MACHINE LEARNING

PROJECT REPORT

- CRISP-DM Methodology
- Business Understanding and Data Understanding
- Data Preparation and Modelling
- Evaluation

https://kmmi.kemdikbud.go.id

DAFTAR ISI

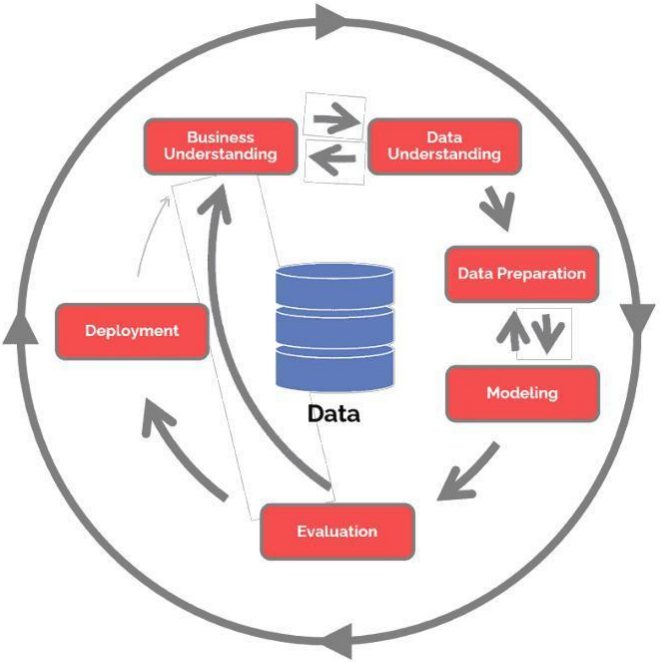
CRISP-DM METHODOLOGY

CRoss Industry Standard Process for Data Mining (CRISP-DM) adalah model proses dengan enam fase yang secara alami menggambarkan life cycle data science yang menggunakan Machine Learning. Metodologi ini akan membantu anda merencanakan, mengatur, dan mengimplementasikan proyek.

CRISP-DM dan Metodologi Data science dari IBM diawali dengan kegiatan Business Understanding yang merupakan proses pemahaman terhadap masalah yang akan diselesaikan. Di dalam kegiatan tersebut juga dilakukan proses pemetaan antara masalah bisnis dengan tugas analitiks (tugas data science yang sesuai).

Berikutnya kegiatan pemahaman terhadap data (Data Understanding) yang meliputi penentuan kebutuhan data, pengumpulan data dan eksplorasi data. Pada Metodologi IBM masing-masing sub kegiatan dijadikan proses tersendiri.

Langkah berikutnya adalah Data Preparation yang dilakukan untuk memperbaiki kualitas data agar sesuai dengan proses Modeling yang akan dilakukan berikutnya. Kualitas model yang dihasilkan di evaluasi (Evaluation) sebelum dideploy menjadi sistem operasional. Rangkaian kegiatan diakhiri dengan proses feedback dan pelaporan.



Gambar 1. Siklus Metode CRISP-DM

Business Understanding

Sejak peluncurannya di tahun 2007, Airbnb telah menjadi pilihan yang sangat populer di kalangan pelancong mancanegara karena menawarkan pengalaman yang unik dan alternatif yang relatif lebih murah dibandingkan dengan hotel. Saat ini, Airbnb sudah memiliki lebih dari 5.6 juta properti yang tersebar di lebih dari 220 negara di dunia dalam katalognya. Jasa dari Airbnb bisa diakses oleh usernya baik melalui webapp, dan aplikasi di platform Android dan iOS.

Masalah yang diharapkan akan diselesaikan dari proyek ini adalah memprediksi harga dari properti yang akan di-listing. Prediksi harga yang akurat diharapkan bisa ditambahkan menjadi fitur rekomendasi harga di Airbnb sehingga bisa membantu para host terutama yang baru mulai mencoba untuk menyewakan propertinya sehingga mereka tidak kesulitan dalam menentukan harga yang sesuai untuk properti yang mereka miliki.

Untuk melakukan prediksi harga ini, metode yang akan digunakan adalah metode Support Vector Machine dan juga Artificial Neural Network. Selain itu dalam melakukan prediksi harga, seringkali prediksi yang diberikan bisa melenceng sangat jauh. Oleh karena itu, evaluation metric yang akan digunakan adalah Mean Absolute Percentage Error (MAPE) yang tidak terlalu dipengaruhi apabila ada prediksinya melenceng cukup jauh.

Data Understanding

Data yang akan digunakan merupakan dataset dari Airbnb Singapura. Dataset ini terdiri dari 2 file yaitu file “listings.csv” yang berisi tabel utama dan juga file “neighbourhoods.csv” yang berisi tabel bantuan yang memberi informasi mengenai daerah yang ada di sekitar suatu lokasi tertentu.

Data Exploration

Berikut adalah seluruh kolom dari tabel yang ada di file “listings.csv”:

- Id : ID dari tempat penginapan
- Listing_url : URL dari tempat penginapan
- Scrape_id : ID dari data yang di scrape dari AirBNB
- Last_scraped : Tanggal terakhir kali proses scrape dilakukan
- Name : Nama dari tempat penginapan
- Description : Deskripsi dari tempat penginapan
- Neighborhood_overview : Gambaran sekilas dalam bentuk deskripsi mengenai lingkungan sekitar daerah penginapan
- Picture_url : URL dari foto tempat penginapan
- Host_id : ID dari pemilik tempat penginapan
- Host_url : URL dari profile pemilik tempat penginapan
- Host_name : Nama dari profile pemilik tempat penginapan
- Host_since : Tanggal dari saat pemilik tempat penginapan mendaftar di AirBNB
- Host_location : Lokasi dari pemilik tempat penginapan
- Host_about : Deskripsi dari pemilik tempat penginapan

- Host_response_time : Waktu yang dibutuhkan oleh pemilik tempat penginapan untuk membalas calon customer
- Host_response_rate : Persentase pembalasan chat oleh pemilik tempat penginapan kepada calon customer
- Host_acceptance_rate : Persentase penerimaan customer untuk menyewa tempat penginapan
- Host_is_superhost : Info bahwa pemilik dari tempat penginapan ini adalah superhost atau tidak (True/False)
- Host_thumbnail_url : URL untuk foto profile dari pemilik tempat penginapan dalam ukuran thumbnail
- Host_picture_url : URL untuk foto profile dari pemilik tempat penginapan
- Host_neighbourhood : Lingkungan dari pemilik tempat penginapan
- Host_listings_count : Jumlah dari banyaknya penyewaan tempat penginapan yang dimiliki oleh pemilik tempat penginapan
- Host_total_listings_count : Jumlah total dari banyaknya penyewaan tempat penginapan yang dimiliki oleh pemilik tempat penginapan
- Host_verification : Daftar informasi mengenai pemilik tempat penginapan yang dapat memverifikasi dirinya
- Host_has_profile : Info mengenai pemilik tempat penginapan ini memiliki profile atau tidak (True/False)
- Host_identity_verification : Info bahwa pemilik dari tempat penginapan ini sudah terverifikasi atau tidak (True/False)
- Neighbourhood : Linkungan
- Neighbourhood_cleansed : Lingkungan yang lebih spesifik
- Neighbourhood_group_cleansed : Grup dari lingkungan yang lebih spesifik
- Latitude : Garis lintang dari lokasi tempat penginapan
- Longitude : Garis bujur dari lokasi tempat penginapan
- Property_type : Tipe properti dari tempat penginapan
- Room_type : Tipe kamar dari tempat penginapan
- Accommodates : Jumlah tamu yang dapat diterima oleh tempat penginapan
- Bathrooms : Jumlah kamar mandi yang ada di tempat penginapan
- Bathrooms_text : Deskripsi mengenai kamar mandi yang ada di tempat penginapan
- Bedrooms : Jumlah kamar tidur yang ada di tempat penginapan
- Beds : Jumlah kasur yang ada di tempat penginapan
- Amenities : Daftar fasilitas yang disediakan oleh tempat penginapan
- Price : Harga dari penyewaan tempat penginapan
- Minimum_nights : Jumlah minimum malam yang harus dipesan saat penyewaan
- Maximum_nights : Jumlah maximum malam yang harus dipesan saat penyewaan
- Has_availability : Available atau tidaknya tempat penginapan
- Availability_30 : Available atau tidaknya tempat penginapan dalam waktu 30 hari
- Availability_60 : Available atau tidaknya tempat penginapan dalam waktu 60 hari
- Availability_90 : Available atau tidaknya tempat penginapan dalam waktu 90 hari
- Availability_365 : Available atau tidaknya tempat penginapan dalam waktu 365 hari

- Number_of_reviews : Jumlah review dari customer yang pernah menginap
- Number_of_reviews_ltm : Jumlah review dari customer dalam waktu 12 bulan yang lalu
- Number_of_reviews_l30d : Jumlah review dari customer dalam waktu 30 hari yang lalu
- First_review : Tanggal pertama adanya review
- Last_review : Tanggal terakhir adanya review
- Review_scores_rating : Rating penilaian review secara keseluruhan
- Review_scores_accuracy : Rating penilaian review dari segi akurasi
- Review_scores_cleanliness : Rating penilaian review dari segi kebersihan
- Review_scores_checkin : Rating penilaian review dari segi check-in
- Review_scores_communication : Rating penilaian review dari segi komunikasi
- Review_scores_location : Rating penilaian review dari segi lokasi
- Review_scores_value : Rating penilaian review dari segi value
- License : Lisensi
- Instant_bookable : Info mengenai proses booking bisa instan atau tidak (True/False)
- Calculated_host_listings_count : Jumlah total transaksi penyewaan dari pemilik penginapan
- Calculated_host_listings_count_entire_homes : Jumlah total transaksi penyewaan dari pemilik penginapan tipe rumah
- Calculated_host_listings_count_private_rooms : Jumlah total transaksi penyewaan dari pemilik penginapan tipe kamar pribadi
- Calculated_host_listings_count_shared_rooms : Jumlah total transaksi penyewaan dari pemilik penginapan tipe kamar berbagi
- Reviews_per_month : Rata-rata review yang di dapat per bulan

Dalam tabel ini, secara keseluruhan terdapat 4387 data tempat penginapan dengan 74 atribut. Dari sekian banyak atribut yang ada dapat ditemukan beberapa kolom yang kosong(Null) sepenuhnya. Contohnya seperti yang ditunjukan dalam gambar berikut

host_id	4387.0	1.162706e+08	1.095252e+08	2.277900e+05	2.542230e+07	1.164704e+07	1.904149e+08	3.619516e+08
host_listings_count	4376.0	4.710032e+01	7.247388e+01	0.000000e+00	2.000000e+00	1.300000e+01	5.300000e+01	2.750000e+02
host_total_listings_count	4376.0	4.710032e+01	7.247388e+01	0.000000e+00	2.000000e+00	1.300000e+01	5.300000e+01	2.750000e+02
latitude	4387.0	1.313209e+00	3.173796e-02	1.245300e+00	1.293215e+00	1.310420e+00	1.322170e+00	1.453280e+00
longitude	4387.0	1.038474e+02	4.093549e-02	1.036469e+02	1.038373e+02	1.038495e+02	1.038622e+02	1.039686e+02
accommodates	4387.0	2.956006e+00	2.395966e+00	1.000000e+00	2.000000e+00	2.000000e+00	4.000000e+00	1.600000e+01
bathrooms	0.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN
bedrooms	3994.0	1.337757e+00	7.188504e-01	1.000000e+00	1.000000e+00	1.000000e+00	1.000000e+00	8.000000e+00
beds	4328.0	1.877311e+00	2.563267e+00	0.000000e+00	1.000000e+00	1.000000e+00	2.000000e+00	5.800000e+01
minimum_nights	4387.0	2.987987e+01	5.613981e+01	1.000000e+00	2.000000e+00	7.000000e+00	3.000000e+01	1.100000e+03

Selain dari beberapa kolom yang seluruh valuenya memiliki nilai Null. Ada beberapa kolom yang sebagian valuenya memiliki nilai Null dengan contoh seperti yang ditunjukan dalam gambar berikut

```
pd.set_option('max_rows', 99999)
data.isnull().sum()
```

```
id                                0
listing_url                       0
scrape_id                         0
last_scraped                      0
name                              0
description                     141
neighborhood_overview          1518
picture_url                       0
host_id                           0
host_url                          0
host_name                        11
host_since                       11
host_location                    12
host_about                     1273
host_response_time              581
host_response_rate              581
host_acceptance_rate            719
host_is_superhost                11
host_thumbnail_url               11
host_picture_url                 11
host_neighbourhood              424
host_listings_count              11
host_total_listings_count        11
host_verifications                0
host_has_profile_pic             11
host_identity_verified           11
neighbourhood                  1518
```

Jika ditotal, dari 74 atribut yang ada terdapat 34 atribut yang masih memiliki Null value di dalamnya. Selanjutnya akan dijelaskan mengenai tabel yang ada dalam file "neighbourhoods.csv". Atribut-atribut yang ada dalam tabel dari file tersebut adalah sebagai berikut:

- Neighbourhood_group : Grup dari lingkungan
- Neighbourhood : Lingkungan

Berikut merupakan contoh dari isi tabel

```
# load the data set ans show the first five transaction
url2 = 'https://raw.githubusercontent.com/sirjvp/housesales/main/neighbourhoods.csv'
data2 = pd.read_csv(url2)
data2.head()
```

	neighbourhood_group	neighbourhood
0	Central Region	Bishan
1	Central Region	Bukit Merah
2	Central Region	Bukit Timah
3	Central Region	Downtown Core
4	Central Region	Geylang

Dalam tabel ini, tidak ditemukan adanya *Null value* di atribut manapun

```
data2.isna().any()
```

```
neighbourhood_group     False
neighbourhood           False
dtype: bool
```

DATA PREPARATION

Dalam pembuatan model yang baik, data-data yang ada perlu dipilah dan dibersihkan terlebih dahulu. Oleh karena itu, langkah pertama yang dilakukan adalah memilih atribut mana saja yang akan dipakai. Atribut yang dipilih hanyalah atribut yang dirasa oleh penulis akan mempengaruhi atribut target.

```
dataset = data[['host_since', 'host_response_time', 'host_response_rate', 'host_acceptance_rate', 'host_listings_count',
                'host_identity_verified', 'neighbourhood_cleansed', 'property_type', 'accommodates', 'bathrooms_text', 'bedrooms',
                'beds', 'number_of_reviews', 'review_scores_rating','review_scores_accuracy','review_scores_cleanliness', 'review_scores_checkin'
                'review_scores_location','review_scores_value','instant_bookable', 'amenities', 'price']]
dataset.head()
```

	host_since	host_response_time	host_response_rate	host_acceptance_rate	host_listings_count	host_identity_verified	neighbourhood_cleansed
0	2010-10-20	within a day	100%	NaN	2.0	t	Woodlands
1	2010-09-08	a few days or more	0%	NaN	1.0	t	Bukit Timah
2	2010-10-20	within a day	100%	NaN	2.0	t	Woodlands
3	2011-01-29	within a few hours	100%	100%	8.0	t	Tampines
4	2011-01-29	within a few hours	100%	100%	8.0	t	Tampines

property_type	accommodates	bathrooms_text	bedrooms	beds	number_of_reviews	review_scores_rating	review_scores_accuracy
Private room in apartment	1	1 bath	1.0	1.0	1	94.0	10.0
Private room in apartment	2	1 bath	1.0	1.0	18	91.0	9.0
Private room in apartment	1	1 bath	1.0	1.0	20	98.0	10.0
Private room in villa	6	1 private bath	2.0	3.0	20	89.0	9.0
Private room in house	3	Shared half-bath	1.0	1.0	24	83.0	8.0

review_scores_cleanliness	review_scores_checkin	review_scores_communication	review_scores_location	review_scores_value	instant_bookable
10.0	10.0	10.0	8.0	8.0	f
10.0	10.0	10.0	9.0	9.0	f
10.0	10.0	10.0	8.0	9.0	f
8.0	9.0	10.0	9.0	9.0	f
8.0	9.0	9.0	8.0	8.0	f

amenities	price
["Elevator", "Cable TV", "Washer", "Wifi", "TV...	$79.00
["Elevator", "Essentials", "Shampoo", "TV", "W...	$80.00
["Dryer", "Elevator", "Wifi", "TV", "Cable TV"...	$66.00
["Dryer", "Dedicated workspace", "Shampoo", "W...	$174.00
["Dryer", "Essentials", "Shampoo", "Wifi", "TV...	$93.00

MISSING VALUES

Setelah memilih kolom yang kemungkinan akan berpengaruh terhadap nilai variabel target, maka yang dilakukan selanjutnya adalah pengecekan mengenai kolom-kolom yang memiliki value Null

```
dataset.isna().sum()
```

```
host_since                      11
host_response_time             581
host_response_rate             581
host_acceptance_rate           719
host_listings_count             11
host_identity_verified          11
neighbourhood_cleansed           0
property_type                    0
accommodates                     0
bathrooms_text                  15
bedrooms                       393
beds                            59
number_of_reviews                0
review_scores_rating          1905
review_scores_accuracy        1910
review_scores_cleanliness     1909
review_scores_checkin         1911
review_scores_communication   1910
review_scores_location        1912
review_scores_value           1912
instant_bookable                 0
amenities                        0
price                            0
dtype: int64
```

Dikarenakan banyaknya jumlah instance yang memiliki value Null dalam tabel yang digunakan, maka penulis memutuskan untuk tidak langsung menghapus semua instance yang memiliki value Null tetapi mengganti instance tersebut ke nilai lainnya. Untuk melakukan itu, Data Transformation perlu dilakukan terlebih dahulu.

DATA TRANSFORMATION

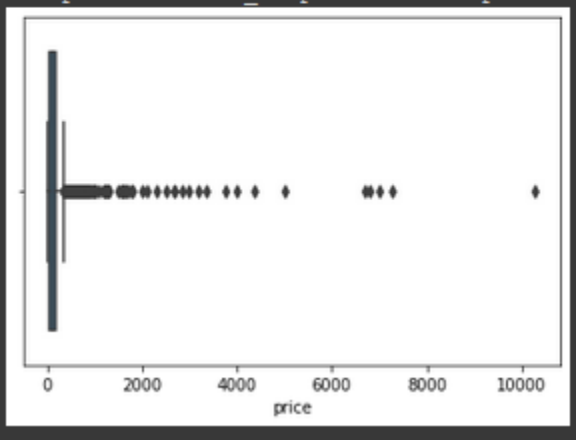
Hal pertama yang dilakukan adalah menghilangkan *outlier* terlebih dahulu. Oleh karena itu penulis mulai melakukan transformasi data dari variabel target yaitu kolom “price”.

```
# Menghilangkan simbol $ pada kolom price dan mengubah tipe datanya dari string menjadi float
dataset['price'] = dataset['price'].astype(str).str.slice(1).str.replace(',','').astype(float)
```

Setelah itu kolom price ditampilkan menggunakan boxplot dan semua outlier dihilangkan

```
sns.boxplot(dataset['price'], fliersize=5)
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py
  FutureWarning
<matplotlib.axes._subplots.AxesSubplot at 0x7f67511948d0>
```

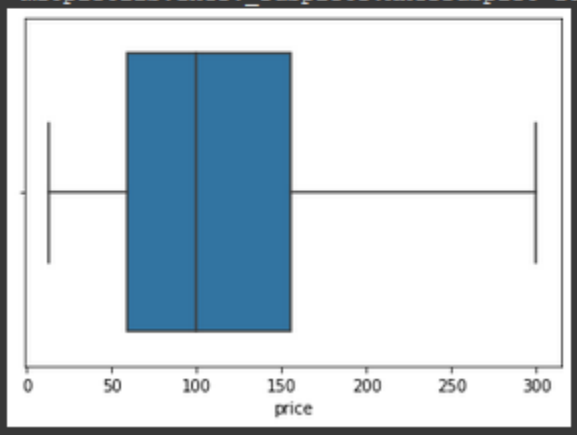



```
[7] dataset['price'].describe()

count     4387.000000
mean       170.926601
std        349.092510
min         13.000000
25%         61.000000
50%        114.000000
75%        179.000000
max      10286.000000
Name: price, dtype: float64
```

```
#remove outlier
dataset = dataset[dataset['price'] <= 300]
```

```
[ ] #boxplot sesudah outlier diremove
    sns.boxplot(dataset['price'])
```

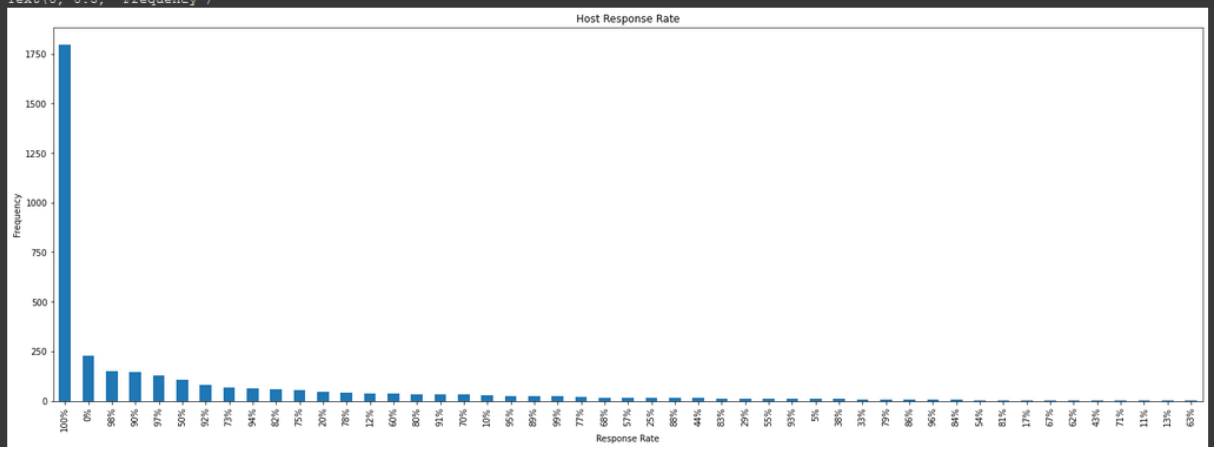
```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorat
  FutureWarning
<matplotlib.axes._subplots.AxesSubplot at 0x7f5a0469b3
```



Selanjutnya, atribut-atribut numerik yang memiliki nilai Null diperiksa satu per satu dan diganti dengan value yang sesuai.

```
[ ] plot = dataset['host_response_rate'].value_counts().plot(kind='bar', figsize=(24,8), title='Host Response Rate')
    plot.set_xlabel('Response Rate')
    plot.set_ylabel('Frequency')
```

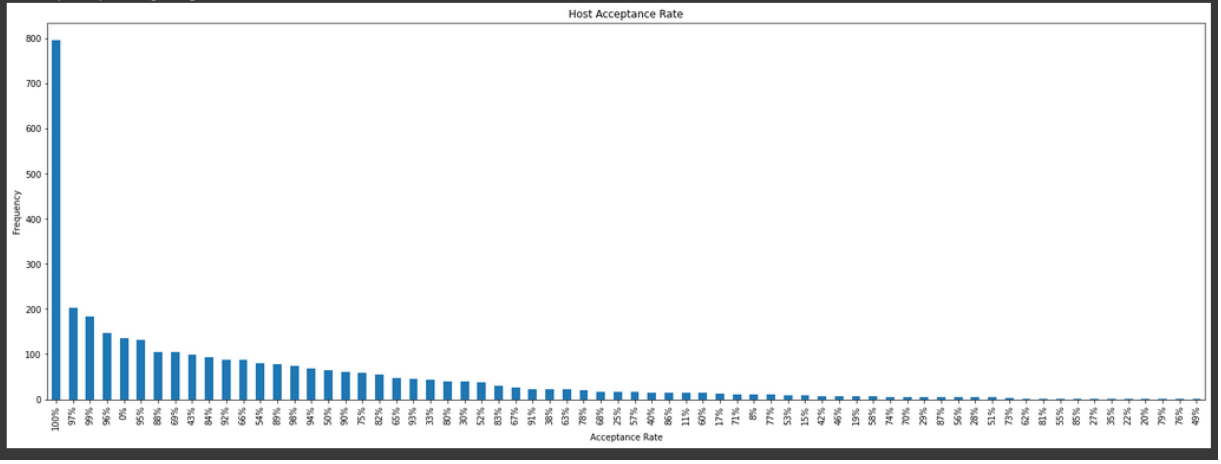
```
Text(0, 0.5, 'Frequency')
```




```
# data NA diganti dengan 100 karena modus dari atribut ini sangat jauh dibandingkan dengan frekuensi angka-angka lain
dataset['host_response_rate'] = dataset['host_response_rate'].astype(str).str[:-1].str.replace('na', '100').astype(float)
```

```
[ ] plot = dataset['host_acceptance_rate'].value_counts().plot(kind='bar', figsize=(24,8), title='Host Acceptance Rate')
    plot.set_xlabel('Acceptance Rate')
    plot.set_ylabel('Frequency')
```

```
Text(0, 0.5, 'Frequency')
```




```
# data NA diganti 100 karena modus dari atribut ini sangat jauh dibandingkan dengan frekuensi angka-angka lain
dataset['host_acceptance_rate'] = dataset['host_acceptance_rate'].astype(str).str[:-1].str.replace('na', '100').astype(float)
```

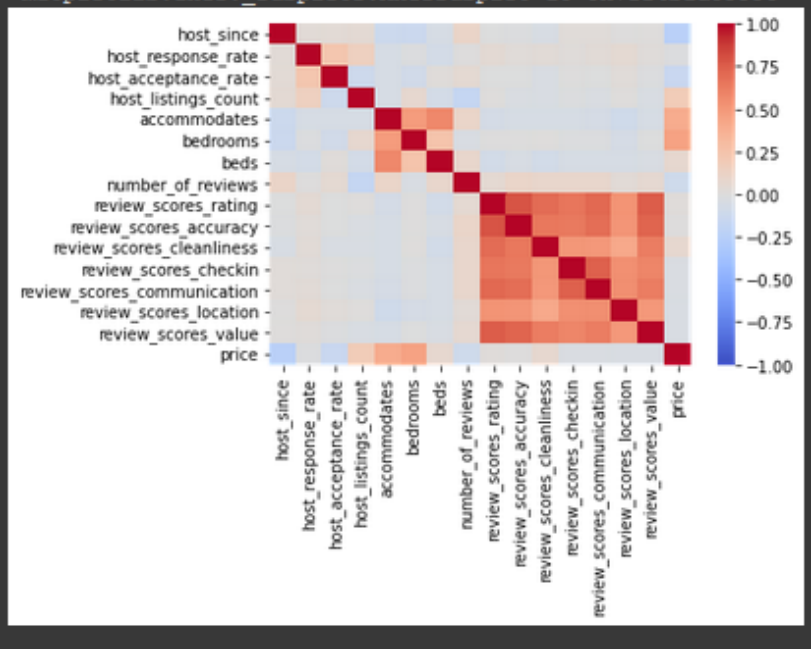
```
#mengganti host since menjadi sudah berapa tahun menjadi host
dataset['host_since'] = 2021 - dataset['host_since'].astype(str).str[:4].astype(float)
```

```
# NA handling untuk continous variabel
dataset.host_since.fillna(value= dataset.host_since.mean(), inplace= True)
dataset.host_listings_count.fillna(value= dataset.host_listings_count.mean(), inplace= True)
dataset.bedrooms.fillna(value= dataset.bedrooms.mean(), inplace= True)
dataset.review_scores_cleanliness.fillna(value= dataset.review_scores_cleanliness.mean(), inplace= True)
dataset.beds.fillna(value= dataset.beds.mean(), inplace= True)
dataset.review_scores_value.fillna(value= dataset.review_scores_value.mean(), inplace= True)
dataset.review_scores_checkin.fillna(value= dataset.review_scores_checkin.mean(), inplace= True)
dataset.review_scores_rating.fillna(value= dataset.review_scores_rating.mean(), inplace= True)
dataset.host_response_rate.fillna(value= dataset.host_response_rate.mean(), inplace= True)
dataset.review_scores_accuracy.fillna(value= dataset.review_scores_accuracy.mean(), inplace= True)
dataset.review_scores_location.fillna(value= dataset.review_scores_location.mean(), inplace= True)
dataset.review_scores_communication.fillna(value= dataset.review_scores_communication.mean(), inplace= True)
```

Selanjutnya, masing-masing atribut dilihat korelasinya dengan variabel target

```
[48] corr = dataset.corr()
     sns.heatmap(corr, vmin=-1, vmax=1, cmap='coolwarm')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7ff4bad09990>
```




```
[49] corr['price']
```

```
host_since                    -0.237725
host_response_rate            -0.013782
host_acceptance_rate          -0.141571
host_listings_count            0.193776
accommodates                   0.401382
bedrooms                       0.458789
beds                           0.073862
number_of_reviews             -0.103771
review_scores_rating           0.020369
review_scores_accuracy         0.004404
review_scores_cleanliness      0.075024
review_scores_checkin         -0.030530
review_scores_communication   -0.028655
review_scores_location        -0.033870
review_scores_value           -0.035848
price                          1.000000
Name: price, dtype: float64
```

Karena korelasi yang didapatkan sangat kecil pada seluruh atributnya, maka penulis memutuskan hanya mengambil atribut yang memiliki nilai korelasi di atas 0.1 atau kurang dari -0.1.

```
[50] # drop data dengan correlation kecil (0.1 s.d. -0.1)
     drop_1 = set(corr.index[corr['price'] <= 0.1])
     drop_2 = set(corr.index[corr['price'] >= -0.1])
     drop = list(drop_1.intersection(drop_2))
     dataset.drop(drop, axis=1, inplace=True)
     drop
```

```
['beds',
 'review_scores_value',
 'review_scores_checkin',
 'review_scores_rating',
 'host_response_rate',
 'review_scores_accuracy',
 'review_scores_location',
 'review_scores_communication',
 'review_scores_cleanliness']
```

Setelah atribut-atribut bernilai numerik sudah selesai diatasi, selanjutnya yang akan diatasi adalah atribut-atribut yang sifatnya *categorical*. Pertama, penulis memutuskan untuk menghapus semua instance yang masih memiliki nilai Null.

```
[66] dataset.dropna(axis = 0, how ='any', inplace=True)
```

Selanjutnya, atribut-atribut yang sifatnya categorical, diolah dengan menggunakan Label Encoding.

```
[ ] label_encoder = preprocessing.LabelEncoder()
    dataset['neighbourhood_cleansed'] = label_encoder.fit_transform(dataset['neighbourhood_cleansed'])
    dataset['property_type'] = label_encoder.fit_transform(dataset['property_type'])
    dataset['bathrooms_text'] = label_encoder.fit_transform(dataset['bathrooms_text'].astype(str).str.replace('<','less than'))

    response_dict = {'within a day':0, 'a few days or more':1, 'within a few hours':2, 'within an hour':3}
    identity_dict = {'f':0, 't':1}
    instant_dict = {'f':0, 't':1}
    dataset.host_response_time = dataset.host_response_time.replace(response_dict)
    dataset.host_identity_verified = dataset.host_identity_verified.replace(identity_dict)
    dataset.instant_bookable = dataset.instant_bookable.replace(instant_dict)
    dataset.head()
```

	host_since	host_response_time	host_acceptance_rate	host_listings_count	host_identity_verified	neighbourhood_cleansed	property_type	ac
0	11.0	0.0	100.0	2.0	1.0	39	13	
1	11.0	1.0	100.0	1.0	1.0	6	13	
2	11.0	0.0	100.0	2.0	1.0	39	13	
3	10.0	2.0	100.0	8.0	1.0	35	27	
4	10.0	2.0	100.0	8.0	1.0	35	22	

Khusus untuk kolom “amenities” karena value di dalamnya berupa list, maka dilakukan One-Hot Encoding untuk bisa mendapatkan semua isinya.

One Hot encoding

```
[52] # Ubah tipe data dari kolom 'amenities' dari string menjadi list
     def remove_petik(isi):
       jadi = ''
       for letter in isi:
         if letter != '"':
           jadi += letter
       return jadi
     test = dataset['amenities'].copy()
     for i in range (len(test.index)):
       data = test.iloc[i]
       data = data[1:-1]
       items = data.split(',')
       for item in items:
         item = remove_petik(item)
       test[i] = items
```

```
[ ]  # Membuat DataFrame baru berisi hasil One-Hot Encoding
     mlb = MultiLabelBinarizer()

     res = pd.DataFrame(mlb.fit_transform(test),
                        columns=mlb.classes_,
                        index=test.index)
     res
```

			"32\" HDTV"	"Air conditioning"	"BBQ grill"	"Baby bath"	"Baby monitor"	"Baby safety gates"	"Babysitter recommendations"	"Baking sheet"	"Barbecue utensils"	"Bathtub"	"Beach essentials"	"Beachfront"	"Bed linens"
0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
3429	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
3430	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
3435	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
3439	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
3461	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0

4229 rows × 235 columns

```
# Menghilangkan kolom-kolom yang tidak perlu dari dataframe
drop=[]
for col in res:
  if len(col) == 1:
    drop.append(col)
res.drop(drop, axis=1, inplace=True)
```

Hasil dari One-Hot Encoding kemudian digabungkan ke dataset awal.

```
new = pd.concat([dataset, res], axis=1)
new.drop('amenities', axis=1, inplace=True)
new.head()
```

	host_since	host_response_time	host_acceptance_rate	host_listings_count	host_identity_verified	neighbourhood_cleansed	property_type	accom
0	11.0	0.0	100.0	2.0	1.0	39.0	13.0	
1	11.0	1.0	100.0	1.0	1.0	6.0	13.0	
2	11.0	0.0	100.0	2.0	1.0	39.0	13.0	
3	10.0	2.0	100.0	8.0	1.0	35.0	27.0	
4	10.0	2.0	100.0	8.0	1.0	35.0	22.0	

5 rows × 190 columns

Ternyata setelah digabungkan, beberapa kolom didapati memiliki nilai Null. Maka dari itu untuk terakhir kalinya, instance-instance yang masih memiliki nilai Null akan dihilangkan.

```
new.isna().sum()
```

```
host_since                          774
host_response_time                  774
host_acceptance_rate                774
host_listings_count                 774
host_identity_verified              774
                                   ...
"Smart lock"                          0
"TV"                                  0
"Washer \u2013\u00a0In building"      0
"Washer"                              0
"Wifi"                                0
Length: 189, dtype: int64
```

```
new.dropna(axis = 0, how ='any', inplace=True)
```

FEATURE ENGINEERING

Sebelum masuk ke modeling, variabel target perlu dipisahkan terlebih dahulu dari atribut-atribut lainnya.

```
[ ]  # split data as X and y
     x, y = new.loc[:,new.columns != 'price'], new.loc[:, 'price']
```

Selanjutnya, supaya model yang dihasilkan bisa lebih akurat, maka dilakukan scaling terhadap semua variabel independen. Untuk itu tim penulis memilih metode standard scaling terhadap seluruh variabel tersebut.

```
[ ]  # standardize feature set
     scaler = StandardScaler()
     standard = pd.DataFrame(scaler.fit_transform(x))
     standard.head()
```

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
0	2.603454	-1.821622	0.736244	-0.679637	0.3951	2.296451	-0.073356	-0.809715	-0.722359	-0.451283	-0.381891	0.995387	-0.705119	-0.051024	-0.016988	0.554909	-0.293564
1	2.603454	-0.927957	0.736244	-0.693586	0.3951	-1.369473	-0.073356	-0.373114	-0.722359	-0.451283	0.120280	0.995387	-0.705119	-0.051024	-0.016988	0.554909	-0.293564
2	2.603454	-1.821622	0.736244	-0.679637	0.3951	2.296451	-0.073356	-0.809715	-0.722359	-0.451283	0.179359	0.995387	-0.705119	-0.051024	-0.016988	0.554909	-0.293564
3	2.170627	-0.034292	0.736244	-0.595942	0.3951	1.852097	1.236736	1.373291	-0.601437	1.134321	0.179359	-1.150861	-0.705119	-0.051024	-0.016988	0.554909	-0.293564
4	2.170627	-0.034292	0.736244	-0.595942	0.3951	1.852097	0.768846	0.063487	3.751768	-0.451283	0.297517	-1.150861	-0.705119	-0.051024	-0.016988	0.554909	-0.293564

5 rows × 189 columns

Terakhir dataset akan dipisahkan menjadi training data dan testing data.

```
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.25, random_state=32)
```

MODELLING

Dalam memilih metode regresi yang digunakan, tim penulis memilih 2 metode yaitu Artificial Neural Network dan juga Support Vector Machine.

1. Artificial Neural Network

   Artificial Neural Network adalah sekumpulan algoritma yang dibuat untuk mencari pola atau informasi dari dataset dengan cara yang mirip dengan cara otak manusia bekerja. Neural network sendiri merupakan sistem dari neuron baik organik maupun buatan. Neural network tersusun dari lapisan node yang saling berkaitan. Masing-masing node adalah perceptron dan cara kerjanya mirip dengan Multiple Linear Regression. Perceptron ini memasukkan input ke dalam activation function. Dalam Multi-Layered Perceptron, akan ada input layer yang mengumpulkan data yang diinputkan. Setelah melalui lapisan node yang disebut hidden layer, nilai yang dihasilkan akan tercermin dari output dari output layer yang merupakan layer terakhir.

   Dalam membuat model Artificial Neural Network, penulis menggunakan class dari library sklearn yaitu MLPRegressor dengan jumlah 2 hidden layer dengan layer pertama terdiri dari 256 node dan layer kedua terdiri dari 128 node. Solver yang digunakan adalah ‘adam’, activation function yang digunakan adalah rectified linear unit function (relu), batch size 128, dan memiliki learning rate 0.001 dan nilainya tidak berubah sampai akhir.

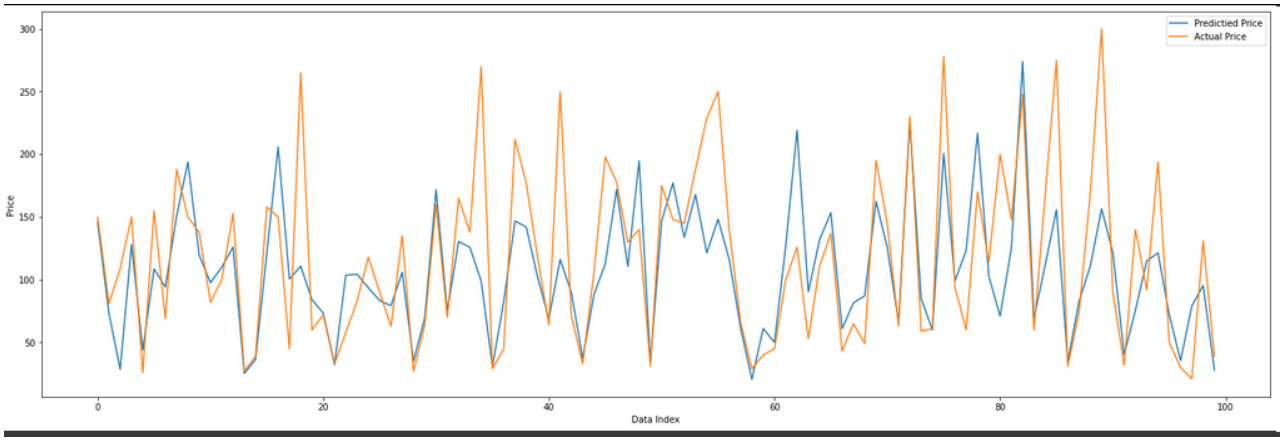
```
[48] model = MLPRegressor(hidden_layer_sizes=(256,128) ,activation='relu', solver='adam',
                          batch_size=128, max_iter=50, learning_rate='constant', learning_rate_init=0.001)
     model.fit(x_train, y_train)

     /usr/local/lib/python3.7/dist-packages/sklearn/neural_network/_multilayer_perceptron.py:571: ConvergenceWar
       % self.max_iter, ConvergenceWarning)
     MLPRegressor(activation='relu', alpha=0.0001, batch_size=128, beta_1=0.9,
                  beta_2=0.999, early_stopping=False, epsilon=1e-08,
                  hidden_layer_sizes=(256, 128), learning_rate='constant',
                  learning_rate_init=0.001, max_fun=15000, max_iter=50, momentum=0.9,
                  n_iter_no_change=10, nesterovs_momentum=True, power_t=0.5,
                  random_state=None, shuffle=True, solver='adam', tol=0.0001,
                  validation_fraction=0.1, verbose=False, warm_start=False)
```

Hasil dari prediksi harga dari model yang dibuat dengan menggunakan Artificial Neural Network ditampilkan dan dibandingkan dengan harga sebenarnya. Dari keseluruhan data yang dipakai untuk testing, diambil 100 sampel data untuk ditampilkan menggunakan grafik.

```
[67] y_pred = model.predict(x_test)
     compare = pd.DataFrame({'prediction': y_pred, 'actual': y_test})
     compare.head()
```

	prediction	actual
2778	72.683547	135.0
2056	77.379153	98.0
4036	164.630145	147.0
1933	55.539522	48.0
1043	59.831487	65.0

```
compare_plt = compare.sample(n=100)
compare_plt.reset_index(drop=True, inplace=True)
plt.figure(figsize=[25,8])
plt.plot(compare_plt.index,compare_plt['prediction'], label='Predictied Price')
plt.plot(compare_plt.index, compare_plt['actual'], label='Actual Price')
plt.xlabel('Data Index')
plt.ylabel('Price')
plt.legend()
plt.show()
```



Dari grafik diatas bila dilihat secara visual dapat disimpulkan bahwa harga hasil prediksi dari model cukup akurat dibandingkan dengan harga sebenarnya. Selain menggunakan line chart, model juga dievaluasi dari nilai Mean Absolute Percentage Error (MAPE) nya. Nilai MAPE yang didapatkan dari model ini adalah 32.08.

```
[74] def MAPE(Y_actual,Y_Predicted):
         mape = np.mean(np.abs((Y_actual - Y_Predicted)/Y_actual))*100
         return mape
     MAPE = MAPE(y_test, y_pred)
     print('MAPE = '+ str(MAPE))

     MAPE = 32.08424516821371
```

2. Support Vector Machine

Support Vector Machine adalah seperangkat metode supervised learning. Metode ini menggunakan teknik yang disebut kernel trick untuk transform data yang nanti dapat ditemukan optical boundary antara possible output. Metode ini menggunakan nonlinear mapping untuk mengubah original data training menjadi dimensi yang lebih tinggi. Kemudian menemukan cara untuk memisahkan data berdasarkan label atau output yang telah kita tentukan. Keuntungan dari metode ini mampu menangani kelangkaan data dan dapat mempelajari decision boundaries yang kompleks dalam feature space dimensi yang lebih tinggi secara efisien

Dalam pembuatan model Support Vector Machine, tim penulis menggunakan class dari library sklearn yaitu SVR. Model ini dibuat dengan menggunakan kernel ‘linear’.

```
[90] from sklearn import svm
     from sklearn.svm import SVR
     # Buat model menggunakan support vector machine
     svmRBF = svm.SVR(kernel='linear')

     # Lakukan training dengan dataset train
     svmRBF.fit(x_train, y_train)
```

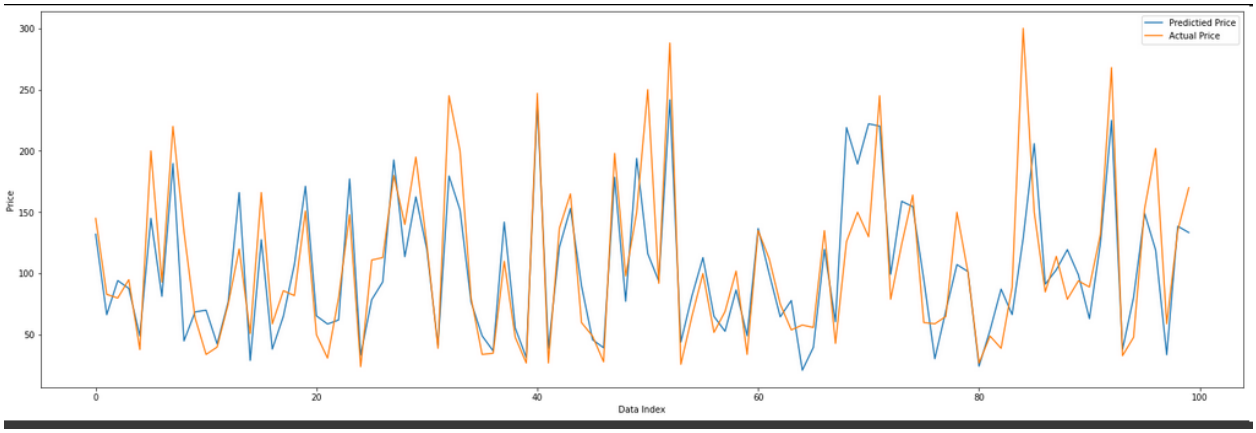
```
SVR(C=1.0, cache_size=200, coef0=0.0, degree=3, epsilon=0.1, gamma='scale',
    kernel='linear', max_iter=-1, shrinking=True, tol=0.001, verbose=False)
```

Hasil dari prediksi harga dari model yang dibuat dengan menggunakan Support Vector Machine ditampilkan dan dibandingkan dengan harga sebenarnya. Dari keseluruhan data yang dipakai untuk testing, diambil 100 sampel data untuk ditampilkan menggunakan grafik.

```
[92] y_predsvm = svmRBF.predict(x_test)
     comparesvm = pd.DataFrame({'prediction': y_predsvm, 'actual': y_test})
     comparesvm.head()
```

	prediction	actual
2778	88.618911	135.0
2056	78.146579	98.0
4036	158.079155	147.0
1933	82.758411	48.0
1043	50.686332	65.0

```
comparesvm_plt = compare.sample(n=100)
comparesvm_plt.reset_index(drop=True, inplace=True)
plt.figure(figsize=[25,8])
plt.plot(comparesvm_plt.index,comparesvm_plt['prediction'], label='Predictied Price')
plt.plot(comparesvm_plt.index, comparesvm_plt['actual'], label='Actual Price')
plt.xlabel('Data Index')
plt.ylabel('Price')
plt.legend()
plt.show()
```



Dari grafik diatas bila dilihat secara visual dapat disimpulkan bahwa harga hasil prediksi dari model cukup akurat bahkan lebih sesuai bila dibandingkan dengan menggunakan Artificial Neural Network dalam memprediksi harga. Namun, jika dilihat dari nilai MAPE nya, nilai MAPE model ini lebih tinggi dibandingkan model yang menggunakan Neural Network yaitu 46.44

```
[93] def MAPE(Y_actual,Y_Predicted):
         mape = np.mean(np.abs((Y_actual - Y_Predicted)/Y_actual))*100
         return mape
     MAPEsvm = MAPE(y_test, y_predsvm)
     print('MAPE = '+ str(MAPEsvm))

     MAPE = 46.4384037656349
```

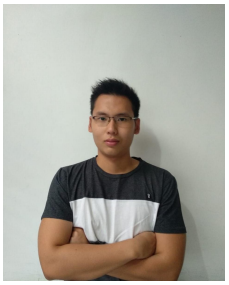
EVALUATION

Dari model machine learning yang kami buat, meskipun belum bisa memprediksi dengan tepat 100%, tapi model tersebut bisa memperkirakan harga properti dari host yang ada di data testing dengan cukup baik. Mungkin dengan model ini, Airbnb bisa menambahkan fitur baru yang memudahkan host-host pemula yaitu dengan memberikan rekomendasi harga. Hal tersebut paling tidak bisa memberikan gambaran dan mempermudah host terutama host-host yang baru pertama kali mencoba masuk di bisnis serupa untuk memasang harga untuk propertinya. Karena sifatnya hanya berupa rekomendasi, maka keputusan akhirnya tetap ada di tangan masing-masing host. Sehingga jika harganya dirasa terlalu tinggi atau terlalu rendah, cukup disesuaikan lagi dengan keinginan host.

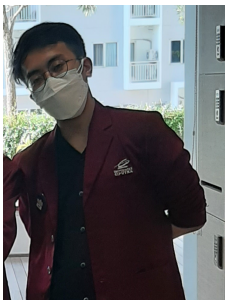
Dalam project yang kelompok kami pada kali ini telah selesaikan terdapat beberapa kendala yang kami alami dalam prosesnya, seperti saat mempersiapkan data terdapat beberapa yang memiliki nilai NaN sehingga perlu kami proses. Selain itu, nilai-nilai outlier yang ada sangat jauh dari nilai-nilai yang lain. Untuk nilai outlier tersebut, seluruh instance yang memilikinya kami hapus. Ada juga data-data yang masih tidak sesuai format sehingga tidak bisa kami gunakan dalam kalkulasi. Untuk data yang seperti itu,

perlu kami lakukan konversi dengan dari satu jenis data ke jenis data lainnya ataupun dengan cara penghapusan karakter yang tidak dapat diproses sehingga hanya menyisakan data yang berupa angka.

TEAM PROFILES



Terrence Pramono adalah mahasiswa Information System for Business (ISB) Universitas Ciputra, Surabaya. Lahir di Salatiga pada tanggal 11 November 2001, Terrence menempuh masa kecilnya di Semarang sebelum melanjutkan kuliah di Surabaya. Tujuannya belajar Machine Learning adalah karena bisnis mulai banyak yang membutuhkan analisis data sehingga belajar Machine Learning bukanlah sesuatu yang merugikan. Malah mungkin bisa menguntungkan di masa yang akan datang.



Marshall Ovierdo Kurniawan adalah mahasiswa Informatics (IMT) Universitas Ciputra, Surabaya. Lahir di Tangerang pada tanggal 03 Oktober 2000. Marshall melewati masa kecilnya di Denpasar, Bali kemudian meneruskan masa remajanya di Surabaya hingga di masa perkuliahan.Tujuannya dalam mendalami Machine Learning adalah untuk menambah wawasannya dalam dunia Artificial Intelligence yang dimana jika dipelajari lebih mendalam, terdapat banyak aspek yang masih memiliki potensi untuk digunakan seperti analisa data dan bahkan hingga proses automasi pada berbagai sektor pekerjaan.

Jonathan Valentino adalah mahasiswa Informatics (IMT) Universitas Ciputra, Surabaya. Lahir di Surabaya pada tanggal 03 Maret 2001. Jonathan melalui masa kecilnya di Surabaya hingga kuliah. Tujuan Belajar Machine learning karena Machine Learning di masa sekarang sangat dibutuhkan dan dapat digunakan di berbagai aspek suatu bisnis atau usaha yang dibutuhkan untuk memproses dan mempelajari data yang dapat digunakan kelanjutannya di Artificial Intelligence sehingga dapat menciptakan sebuah sistem yang pintar.dalam mengambil sebuah keputusan.