predicting customer satisfaction with

RANDOM FOREST

data science portfolio

- Ayatullah Reza Chalid
- Lutfia Husna Khoirunnisa

What is

CLASSIFICATION?

Klasifikasi merupakan suatu **algortima atau teknik** yang dapat digunakan untuk membuat skema atau membuat suatu kategori dari data yang berlabel

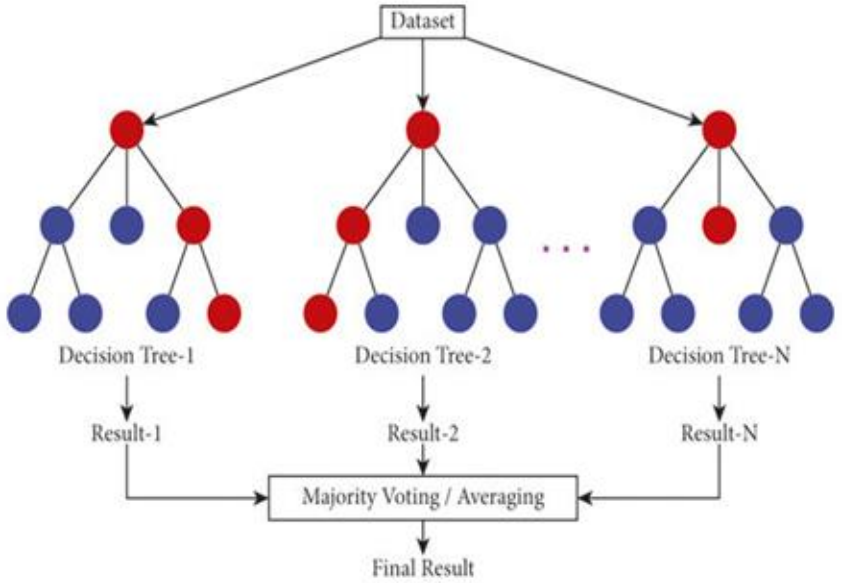
Klasifikasi dapat membantu kita dalam berbagai bidang, dalam project ini klasifikasi digunakan dalam **memprediksi kepuasan** penumpang suatu maskapai pesawat terbang

RANDOM FOREST

Random Forest merupakan salah satu metode klasifikasi yang berisikan berisikan kumpulan **pohon-pohon klasifikasi** dengan random sub sampling atau pemilihan m variabel yang digunakan dalam membangun pohon.

RANDOM FOREST

Pohon klasifikasi sendiri dapat diartikan sebagai suatu diagram yang memiliki bentuk seperti **pohon** dan memiliki *root node* sebagai sumber data, *inner node* yang berisi pertanyaan sebagai pengklasifikasi data, dan *leaf node* yang merupakan hasil keputusan pengklasifikasian data tersebut.



RANDOM FOREST

Algoritma random forest diawali dengan memilih sample acak dari data, yang kemudian dilanjutkan dengan membuat *decision tree* untuk tiap sample yang dipilih dan setelah diperoleh hasil prediksi dari setiap *decision tree* yang dibuat. Selanjutnya akan diambil hasil klasifikasi menggunakan nilai yang paling sering muncul (modus) dari hasil prediksi sebagai nilai prediksi akhir.

Introduction to the **PROJECT**

Projek ini bertujuan untuk mengetahui tingkat akurasi dari penggunaan *Random forest clasifier* pada kasus kepuasan pelanggan dalam maskapai penerbangan. Data yang digunakan dapat diakses ***di sini.***

Tools:

Python

Google Slides

Note:

Coding yang digunakan dalam *project* ini dapat diakses https://colab.research.google.com/drive/1ppEZUnMqUVyZsN0N7zP7mLTlZWKeRIDB?usp=sharing

steps:

Data and Problem Understanding

Melihat masalah awal, dan data yang akan digunakan dalam pemecahan masalah tersebut

Data Preparation and Overview

Melakukan *data cleansing* dan melihat gambaran secara umum maupun statistik dari data

Modeling

Memodelkan data untuk mencari penyelesaian dari masalah

Evaluation

Melakukan evaluasi dari model yang dibuat

Data and Problem Understanding

Problem:

Sebuah perusahaan maskapai penerbangan ingin melakukan prediksi kepuasan dari *customer* hanya dengan melihat hasil survey berdasarkan beberapa kriteria yang diajukan kepada *customernya.*

Data:

Data yang akan digunakan memiliki 25 field, dengan 103903 record. Data tersebut merupakan hasil survey kepuasan dari beberapa kriteria yang diajukan kepada *customer* suatu maskapai penerbangan dengan kepuasan final dari *customer* yang dinyatakan pada field *'satisfaction'*.

```
#Data overview
print(dataset.info())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 103904 entries, 0 to 103903
Data columns (total 25 columns):
 #   Column                            Non-Null Count   Dtype
---  ------                            --------------   -----
 0   Unnamed: 0                        103904 non-null  int64
 1   id                                103904 non-null  int64
 2   Gender                            103904 non-null  object
 3   Customer Type                     103904 non-null  object
 4   Age                               103904 non-null  int64
 5   Type of Travel                    103904 non-null  object
 6   Class                             103904 non-null  object
 7   Flight Distance                   103904 non-null  int64
 8   Inflight wifi service             103904 non-null  int64
 9   Departure/Arrival time convenient 103904 non-null  int64
 10  Ease of Online booking            103904 non-null  int64
 11  Gate location                     103904 non-null  int64
 12  Food and drink                    103904 non-null  int64
 13  Online boarding                   103904 non-null  int64
 14  Seat comfort                      103904 non-null  int64
 15  Inflight entertainment            103904 non-null  int64
 16  On-board service                  103904 non-null  int64
 17  Leg room service                  103904 non-null  int64
 18  Baggage handling                  103904 non-null  int64
 19  Checkin service                   103904 non-null  int64
 20  Inflight service                  103904 non-null  int64
 21  Cleanliness                       103904 non-null  int64
 22  Departure Delay in Minutes        103904 non-null  int64
 23  Arrival Delay in Minutes          103594 non-null  float64
 24  satisfaction                      103904 non-null  object
dtypes: float64(1), int64(19), object(5)
memory usage: 19.8+ MB
None
```

```
Checking missing value for each feature:
Unnamed: 0                            0
id                                    0
Gender                                0
Customer Type                         0
Age                                   0
Type of Travel                        0
Class                                 0
Flight Distance                       0
Inflight wifi service                 0
Departure/Arrival time convenient     0
Ease of Online booking                0
Gate location                         0
Food and drink                        0
Online boarding                       0
Seat comfort                          0
Inflight entertainment                0
On-board service                      0
Leg room service                      0
Baggage handling                      0
Checkin service                       0
Inflight service                      0
Cleanliness                           0
Departure Delay in Minutes            0
Arrival Delay in Minutes            310
satisfaction                          0
dtype: int64

Counting total missing value:
310
Duplicated data count: 0
```

Data Preparation

Diketahui terdapat 310 data hilang pada field 'Arrival Delay in Minutes', dan tidak terdapat data duplikat.

Selanjutnya *records* data yang memiliki *missing value* akan dihapus untuk menghindari bias pada hasil pemodelan.

Data Preparation

```
[7]  #Delete the missing datas
     dataset = dataset.dropna()
```

```
[27] print('Counting total missing value:')
     print(dataset.isna().sum().sum())

     Counting total missing value:
     0
```

Record data yang memiliki *missing value* dihapus agar tidak terjadi bias, dan tidak mengurangi keakuratan pemodelan data

```
#print the number of columns and rows of the dataset
print('Num of Rows, Num of Columns:', dataset.shape)

Num of Rows, Num of Columns: (103594, 25)
```

Setelah dilakukan data cleansing, record data yang awalnya sebanyak 103903 berubah menjadi 103594

Data Overview

(computes and displays summary statistics for the dataset)

	Age	Flight Distance	Inflight wifi service	Departure/Arrival time convenient	Ease of Online booking	Gate location
count	103594.000000	103594.000000	103594.000000	103594.000000	103594.000000	103594.000000
mean	39.380466	1189.325202	2.729753	3.060081	2.756984	2.977026
std	15.113125	997.297235	1.327866	1.525233	1.398934	1.277723
min	7.000000	31.000000	0.000000	0.000000	0.000000	0.000000
25%	27.000000	414.000000	2.000000	2.000000	2.000000	2.000000
50%	40.000000	842.000000	3.000000	3.000000	3.000000	3.000000
75%	51.000000	1743.000000	4.000000	4.000000	4.000000	4.000000
max	85.000000	4983.000000	5.000000	5.000000	5.000000	5.000000

Data Overview

(computes and displays summary statistics for the dataset)

	Food and drink	Online boarding	Seat comfort	Inflight entertainment	On-board service	Leg room service
count	103594.000000	103594.000000	103594.000000	103594.000000	103594.000000	103594.000000
mean	3.202126	3.250497	3.439765	3.358341	3.382609	3.351401
std	1.329401	1.349433	1.318896	1.333030	1.288284	1.315409
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	2.000000	2.000000	2.000000	2.000000	2.000000	2.000000
50%	3.000000	3.000000	4.000000	4.000000	4.000000	4.000000
75%	4.000000	4.000000	5.000000	4.000000	4.000000	4.000000
max	5.000000	5.000000	5.000000	5.000000	5.000000	5.000000

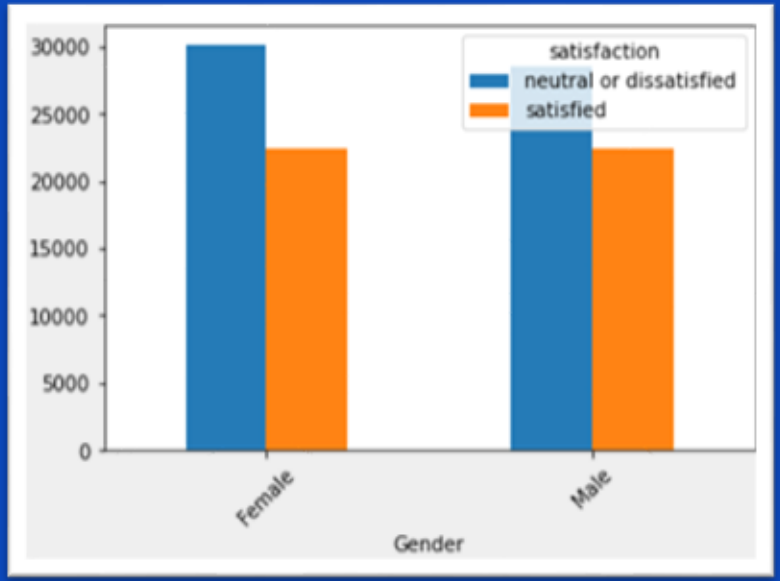
Data Overview

(computes and displays summary statistics for the dataset)

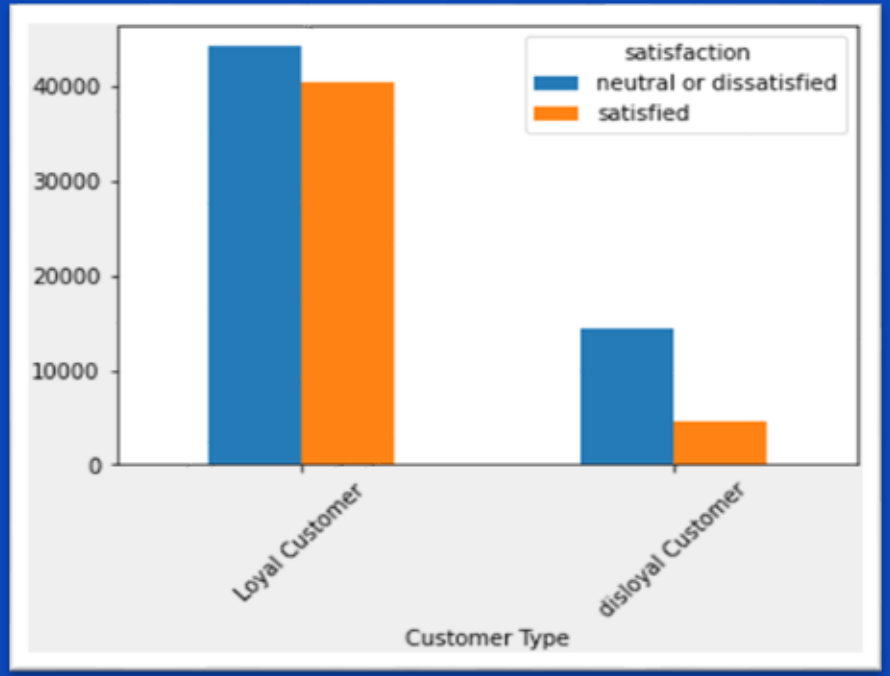
	Baggage handling	Checkin service	Inflight service	Cleanliness	Departure Delay in Minutes	Arrival Delay in Minutes
count	103594.000000	103594.000000	103594.000000	103594.000000	103594.000000	103594.000000
mean	3.631687	3.304323	3.640761	3.286397	14.747939	15.178678
std	1.181051	1.265396	1.175603	1.312194	38.116737	38.698682
min	1.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	3.000000	3.000000	3.000000	2.000000	0.000000	0.000000
50%	4.000000	3.000000	4.000000	3.000000	0.000000	0.000000
75%	5.000000	4.000000	5.000000	4.000000	12.000000	13.000000
max	5.000000	5.000000	5.000000	5.000000	1592.000000	1584.000000

Data Overview

(visualizing the distribution of the data)



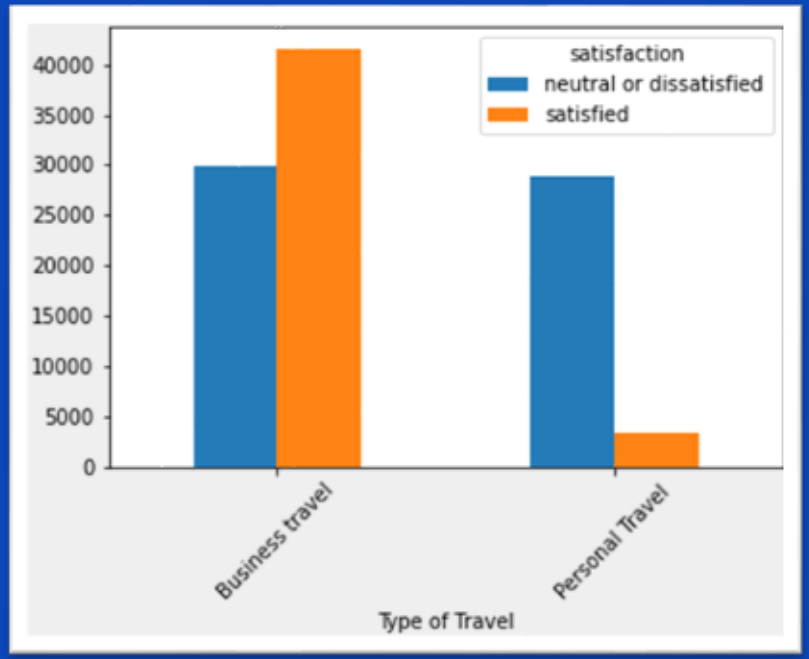
Gender distribution



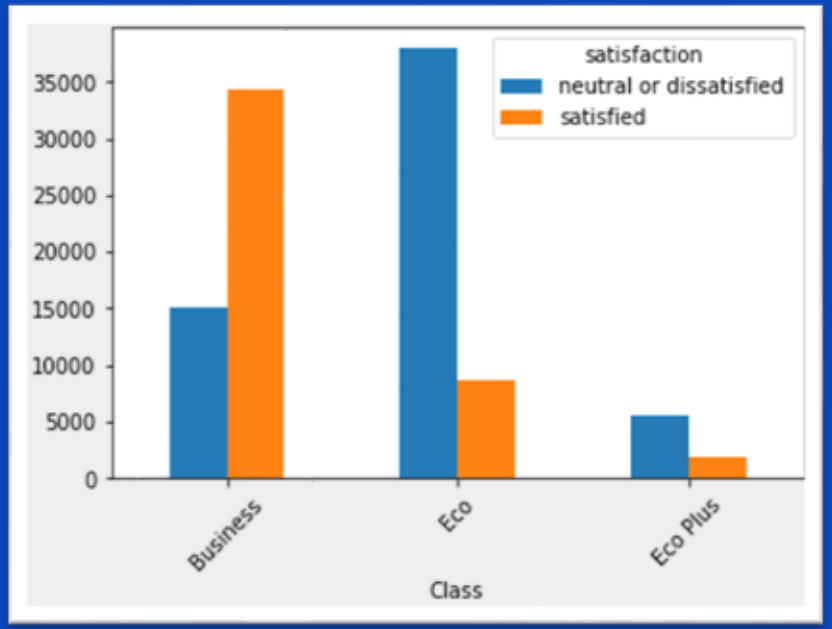
Customer Type distribution

Data Overview

(visualizing the distribution of the data)



Type of Travel distribution



Class distribution

Data Preparation

Untuk mempermudah, kita mengganti “satisfaction” dengan angka “1” sedangkan “dissatisfied” dengan angka “0”

```
# change data to categorical = binary-class
# satisfied -> 1, dissatisfied -> 0

dataset['satisfaction'] = dataset['satisfaction'].apply(lambda x: 0 if 'dissatisfied' in x else 1)
```

Memisahkan antara kolom “target” dan “features”

```
#Separate the target and features column
target = dataset[['satisfaction']]
features = dataset.drop(['Unnamed: 0','id','satisfaction'], axis=1)
```

Mengubah data yang bersifat kategorik menjadi indikator variabel

```
#converts categorical data of features dataset into dummy or indicator variables
features = pd.get_dummies(features)
```

Data Preparation (data splitting)

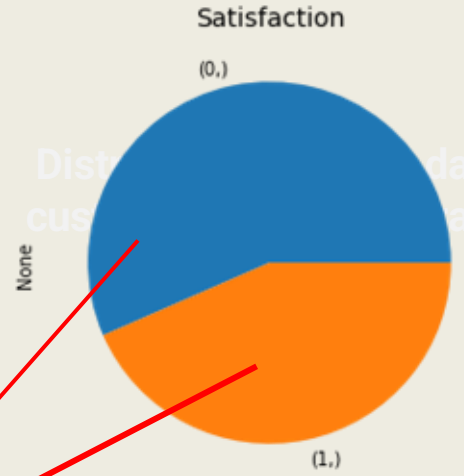
```
# split data
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(features, target, test_size = 0.3, random_state = 42)
print('length of X_train :', len(X_train))
print('length of y_train :', len(y_train))
print('length of X_test', len(X_test))
print('length of y_test', len(y_test))
```

```
length of X_train : 72515
length of y_train : 72515
length of X_test 31079
length of y_test 31079
```

Untuk membuat *machine learning* bekerja dengan baik, kami membagi dataset menjadi dua bagian, sehingga kami mendapat hasil data berupa data train yang berisikan 72515 record dan data test berisikan 31079 record

```
Value Counts:
 satisfaction
0                17583
1                13496
```

Distribusi kepuasan dari customer pada *data train*



Modeling

```
# Import the model we are using
from sklearn.ensemble import RandomForestClassifier

# Instantiate model with 1000 decision trees
rf = RandomForestClassifier(n_estimators = 1000, random_state = 42)
# Train the model on training data
rf.fit(X_train, y_train);
```

n_estimator adalah banyak pohon di dalam hutan

Prediksi pada data testing dilakukan dan hasilnya disimpan dalam variable y_predict

```
# Predict y data with classifier:
y_predict = rf.predict(X_test)
```

Evaluation

```
# Print results:
from sklearn.metrics import classification_report, confusion_matrix

print("confusion matrix:\n",confusion_matrix(y_test, y_predict))
print("\nModel report:\n",classification_report(y_test, y_predict))
```

```
confusion matrix:
 [[17223   360]
 [  803 12693]]

Model report:
              precision    recall  f1-score   support

           0       0.96      0.98      0.97     17583
           1       0.97      0.94      0.96     13496

    accuracy                           0.96     31079
   macro avg       0.96      0.96      0.96     31079
weighted avg       0.96      0.96      0.96     31079
```

Terdapat hasil prediksi yang tidak sesuai dengan data aslinya

Model yang dibangun memiliki akurasi sebesar

96%

Conclusion

Output yang kami dapat ialah penggunaan metode *random forest* pada kasus prediksi kepuasan *customer* maskapai penerbangan memiliki tingkat akurasi sebesar 96%

Thank you!

linkedin.com/ayatullahreza
linkedin.com/lutfiahusnakhoirunnisa