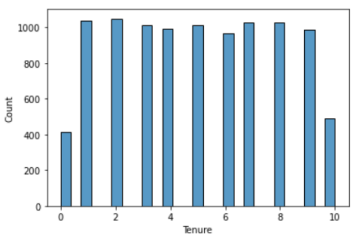
```
[16]: df = data[['CreditScore', 'Age', 'Tenure', 'Balance', 'NumOfProducts', 'HasCrCard', 'IsActiveMember',
                 'EstimatedSalary', 'Exited']]
      corr = df.corr()
      sns.heatmap(corr)

[16]: <AxesSubplot:>
```



Mencari Korelasi Menggunakan Visualiasasi

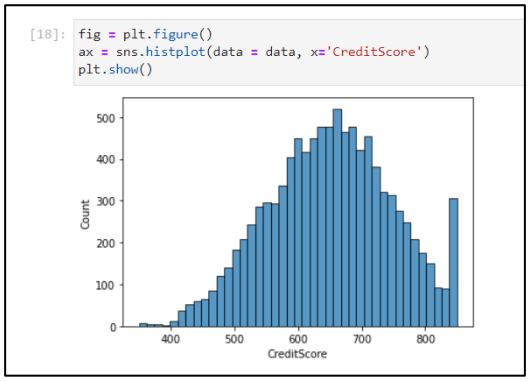
- **Atribut Tenure**

```
[20]: fig = plt.figure()
      ax = sns.histplot(data = data, x='Tenure')
      plt.show()
```



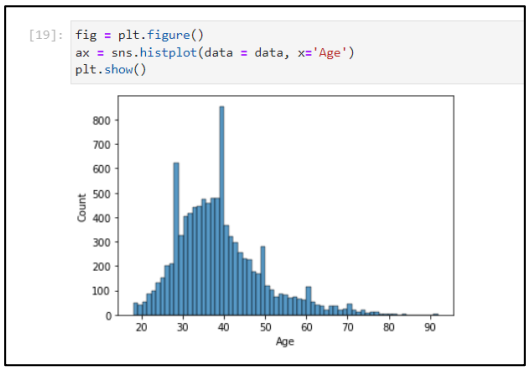
Dapat dilihat pelanggan dengan tenure lama dan singkat sedikit berdasarkan visualisasi diatas.

- **Atribut Credit Score**



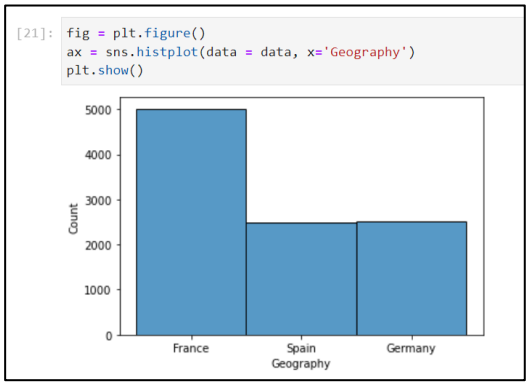
Dapat dilihat bahwa banyak pelanggan yang memiliki credit score tinggi.

- **Atribut Age**



Dapat dilihat bahwa pelanggan dengan umur 29-39 tahun banyak mendominasi pada data.

- **Atribut Geography**

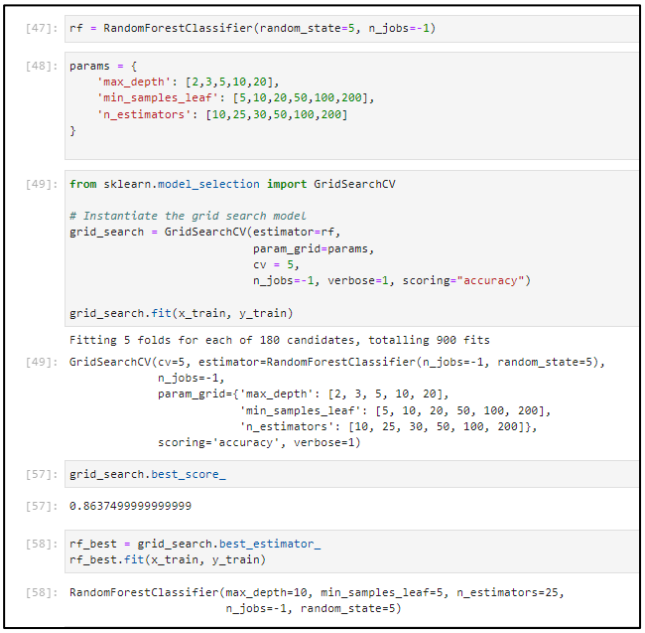


Dapat dilihat bahwa pelanggan yang berlokasi di negara France mendominasi data.

4. Modeling

Setelah data dibersihkan maka data siap untuk diolah dengan menggunakan algoritma yang selaras dengan apa yang dibutuhkan. Pada tahap ini juga pihak bank harus menentukan Teknik Data Mining yang ingin dilakukan. Secara garis besar terdapat empat Teknik yaitu klasifikasi, klusteriasi, forcasting dan estimasi. Setiap Teknik tersebut juga memiliki algoritma yang bisa digunakan untuk melakukan pemodelan. Pihak bank bisa melakukan perbandingan beberapa algoritma untuk mendapatkan hasil akruasi yang terbaik dari semua algoritma yang telah dilakukan pengujian. Adapun model yang digunakan pada tahap ini adalah Decision Tree, Logistic Regression, Hyperparameter Tuning for random forest dan Random Forest.

Tahap Modeling



```
[47]: rf = RandomForestClassifier(random_state=5, n_jobs=-1)

[48]: params = {
          'max_depth': [2,3,5,10,20],
          'min_samples_leaf': [5,10,20,50,100,200],
          'n_estimators': [10,25,30,50,100,200]
      }

[49]: from sklearn.model_selection import GridSearchCV

      # Instantiate the grid search model
      grid_search = GridSearchCV(estimator=rf,
                                 param_grid=params,
                                 cv = 5,
                                 n_jobs=-1, verbose=1, scoring="accuracy")

      grid_search.fit(x_train, y_train)

      Fitting 5 folds for each of 180 candidates, totalling 900 fits

[49]: GridSearchCV(cv=5, estimator=RandomForestClassifier(n_jobs=-1, random_state=5),
                   n_jobs=-1,
                   param_grid={'max_depth': [2, 3, 5, 10, 20],
                               'min_samples_leaf': [5, 10, 20, 50, 100, 200],
                               'n_estimators': [10, 25, 30, 50, 100, 200]},
                   scoring='accuracy', verbose=1)

[57]: grid_search.best_score_

[57]: 0.8637499999999999

[58]: rf_best = grid_search.best_estimator_
      rf_best.fit(x_train, y_train)

[58]: RandomForestClassifier(max_depth=10, min_samples_leaf=5, n_estimators=25,
                             n_jobs=-1, random_state=5)
```

5. Evaluation

Setelah melakukan pemodelan pada fase sebelumnya selanjutnya pihak bank harus melakukan evaluasi dari hasil pemodelan tersebut apakah sudah sesuai dengan apa yang dinginkan pada tahap awal atau tidak. Apabila sudah sesuai hasillnya maka

akan didapatkan sebuah keputusan untuk diambil. Evaluasi bisa menggunakan *Confussion Matrix, Classification Report, AUC.* Berikut hasil evaluasi dari ketiga model matik :

- *Confussion Matrix*

```
*[60]: from sklearn.metrics import confusion_matrix
       print("Logistic Regression : \n", confusion_matrix(y_test, y_lr))
       print("Decision Tree : \n", confusion_matrix(y_test, y_dtree))
       print("Random Forest  : \n", confusion_matrix(y_test, y_rf_before))
       print("Random Forest dengan Hyperparameter Tuning: \n", confusion_matrix(y_test, y_rf_after))

       Logistic Regression :
        [[1565   30]
        [ 388   17]]
       Decision Tree :
        [[1369  226]
        [ 193  212]]
       Random Forest  :
        [[1570   25]
        [ 284  121]]
       Random Forest dengan Hyperparameter Tuning:
        [[1546   49]
        [ 243  162]]
```

Dapat dilihat bahwa confussion matrix dari semua metode yang diujikan menghasilkan confussion matrix yang berbeda.

- *Classification Report*

```
[61]: from sklearn.metrics import classification_report
      print("Logistic Regression : \n\n", classification_report(y_test, y_lr))
      print("Decision Tree : \n\n", classification_report(y_test, y_dtree))
      print("Random Forest : \n\n", classification_report(y_test, y_rf_before))
      print("Random Forest dengan Hyperparameter Tuning: \n\n", classification_report(y_test, y_rf_after))

      Logistic Regression :

                    precision    recall  f1-score   support

                 0       0.80      0.98      0.88      1595
                 1       0.36      0.04      0.08       405

          accuracy                           0.79      2000
         macro avg       0.58      0.51      0.48      2000
      weighted avg       0.71      0.79      0.72      2000

      Decision Tree :

                    precision    recall  f1-score   support

                 0       0.88      0.86      0.87      1595
                 1       0.48      0.52      0.50       405

          accuracy                           0.79      2000
         macro avg       0.68      0.69      0.69      2000
      weighted avg       0.80      0.79      0.79      2000

      Random Forest :

                    precision    recall  f1-score   support

                 0       0.85      0.98      0.91      1595
                 1       0.83      0.30      0.44       405

          accuracy                           0.85      2000
         macro avg       0.84      0.64      0.67      2000
      weighted avg       0.84      0.85      0.81      2000

      Random Forest dengan Hyperparameter Tuning:

                    precision    recall  f1-score   support

                 0       0.86      0.97      0.91      1595
                 1       0.77      0.40      0.53       405

          accuracy                           0.85      2000
         macro avg       0.82      0.68      0.72      2000
      weighted avg       0.84      0.85      0.84      2000
```

Dapat dilihat pada gambar bahwa nilai *precision, recall, F1-Score* dan *support*.

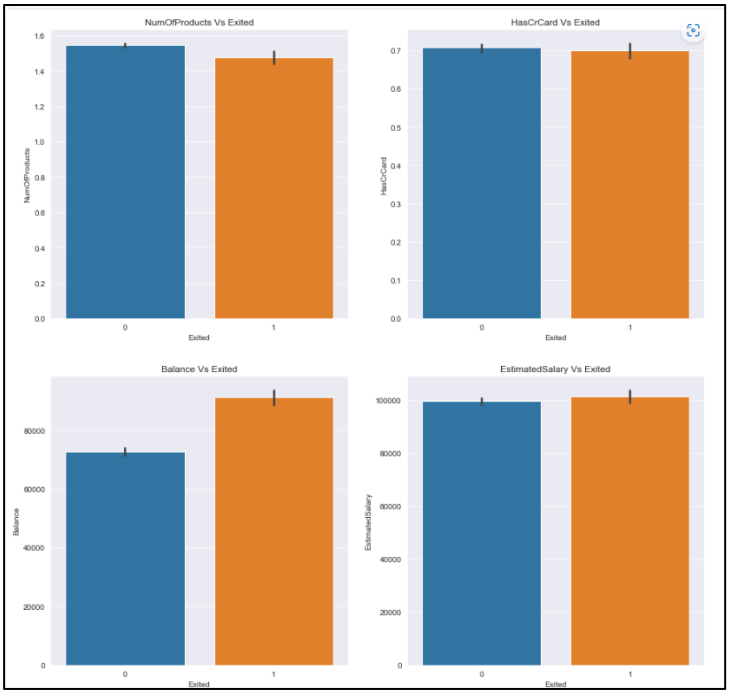
- *AUC*

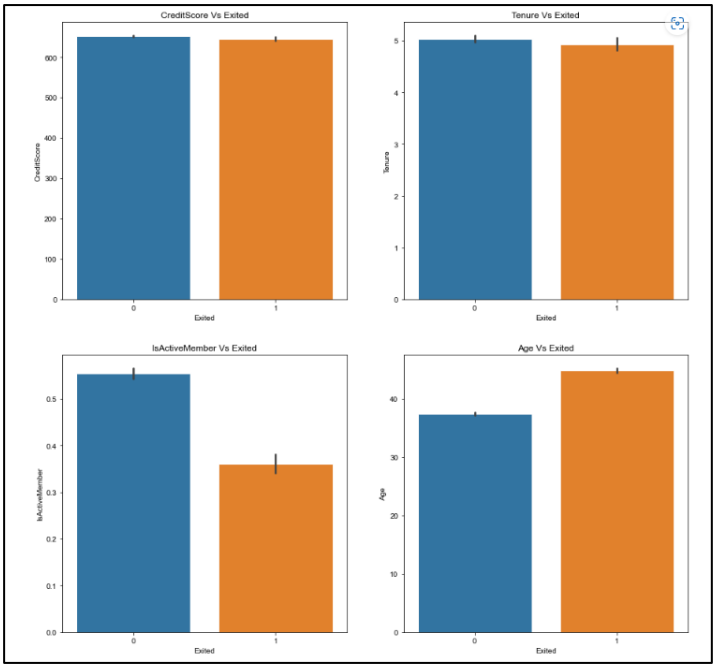
```
[62]: from sklearn.metrics import roc_curve, auc
fpr, tpr, thresholds = roc_curve(y_test, y_lr, pos_label=1) # pos_label: positive label
print("Logistic Regression :", auc(fpr, tpr))
fpr, tpr, thresholds = roc_curve(y_test, y_dtree, pos_label=1) # pos_label: positive label
print("Decision Tree :", auc(fpr, tpr))
fpr, tpr, thresholds = roc_curve(y_test, y_rf_before, pos_label=1) # pos_label: positive label
print("Random Forest :", auc(fpr, tpr))
fpr, tpr, thresholds = roc_curve(y_test, y_rf_after, pos_label=1) # pos_label: positive label
print("Random Forest dengan Hyperparameter Tuning:", auc(fpr, tpr))

Logistic Regression : 0.5115832656062542
Decision Tree : 0.6908820000774024
Random Forest : 0.6415457254537714
Random Forest dengan Hyperparameter Tuning: 0.6846394984326019
```

Dapat dilihat pada gambar untuk nilai Logistic Regerssion adalah 0.5115832656062542, Decision Tree adalah 0.6908820000774024, Random Forest adalah 0.6415457254537714 dan Random Forest dengan Hyperparameter Tuning adalah 0.6846394984326019.

6. Deployment

Apabila sebuah keputusan telah diambil maka hasil dari keputusan tadi harus disebarkan melalui sistem atau aplikasi kepada department yang mengurusi permasalahan peminjaman uang tersebut. Dapat berupa visualisasi yang ditampilkan pada sistem atau dashboard diwebsite.




CreditScore Vs Exited
CreditScore
Exited
Tenure Vs Exited
Tenure
Exited
IsActiveMember Vs Exited
IsActiveMember
Exited
Age Vs Exited
Age
Exited

Daftar Pustaka

[1] Apa itu Data Mining? Pengertian, Metode, Tahapan, dan Contoh Terbaru
https://info.populix.co/articles/data-mining-adalah

[2] Cross-Industry Standard Process for Data Mining (CRISP-DM)

https://mmsi.binus.ac.id/2020/09/18/cross-industry-standard-process-for-data-mining-crisp-dm/

Github Link Julio Andarestu : https://github.com/JulioAndarestu/Zenius#readme