![Title and text
](data:image/png;base64,iVBORw0KGgoAAAANSUhEUgAAAAEAAAABCAYAAAAfFcSJAAAADUlEQVR4XmP4//8XAwAI8gL5c60pfQAAAABJRU5ErkJggg==)

# Project Team

|  |
| --- |
| **Mentor:**    **Team:** |

# Project Timeline

|  |  |  |
| --- | --- | --- |
| **Stages** | **Topics** | **Agenda** |
| Stage 0 | * Problem * Roles * Goals * Objectives * Business Metrics | **June 11, 2021:**  Problem Identification |
| Stage 1 | * Problem Statement * Data Exploration * Data Cleansing * Feature Engineering | **June 13, 2021:**  1st Internal Meeting: Kick-Off, Create an internal Commitment.  **June 15, 2021:**  2nd Internal Meeting: Stage 1 progress discussion  **June 19, 2021:**  1st Mentoring: Stage 1 Review  **June 20, 2021:**  3rd Internal Meeting: Stage 1 finalization |
| Stage 2 | * Exploratory Data Analysis * Insight and Visualization | **June 23, 2021:**  4th Internal Meeting: Stage 2 progress discussion  **June 26, 2021:**  2nd Mentoring: Stage 2 Review  **June 27, 2021:**  5th Internal Meeting: Stage 2 Finalization |
| Stage 3 | * Modelling * Evaluation | **July 2, 2021:**  6th Internal Meeting: Stage 3 progress discussion  **July 3, 2021:**  3rd Mentoring: Stage 3 Review  **July 4, 2021:**  7th Internal Meeting: Stage 3 Finalization |
| Stage 4 | * Presentation Material * Speech Practice | **July 7, 2021:**  8th Internal Meeting: Stage 4 progress discussion  **July 8, 2021:**  4th Mentoring: Stage 4 Review  **July 9, 2021:**  9th Internal Meeting: Stage 4 Finalization |

# Problem Statement

To identify the problem, we look out the churn rate first:

|  |
| --- |
| Import pandas as pd  df = pd.read\_csv('Churn\_Modelling.csv')  churn\_rate = df[df['Exited']==1].shape[0] \*100 / df.shape[0]  print('Churn Rate: ', churn\_rate,'%') |

The result is churn rate ratio is **20.37%.**

Berikutnya kami mencari jumlah customer yang hilang:

|  |
| --- |
| churn = df[df['Exited']==1]  churn['CustomerId'].count() |

Jumlah customer yang keluar adalah **2037.**

Berikutnya kami mencari besarnya kerugian dana yang dialami bank JOGA:

|  |
| --- |
| churn['Balance'].sum() |

Jumlah biaya yang hilang adalah 185.588.094,63.

Dari informasi diatas maka problem statement yang kami hadapi yaitu:

Bank JOGA has a **Churn rate** of **20.37%.** They **loss 2.037 customers** which impact to **profit loss $ 185M**.

Dalam menyelesaikan project ini kami sebagai Data Science Consultant diminta untuk membantu menyelesaikan permasalahan ini.

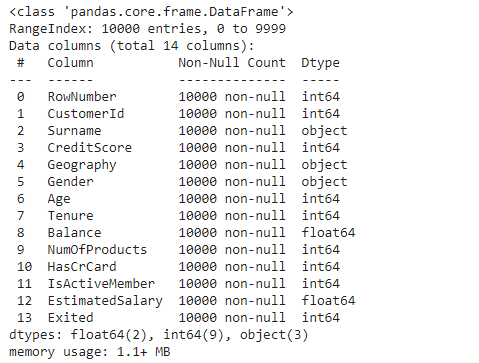
Beberapa hal yang perlu diperhatikan dalam project ini adalah:

|  |  |  |
| --- | --- | --- |
| Goal | : | Decrease Churn Rate in Bank JOGA |
| Objectives | : | * Make a **prediction** of Bank JOGA’s customer **probability to churn**. * Provide recommendations for developing bank service strategies to **increase bank customer loyalty** |
| Business Metrics | : | * Churn Rate * Profit |

# Pre-Processing

At first, kami coba mencari tau jumlah fitur dan target yang dimiliki:

|  |
| --- |
| df.info() |



Dataset memiliki 10.000 baris yang terdiri dari 13 Fitur dan 1 Target

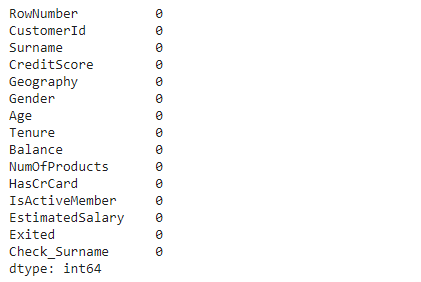
**Target:** Exited, apakah pelanggan meninggalkan bank atau tidak. (0=Tidak,1=Ya)

**Fitur:**

1. RowNumber : nomor baris pada masing-masing data yang ditampilkan, sama seperti index. Fitur ini akan dihapus.
2. CustomerID : Nomer identitas customer, berisi nilai acak dan tidak berpengaruh pada pelanggan yang meninggalkan bank. Kolom ini akan dihapus.
3. Surname : Nama belakang / family name, nama keluarga pelanggan tidak berdampak pada keputusan mereka untuk meninggalkan bank. Kolom ini akan dihapus.
4. CreditScore : Nilai skor kredit.
5. Geography : Negara dimana customer terdaftar.
6. Gender : Jenis kelamin.
7. Age : Usia / Umur customer.
8. Tenure : Seberapa lama (tahun) customer menjadi nasabah di Bank JOGA.
9. Balance : Jumlah rata-rata saldo di rekening.
10. NumOfProducts : jumlah produk yang digunakan customer.
11. HasCrCard : Status kepemilikan kartu kredit. (0=Tidak,1=Ya)
12. IsActiveMember : Status keaktifan customer. (0=Tidak,1=Ya)
13. EstimatedSalary : Perkiraan gaji customer.

Berikutnya mencari berapa banyak nilai kosong pada dataset.

|  |
| --- |
| df.isna().sum() |



Dari hasil pengecekan terlihat bahwa tidak ada data yang kosong atau missing. Sehingga tidak dilakukan proses drop atau imputasi row yang terdapat data yang kosong.

Berikutnya kami mencari data yg duplicate.

|  |
| --- |
| df.duplicated().sum() |



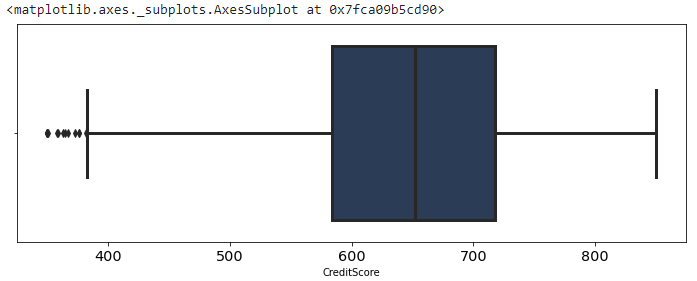
|  |
| --- |
| df.duplicated(subset=['CustomerId','Geography','Age']).sum() |



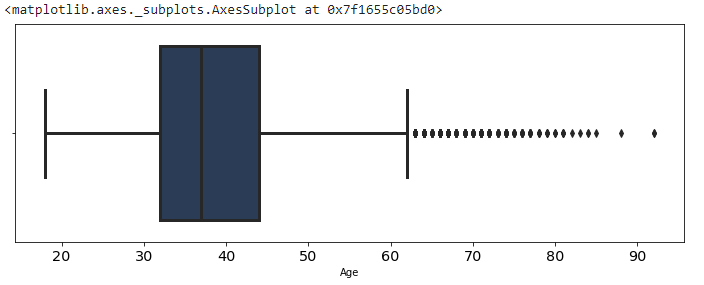
Tidak ada duplicate data pada keseluruhan dataset dan beberapa Feature di atas.

Berikutnya kami mencari outliers pada beberapa fitur numerical.

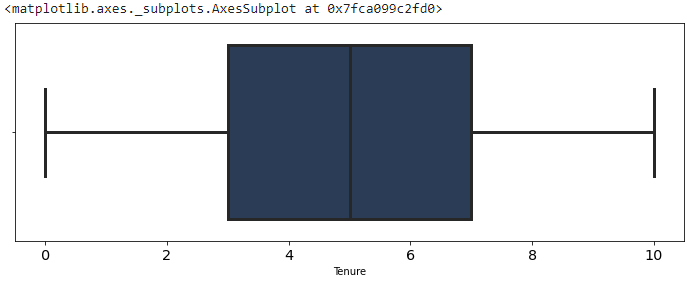
|  |
| --- |
| sns.boxplot(x='CreditScore', data=df) |



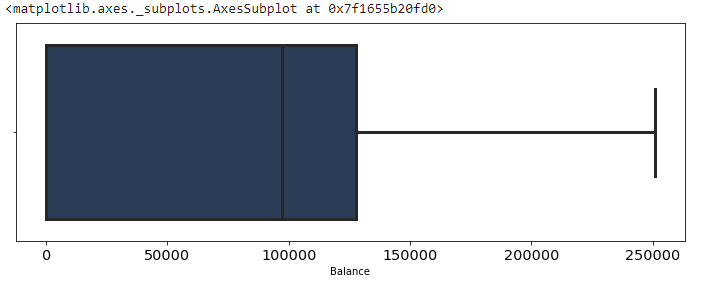
|  |
| --- |
| sns.boxplot(x='Age', data=df) |



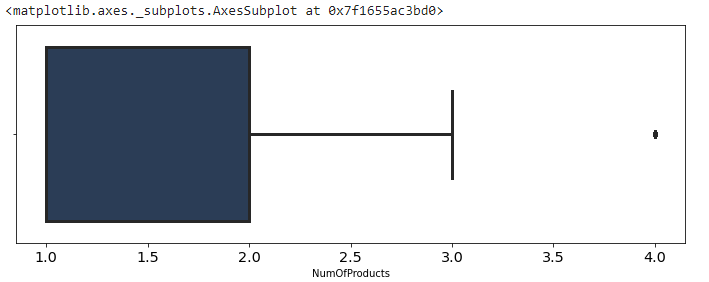
|  |
| --- |
| sns.boxplot(x='Tenure', data=df) |



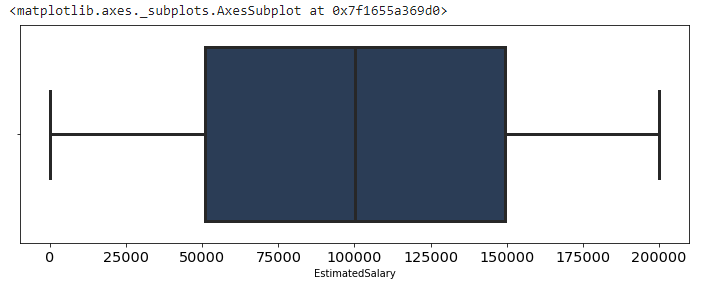
|  |
| --- |
| sns.boxplot(x='Balance', data=df) |



|  |
| --- |
| sns.boxplot(x='NumOfProducts', data=df) |



|  |
| --- |
| sns.boxplot(x='EstimatedSalary', data=df) |



Dari data di atas, terdapat outliers di beberapa data yaitu:

* CreditScore, akan kami eliminasi datanya menggunakan Z-Score
* Age, tidak kami olah karena masih berada di rentang usia hidup
* NumOfProduct, tidak kami olah karena 4 jenis produk masih memungkinkan dimiliki customer

Berikutnya kami menghilangkan outliers dengan Z-score.

|  |
| --- |
| from scipy import stats  print(f'Jumlah baris sebelum memfilter outlier: {len(df)}')  filtered\_entries = np.array([True] \* len(df))  for col in ['CreditScore']:      zscore = abs(stats.zscore(df[col]))      filtered\_entries = (zscore < 3) & filtered\_entries    df = df[filtered\_entries]  print(f'Jumlah baris setelah memfilter outlier: {len(df)}') |

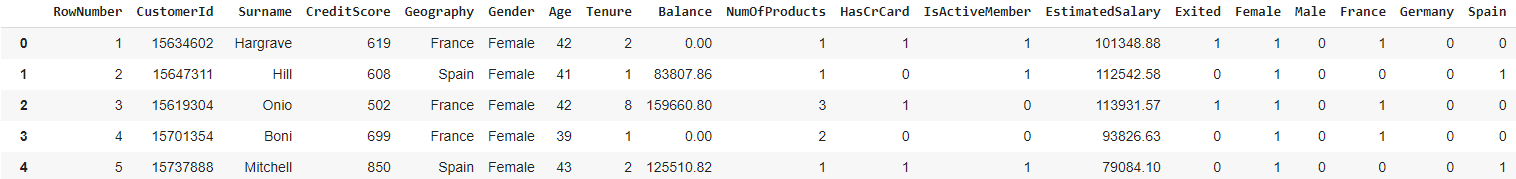


Jika dilihat dari hasil diatas, maka outliers yang berhasil dihilangkan berjumlah 8 baris.

## Feature Engineering

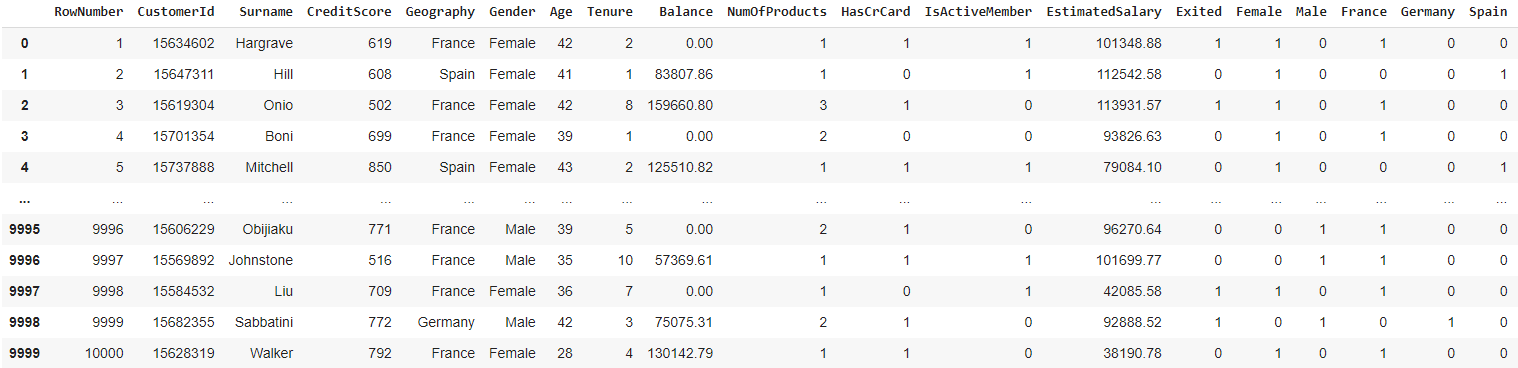
One-hot Encoding dilakukan untuk merubah data kategorikal pada fitur Geography dan Gender menjadi data numerical.

|  |
| --- |
| for i in cats:      onehots = pd.get\_dummies(df[i])      df = df.join(onehots)  df.head(5) |



Berikutnya kami melakukan segmentasi berdarsarkan Balance yang dimiliki nasabah.

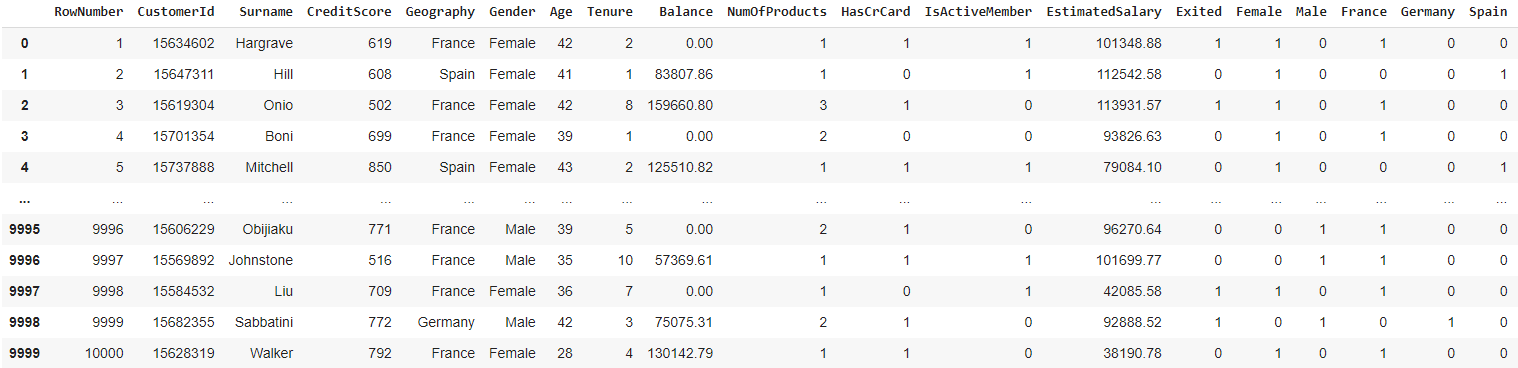
|  |
| --- |
| #Membuat segmentasi nasabah berdasarkan Percentile `Balance` (Q1, Q2, Q3).  def segment(x):      if x['Balance'] == 0:          segment = 'Bronze'      elif x['Balance'] > 0 and x[ 'Balance'] <= 97198 :          segment = 'Silver'      elif x['Balance'] > 97198 and x['Balance'] <= 127644 :          segment = 'Gold'      else:          segment = 'Platinum'      return segment  df['SegmentCust'] = df.apply(lambda x: segment(x),axis =1)  df |



Berikutnya kami melakukan segmentasi pada fitur Usia untuk melihat karakteristik customer Berdasarkan generasi.

* Sampai dengan usia 24 : Generasi Gen Z
* Usia 25 – usia 40 : Generasi Millenials
* Usia 41 – 56 : Generasi Gen X
* Usia 57 – 66 : Generasi Boomers type 2
* Usia 67 dan seterusnya : Generasi Boomers type 1

|  |
| --- |
| #Melakukan Penambahan Feature Segment Age  def segment(x):      if x['Age'] > 0  and x[ 'Age'] <= 24:          segment = 'Gen Z'      elif x['Age'] > 24  and x[ 'Age'] <= 40:          segment = 'Millenials'      elif x['Age'] > 40 and x['Age'] <= 56 :          segment = 'Gen X'      elif x['Age'] > 56 and x['Age'] <= 66 :          segment = 'Boomers II'      else:          segment = 'Boomers I' # Age >66      return segment  df['SegmentAge'] = df.apply(lambda x: segment(x),axis =1) |



Berikutnya kami menambahkan 1 feature dengan melakukan segmentasi pada CreditScore untuk melihat karakteristik customer berdasarkan nilai Credit Score. Segmentasi pelanggan berdasarkan Skor FICO (Skor Frank Kriticos):

* 740 – 850: Excellent
* 670 – 739: Good
* 350 – 669: Poor/Fair

|  |
| --- |
| #Melakukan Penambahan Feature Segment CreditScore  def segment(x):      if x['CreditScore'] >= 350 and x[ 'CreditScore'] <= 669:          segment = 'Poor/Fair'      elif x['CreditScore'] > 670  and x[ 'CreditScore'] <= 739:          segment = 'Good'      else:          segment = 'Excellent' # 740 -850      return segment  df['SegmentCreditScore'] = df.apply(lambda x: segment(x),axis =1) |

Berikutnya kami menambahkan 1 feature dengan melakukan segmentasi pada Tenure untuk melihat karakteristik customer berdasarkan lama nasabah menggunakan layanan Bank JOGA.

|  |
| --- |
| #Membuat segmentasi nasabah berdasarkan Tenure  def segment(x):      if x['Tenure'] == 0:          segment = 'low\_term'      elif x['Tenure'] > 0 and x[ 'Tenure'] <= 5 :          segment = 'short\_term'      elif x['Tenure'] > 5 and x['Tenure'] <= 10 :          segment = 'medium\_term'      return segment  df['Tenure\_Term'] = df.apply(lambda x: segment(x),axis =1) |

Kami melakukan standarisasi dan Normalisasi pada fitur Age, Balance, IsActiveMember, Female, Germany, dan NumOfProducts sebelum nantinya fitur ini akan digunakan pada pemodelan.

|  |
| --- |
| features = ['Age','Balance','IsActiveMember','Female','Germany','NumOfProducts']  for var in features:    df[var]= (df[var]+1).apply(np.log)  from sklearn.preprocessing import MinMaxScaler, StandardScaler  for var in features:    df[var]= MinMaxScaler().fit\_transform(df[var].values.reshape(len(df), 1)) |

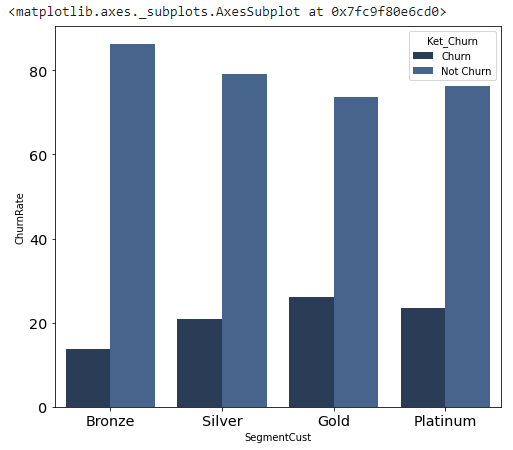
# EDA and Insight

At first, kami melihat profil dari customer yang ada di Bank JOGA.

|  |
| --- |
|  |

Berikutnya kami melihat data customer berdasarkan kategori balance

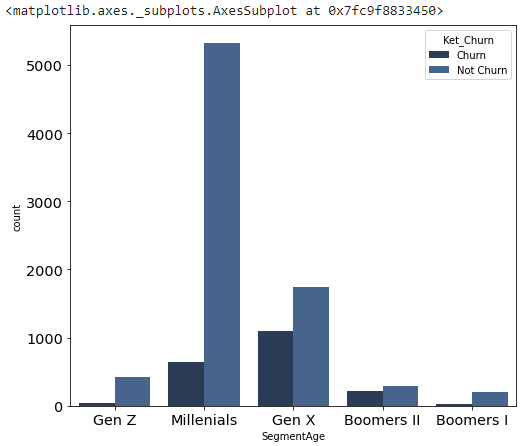
|  |
| --- |
| #Visualisasi Segmentasi Nasabah berdasarkan Balance terhadap Peluang Churn 'ChurnRate'  plt.figure(figsize=(8,7))  sns.barplot(x='SegmentCust', y='ChurnRate', data=df2,              estimator=sum, hue='Ket\_Churn', order=[ 'Bronze', 'Silver', 'Gold', 'Platinum']) |



Banyaknya nasabah yang memiliki Saldo Kategori ‘Emas’ lebih besar kemungkinannya untuk keluar dari Layanan Bank. Mereka memiliki keseimbangan yang baik yang memberikan kontribusi bagi keuntungan Bank ($97.198 - $127.644).

Berikutnya kami melihat data customer berdasarkan kategori usia

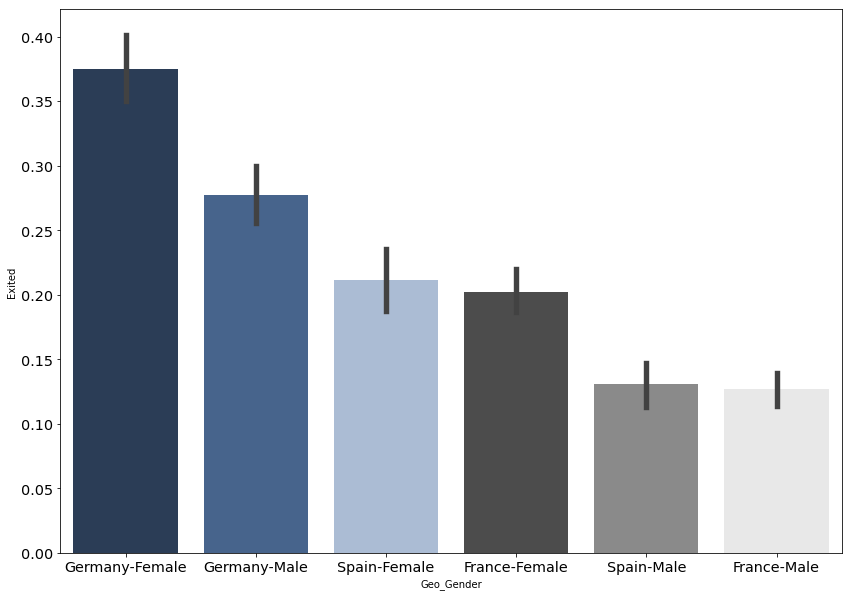
|  |
| --- |
| plt.figure(figsize=(8,7))  sns.countplot(x='SegmentAge', data=df, hue='Ket\_Churn', order=[ 'Gen Z', 'Millenials', 'Gen X', 'Boomers II', 'Boomers I']) |



Meski nasabah didominasi oleh kaum Milenial, kategori 'Gen X' lebih berpotensi meninggalkan layanan bank. Jumlah nasabah mencapai 1.038 customer.

Berikutnya kami melihat data customer berdasarkan Gender dan Geography

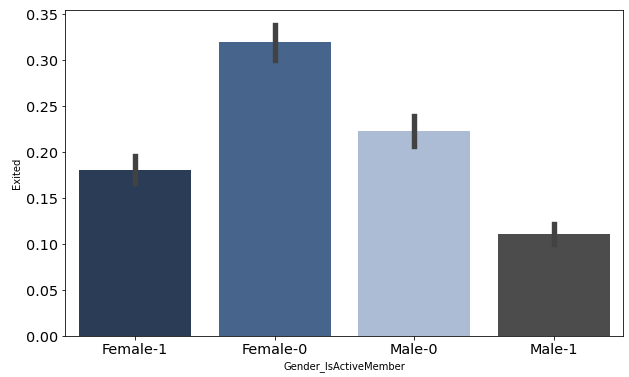
|  |
| --- |
| df['Geo\_Gender']= df['Geography'].str.cat(df['Gender'],sep='-')  plt.figure(figsize=(14,10))  sns.barplot(x = 'Geo\_Gender', y= 'Exited', data=df,order =['Germany-Female','Germany-Male','Spain-Female','France-Female','Spain-Male','France-Male']); |



Pelanggan wanita di Jerman lebih berpotensi untuk Churn daripada kategori mana pun.

Berikutnya kami melihat data customer berdasarkan Gender dan status keaktifan customer

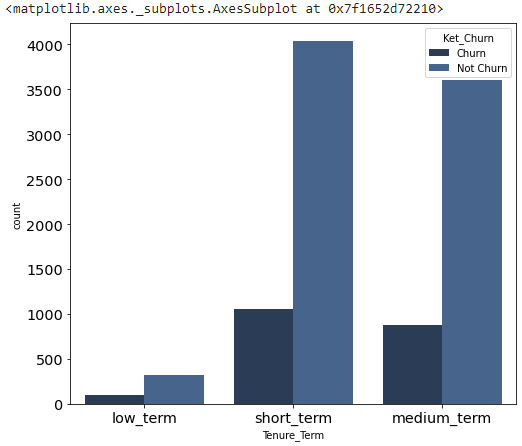
|  |
| --- |
| df['Gender\_IsActiveMember']= df['Gender'].str.cat(df['IsActiveMember'].astype('str'),sep='-')  plt.figure(figsize=(10,6))  sns.barplot(x = 'Gender\_IsActiveMember', y= 'Exited', data=df); |



Pelanggan wanita dan memiliki status 'Non-Aktif' lebih berpotensi untuk Churn daripada kategori mana pun.

Berikutnya kami melihat data customer berdasarkan kategori lamanya nasabah menggunakan produk Bank JOGA

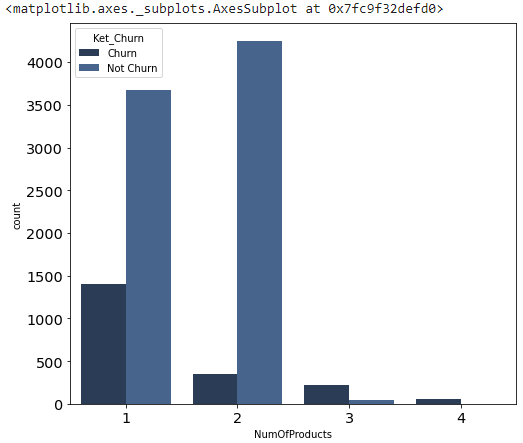
|  |
| --- |
| #Data Visualisasi Tenure Term terhadap Potensi Churn  plt.figure(figsize=(8,7))  sns.countplot(x='Tenure\_Term', data=df, hue='Ket\_Churn', order=[ 'low\_term', 'short\_term', 'medium\_term']) |



Nasabah dengan tenor antara 1-5 tahun kemungkinan besar akan pindah ke bank lain.

Berikutnya kami melihat data customer berdasarkan Jumlah Produk yang dimiliki customer

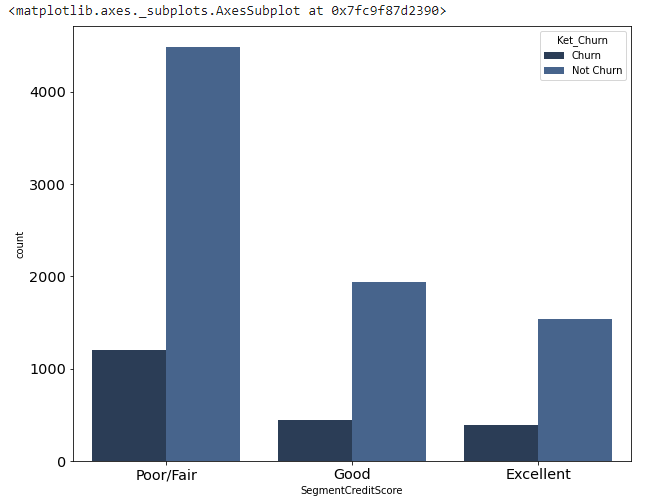
|  |
| --- |
| #Data Visualisasi Tenure Term terhadap Potensi Churn  plt.figure(figsize=(8,7))  sns.countplot(x='NumOfProducts', data=df, hue='Ket\_Churn') |



Nasabah yang hanya memiliki 'satu' produk cenderung membandingkan Bank Joga dengan layanan Bank lain, mereka lebih berpotensi untuk Churn.

Berikutnya kami melihat data customer berdasarkan kategori skor kredit

|  |
| --- |
| #Data Visualisasi Segmentasi CreditScore terhadap Churn  plt.figure(figsize=(10,8))  sns.countplot(x='SegmentCreditScore', data=df, hue='Ket\_Churn') |



Customer dengan skor kredit 'Buruk/Cukup' memiliki jumlah pelanggan yang paling banyak melakukan churn.

# Modelling and Evaluation

Sebelum melakukan pemodelan, kami harus membagi dataset menjadi data train dan data test.

|  |
| --- |
| #Splitting the data into Train and Test  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, random\_state = 42) |

Setelah itu kami mengecek jumlah baris pada data train dan data test.

|  |
| --- |
| X\_train.shape |



|  |
| --- |
| X\_test.shape |



Berikutnya kami melakukan pengecekan nilai target pada data train untuk mengecek apakah terjadi imbalance class atau tidak.

|  |
| --- |
| y\_train.value\_counts() |

Karena nilai target pada data train tidak seimbang, maka kami lakukan over sampling SMOTE agar nilai pada target menjadi seimbang. Hal ini penting mengingat evaluasi metrics yang kami gunakan adalah **Recall.** Jika nilai 1 pada target tidak seimbang atau bahkan terlalu kecil, maka nilai Recall akan menjadi kecil.

|  |
| --- |
| from imblearn import over\_sampling  X\_train, y\_train = over\_sampling.SMOTE().fit\_resample(X\_train, y\_train) |

Setelah itu kami melakukan pengecekan jumlah baris pada data train.

|  |
| --- |
| print(X\_train.shape, y\_train.shape) |



Selain itu, kami juga melakukan pengecekan nilai target pada data train.

|  |
| --- |
| y\_train.value\_counts() |

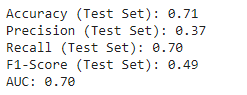
Setelah memastikan nilai target pada data train sudah seimbang, maka kita masuk ke pemodelan.

Sehubungan dengan dataset yang memiliki 1 target, churn / no churn, maka kami melakukan pemodelan Classification menggunakan model, yaitu Logistic Regression, Decision Tree, K-Nearest Neighbors (KNN), Random Forest, Ada Boost, dan XG Boost.

|  |
| --- |
| from sklearn.metrics import accuracy\_score, precision\_score, recall\_score, f1\_score  from sklearn.metrics import roc\_curve, auc  def eval\_classification(model, pred, xtrain, ytrain, xtest, ytest):      print("Accuracy (Test Set): %.2f" % accuracy\_score(ytest, pred))      print("Precision (Test Set): %.2f" % precision\_score(ytest, pred))      print("Recall (Test Set): %.2f" % recall\_score(ytest, pred))      print("F1-Score (Test Set): %.2f" % f1\_score(ytest, pred))        fpr, tpr, thresholds = roc\_curve(ytest, pred, pos\_label=1) # pos\_label: label yang kita anggap positive      print("AUC: %.2f" % auc(fpr, tpr))  def show\_feature\_importance(model):      feat\_importances = pd.Series(model.feature\_importances\_, index=X.columns)      ax = feat\_importances.nlargest(25).plot(kind='barh', figsize=(10, 8))      ax.invert\_yaxis()      plt.xlabel('score')      plt.ylabel('feature')      plt.title('feature importance score')  def show\_best\_hyperparameter(model, hyperparameters):      for key, value in hyperparameters.items() :          print('Best '+key+':', model.get\_params()[key]) |

Pertama, pemodelan menggunakan Logistic Regression:

|  |
| --- |
| from sklearn.linear\_model import LogisticRegression  logres = LogisticRegression(random\_state=42)  logres.fit(X\_train, y\_train)  y\_pred = logres.predict(X\_test)  eval\_classification(logres, y\_pred, X\_train, y\_train, X\_test, y\_test) |

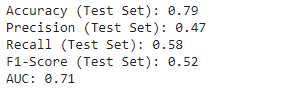


|  |
| --- |
| print('Train score: ' + str(logres.score(X\_train, y\_train)))  print('Test score:' + str(logres.score(X\_test, y\_test))) |



Kedua, pemodelan menggunakan Decision Tree:

|  |
| --- |
| from sklearn.tree import DecisionTreeClassifier  dt = DecisionTreeClassifier(random\_state=42)  dt.fit(X\_train,y\_train)  y\_pred\_dt = dt.predict(X\_test)  eval\_classification(dt, y\_pred\_dt, X\_train, y\_train, X\_test, y\_test) |

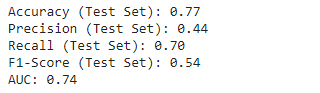


|  |
| --- |
| print('Train score: ' + str(dt.score(X\_train, y\_train)))  print('Test score:' + str(dt.score(X\_test, y\_test))) |



Ketiga, pemodelan menggunakan KNN:

|  |
| --- |
| from sklearn.neighbors import KNeighborsClassifier  knn = KNeighborsClassifier(n\_neighbors=13)  knn.fit(X\_train, y\_train)  y\_pred\_knn = knn.predict(X\_test)  eval\_classification(knn, y\_pred\_knn, X\_train, y\_train, X\_test, y\_test) |

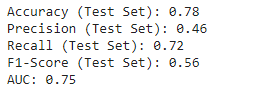


|  |
| --- |
| print('Train score: ' + str(knn.score(X\_train, y\_train)))  print('Test score:' + str(knn.score(X\_test, y\_test))) |



Keempat, pemodelan menggunakan Ada Boost:

|  |
| --- |
| from sklearn.ensemble import AdaBoostClassifier  ab = AdaBoostClassifier(random\_state=42)  ab.fit(X\_train,y\_train)  y\_pred\_ab = ab.predict(X\_test)  eval\_classification(ab, y\_pred\_ab, X\_train, y\_train, X\_test, y\_test) |

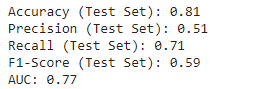


|  |
| --- |
| print('Train score: ' + str(ab.score(X\_train, y\_train)))  print('Test score:' + str(ab.score(X\_test, y\_test))) |



Kelima, pemodelan menggunakan XG Boost:

|  |
| --- |
| from xgboost import XGBClassifier  xg = XGBClassifier(random\_state=42)  xg.fit(X\_train, y\_train)  y\_pred\_xg = xg.predict(X\_test)  eval\_classification(xg, y\_pred\_xg, X\_train, y\_train, X\_test, y\_test) |



|  |
| --- |
| print('Train score: ' + str(xg.score(X\_train, y\_train)))  print('Test score:' + str(xg.score(X\_test, y\_test))) |



Dari beberapa model di atas, dapat disimpulkan hasil datanya adalah sebagai berikut.

Kami menggunakan evaluasi metrics recall untuk memilih model mana yang terbaik, karena kami ingin meminimalisir kesalahan untuk menduga customer yang aktualnya churn tapi terprediksi tidak churn.

Oleh karena itu, kami memilih model XG Boost, karena memiliki nilai Recall tertinggi diantara model lain, yaitu 0.72. Nilai skor data train dan data test menunjukkan hasil yang fit, tidak under fit maupun over fit.

Berikutnya kami melihat confusion matrix.

|  |
| --- |
| from sklearn.metrics import accuracy\_score, f1\_score, precision\_score,confusion\_matrix, recall\_score, roc\_auc\_score    cm\_xgb = confusion\_matrix(y\_test, y\_pred=y\_pred\_xg)    TP = cm\_xgb[1, 1]  TN = cm\_xgb[0, 0]  FP = cm\_xgb[0, 1]  FN = cm\_xgb[1, 0] |

|  |
| --- |
| from matplotlib import rc,rcParams  def plot\_confusion\_matrix(cm, classes,  title='Confusion matrix',  cmap=plt.cm.Blues):    plt.rcParams.update({'font.size': 12})  plt.imshow(cm, interpolation='nearest', cmap=cmap)  plt.title(title,fontdict={'size':'24'})  plt.colorbar()  tick\_marks = np.arange(len(classes))  plt.xticks(tick\_marks, classes,fontsize=12,color="blue")  plt.yticks(tick\_marks, classes,fontsize=12,color="blue")  rc('font', weight='bold')  fmt = '.0f'  thresh = cm.max()  for i, j in itertools.product(range(cm.shape[0]), range(cm.shape[1])):  plt.text(j, i, format(cm[i, j], fmt),  horizontalalignment="center",  color="orange")    plt.ylabel('True label',fontdict={'size':'16'})  plt.xlabel('Predicted label',fontdict={'size':'16'})  plt.tight\_layout() |

|  |
| --- |
| import itertools  plot\_confusion\_matrix(confusion\_matrix(y\_test, y\_pred=y\_pred), classes=['Non Churn','Churn'],  title='Confusion matrix') |

Dari Confusion Matrix di atas, didapatkan bahwa customer yang terprediksi churn ada 232, sedangkan yang salah terprediksi ada 164.

Berikutnya kami melakukan feature importance untuk melihat fitur mana yang paling penting dalam pemodelan ini.

|  |
| --- |
| show\_feature\_importance(xg) |

Jumlah produk yang dimiliki customer adalah fitur yang paling berkontribusi dalam membuat churn (>0.25). Status keaktifan customer dan Usia juga berpengaruh signifikan terhadap churn.

Dari insight dan pemodelan yang kami lakukan, kami dapatkan kesimpulan seperti berikut:

|  |  |
| --- | --- |
| Customer yang berpeluang untuk churn | Customer yang menggunakan 1 produk |
| Customer yang tidak aktif |
| Customer dari Gen X |
| Customer dari Jerman |
| Customer berjenis kelamin Perempuan |
| Customer kategori Gold |

Kami merekomendasikan agar Bank menghubungi secara personal ke semua Nasabah yang memiliki kategori berpotensi churn. Bank akan memberikan promo dan rekomendasi kepada nasabah untuk mengurangi jumlah churn.

# Business Recommendation

**Preferensi Industri Keuangan Perbankan di Jerman**

Industri keuangan di Jerman sedikit berbeda dari negara lainnya. Segmentasi industri perbankan di Jerman sangat beragam, tersegmentasi dan sangat kompetitif. Selain itu juga, masyarakat di Jerman sebagian besar masyarakat memilih menyimpan uang di Bank. Sehingga, sensitivitas terhadap bunga Bank sangat mempengaruhi nasabah dalam memilih layanan Bank di Jerman (McKinsey and Company, 2016 ). Kebijakan moneter di Bank Sentral Eropa telah memperkuat tren “Suku Bunga Rendah” termasuk di Jerman yakni mempertahankan suku bunga rendah dan semakin menekan margin bunga bersih.

Berdasarkan artikel pada jurnal di Cision tahun 2017 menyatakan bahwa kurang dari 10% penduduk Jerman yang memiliki kartu kredit dan 80% selebihnya memilih membayar tepat waktu dan cash. Sekitar 80 persen dari semua pinjaman di Bank untuk konsumsi rumah tangga dan perusahaan domestik memiliki jatuh tempo sebesar diatas 5 tahun (tenor), sementara Dana Pihak Ketiga (DPK) di Bank juga tinggi yakni (dana mahal) deposito, dana pensiun dan (dana murah) tabungan. Dengan begitu, kegiatan lending Bank giat dilakukan pada sektor perusahaan dan konsumsi rumah tangga. Selain itu, literasi keuangan masyarakat di Jerman juga tinggi (dana murah), mereka senang menabung di Bank. Pola konsumsi masyarakat Jerman umumnya tidak konsumtif, pengeluaran didominasi untuk hal produktif seperti cicilan rumah, kredit mobil, dan barang elektronik lainnya dengan risiko yang rendah. Sehingga kecenderungan untuk meninggalkan layanan juga cukup besar jika terdapat penawaran dari Bank lain yang lebih menarik.

**Pendekatan Bisnis terhadap Nasabah yang Potensi ‘Churn’**

Berdasarkan kriteria nasabah yang berpotensi meninggalkan layanan Bank Joga yakni nasabah yang memiliki kriteria : perempuan, generasi X (41-56 tahun), di Jerman, status akun Bank tidak aktif, memiliki 1 produk Bank dan di kategori Gold Card. Hal tersebut menjadi input yang baik untuk Tim Bisnis Bank Joga dalam meningkatkan pelayanan untuk mempertahankan nasabah. Dari kategori pemodelan churn yang telah diuji, maka tim bisnis telah mendapatkan kriteria nasabah yang berpotensi meninggalkan layanan Bank. Dari total 2.037 nasabah yang churn saat ini, dengan modeling dapat diprediksi nasabah yang akan churn sebesar 72% (Nilai Recall). Selanjutnya dari prediksi nasabah sebesar 1.467 akan dilakukan pendekatan bisnis untuk mempertahankan nasabah potensial tersebut. Pihak Bank Joga akan mencoba memilih nasabah yang berpotensi churn sesuai kriteria model dan melakukan pendekatan untuk menawarkan produk dan promo. Berikut rekomendasi bisnis yang akan dilakukan Bank Joga.

Pendekatan Bisnis Bank Joga terhadap Nasabah yang Potensi Churn

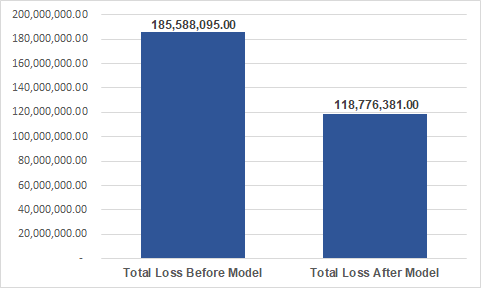
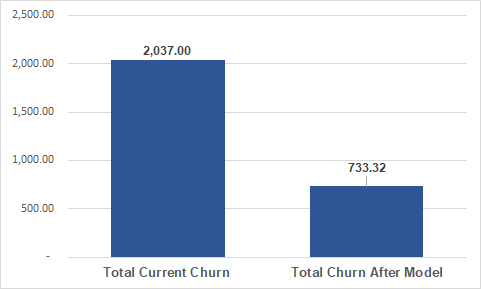
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Penawaran Produk** | **Pendekatan Nasabah** | **Mekanisme** | **Biaya** |
| 1 | 5% Cashback pada transaksi menggunakan kartu debit | Meningkatkan transaksi dan stimulus **nasabah yang tidak aktif** | Maksimum **15$ (3x)**  Target : **100** nasabah | $ 1.500 |
| 2 | 10% Diskon pada transaksi menggunakan kartu kredit (retail, food, beauty and travel) | Meningkatkan jumlah transaksi kredit pada nasabah **perempuan pemilik Gold-Card di Jerman** | Maksimum **25$ (2x)**  Target : **250** nasabah | $ 6.250 |
| 3 | 50% Diskon premi bulan pertama Asuransi Jiwa Bank Joga | Menstimulus nasabah **Generasi X** untuk menggunakan produk asuransi jiwa dari Bank | Maksimum **50 (1 x)**  Target : **150** nasabah | $ 10.000 |
| 4 | Deposito spesial rate 6% | Meningkatkan **jumlah produk dari nasabah** dan meningkatkan ‘Funding’ pada bank | Minimum depostio $ **25.000** di **3** bulan pertama **(1x)**  Target : **250** nasabah | $ 72.000 |
| Total | | | **750 nasabah** | **$ 89.750** |

Pendekatan bisnis tersebut dilakukan untuk mempertahankan nasabah untuk tetap menggunakan layanan Bank Joga. Dari prediksi 1.467 nasabah yang akan ‘Churn’ maka diasumsikan 50% diantaranya dapat diselamatkan melalui pendekatan penawaran promo Bank, yakni 733 nasabah (dibulatkan 750 nasabah). Asumsi tersebut jika promo dapat diterima oleh nasabah yang berpotensi ‘Churn’ sebesar 50% dari total prediksi nasabah.

Produk yang ditawarkan Bank Joga untuk nasabah tersebut telah disesuaikan dengan prediksi kriteria nasabah yang akan ‘Churn’. Sehubungan dengan referensi nasabah yang Churn itu kebanyakan adalah wanita pemilik Gold-Card di Jerman berusia 41-56 tahun, maka produk yang ditawarkan berupa *cashback* untuk transaksi pakai kartu debit, diskon belanja menggunakan kartu kredit di *retail, food, beauty dan travel* untuk meningkatkan jumlah nasabah menggunakan kartu kredit. Selain itu, di usia yang sudah mapan secara finansial, bank menawarkan promo produk asuransi Jiwa guna mempertahankan nasabah sekaligus menjual produk asuransi Bank Joga. Promo yang berkaitan dengan funding ialah deposito dengan bunga spesial yakni 6%, untuk menarik nasabah untuk menambah jumlah produk bank yang digunakan dan juga dapat menambah pemasukan untuk pendanaan bank. Hal tersebut dinilai dapat memicu ketertarikan nasabah potensial ‘Churn’ untuk tetap menggunakan layanan Bank Joga.

Melalui pendekatan bisnis yang dilakukan Bank Joga, dapat terlihat penurunan kerugian dan model dapat menyelamatkan kerugian Bank tersebut, jika asumsi 50% program berhasil terlaksana seperti pada grafik dibawah ini.

Grafik Penurunan Jumlah Nasabah dan Kerugian Sebelum dan Sesudah Modeling

****

Dari grafik diatas terlihat penurunan nasabah yang ‘Churn’ dari 2.037 ke 733 nasabah (36%). Selain itu juga Bank Joga dapat menyelamatkan total kerugian akibat nasabah yang ‘Churn’ sebesar **$ 66,811,714 (35%)** dari yang awalnya **$ 185.588.095.** Perhitungan prediksi profit loss tersebut didapat dari **selisih nasabah yang Churn (1.034 nasabah) x rata-rata balance nasabah ($ 91.086),** sehingga didapatkan total penurunan kerugian bank setelah pemodelan yakni sebesar **$ 118.776.381**.

Penurunan kerugian tersebut dapat menjadi sisi positif dari pemodelan untuk melakukan peningkatan layanan Bank Joga.

Dari total pengeluaran dari Bank Joga untuk mempertahankan nasabah dalam bentuk penawaran *cashback*, diskon dan produk lain, maka sesuai perhitungan (pada lampiran) maka Bank Joga akan mendapatkan dana kembali*(Break Even Point)* pada bulan ke-9 setelah produk disampaikan pada nasabah.

Pada pemodelan menggunakan Machine Learning, Bank Joga dapat menganalisis dan memprediksi nasabah yang memiliki potensi untuk Churn, sehingga prediksi dari data nasabah dapat menjadi masukan untuk tim Bisnis melakukan pendekatan bisnis terhadap nasabah untuk tetap mempertahankan nasabah serta menambah pendapatan *(revenue)* terhadap Bank. Dengan demikian, analisis serta prediksi model ‘nasabah churn’ dapat memberikan dampak langsung terhadap strategi Bank dalam peningkatan layanan serta meningkatkan pemasukan Bank Joga.

**DAFTAR PUSTAKA**

McKinsey and Company. *“The road ahead Perspectives on German banking”*. 2016

<https://www.prnewswire.com/news-releases/consumer-payments-country-snapshot-germany-2016-300483810.html> diakses pada Minggu, 11 Juli 11.25 WIB

<https://www.theglobaleconomy.com/Germany/consumer_credit/> diakses pada Minggu, 11 Juli 11:45 WIB