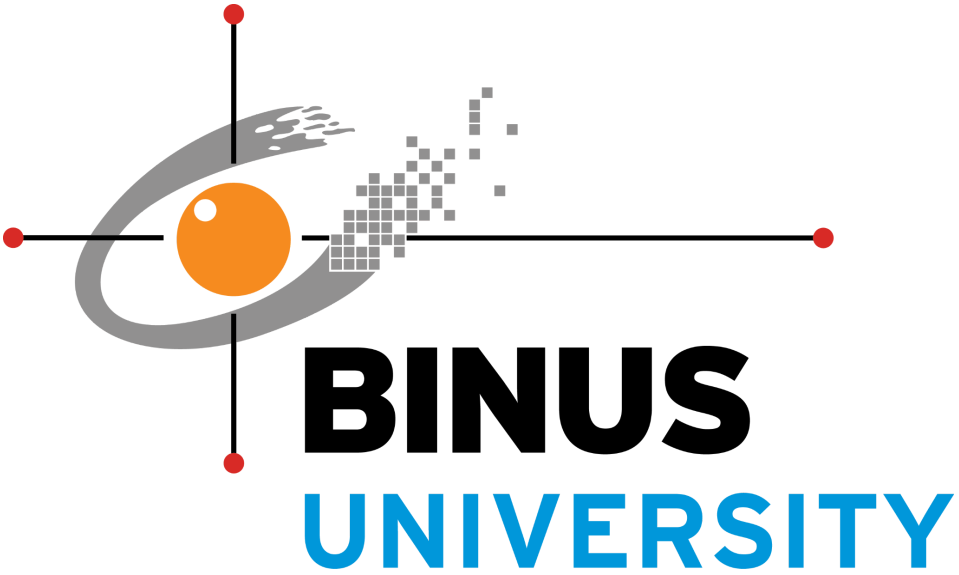
TUGAS KELOMPOK DATA MINING LB01:
"AIRLINE CUSTOMER SATISFACTION CLASSIFICATION USING SUPPORT VECTOR MACHINE"



Dosen Mata Kuliah:
D6198 Fepri Putra Panghurian, S.Kom, M.T.I.

Penyusun:

Alden Ardiwinata Putra	**2501977286**
Princessa Fortunata Fusanto	**2501984511**
Ricky Krisdianto	**2501974385**
Celina Josephine	**2540130365**
Ivana Apriani	**2540129855**
Hossey Masada	**2540128165**

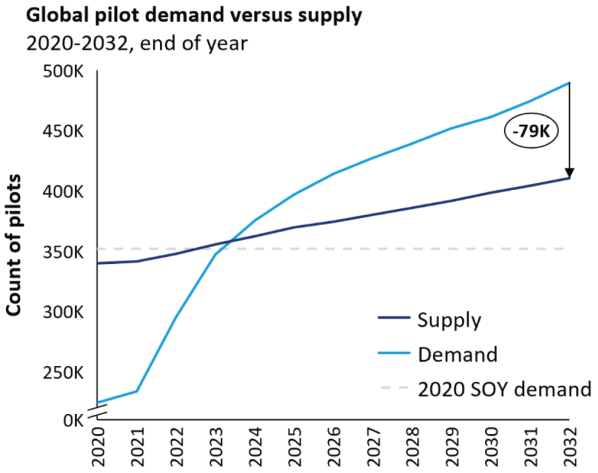
COMPUTER SCIENCE
SCHOOL OF COMPUTER SCIENCE
UNIVERSITAS BINA NUSANTARA
JAKARTA 2023

DAFTAR PUSTAKA

BAB I
PENDAHULUAN

1.1. Latar Belakang

Pandemi COVID-19 kini dirasakan mereda secara global dan hal ini berdampak langsung pada perkembangan dunia aviasi mengalami peningkatan *pilot demand* yang tinggi, terutama Asia. Oliver Wyman memprediksi bahwa permintaan tinggi ini akan terus mengalami perkembangan surplus sampai 2032.



Gambar 1. Prediksi Supply & Demand Pilot Global

Bagaimana dengan di Indonesia? Dilansir dari Kompas, saat ini terdapat 53 maskapai penerbangan di Indonesia dan jumlah ini diperkirakan akan terus bertumbuh kedepannya. Dengan pesatnya perkembangan industri penerbangan, pemahaman dan peningkatan kepuasan pelanggan (*customer satisfaction*) menjadi sangat penting untuk mengetahui pola dan perilaku pelanggan saat ini, terutama pada era pra-pandemi ini dimana dunia aviasi sedang mengalami pemulihan. Hal ini bertujuan agar industri penerbangan bisa tetap berkembang sesuai dengan keinginan dan kebutuhan pelanggan.

Demikian, dalam projek ini kami berusaha mengimplementasikan proses *knowledge discovery in databases (KDD)*, atau umum dikenal sebagai data mining, dalam meneliti dataset kepuasan pelanggan maskapai (*aviation satisfaction survey dataset*). Adapun algoritma yang akan kami eksplorasi adalah salah satu algoritma *supervised learning* yang dikenal akan kestabilan dan pendekatannya yang mumpuni untuk berbagai karakteristik dataset, yakni Support Vector Machine (SVM). Analisis klasifikasi ini kemudian akan kami turunkan faktor-faktor kunci yang berkontribusi terhadap kepuasan pelanggan berdasarkan *existing data*. Analisis ini diharapkan dapat membantu maskapai dalam *business decision making*, optimalisasi pelayanan ataupun operasional, dan analisis faktor serta aspek *influential* dalam trend pasar aviasi.

1.2. Tujuan

Adapun tujuan dari projek ini:

- Melakukan *exploratory data analysis (EDA)* untuk memahami lebih lanjut karakteristik dataset.
- Menentukan secara kuantitatif (dengan pendekatan *machine learning*) atribut apa saja yang memiliki pengaruh terbesar dalam penentuan kepuasan pelanggan aviasi serta parameter model mana yang paling sesuai untuk menangani prediksi klasifikasi model dataset kepuasan pelanggan.
- Melakukan *model training, fitting,* dan *testing* untuk mengeksplorasi algoritma SVM dalam implementasinya untuk prediksi klasifikasi kepuasan pelanggan maskapai.
- Menurunkan analisis KDD tersebut menjadi sebuah *insight* dan keputusan bisnis yang berbasis data (data driven) dan konkrit secara statistik (terkini) ataupun prediktif (di masa depan).

1.3. Manfaat

Berikut manfaat yang dapat diperoleh dari pengerjaan projek ini:

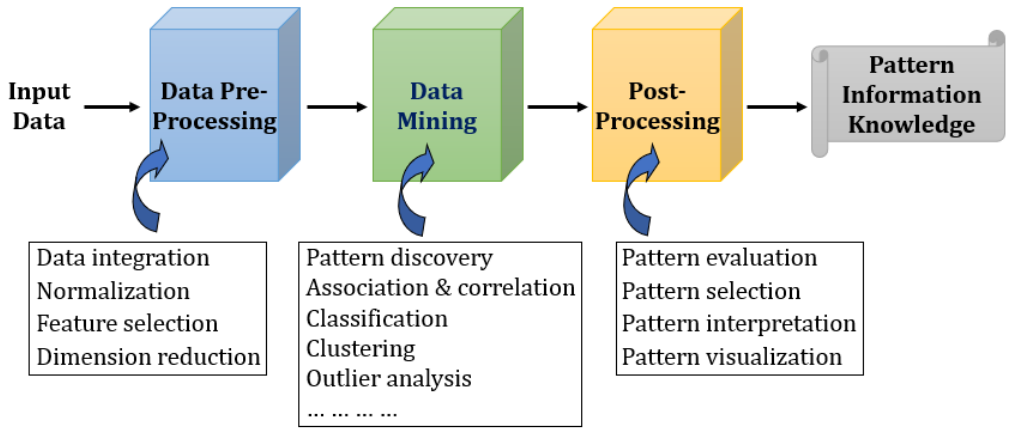
- Membantu maskapai untuk melakukan inspeksi terhadap atribut dataset (parameter kepuasan) yang masih memiliki performa dibawah rata-rata.
- Memberikan sebuah gambaran kuantitatif mengenai keputusan operasional apa yang bisa diambil untuk meningkatkan kualitas pelayanan maskapai kepada pelanggan.
- Memahami kinerja dan mendapatkan pengetahuan bagaimana machine learning, terutama SVM beserta karakteristik algoritmanya, digunakan untuk mengelola dataset kepuasan pelanggan (kekurangan dan kelebihan SVM dalam prediksi klasifikasi dataset kepuasan pelanggan maskapai).
- Memiliki visualisasi mengenai bagaimana suatu atribut dapat mempengaruhi keakuratan model *machine learning* SVM dalam melakukan prediksi klasifikasi.

BAB II
METODOLOGI

2.1. Kerangka Kerja: *Knowledge Discovery in Databases Process*

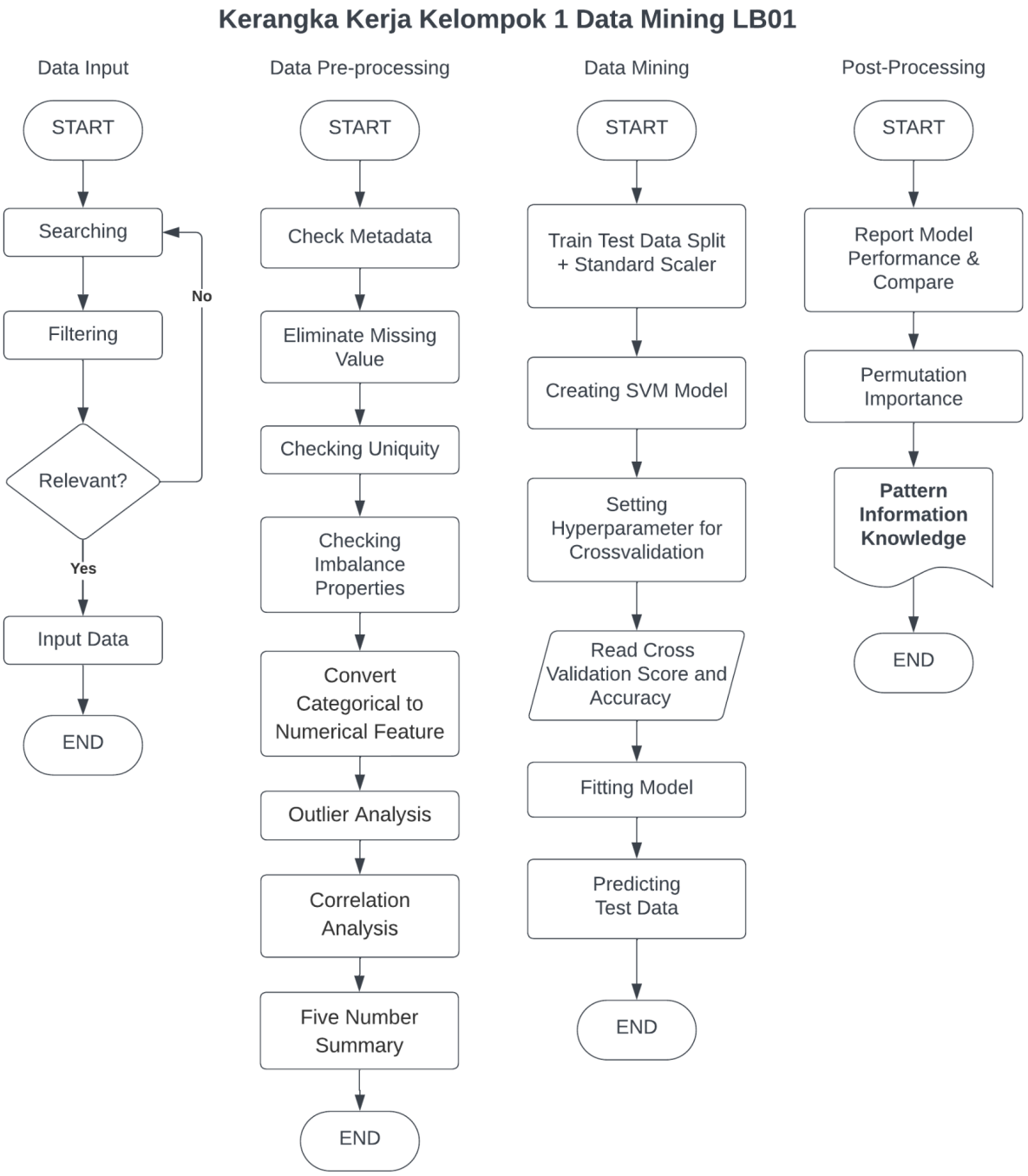
Metodologi yang akan kami gunakan dalam projek ini dikenal sebagai *KDD Process*. Berdasarkan *The Morgan Kaufmann Series, Data Mining: Concepts and Techniques*, proses KDD dapat dijelaskan dalam poin-poin berikut:

- **Data Input:** Pencarian dataset dari berbagai sumber untuk kebutuhan analisis yang relevan.
- **Data Preprocessing:** Pemrosesan awal data dimana dataset akan mengalami pembersihan terlebih dahulu sebelum diproses lebih lanjut. Pembersihan yang dimaksud melibatkan *data integration, data cleaning, feature selection,* dan sebagainya.
- **Data Mining:** Inti dari proses KDD, implementasi algoritma dan berbagai metode komputatif dilakukan di tahap ini untuk mengidentifikasi berbagai informasi ataupun pattern.
- **Post-Processing:** Melakukan asesmen model algoritma yang telah dijalankan sebelumnya untuk memastikan berbagai metrik dan parameter yang relevan (accuracy, f1-score, recall, precision, dan sebagainya).
- **Pattern Information Knowledge:** Tahap terakhir yang melibatkan keseluruhan proses divisualisasikan menjadi media yang informatif. Interpretasi proses KDD ini dapat digambarkan melalui presentasi ataupun infografis kepada *user* ataupun *expert*.



Gambar 2. Prediksi Supply & Demand Pilot Global

Metodologi yang telah dijelaskan di atas akan kami aplikasikan dalam kerangka kerja projek kami. Flowchart kerangka kerja projek ini mulai dari data input, preprocessing, data mining, post-processing, sampai terbentuknya dokumen *pattern information knowledge* dapat dilihat pada Gambar 3.



Gambar 3. Flowchart Kerangka Kerja

2.2. Algoritma

Kami akan mengeksplorasi berbagai algoritma komputasi dan *machine learning* dalam projek ini, antara lain sebagai berikut:

2.2.1. *Support Vector Machine (SVC from sklearn.svm)*

SVM merupakan salah satu algoritma machine learning dengan pendekatan supervised learning yang bekerja dengan mencari hyperplane atau fungsi pemisah terbaik untuk memisahkan kelas atribut. Dalam projek ini kami

mengeksplorasi prediksi klasifikasi menggunakan *Kernel SVM* dari library *sklearn* yang terbagi menjadi beberapa jenis sesuai dengan fungsinya:

- **Linear:** menggunakan fungsi hyperplane linear yang membagi kelas-kelas data dengan garis lurus.
- **Radial Basis Function:** menggunakan fungsi hyperplane dengan dimensi tak hingga (persamaan lingkaran/elips) yang mampu menangani data non-linear.
- **Polynomial:** menggunakan fungsi hyperplane banyak suku/eksponen untuk mencari persamaan terbaik yang mampu memisahkan kelas-kelas data non-linear.
- **Sigmoid:** menggunakan fungsi hyperplane dengan persamaan garis sigmoid yang membagi data secara sinusoidal (ada kelas positif ada kelas negatif).

2.2.2. *Permutation Importance (from sklearn.inspection)*

Model machine learning akan memberikan semacam 'koefisien' untuk menandai fitur mana yang bisa meningkatkan ataupun merusak akurasi prediksi. Koefisien ini tidak dapat diperoleh secara langsung, melainkan perlu diolah kembali menggunakan *permutation importance*. Seperti namanya, *permutation importance* menggunakan prinsip matematika permutasi untuk mengacak data beserta atributnya. Algoritma ini akan mengacaukan relasi antar-atribut dan mencatat perubahan error setiap permutasi iterasinya sehingga didapatkan fitur mana yang berkontribusi paling besar terhadap akurasi model *machine learning*.

2.3. *Tools*

Sumber dataset yang akan diimplementasikan pada proses KDD ini berasal dari *Kaggle*. Dataset ini kemudian diolah lebih lanjut menggunakan platform *Google Colaboratory* dengan bahasa pemrograman *Python* dengan library utama *pandas* dan *sklearn*. Link *source code* project ini terlampir pada halaman paling terakhir laporan.

BAB III
PEMBAHASAN

3.1. Data Input

Pada tahap pencarian, kami memutuskan untuk menggunakan dataset *Airlines Customer Satisfaction*. Dataset ini diperoleh dari *Kaggle* dan diakses pada tanggal 25 November 2023. Berikut link sumber dataset yang dimaksud:

https://www.kaggle.com/datasets/sjleshrac/airlines-customer-satisfaction

Dataset *Airlines Customer Satisfaction* berisikan data survei kepuasan pelanggan yang telah terbang bersama maskapai penerbangan dari berbagai aspek yang ada, termasuk juga di dalamnya informasi mengenai demografis singkat pelanggan. Dataset ini memiliki format CSV dengan ukuran 2 MB berisikan 129.880 baris data. Terdapat 22 features pada dataset, yaitu:

1. *Satisfaction* (target)
2. *Customer type*
3. *Age*
4. *Type of travel*
5. *Class*
6. *Flight distance*
7. *Seat comfort*
8. *Departure/Arrival time convenient*
9. *Food and drink*
10. *Gate location*
11. *Inflight wifi service*
12. *Inflight entertainment*
13. *Online support*
14. *Ease of online booking*
15. *On-board service*
16. *Leg room service*
17. *Baggage handling*
18. *Check-in service*
19. *Cleanliness*
20. *Online boarding*
21. *Departure Delay in Minutes*
22. *Arrival Delay in Minutes*

Pada tahap ini, dilakukan import library *Python* yang akan digunakan pada tahap selanjutnya, yakni: ***Pandas*** (data analysis utility), ***Numpy*** (mathematical function), ***Seaborn*** (visualization), ***Matplotlib*** (visualization), ***Sklearn*** (machine learning utility).

3.2. Data Preprocessing

Pada tahap ini dilakukan serangkaian prosedur pre-processing pada dataset Airlines Customer Satisfaction sebelum pemrosesan algoritma lebih lanjut. Proses-proses yang kami lakukan antara lain:

3.2.1. Metadata Information

Sebelum memulai pre-processing, penting kita ketahui lebih dahulu karakteristik dataset kita melalui metadatanya. Informasi yang dapat kita peroleh antara lain dimensi dataset (jumlah kolom dan baris), tipe data atribut, dan jumlah elemen per atribut.

```
[ ]  # check dataset size, metadata and a brief look into dataset
     print("Dataset Size:", df.shape)
     print(df.info())
     df.head()
```

```
Dataset Size: (129880, 22)
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 129880 entries, 0 to 129879
Data columns (total 22 columns):
 #   Column                            Non-Null Count   Dtype
---  ------                            --------------   -----
 0   satisfaction                      129880 non-null  object
 1   Customer Type                     129880 non-null  object
 2   Age                               129880 non-null  int64
 3   Type of Travel                    129880 non-null  object
 4   Class                             129880 non-null  object
 5   Flight Distance                   129880 non-null  int64
 6   Seat comfort                      129880 non-null  int64
 7   Departure/Arrival time convenient 129880 non-null  int64
 8   Food and drink                    129880 non-null  int64
 9   Gate location                     129880 non-null  int64
 10  Inflight wifi service             129880 non-null  int64
 11  Inflight entertainment            129880 non-null  int64
 12  Online support                    129880 non-null  int64
 13  Ease of Online booking            129880 non-null  int64
 14  On-board service                  129880 non-null  int64
 15  Leg room service                  129880 non-null  int64
 16  Baggage handling                  129880 non-null  int64
 17  Checkin service                   129880 non-null  int64
 18  Cleanliness                       129880 non-null  int64
 19  Online boarding                   129880 non-null  int64
 20  Departure Delay in Minutes        129880 non-null  int64
 21  Arrival Delay in Minutes          129487 non-null  float64
dtypes: float64(1), int64(17), object(4)
memory usage: 21.8+ MB
None
```

Gambar 4. Informasi Komponen Dataset

3.2.2. Eliminate Missing Value

Atribut *Arrival Delay in Minutes* memiliki 393 nilai kosong dan kami memutuskan untuk membuang baris tersebut. Jumlah awal 129,880 baris berkurang sebesar 0.3% menjadi 129,487 setelah missing value dihilangkan.

```
[ ]  # eliminate missing value
     df = df.dropna()
     df.shape
```

```
(129487, 22)
```

Gambar 5. Eliminasi Missing Value

3.2.3. *Check Uniquity*

Setelah menghilangkan *missing value*, kami melakukan *uniquity check* untuk melihat ada berapa nilai unik untuk masing-masing atribut dan mengidentifikasi apakah ada nominal data yang *imbalance*.

```
[ ] # check uniquity and any imbalance properties in dataset
    print(df.nunique(), "\n")
    features = df.columns

    features = features.drop('Age')
    features = features.drop('Flight Distance')
    features = features.drop('Departure Delay in Minutes')
    features = features.drop('Arrival Delay in Minutes')

    for feature in features:
      print(df[(feature)].value_counts(), "\n")
```

```
satisfaction                       2
Customer Type                      2
Age                               75
Type of Travel                     2
Class                              3
Flight Distance                 5397
Seat comfort                       6
Departure/Arrival time convenient  6
Food and drink                     6
Gate location                      6
Inflight wifi service              6
Inflight entertainment             6
Online support                     6
Ease of Online booking             6
On-board service                   6
Leg room service                   6
Baggage handling                   5
Checkin service                    6
Cleanliness                        6
Online boarding                    6
Departure Delay in Minutes       464
Arrival Delay in Minutes         472
dtype: int64
```

```
satisfied       70882
dissatisfied    58605
Name: satisfaction, dtype: int64

Loyal Customer       105773
disloyal Customer     23714
Name: Customer Type, dtype: int64

Business travel    89445
Personal Travel    40042
Name: Type of Travel, dtype: int64

Business    61990
Eco         58117
Eco Plus     9380
Name: Class, dtype: int64

3    29096
2    28645
4    28315
1    20882
5    17768
0     4781
Name: Seat comfort, dtype: int64

4    29504
5    26723
3    23110
2    22735
1    20771
0     6644
Name: Departure/Arrival time convenient, dtype: int64

3    28065
4    27129
2    27078
1    21008
5    20285
0     5922
Name: Food and drink, dtype: int64

3    33451
4    29997
2    24441
1    22497
5    19099
0        2
Name: Gate location, dtype: int64
```

Gambar 6. Uniquity Check & Imbalance

3.2.4. *Label Encoder*

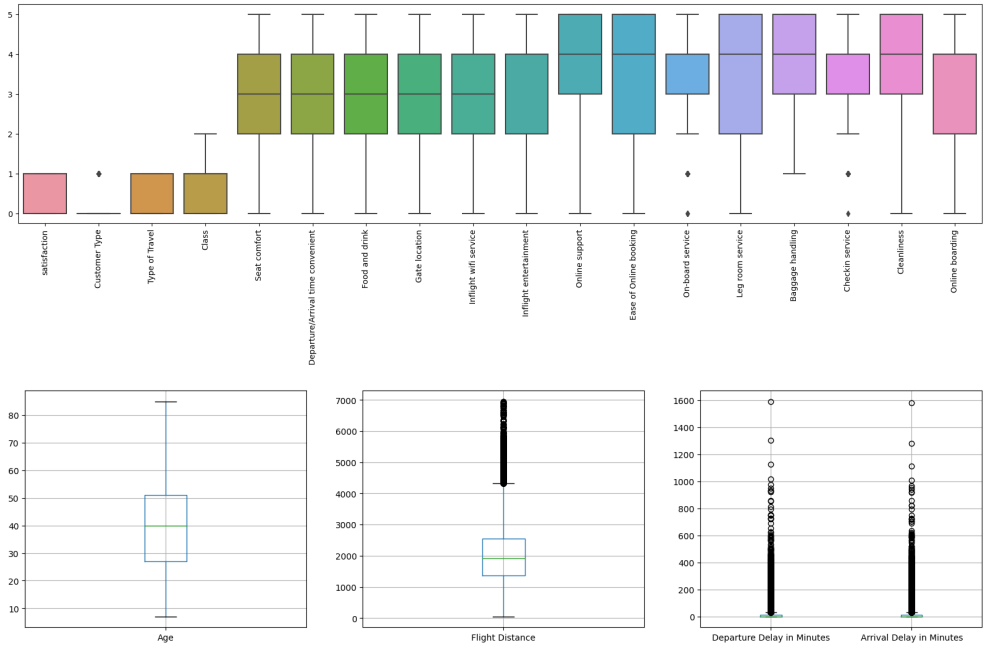
Dari pengecekan *uniquity*, kami menemukan sebanyak 4 atribut categorical yang perlu diubah menjadi numerik. Demikian kami menggunakan *Label Encoder* untuk mengubah 4 atribut tersebut ke tipe data numerik yang sesuai.

```
[ ] # convert categorical feature into numerical feature
    encoder = LabelEncoder()
    df['satisfaction'] = encoder.fit_transform(df['satisfaction'])
    df['Customer Type'] = encoder.fit_transform(df['Customer Type'])
    df['Type of Travel'] = encoder.fit_transform(df['Type of Travel'])
    df['Class'] = encoder.fit_transform(df['Class'])
```

Gambar 7. Label Encoder

3.2.5. *Outlier Analysis*

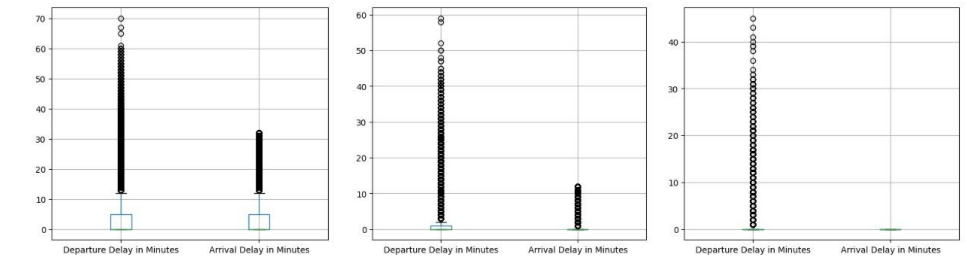
Pada tahap ini kami mengidentifikasi nilai-nilai ekstrim yang dapat mempengaruhi analisis. Data divisualisasikan dengan boxplot yang kemudian dipertimbangkan untuk mengeliminasi outlier atau tidak.



Gambar 8. Boxplot Atribut Data

Kami memutuskan untuk tidak menghilangkan outlier karena adanya anomali. Menggunakan metode *interquartil range outlier removal*, jumlah outlier belum mencapai rasio optimal meski sudah mengalami 3 iterasi pembersihan. Pertimbangan lainnya adalah jumlah data yang berkurang sangat signifikan dan berdampak negatif pada performa model prediksi, yakni sebesar 43.8% dari 129.487 menjadi 72.753 data.

```
distance = 1.5 * (np.percentile(df['Arrival Delay in Minutes'], 75) - np.percentile(df['Arrival Delay in Minutes'], 25))
df.drop(df[df['Arrival Delay in Minutes'] > distance + np.percentile(df['Arrival Delay in Minutes'], 75)].index, inplace=True)
df.drop(df[df['Arrival Delay in Minutes'] < np.percentile(df['Arrival Delay in Minutes'], 25) - distance].index, inplace=True)
```



Gambar 9. Anomali Pembersihan Outlier

3.2.6. Correlation Analysis

Correlation matrix digunakan untuk melihat korelasi antara masing-masing atribut data pada dataset. Terdapat 2 atribut data yang saling berkorelasi, *Departure Delay* & *Arrival Delay in Minutes* sebesar 0.97. Salah satu atribut data tersebut harus dihapus karena korelasi diantaranya terlalu dekat sehingga informasi yang diperoleh model tidak menambah koefisien akurasi yang signifikan (perilaku kedua fitur terlalu mirip, sehingga salah satu bisa diabaikan).

```
[ ]  # drop the highly correlated features
     df.drop('Departure Delay in Minutes', axis=1, inplace=True)
```

Gambar 10. Correlation Matrix Setelah Drop Attribute

3.2.7. Five Numbers Summary

Setelah semua proses pre-processing dijalankan, pengecekan terakhir perlu dilakukan dalam format *five numbers summary* untuk memberikan kita gambaran akhir distribusi dan informasi statistik data secara global.

```
[ ]  # final 5 numbers summary
     df.describe()
```

	satisfaction	Customer Type	Age	Type of Travel	Class	Flight Distance	Seat comfort	Departure/Arrival time convenient	Food and drink	loca
count	129487.000000	129487.000000	129487.000000	129487.000000	129487.000000	129487.000000	129487.000000	129487.000000	129487.000000	129487.00
mean	0.547406	0.183138	39.428761	0.309236	0.593704	1981.008974	2.838586	2.990277	2.852024	2.99
std	0.497749	0.386781	15.117597	0.462180	0.621371	1026.884131	1.392873	1.527183	1.443587	1.30
min	0.000000	0.000000	7.000000	0.000000	0.000000	50.000000	0.000000	0.000000	0.000000	0.00
25%	0.000000	0.000000	27.000000	0.000000	0.000000	1359.000000	2.000000	2.000000	2.000000	2.00
50%	1.000000	0.000000	40.000000	0.000000	1.000000	1924.000000	3.000000	3.000000	3.000000	3.00
75%	1.000000	0.000000	51.000000	1.000000	1.000000	2543.000000	4.000000	4.000000	4.000000	4.00
max	1.000000	1.000000	85.000000	1.000000	2.000000	6951.000000	5.000000	5.000000	5.000000	5.00

Gambar 11. Five Numbers Summary

3.3. Data Mining

Pada tahap ini, algoritma dan proses komputasi yang melibatkan *machine learning* diimplementasikan untuk membuat model prediksi pada dataset yang sudah di *pre-process* sebelumnya.

3.3.1. Train and Test Data Split

Dataset akan dibagi menjadi dua dataframe terpisah, dataframe yang pertama berisi semua atribut non-target, atribut kedua berisi hanya atribut target. Kemudian kedua dataframe tersebut akan dibagi dua kembali, menjadi data untuk *training* dan data untuk *testing*. Menggunakan fungsi yang sudah disediakan oleh *sklearn*, kami membagi dataset menjadi *x_train, x_test,*

y_train, dan *y_test*. Kami memutuskan untuk membagi dataset dengan rasio 80% dialokasikan untuk *training* dan 20% untuk *testing*.

```
# preparation for machine learning model (normalization by standard scaler and test split)
x = df.drop('satisfaction', axis=1) # features
y = df['satisfaction'] # target

x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.2, random_state=42)
```

```
Train Data:
1    56805
0    46784
Name: satisfaction, dtype: int64

Test Data:
1    14077
0    11821
Name: satisfaction, dtype: int64
```

Gambar 12. Train Test Split

3.3.2. Standard Scaler

Data sudah bisa digunakan untuk training model machine learning, namun kami memutuskan untuk melakukan standarisasi ulang format data menggunakan *standard scaler*. *Standard scaler* akan mentransformasikan data sehingga data tidak lagi memiliki rata-rata dan membuat data terkalibrasi sesuai variansi unitnya. Standarisasi akan meningkatkan performa model *machine learning*, karena data yang terstandarisasi tidak memiliki satuan ataupun skala yang berbeda tiap fiturnya dan dampak penyimpangan outlier mampu diatasi oleh model dengan baik dengan data yang terstandarisasi.

```
scaler = StandardScaler()
x_train = scaler.fit_transform(x_train)
x_test = scaler.transform(x_test)

x_train
```

```
array([[ 2.11476084, -1.21903868, -0.66826021, ...,  0.25590362,
         1.26904953, -0.2639535 ],
       [-0.47286671,  1.22745346, -0.66826021, ...,  1.12575758,
         1.26904953, -0.39493727],
       [-0.47286671,  0.1033895 , -0.66826021, ..., -0.61395035,
         0.49833755, -0.39493727],
       ...,
       [-0.47286671,  0.30175373,  1.49642308, ...,  0.25590362,
        -1.81379838, -0.15916648],
       [-0.47286671, -1.81413137,  1.49642308, ...,  0.25590362,
        -1.81379838, -0.29015025],
       [-0.47286671, -0.623946  , -0.66826021, ...,  1.12575758,
         1.26904953, -0.39493727]])
```

Gambar 13. Standard Scaler

3.3.3. Cross-validation (K-Fold Cross-validation)

Dataset akan dibagi menjadi 5 fold subset sama besar yang memiliki proporsi train data dan test data pada masing-masing subset. Setiap fold akan dihitung akurasinya, sehingga akan ada 5 iterasi *training* dan *testing* untuk setiap fold yang *score*-nya akan dirata-ratakan. Apabila *cross-validation* menunjukkan score yang konsisten dengan rata-rata akurasi tinggi, maka model *machine learning* telah bekerja dengan baik dan bisa digunakan untuk memprediksi data tanpa menghasilkan bias yang signifikan untuk pengujian tunggal.

```
[ ]  # set hyperparameter for cross-validation (window = 5)
     crossValidation = cross_val_score(SVM_model, x_train, y_train, cv=5)

     [LibSVM][LibSVM][LibSVM][LibSVM][LibSVM]
```

Gambar 14. Cross-validation

3.3.4. SVM Model Training, Fitting and Predict

Tahap terakhir adalah melakukan training model, fitting model, dan memprediksi train data yang kemudian akan dibandingkan performanya.

```
[ ]  # model fitting and predict test data
     SVM_model.fit(x_train, y_train)
     y_pred = SVM_model.predict(x_test)

     [LibSVM]
```

Gambar 15. Model Train and Predict

3.4. Post-Processing

Pada tahap ini, kami akan melakukan analisis kembali dari model yang sudah diimplementasikan. Kami akan mengambil pola dan metrik yang kemudian bisa divisualisasikan menjadi sebuah *insight*. Bagian ini masih cenderung berisi analisis teknikal dari model kernel SVM yang diuji.

3.4.1. Cross-validation Score

Setiap kernel menunjukkan *cross-validation score* yang konsisten untuk 5-*fold* iterasi. Berikut hasil *cross-validation* setiap kernel SVM yang diuji:

Linear Kernel

```
Cross-validation Score:  [0.83260933 0.83516749 0.8332368  0.83299546 0.83168412]
Overall Accuracy:  0.8331386396768428
```

Radial Basis Function Kernel

```
Cross-validation Score:  [0.94232069 0.9431895  0.94154841 0.94429964 0.94284887]
Overall Accuracy:  0.9428414214583268
```

Polynomial Kernel

```
Cross-validation Score:  [0.92089005 0.92296554 0.92122792 0.92325514 0.92074142]
Overall Accuracy:  0.9218160125631826
```

Sigmoid Kernel

```
Cross-validation Score:  [0.71807124 0.71985713 0.71575442 0.71874698 0.75184631]
Overall Accuracy:  0.7248552162014653
```

Gambar 16. Cross-validation Score setiap Kernel

3.4.2. Prediction Accuracy

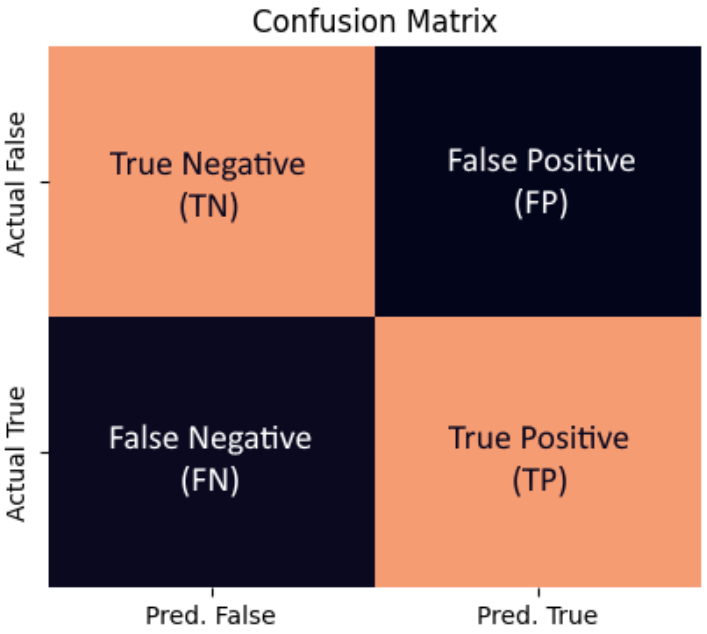
Akurasi model untuk setiap kernel SVM berbanding lurus dengan *cross-validation score* yang telah dijabarkan sebelumnya. Berikut adalah nilai akurasi model untuk setiap kernel SVM:

Kernel SVM	Akurasi Prediksi
Linear	0.8335778824619662
Radial Basis Function	0.9447061549154375
Polynomial	0.9259402270445595
Sigmoid	0.7183952428758977

Tabel 1. Perbandingan Akurasi Model

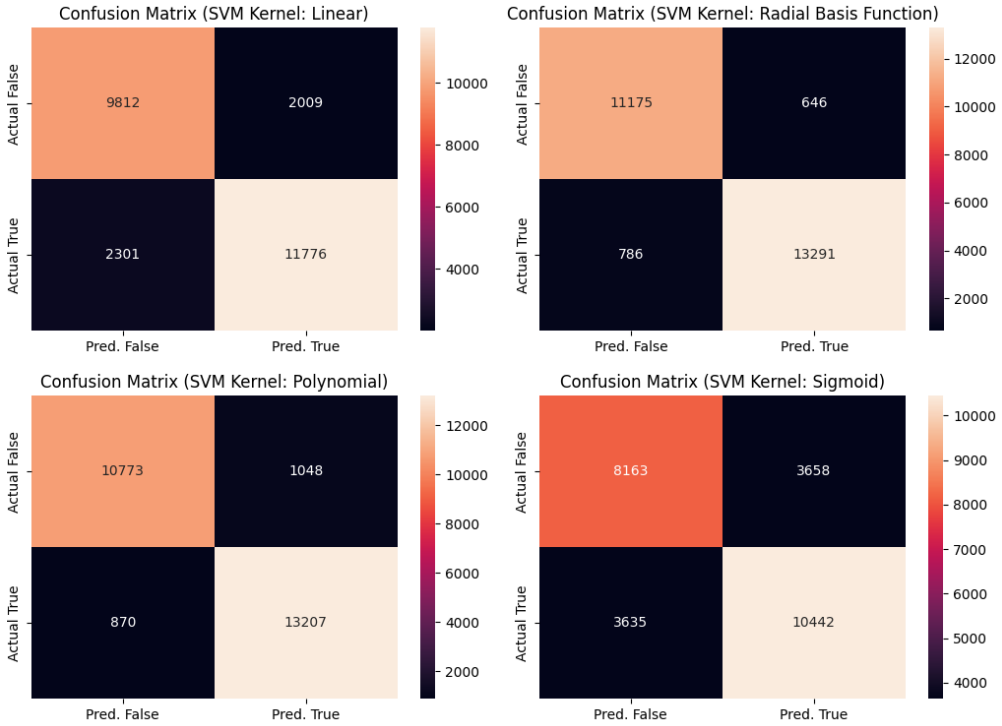
3.4.3. Confusion Matrix

Akurasi hanya memberikan kita sebagian kecil informasi mengenai performa prediksi. Oleh karena itu, kita memerlukan *confusion matrix* untuk menjelaskan lebih jauh perilaku model dalam melakukan prediksi yang dijabarkan melalui terminologi *true positive, true negative, false positive,* dan *false negative*. Perlu diketahui bahwa cara menginterpretasikan *confusion matrix* dari library *sklearn* adalah sebagai berikut:



Gambar 17. Struktur Confusion Matrix sklearn

Berikut adalah kompilasi *confusion matrix* yang diperoleh dari masing-masing kernel SVM:

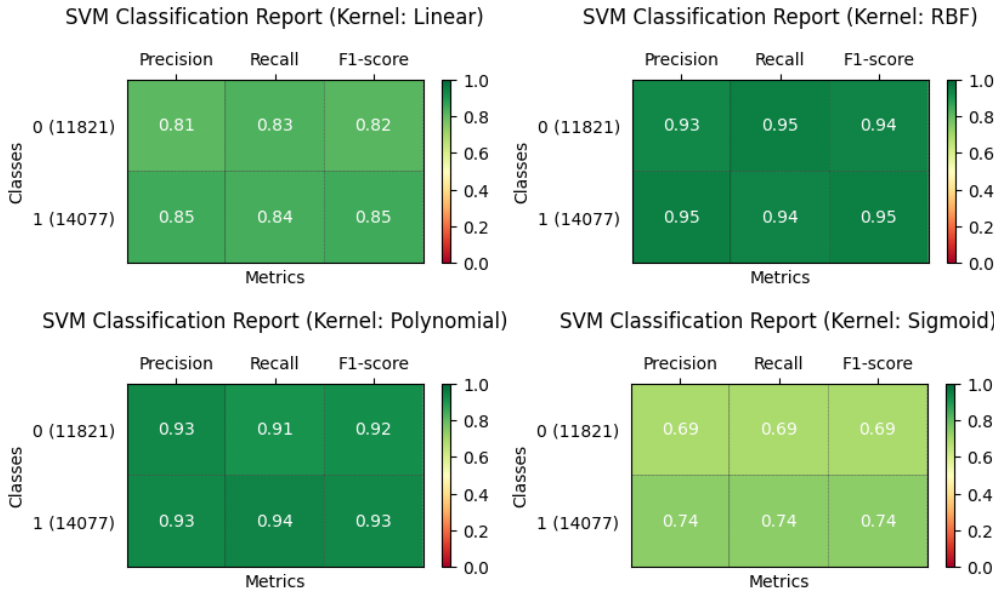


Gambar 18. Confusion Matrix setiap Kernel

3.4.4. Classification Report

Confusion matrix dapat membantu kita untuk menghitung metrik-metrik yang digunakan mengevaluasi model kernel SVM. Metrik-metrik tersebut adalah *precision, recall,* dan *f1-score* yang akan memberikan kita gambaran tentang performa model *machine learning* dalam melakukan prediksi.

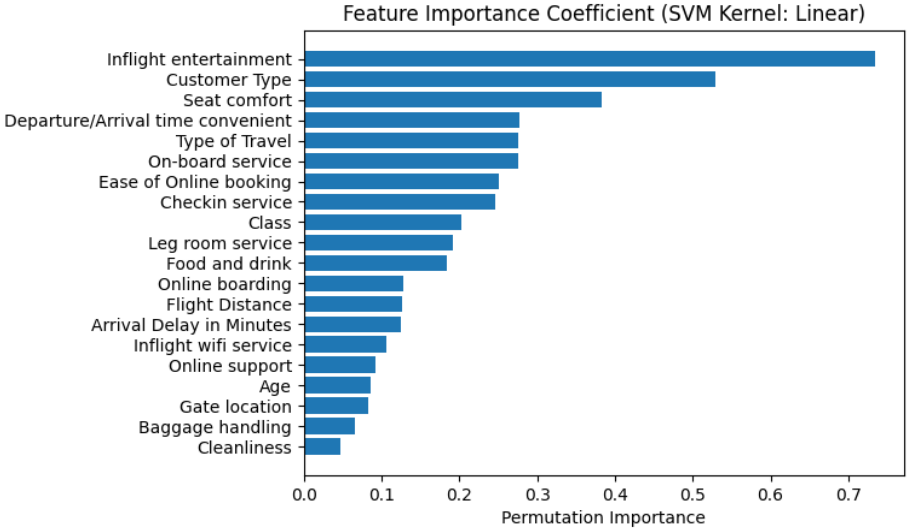
Metrik-metrik dirangkum dalam *classification report* dan berikut adalah *classification report* untuk setiap kernel SVM:



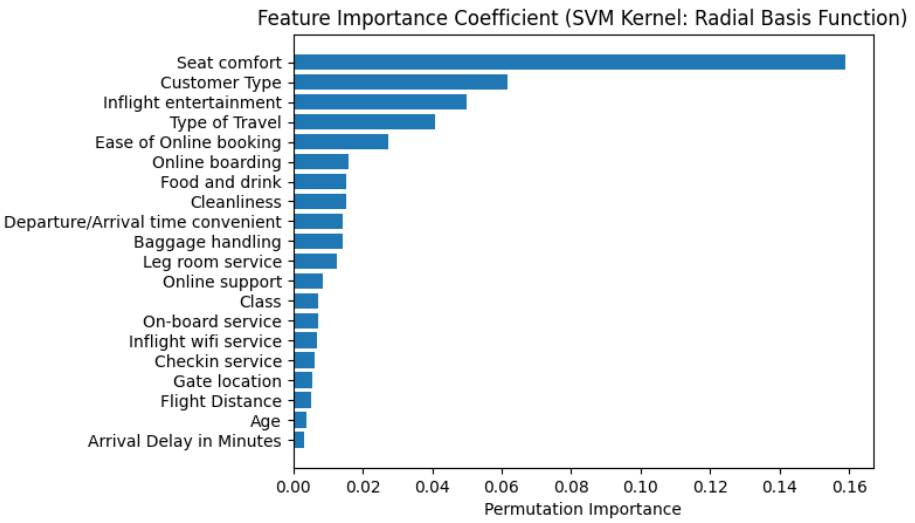
Gambar 19. Classification Report setiap Kernel

3.4.5. Permutation Importance

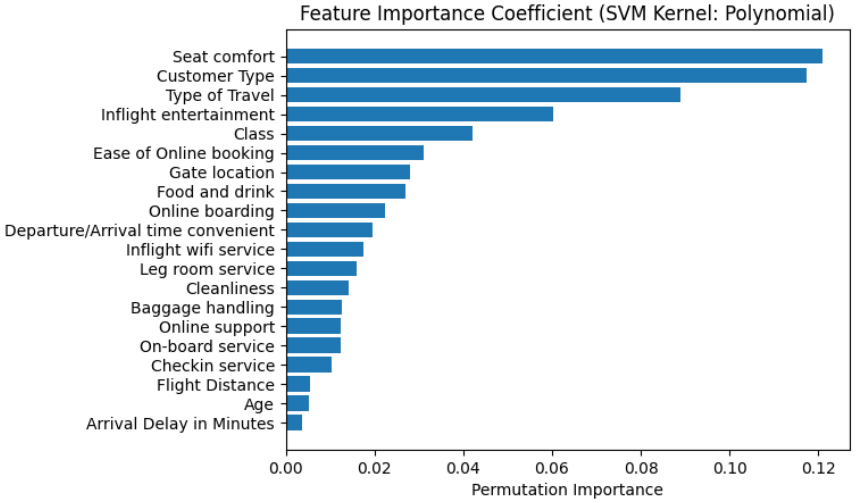
Kegiatan *post-processing* yang terakhir ialah mengidentifikasi fitur apa saja yang menurut masing-masing model SVM memiliki kontribusi terbesar terhadap peningkatan akurasi prediksi. Pola dan informasi ini dapat diperoleh melalui *permutation importance* dari masing-masing model kernel SVM:



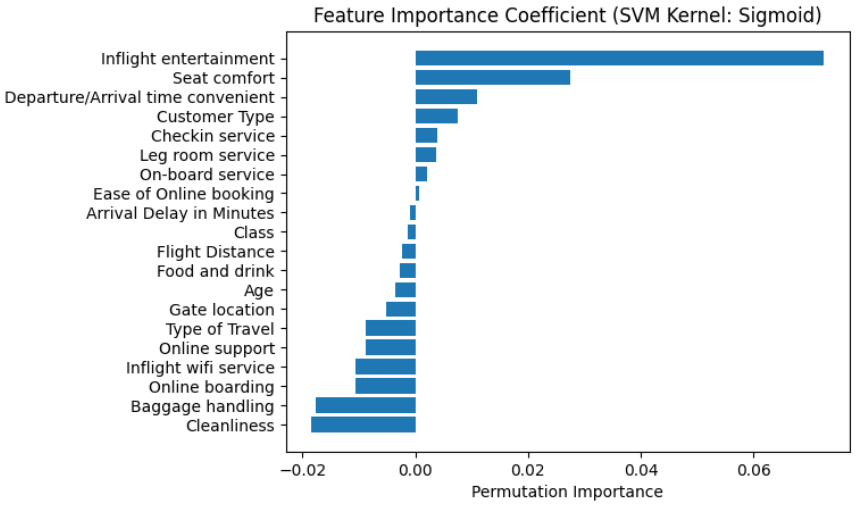
Gambar 20. Permutation Importance Kernel Linear



Gambar 21. Permutation Importance Kernel Radial Basis Function



*Gambar 22. Permutation Imp*ortance Kernel Polynomial



Gambar 23. Permutation Importance Kernel Sigmoid

3.5. Pattern Information Knowledge

Analisis pola dari bagian *post-processing* project ini kami bagi menjadi dua aspek untuk dipresentasikan sebagai *pattern information knowledge*: *insight* teknikal dan *insight* bisnis.

3.5.1. *Insight* Teknikal

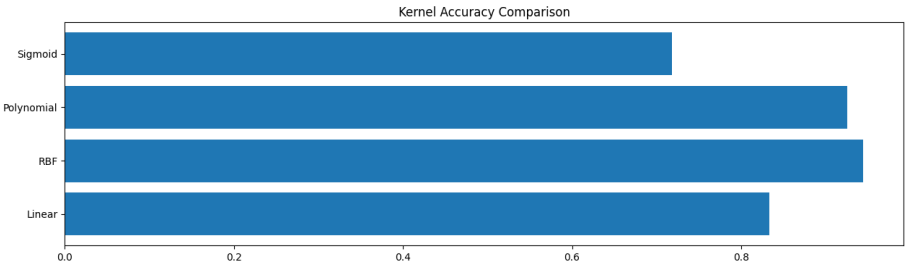
Berbagai informasi teknikal dapat diturunkan dari proses-proses yang telah dilakukan. Berikut adalah poin-poin penting mengenai *insight* teknikal:

a. Karakteristik dan perilaku dataset

Dataset menunjukkan sedikit *missing value*, namun memiliki anomali outlier. Ukuran dataset dapat dikatakan cukup besar dan berdimensi tinggi (129.880 baris, 23 kolom). Distribusi dan variansi data dapat dikatakan cukup seimbang. Beberapa fitur menunjukkan keterkaitan dalam analisis korelasi namun tidak mengganggu kinerja keempat kernel SVM dalam melakukan prediksi.

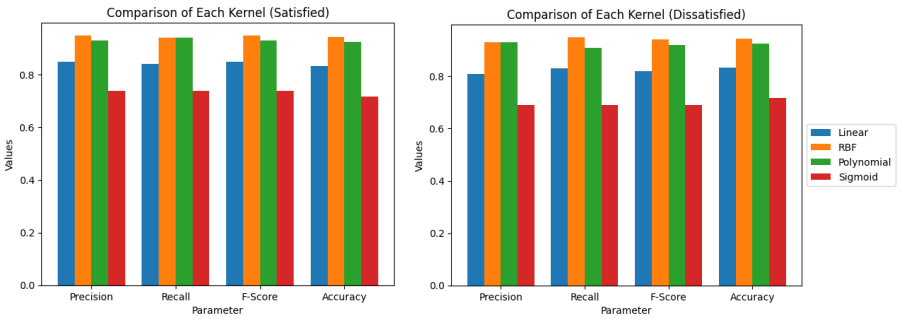
b. Perbandingan metrik antar-kernel SVM

Secara global, akurasi dan metrik yang diperoleh dari keempat kernel SVM menunjukkan metrik (*accuracy, precision, recall, f1-score*) yang baik (rentang ~0.7 sampai ~0.9). Akurasi tertinggi dicapai oleh kernel RBF (94%) dan kernel Polynomial (92%) kemudian diikuti oleh kernel Linear (83%) dan kernel Sigmoid (71%).



Gambar 24. Perbandingan Akurasi Kernel SVM

Keempat model kernel secara umum menunjukkan selisih dan disparitas yang mirip untuk setiap metrik dan kernel RBF mengungguli hampir semua metrik tersebut.



Gambar 25. Perbandingan Semua Metrik Kernel SVM

c. Analisis kesesuaian model terhadap dataset

Terdapat perbedaan durasi *runtime training* dalam menjalankan masing-masing kernel. Kernel yang memiliki *runtime training* paling singkat yakni RBF (4 menit) dan Polynomial (5 menit), sementara *runtime* terlama ada pada kernel Linear (24 menit) dan Sigmoid (31 menit). Hal ini dikarenakan kernel RBF dan kernel Polynomial memiliki pendekatan paling adaptif terhadap dataset berdimensi tinggi dan kemampuan untuk menangani klasifikasi yang tidak bisa dioptimalkan dengan baik oleh fungsi linear ataupun sigmoid. Demikian, kernel RBF dan kernel Polynomial lebih efektif dalam mengidentifikasi kelas-kelas dalam dataset kami.

Kesesuaian antara model kernel dan dataset dapat dilihat dari *f1-score* yang merupakan indikator keseimbangan (*harmonization*) antara rasio *recall* (*TP/(TP+FN)*) dan *precision* (*TP/(TP+FP)*). *F1-score* tertinggi dimiliki oleh kernel RBF (0.94 dan 0.95) yang dapat diartikan bahwa model SVM kernel RBF melakukan analisis prediktif dengan *margin of error* (*recall = 0.95 dan 0.94; precision* = 0.93 dan 0.95) yang minimum pada dataset.

Permutation importance juga dapat menjadi indikator kesesuaian model kernel terhadap dataset. Hal ini dapat dilihat pada kernel sigmoid yang memiliki koefisien negatif pada analis *permutation importance*. Koefisien negatif memiliki arti bahwa suatu fitur bersifat meniadakan, merusak, atau mengacaukan algoritma model dalam mempelajari data training, demikian koefisien negatif dihindari dalam menentukan model machine learning yang sesuai.

3.5.2. *Insight* Bisnis

Informasi bisnis diperoleh dari analisis korelasi antar-fitur yang kemudian akan dicocokkan lagi dengan hasil dari *permutation importance* dari model yang telah dibuat menggunakan kernel RBF dan Polynomial. Berikut adalah beberapa informasi penting yang bisa dijadikan pertimbangan dari maskapai sebagai penggerak bisnis aviasi agar bisa meningkatkan profit sebagai pengaruh dari meningkatnya kepuasan pelanggan:

a. Korelasi *departure delay in minute* dan *arrival delay in minute*

Korelasi antar kedua fitur ini dengan nilai 0.97 menunjukkan meningkatnya durasi lama atau singkatnya *departure delay* akan berbanding lurus dengan *arrival delay*. Kita dapat berasumsi bahwa kedua faktor sama-sama tidak disukai oleh pelanggan. Demikian, semakin lama delay yang dialami oleh pelanggan, bisa menjadi faktor penting penentu kepuasan pelanggan terhadap maskapai.

b. Fasilitas yang dipilih pelanggan

Kenyamanan selama di dalam pesawat adalah yang terpenting bagi pelanggan. Terlihat dari *correlation matrix* bahwa ***inflight entertainment*** adalah fitur dengan korelasi tertinggi (0.52) dengan ***satisfaction,*** dan ***seat comfort*** memiliki korelasi erat (0.72) dengan ***food and drink***. Hal ini juga didukung dalam analisis *permutation importance* pada kernel RBF dan Polynomial, dimana 2 dari 4 fitur terpenting yang diperoleh ialah ***seat comfort*** dan ***inflight entertainment,*** sisanya adalah ***customer type*** dan ***type of travel***. Maka dari itu, pihak maskapai bisa meningkatkan strategi marketingnya seperti memberikan bundling pemilihan kursi dan *benefit* mendapatkan makanan/minuman atau menambah jumlah *entertainment* dalam pesawat.

c. Layanan pada platform online

Terlihat pada *feature ease of online booking* dengan *online support* dan *ease of online booking* dengan *online boarding* cukup mempengaruhi satu sama lain menandakan bahwa layanan *online* lebih efisien dan memudahkan pelanggan dibandingkan harus melakukannya secara *onsite*. Untuk itu, maskapai harus bisa menjaga konsistensi dan keselarasan informasi seperti informasi *boarding*, *gate* yang dituju serta *departure/arrival time* kepada pelanggan yang nantinya ini akan menjadi nilai tambah dari pelanggan kepada maskapai karena keefisiensian layanan onlinenya.

BAB IV
PENUTUP

4.1. Hasil

Berdasarkan serangkaian proses analisis yang telah dilakukan pada dataset "*Airline Customer Satisfaction*" dengan menggunakan 4 jenis kernel dari model SVM, yaitu; *Linear, RBF, Polynomial* dan *Sigmoid*, ditemukan hasil bahwa kernel RBF memiliki hasil akurasi yang paling baik diantara kernel yang lainnya dengan hasil tingkat akurasi 94% dan *f1-score* paling mumpuni (0.95 dan 0.94) yang menunjukkan keseimbangan *recall* dan *precision* dengan *error* paling minimum. Dimana, ini menandakan bahwa kernel RBF lebih tepat dalam memodelkan dan memprediksi kepuasan pelanggan dalam konteks data yang dianalisis. Kenyamanan di dalam pesawat sangat penting bagi pelanggan. *Inflight entertainment* memiliki korelasi tertinggi (0,52) dengan *satisfaction*, sementara *seat comfort* berhubungan erat (0,72) dengan *food and drink*. Analisis menunjukkan bahwa *seat comfort, food and drink, inflight entertainment,* dan *type of travel* krusial dalam menentukan *satisfaction*. Korelasi tinggi (0,97) antara *departure delay* dan *arrival delay* bahwa pelanggan tidak menyukai keterlambatan. Semakin lama keterlambatan, semakin berpengaruh terhadap kepuasan pelanggan terhadap maskapai.

4.2. Evaluasi

Selama berlangsungnya serangkaian proses analisis, ada beberapa kendala yang muncul dimana salah satunya pada saat visualisasi boxplot dimana terdapat pertimbangan dalam mengeliminasi outliernya yang mana pada akhirnya diputuskan untuk tidak menghilangkan outlier dikarenakan anomali. Untuk mengatasi problem ini dapat dilakukan segmentasi data, dimana model akan dibuat menjadi 2 yaitu model dengan data dengan outlier dan tanpa outlier, yang mana ini dapat membantu menentukan seberapa berpengaruhnya outlier tersebut bagi model.

4.3. Kesimpulan

Kami menganalisis dataset "*Airlines Customer Satisfaction"* menggunakan metode SVM dan mengidentifikasi data dalam beberapa tahapan penting dalam proses data mining. Dimulai dengan data input dan dilanjutkan dengan data preprocessing, kami melakukan eliminasi *missing values*, penanganan *outlier*, *label encoding*, dan analisis korelasi antar atribut. Pada tahap data mining, kami membagi dataset menjadi *train dan test set*, melakukan S*tandard Scaler*, dan menerapkan SVM dengan berbagai jenis kernel seperti *Linear, RBF, Polynomial* dan *Sigmoid*. Dari hasil cross-validation dan evaluasi model, dapat disimpulkan bahwa hasil perhitungan analisis dengan kernel RBF (94%) dan Polynomial (92%) memberikan tingkat akurasi yang lebih tinggi dibandingkan dengan kernel Linear (83%) dan Sigmoid (71%). Hal itu menandakan bahwa kedua jenis kernel ini memiliki fleksibilitas yang tinggi sehingga lebih efektif dalam mengidentifikasi pola-pola yang kemudian bisa diturunkan menjadi *valuable insight* dalam dataset yang bersifat non-linear yang kami miliki.

DAFTAR PUSTAKA

Jana, S. (2020, March 19). Airlines Customer satisfaction [Dataset]. Kaggle. URL: https://www.kaggle.com/datasets/sjleshrac/airlines-customer-satisfaction

Scikit-learn developers (BSD License). (2007 - 2023). sklearn.svm.SVC. scikit-learn. URL: sklearn.svm.SVC — scikit-learn 1.3.2 documentation

Scikit-learn developers (BSD License). (2007 - 2023). Permutation feature importance. scikit-learn. URL: https://scikit-learn.org/stable/modules/permutation_importance.html

Murray, G. & Heilakka, R. (n.d.). The Airline Pilot Shortage Will Get Worse. OliverWyman. URL: https://www.oliverwyman.com/our-expertise/insights/2022/jul/airline-pilot-shortage-will-get-worse.html

Rahayu, I. R. S., & Idris, M. (2021, December 30). Daftar Nama Maskapai Penerbangan di Indonesia. Kompas. URL: https://money.kompas.com/read/2021/12/30/154240326/daftar-nama-maskapai-penerbangan-di-indonesia#google_vignette

Bouwer, J. , Saxon, S. , Wittkam, N. (2021, April 2). Back to the future? Airline sector poised for change post-COVID-19. McKinsey. URL : https://www.mckinsey.com/industries/travel-logistics-and-infrastructure/our-insights/back-to-the-future-airline-sector-poised-for-change-post-covid-19

Donovan, D. (2020, Mach 30). How The Airline Industry Will Transform Itself As It Comes Back From Coronavirus. Forbes. URL : https://www.forbes.com/sites/deandonovan/2020/03/30/how-the-airline-industry-will-transform-itself-as-it-comes-back-from-cornonavirus/?sh=307bcb3067b9

LAMPIRAN

SVM Linear

https://colab.research.google.com/drive/17_oCH5WZFOKQTgjnox6UWTsBr5c61tGF?usp=sharing

SVM RBF

https://colab.research.google.com/drive/1Gyr5wPMmB301jIhw3eUe4QVc3agbm4Cf?usp=sharing

SVM Polynomial

https://colab.research.google.com/drive/1K_LIGAXR0eEy6WMDDiim8kTLj0jl2WMj?usp=sharing

SVM Sigmoid

https://colab.research.google.com/drive/1skV12samLSySSoCKDnzJsb90mV39Wi6f?usp=sharing

Post Processing:

1. Plot Classification Report per Kernel:

https://colab.research.google.com/drive/1O3AqPmyZEUUa9wUDKvXRujG4TyTxZyXG?usp=sharing

2. Analisis Tambahan

https://colab.research.google.com/drive/1aarnEme2R1EtpfVmmDA89Um5X8wgenUQ?usp=sharing