Big Data Analytics terhadap Harga Jual Produk Shopee Menggunakan *Random Forest*

Kelompok 11

Anggota:

1. Dzakiyyah Al Kaazhim [Institut Teknologi Telkom Purwokerto]
2. Fayza Apriliza [Institut Teknologi Telkom Purwokerto]
3. Naufal Mufid F. [Institut Teknologi Nasional Bandung]

AI-HACKER

BISA AI ACADEMY

STUDI INDEPENDEN KAMPUS MERDEKA BATCH 3

2022

RINGKASAN

E-commerce merupakan suatu transaksi jual beli yang dilakukan secara elektronik atau *online* melalui media internet. Hal ini dilakukan karena banyaknya masyarakat yang lebih tertarik untuk berbelanja secara *online* melalui *website e-commerce* dibandingkan dengan berbelanja di toko *offline* secara langsung. Dengan pesatnya pertumbuhan e-commerce mendorong berbagai produk bergelimpangan baik dari *brand* luar maupun dalam negeri menyebabkan ketatnya persaingan untuk memperluas pasar. Persaingan tersebut mengharuskan brand-brand tersebut menawarkan harga jual yang kompetitif atau harga yang melampaui harga pokok produksi namun diterima oleh pasar. Oleh karena itu, penulis melakukan analisis dan membuat model yang dapat memprediksi harga jual suatu produk menggunakan variabel-variabel tertentu. Metode utama yang digunakan adalah Random Forest yang kemudian dibandingkan dengan aloritma lain, seperti Linear Regression, Decision Tree, K-Neigbors, Lasso, dan Ridge. Hasil yang diperoleh adalah berdasarkan *dataset sample* didapatkan algoritma terbaik adalah Random Forest dengan *training score* sebesar 97,77%, *testing score* sebesar 94,76%, dan MAPE 0,084. Sedangkan hasil yang diperoleh menggunakan *new dataset* adalah *training score* sebesar 99%, *testing score* sebesar 94,81%, dan MAPE 0,19.

DAFTAR ISI

BAB I

PENDAHULUAN

1.1. Latar Belakang

Berkembangnya teknologi informasi saat ini sangatlah berpengaruh pada kemajuan suatu usaha. Berbagai cara promosi dilakukan untuk mengenalkan usaha atau produk yang dimilikinya ke masyarakat luas. Salah satu cara promosi yang dilakukan, yaitu menggunakan situs atau *website e-commerce* untuk memasarkan produknya. *E-commerce* merupakan suatu transaksi jual beli yang dilakukan secara elektronik atau *online* melalui media internet. Hal ini dilakukan karena banyaknya masyarakat yang lebih tertarik untuk berbelanja secara *online* melalui *website e-commerce* dibandingkan dengan berbelanja di toko *offline* secara langsung. Menurut APJII lebih dari 74% konsumen di Indonesia memilih belanja secara *online* sehingga transaksi dagang di *e-commerce* mencapai Rp401 triliun di tahun 2021 (APJI, 2022). Septriana Tangkary selaku Direktur Pemberdayaan Informatika, Direktorat Jenderal Aplikasi Informatika Kementerian Kominfo, mengatakan pertumbuhan nilai *e-commerce* tertinggi di dunia, yakni mencapai 78% (Kemkominfo, 2019).

Dengan pesatnya pertumbuhan *e-commerce* mendorong berbagai produk bergelimpangan baik dari *brand* luar maupun dalam negeri menyebabkan ketatnya persaingan untuk memperluas pasar. Persaingan tersebut mengharuskan brand-brand tersebut menawarkan harga jual yang kompetitif atau harga yang melampaui harga pokok produksi namun diterima oleh pasar (Patras, 2018). Data memiliki peran penting dalam pengambilan keputusan strategi, salah satunya pengambilan keputusan mengenai harga jual produk.

Oleh karena itu, penulis melakukan analisis dan membuat model yang dapat memprediksi harga jual suatu produk menggunakan variabel-variabel tertentu. Penulisan ini bertujuan untuk mengetahui apa saja yang mempengaruhi harga jual suatu produk di *marketplace*, khususnya Shopee Indonesia dan bagaimana hasil akurasi dari model yang telah dibuat.

1.2. Rumusan Masalah

1) Apa saja yang mempengaruhi harga jual suatu produk di *marketplace*, khususnya Shopee Indonesia?
2) Apa metode yang digunakan untuk memprediksi harga jual suatu produk?
3) Bagaimana tingkat *error* dari model yang telah dibuat?

1.3. Tujuan

Berdasarkan latar belakang dan rumusan masalah maka tujuan yang ingin dicapai sebagai berikut.

1) Untuk mengetahui apa saja yang mempengaruhi harga jual suatu produk di Shopee Indonesia.
2) Untuk mengetahui metode yang digunakan untuk memprediksi harga jual suatu produk.
3) Untuk mengetahui tingkat *error* dari model yang telah dibuat.

BAB II

LANDASAN TEORI

2.1 Big Data Analytics

Dalam terminologi Layman, Big Data Analytics adalah analisis masalah yang sulit dipecahkan atau analisis data yang melibatkan data dalam jumlah besar, bahkan sampai triliunan baris. Baik data maupun big data tidak akan menghasilkan informasi apapun jika tidak dilakukan proses data atau analis data. Tujuan dilakukannya Big Data Analytics ini adalah untuk mendapatkan informasi yang berharga (big value), sehingga dapat dimanfaatkan pada proses pengambilan keputusan dan strategi bisnis yang lebih baik dalam berbagai bidang.

2.2 Harga Jual

Menurut Murti dan Soeprihanto (2007:281) untuk mendapatkan sejumlah kombinasi dari barang beserta pelayanannya dibutuhkan sejumlah uang yang dalam hal ini disebut harga. Krismiaji dan Anni (2011:326) menyatakan harga jual merupakan upaya untuk menyeimbangkan keinginan dalam memperoleh manfaat sebesar-besarnya dari perolehan pendapatan yang tinggi dan akan terjadi penurunan volume penjualan jika harga jual yang dibebankan ke konsumen terlalu mahal.

2.3 Big Basket

Big Basket adalah salah satu online supermarket grosir di India yang didirikan pada tahun 2011. Big Basket beroperasi di lebih dari 30 kota di India. Big Basket menawarkan berbagai macam produk mulai dari buah dan sayur-mayur, makanan, keperluan rumah tangga, hingga obat-obatan. Gambar 2.1. merupakan logo Big Basket.

Gambar 2.1. Logo Big Basket

2.4 Shopee

Shopee merupakan salah satu *marketplace* atau platform belanja *online* di Asia Tenggara dan Taiwan yang didirikan pada tahun 2015. Shopee menawarkan berbagai jenis kategori produk mulai dari elektronik hingga produk untuk hobby dan mainan. Gambar 2.2. merupakan logo Shopee.

Gambar 2.2. Logo Shopee

2.5 Random Forest

Random Forest adalah algoritma turunan dari pohon keputusan yang membagi *dataset* menjadi beberapa bagian dan menggunakan algoritma pohon keputusan pada setiap *subset*. *Random Forest* dapat mengatasi *overfitting* data (Serengil, 2017). *Random Forest* merupakan salah satu algoritma *supervised learning* yang dalam prosesnya dilakukan pemecahan data secara random ke dalam *decision tree* dan menggabungkan *tree* dengan *training* pada data yang dimiliki. Algoritma ini berupa kombinasi dari beberapa tree predictors atau bisa disebut *decision trees* dimana setiap *tree* bergantung pada nilai random *vector* yang dijadikan sampel secara bebas dan merata pada semua tree dalam forest tersebut. Hasil prediksi dari *Random Forest* didapatkan melalui hasil terbanyak dari setiap individual *decision tree* (voting untuk klasifikasi dan rata-rata untuk regresi). Untuk RF yang terdiri dari N *trees* dirumuskan sebagai:

$$l(y) = argmax_c \left(\sum_{n=1}^{N} I_{h_n(y)=c} \right)$$

Dimana *I* adalah fungsi indicator dan h_n adalah *tree* ke-n dari RF.

2.6 Metode Linear Regression

Metode Linear Regression adalah salah satu metode prediksi menggunakan perhitungan dua atau lebih variable secara matematis, yakni variable predictor dan variable kriterium. Berikut rumus metode Linear Regression.

$$\hat{Y} = a + bx$$

$$b = \frac{n(\sum XY) - (\sum X)(\sum Y)}{n(\sum X^2) - (\sum X)^2}$$

$$a = \frac{\sum Y - b \sum X}{n}$$

Keterangan : Y = variabel kriterium, X = variabel predictor, a = bilangan konstan, dan b = koefisien arah regresi linear

2.7 Metode Decision Tree

Decision Tree adalah salah satu algoritma prediksi yang dapat diterapkan pada data kategorikal ataupun numerical. Algoritma ini termasuk ke metode supervised learning. Algoritma ini akan memberikan hasil akhir berupa model yang dapat memprediksi dengan mempelajari aturan-aturan penentuan kategori berdasarkan fitru-fitur yang dimiliki oleh data. Algoritma yang diterapkan di scikit learn python adalah Gini Index. Berikut rumus Gini Index.

$$Gini\ (D) = 1 - \sum_{i=1}^{m} P_i^{\,2}$$

Dimana pi adalah perluang suatu pasangan data di D yang terdapat di kelas Ci.

2.8 Metode K-Neighbors

K-Neighbors Regression merupakan salah satu metode algoritma yang digunakan untuk masalah regresi, yakni memprediksi output berdasarkan variable-variabel independent tertentu. K-Neighbors Regression membuat prediksi berdasarkan hasil dari k tetangga yang paling dekat titik tersebut. Jarak Euclidean digunakan untuk menghitung prediksi numerik. Berikut persamaan rumus jarak Euclidean yang digunakan pada metode K-Neighbors Regression.

$$D(p,q) = \sqrt{\sum_{i=1}^{n}(q_i - P_i)^2}$$

2.9 Metode Lasso

Lasso (Least Absolute Shrinkage and Selection Operator) merupakan salah satu metode algoritma yang termasuk ke dalam regresi berkendala. Lasso digunakan untuk melakukan seleksi variable dengan cara menyusutkan koefisien regresi dari variable bebas yang sangat mempengaruhi galat menjadi sama dengan nol atau mendekati nol. Persamaan rumus metode Lasso sebagai berikut.

$$(\hat{\beta}) = argmin \sum_{i=1}^{n}(y_i - \beta_0 - \sum_{k=1}^{p} x_{ik}\,\beta_k)$$

2.10 Metode Ridge

Regresi ridge termasuk metode regresi yang memberikan hasil koefisien regresi dengan memodifikasi metode kuadrat terkecil. Ridge digunakan untuk menangani ketidakstabilan estimator kuadrat yang muncul karena adanya multicollinearity. Persamaan rumus metode Ridge sebagai berikut.

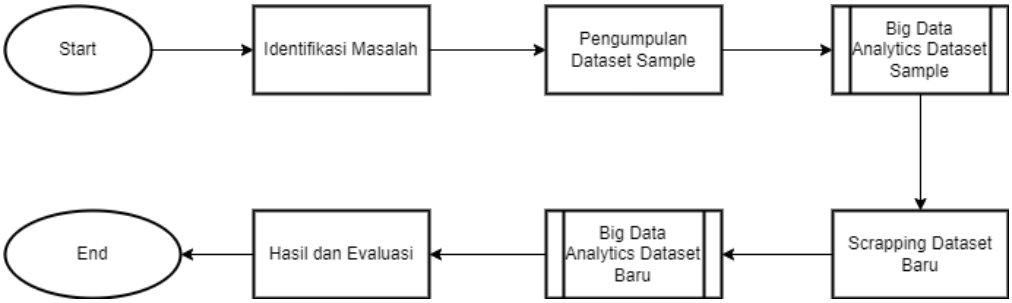
$$\widehat{\beta_R} = \sum_{i=1}^{k}(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\,\beta_j)^2$$

BAB III

METODOLOGI

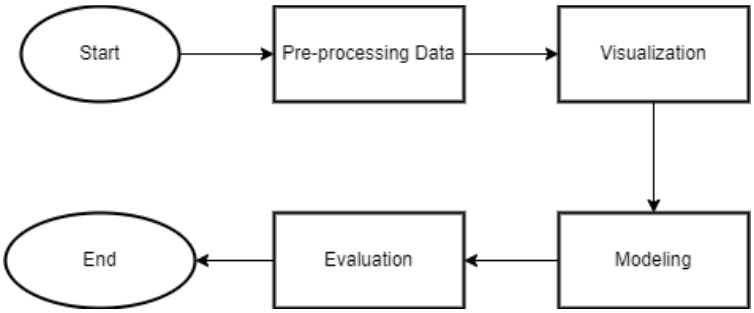
Penelitian ini memiliki beberapa tahapan di antaranya: (1) identifikasi masalah; (2) pengumpulan dataset *sample*, dataset *sampel* yang digunakan adalah Big Basket Product yang diunduh melalui *platform* kaggle.com; (3) *big data analytics dataset sample*; (4) *scrapping* dataset

baru menggunakan beautifulsoup pada *website* Shopee; (5) melakukan *big data analytics* pada dataset baru; dan (6) Hasil dan evaluasi. Alur penelitian ditunjukan pada Gambar 3.1.



Gambar 3.1. Alur penelitian

Pada tahapan *big data analytics* terhadap dataset *sample* ataupun baru terdapat sejumlah sub-proses di dalamnya yang ditunjukan pada Gambar 3.2.



Gambar 3.2. Proses dalam tahapan *Big Data Analytics*

Tahapan dalam *big data analytics* sebagai berikut:

1) *Pre-processing Data*

   Pada tahap *pre-processing data* dilakukan *data cleaning, customization data, handling missing value, label encoder,* dan *rescale data* menggunakan standarisasi.

2) *Visualization*

Pada tahapan ini dilakukan visualisasi data yang bertujuan untuk mengambil *insight* ataupun informasi dari data tersebut. Hal ini juga bertujuan untuk mempermudah dalam menganalisis data.

3) *Modeling*

Pada tahap ini dibangun model algoritma regresi, latih dan prediksi model, serta dilakukan *tuning hyperparameter*.

4) *Evaluation*

Pada tahapan dilakukan evaluasi terhadap model yang dibangun dengan menggunakan MAPE (*Mean Absolute Percentage Error*).

BAB IV

HASIL DAN PEMBAHASAN

4.1 BigBasket (Dataset Sample)

4.1.1 Pengumpulan Data

Dataset yang digunakan yaitu terkait dengan data penjualan di salah satu marketplace di India, BigBasket. *Dataset* tersebut bersumber dari kaggle, dimana *dataset* tersebut memiliki 27554 data di dalam 8 kolom.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 27555 entries, 0 to 27554
Data columns (total 8 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   product       27555 non-null  object
 1   category      27555 non-null  object
 2   sub_category  27555 non-null  object
 3   brand         27555 non-null  object
 4   sale_price    27555 non-null  float64
 5   market_price  27555 non-null  float64
 6   type          27555 non-null  object
 7   rating        27555 non-null  float64
dtypes: float64(3), object(5)
memory usage: 1.7+ MB
```

Gambar 4.1. Informasi lebih detail mengenai struktur DataFrame

4.1.2 Pre-processing Data

Pre-processing adalah tahapan dimana data atau *dataset* dipersiapkan untuk diolah lebih lanjut. Pada tahapan ini data yang tidak diinginkan akan dieliminasi dan data yang akan digunakan akan ditingkatkan kualitasnya dengan beberapa proses seperti *data cleansing, customization data, handling missing value, label encoder,* dan *rescale data* menggunakan standarisasi.

A. Handling Missing Value

```
[ ]  df['rating']= df['rating'].fillna(df['rating'].median())
```

Gambar 4.2 Replace Missing Value Rating dengan nilai median

```
[ ]  df['description'] = df['description'].fillna(df['description'].mode()[0])
     df['product'] = df['product'].fillna(df['product'].mode()[0])
     df['brand'] = df['brand'].fillna(df['brand'].mode()[0])
```

Gambar 4.3. Mengisi nilai yang kosong pada kolom description, product, dan brand

B. Drop Column

```
df.drop(columns=["index", "description"], inplace=True)
df
```

Gambar 4.4. Drop Kolom Index dan Description

C. Label Encoder

```
[ ] from sklearn.preprocessing import LabelEncoder
    label_encoder = LabelEncoder()
```

```
[ ] df['product'] = label_encoder.fit_transform(df['product'])
    df['category'] = label_encoder.fit_transform(df['category'])
    df['sub_category'] = label_encoder.fit_transform(df['sub_category'])
    df['brand'] = label_encoder.fit_transform(df['brand'])
    df['type'] = label_encoder.fit_transform(df['type'])
```

Gambar 4.5. Label Encoder

	product	category	sub_category	brand	sale_price	market_price	type	rating
0	8277	2	49	1959	220.0	220.0	204	4.1
1	22935	9	86	1258	180.0	180.0	420	2.3
2	2957	4	73	2125	119.0	250.0	249	3.4
3	3573	4	9	1386	149.0	176.0	250	3.7
4	5476	2	8	1455	162.0	162.0	39	4.4

Gambar 4.6. Hasil label encoding

D. Split Data Target and Features

```
[ ] X = df[['product','category','sub_category','brand','market_price','type','rating']]
```

Gambar 4.7. Features

```
[ ] y = df['sale_price']
    y
```

Gambar 4.8. Data Target

E. Standarisasi

```
[ ] from sklearn.preprocessing import StandardScaler
    scaler = StandardScaler()
    # transform data
    scaled = scaler.fit_transform(X)
    X = pd.DataFrame(scaled, columns=['product','category','sub_category','brand','market_price','type','rating'])
    X.head()
```

	product	category	sub_category	brand	market_price	type	rating
0	-0.508260	-1.083396	-0.021181	1.180914	-0.278582	-0.034887	0.174393
1	1.655778	1.129927	1.352182	0.136524	-0.347343	1.758819	-2.743762
2	-1.293680	-0.451018	0.869649	1.428230	-0.227011	0.338802	-0.960445
3	-1.202737	-0.451018	-1.505898	0.327225	-0.354220	0.347106	-0.474086
4	-0.921787	-1.083396	-1.543016	0.430026	-0.378286	-1.405079	0.660752

Gambar 4.9. Standarisasi

F. Feature Selection

```
[ ] # menerapkan SelectKBest class to mendapatkan top 10 best feature
    from sklearn.feature_selection import SelectKBest
    from sklearn.feature_selection import f_classif

    # Metode Filter -> Chi Square
    bestfeatures = SelectKBest(score_func=f_classif, k=3) #k = number of top features to select
    fit = bestfeatures.fit(X,y)
    dfscores = pd.DataFrame(fit.scores_)
    dfcolumns = pd.DataFrame(X.columns)

    featureScore = pd.concat([dfcolumns, dfscores], axis=1)
    featureScore.columns = ['Attr', 'Score']
    print(featureScore.nlargest(3,'Score'))
```

```
          Attr       Score
4  market_price  239.916802
1      category    2.626493
3         brand    1.799624
```

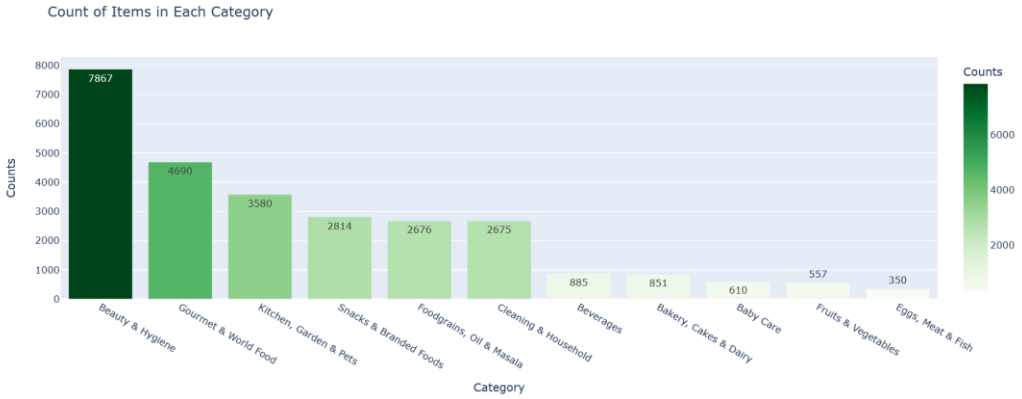
Gambar 4.10. Feature Selection

Berdasarkan feature selection tersebut didapatkan bahwa market_price, category, dan brand merupakan tiga feature atau atribut yang sangat mempengaruhi harga jual produk berdasarkan dataset *sample*.

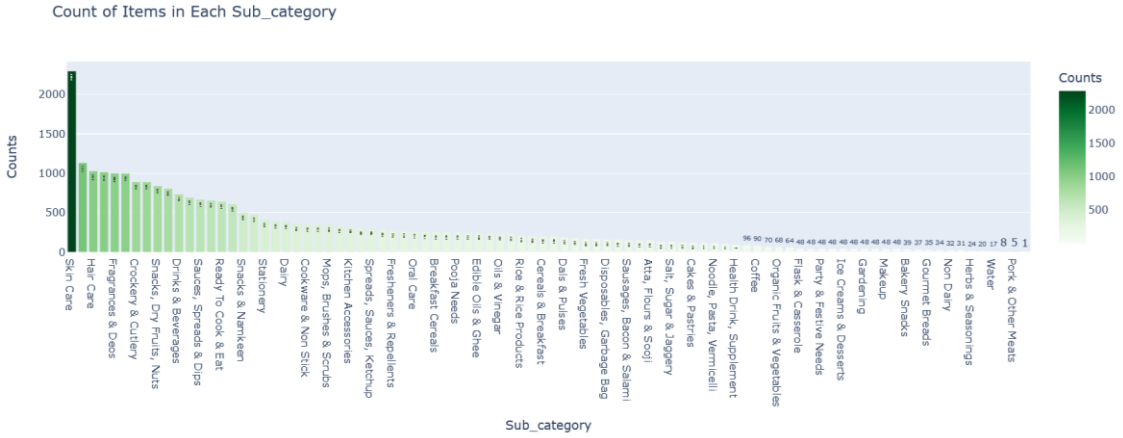
4.1.3 Visualization

Visualisasi adalah konversi data ke dalam format visual untuk menampilkan suatu informasi sehingga karakteristik dari data dapat dianalisis atau dilaporkan. Visualisasi data adalah salah satu teknik paling baik dan menarik untuk eksplorasi data. Berikut visualisasi dari *dataset* yang digunakan.

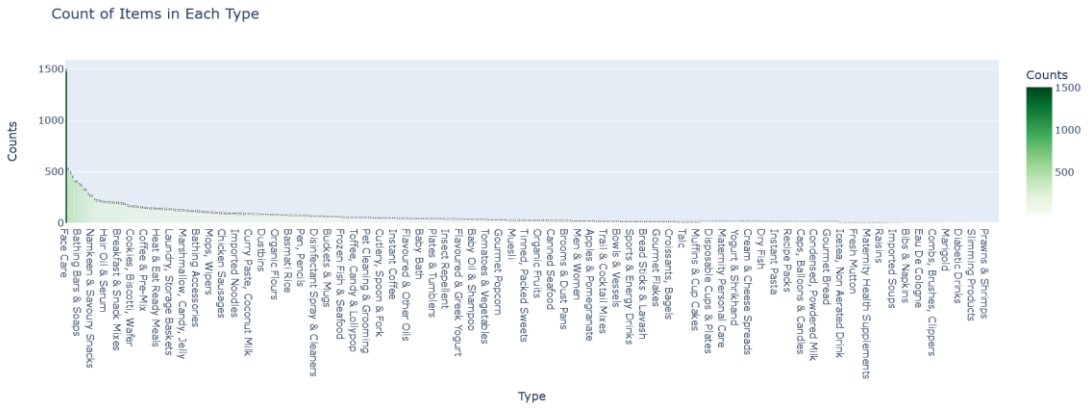
A. Visualisasi Data Count



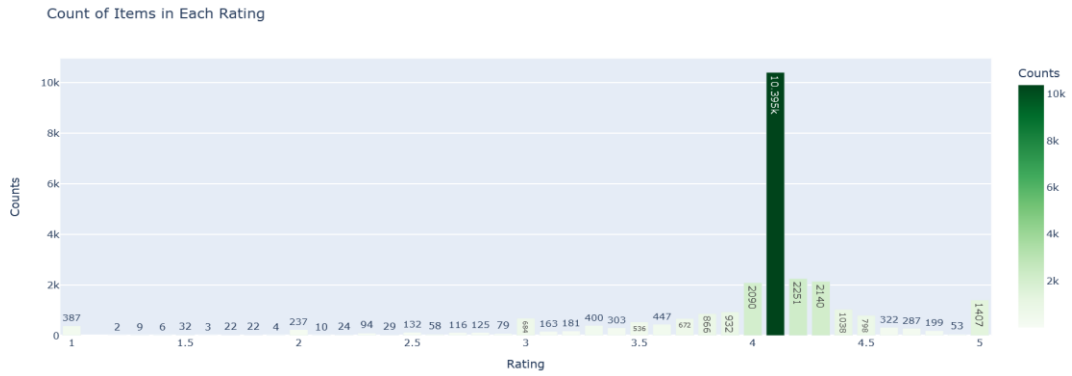
Gambar 4.11. Visualisasi Data Count Category



Gambar 4.12. Visualisasi Data Count Sub Category

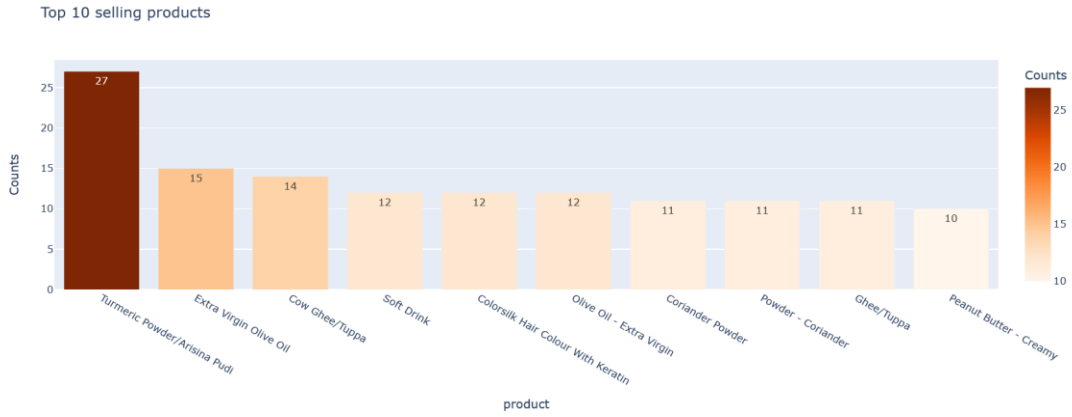


Gambar 4.13. Visualisasi Data Count Type

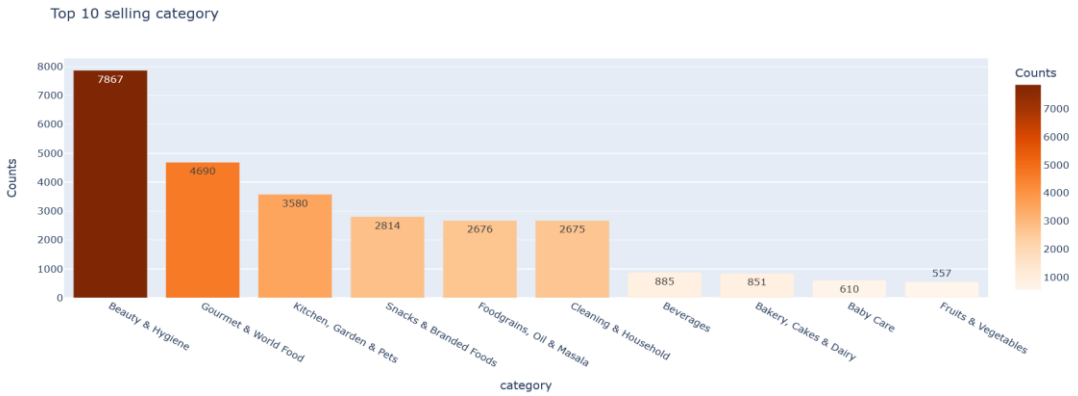


Gambar 4.14. Visualisasi Data Count Rating

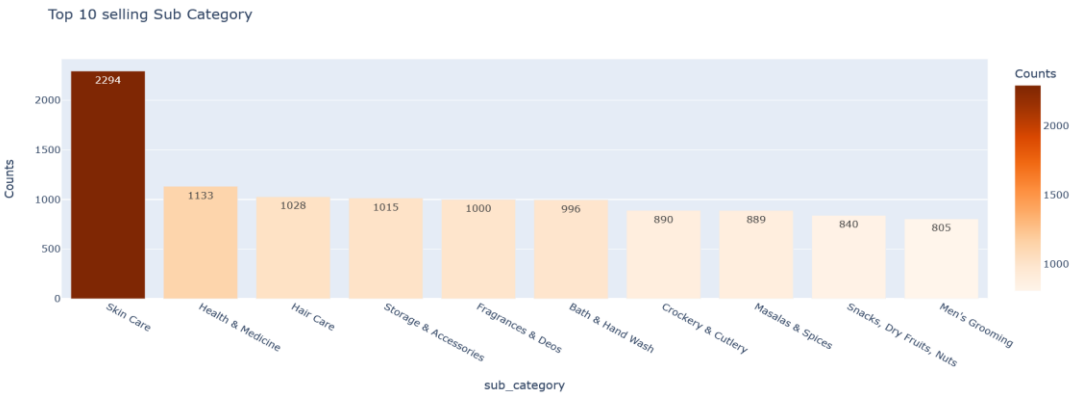
B. Visualisasi Top 10



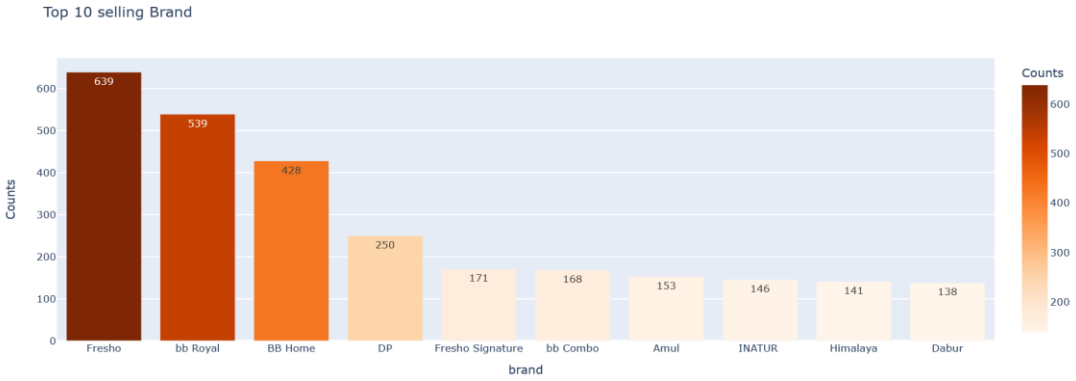
Gambar 4.15 Visualisasi Top 10 Product



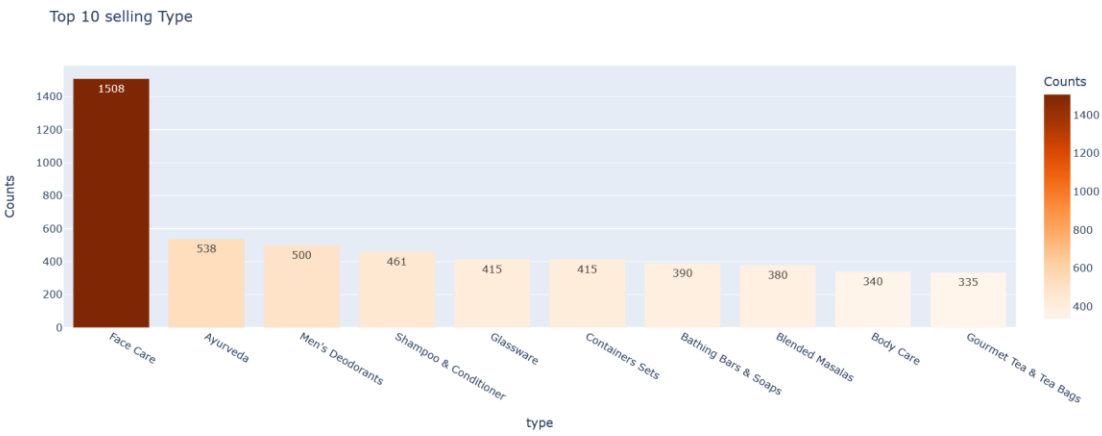
Gambar 4.16 Visualisasi Top 10 Category



Gambar 4.17 Visualisasi Top 10 Sub Category



Gambar 4.18 Visualisasi Top 10 Brand



Gambar 4.15 Visualisasi Top 10 Type

4.1.4 Modeling

Kasus prediksi harga jual suatu produk ini termasuk ke kategori regresi sehingga model yang digunakan adalah model algoritma regresi. Beberapa model algoritma regression digunakan dan model dilatih dengan parameter *default*. Hasil *train score, test score,* dan MAPE dari setiap model ditunjukan pada Gambar 4.16.

	Model	Train Score	Test Score	MAPE
0	Random Forest	0.990759	0.944853	0.093778
1	Linear Regression	0.929831	0.936229	0.194145
2	Decision Tree	0.997776	0.827926	0.097419
3	K-Neighbors	0.956712	0.931289	0.127387
4	Lasso	0.929828	0.936200	0.194698
5	Ridge	0.929831	0.936227	0.194206

Gambar 4.16. Hasil *train score, test score,* dan MAPE dari setiap model

Berdasarkan hasil tersebut, random forest merupakan model dengan MAPE paling rendah sehingga model tersebut yang digunakan. Untuk mendapatkan tingkat error yang lebih minimum dilakukan *tuning hyperparameter* dengan menggunakan *randomized search.* Dengan *tuning hyperparameter* didapatkan *best estimator*, yakni bootstrap=False, max_depth=80, max_features='sqrt', dan n_estimators=1350.

4.1.5 Evaluation

Setelah model random forest dilakukan *tuning hyperparameter*, model dilatih dan diuji kembali yang mana didapatkan hasil *training score* sebesar 97,77%, *testing score* sebesar 94,76%, dan MAPE 0,084.

4.2 Shopee (Dataset Baru)

4.1.1 Pengumpulan Data

Dataset baru diperoleh melalui *scraping website* Shopee Indonesia dengan menggunakan BeautifulSoup. Dataset produk yang di-*scrape* hanya terbatas pada produk-produk yang terdapat di Shopee Mall dengan data yang diambil berupa nama produk, kategori, *brand*, jumlah yang terjual, dan harga jual produk. Gambar 4.17 adalah dataset mentah hasil *scraping* yang diperoleh.

	Unnamed: 0	product_name	category	brand	item_sold	city	price
0	0	Samsung HD TV 32" UA32T4001 (2020)	Elektronik	Samsung Official Shop	4,9RB Terjual	KOTA JAKARTA UTARA	Rp2.699.000
1	1	Samsung Smart HD TV 32" T4500 - UA32T4500AK	Elektronik	Samsung Official Shop	2,9RB Terjual	KOTA JAKARTA UTARA	Rp2.729.000
2	2	Samsung Galaxy Tab A7 Lite 3+32 GB Gray	Elektronik	Samsung Official Shop	4,8RB Terjual	KOTA JAKARTA UTARA	Rp2.149.000
3	3	Samsung AC 1/2 PK Standard R32 dengan Fast Coo...	Elektronik	Samsung Official Shop	1,5RB Terjual	KOTA JAKARTA UTARA	Rp3.079.000
4	4	Samsung Galaxy Tab A7 Lite 3+32 GB Silver	Elektronik	Samsung Official Shop	1,4RB Terjual	KOTA JAKARTA UTARA	Rp2.149.000

Gambar 4.17. dataset mentah hasil *scraping*

Dataset terdiri atas tujuh kolom dan 13680 baris dengan bertipe data numerik ataupun objek / kategori.

4.1.2 Pre-processing Data

Pre-processing adalah tahapan dimana data atau *dataset* dipersiapkan untuk diolah lebih lanjut. Pada tahapan ini data yang tidak diinginkan akan dieliminasi dan data yang akan digunakan akan ditingkatkan kualitasnya dengan beberapa proses seperti *data cleansing, customization data, handling missing value, label encoder,* dan *rescale data* menggunakan standarisasi.

A. Data Cleansing

```
[ ]  def extract_nilai(nilai):
       # a = str(nilai)
       a = str(nilai).split()
       a_min = a[0].replace('.', '')
       a_max = a[-1].replace('.', '')
       return a[-1].replace('.', '')

     df['price'] = df['price'].map(extract_nilai)
```

Gambar 4.18. Extract Nilai

Gambar 4.19. Data Cleansing

B. Create Column Market Price

Membuat kolom market price dari masing-masing nilai rerata kolom price berdasarkan brand, category, dan city.

	product_name	category	brand	item_sold	city	price	market_price_brand	market_price_category	market_price_city
60	Smartwatch Aukey Fitnes Tracker 12 Activity - ...	Elektronik	Aukey Indonesia Official Shop	10000.0	KOTA JAKARTA BARAT	298000	150561.02	174342.14	122326.8
61	Car Charger Aukey Expedition Series CC-S1 2 Port	Elektronik	Aukey Indonesia Official Shop	3700.0	KOTA JAKARTA BARAT	98000	150561.02	174342.14	122326.8
62	Aukey Charger 20W PD 3.0 Fast Charging	Elektronik	Aukey Indonesia Official Shop	10000.0	KOTA JAKARTA BARAT	248000	150561.02	174342.14	122326.8
63	Aukey Cable Micro USB 2.0 1M (NO PACKING & NO ...	Elektronik	Aukey Indonesia Official Shop	10000.0	KOTA JAKARTA BARAT	24000	150561.02	174342.14	122326.8
64	Sarung / Pouch Powerbank	Elektronik	Aukey Indonesia Official Shop	6800.0	KOTA JAKARTA BARAT	14000	150561.02	174342.14	122326.8

Gambar 4.20. Create Column Market Price

C. Handling missing value

```
total=newdf.isnull().sum().sort_values(ascending=False)
print(total)
```

```
item_sold               279
product_name              0
category                  0
brand                     0
city                      0
price                     0
market_price_brand        0
market_price_category     0
market_price_city         0
dtype: int64
```

Gambar 4.21. Jumlah missing value sebelum ditangani

```
total=newdf.isnull().sum().sort_values(ascending=False)
print(total)
```

```
product_name            0
category                0
brand                   0
item_sold               0
city                    0
price                   0
market_price_brand      0
market_price_category   0
market_price_city       0
dtype: int64
```

Gambar 4.22. Jumlah missing value setelah ditangani

D. Mengatasi *Outliers* pada kolom *price* dan *item_sold*

```
# Hitung nilai Q1 dan Q3 untuk kolom price
Q1 = np.quantile(df['price'], .25)
Q3 = np.quantile(df['price'], .75)
```

```
# Hitung nilai IQR kolom price
IQR = Q3 - Q1
```

```
print('Kuartil 1 = ', Q1)
print('Kuartil 3 = ', Q3)
print('IQR = ', IQR)
```

```
Kuartil 1 =  56500.0
Kuartil 3 =  279900.0
IQR =  223400.0
```

```
min_IQR = Q1 - 1.5 * IQR
max_IQR = Q3 + 1.5 * IQR

print('IQR minimum = ', min_IQR)
print('IQR maksimum = ', max_IQR)
```

```
IQR minimum =  -278600.0
IQR maksimum =  615000.0
```

```
newdf = df[(df['price'] >= min_IQR) & (df['price'] <= max_IQR)]
```

Gambar 4.23. *Handling outliers* pada kolom *price*

```
# Hitung nilai Q1, Q3, dan IQR pada item_sold
Q1 = np.quantile(newdf['item_sold'], .25)
Q3 = np.quantile(newdf['item_sold'], .75)
IQR = Q3 - Q1
```

```
min_IQR = Q1 - 1.5 * IQR
max_IQR = Q3 + 1.5 * IQR

print('IQR minimum = ', min_IQR)
print('IQR maksimum = ', max_IQR)
```

```
IQR minimum =  -6842.5
IQR maksimum =  11785.5
```

```
newdf = newdf[(newdf['item_sold'] >= min_IQR) & (newdf['item_sold'] <= max_IQR)]
newdf
```

Gambar 4.24. *Handling outliers* pada kolom *item_sold*

```
columns=['price']

for c in columns:

    Range=c+'_RANGE'
    newdf[Range]=0
    newdf.loc[((df[c]>0)&(df[c]<=50000)),Range]=1
    newdf.loc[((df[c]>50000)&(df[c]<=100000)),Range]=2
    newdf.loc[((df[c]>100000)&(df[c]<=250000)),Range]=3
    newdf.loc[((df[c]>250000)&(df[c]<=400000)),Range]=4
    newdf.loc[((df[c]>400000)),Range]=5
```

Gambar 4.25. Membuat kolom *price range*

Pada Gambar 4.25 menunjukan membuat kolom baru dengan nama price_range bertujuan untuk mengatasi *outliers* pada kolom *price.* Kolom *price_range* berisi pengelompokan range harga produk menjadi 5 kategori yang berbeda. Kategori 1 adalah produk dengan harga dibawah Rp50.000 dan kategori 5 adalah produk dengan harga di atas Rp400.0000.

E. Label Encoder

```
from sklearn.preprocessing import LabelEncoder
label_encoder = LabelEncoder()
```

```
newdf['product_name'] = label_encoder.fit_transform(newdf['product_name'])
newdf['category'] = label_encoder.fit_transform(newdf['category'])
newdf['brand'] = label_encoder.fit_transform(newdf['brand'])
newdf['city'] = label_encoder.fit_transform(newdf['city'])
```

Gambar 4.26. Proses label encoder

```
newdf.tail()
```

	product_name	category	brand	item_sold	city	price	market_price_brand	market_price_category	market_price_city	price_RANGE
11736	206	2	37	3400.0	3	46400	78980.0	76571.12	116731.57	1
11737	202	2	37	5200.0	3	13700	78980.0	76571.12	116731.57	1
11738	195	2	37	8000.0	3	93800	78980.0	76571.12	116731.57	2
11739	200	2	37	7000.0	3	55200	78980.0	76571.12	116731.57	2
11740	481	2	37	4100.0	3	70500	78980.0	76571.12	116731.57	2

Gambar 4.27. Hasil label encoder

F. Split Data Target

	product_name	category	brand	item_sold	city	market_price_brand	market_price_category	market_price_city	price_RANGE
814	1142	4	29	4500.0	1	123466.67	223724.69	111580.49	3
815	1112	4	29	7500.0	1	123466.67	223724.69	111580.49	3
816	1139	4	29	10000.0	1	123466.67	223724.69	111580.49	2
817	1155	4	29	5400.0	1	123466.67	223724.69	111580.49	3
818	1107	4	29	7900.0	1	123466.67	223724.69	111580.49	3
...	...	...	...	...	...	...	...	...	...
11736	206	2	37	3400.0	3	78980.00	76571.12	116731.57	1
11737	202	2	37	5200.0	3	78980.00	76571.12	116731.57	1
11738	195	2	37	8000.0	3	78980.00	76571.12	116731.57	2
11739	200	2	37	7000.0	3	78980.00	76571.12	116731.57	2
11740	481	2	37	4100.0	3	78980.00	76571.12	116731.57	2

Gambar 4.28. Features

```
814      101000
815      160000
816       75000
817      183000
818      130000
          ...
11736     46400
11737     13700
11738     93800
11739     55200
11740     70500
```

Gambar 4.29. Data Target

G. Standarisasi

```
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
# transform data
scaled = scaler.fit_transform(X)
X = pd.DataFrame(scaled, columns=['product_name','category','brand','item_sold','city', 'market_price_brand','market_price_categ
X.head()
```

	product_name	category	brand	item_sold	city	market_price_brand	market_price_category	market_price_city	price_RANGE
0	-0.933936	-0.249072	-0.972409	0.334633	-1.252333	-0.203641	2.027626	-1.333617	0.469692
1	-0.955190	-0.249072	-0.972409	1.116317	-1.252333	-0.203641	2.027626	-1.333617	0.469692
2	-0.936062	-0.249072	-0.972409	1.767720	-1.252333	-0.203641	2.027626	-1.333617	-0.451755
3	-0.924726	-0.249072	-0.972409	0.569138	-1.252333	-0.203641	2.027626	-1.333617	0.469692
4	-0.958733	-0.249072	-0.972409	1.220541	-1.252333	-0.203641	2.027626	-1.333617	0.469692

Gambar 4.30. Standarisasi

H. Feature Selection

Gambar 4.30. *Correlation map*

```
# menerapkan SelectKBest class to mendapatkan top 4 best feature
from sklearn.feature_selection import SelectKBest
from sklearn.feature_selection import f_classif

# Metode Filter -> Chi Square
bestfeatures = SelectKBest(score_func=f_classif, k=4) #k = number of top features to select
fit = bestfeatures.fit(X,y)
dfscores = pd.DataFrame(fit.scores_)
dfcolumns = pd.DataFrame(X.columns)

featureScore = pd.concat([dfcolumns, dfscores], axis=1)
featureScore.columns = ['Attr', 'Score']
print(featureScore.nlargest(4,'Score'))
```

```
                  Attr         Score
8           price_RANGE  1.224822e+16
5    market_price_brand  1.083353e+01
6  market_price_category  4.262434e+00
7     market_price_city  3.745648e+00
```

Gambar 4.31. Features Selection

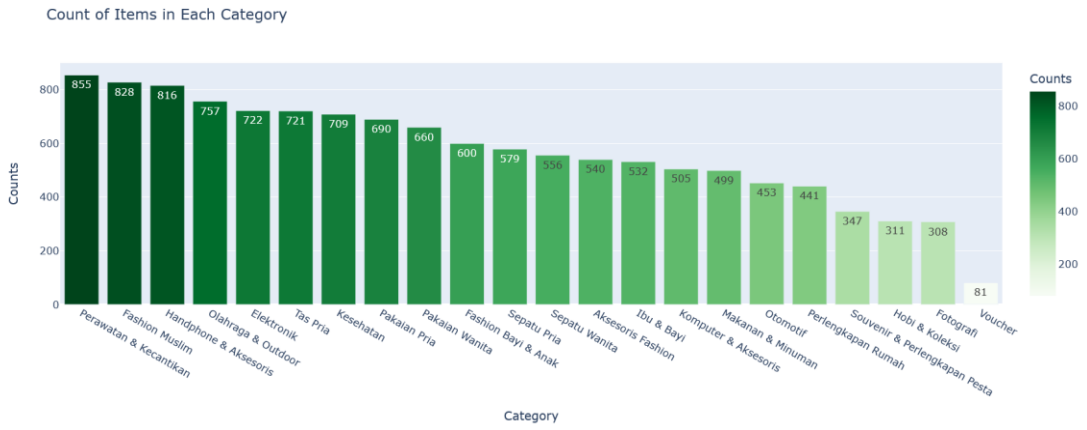
Berdasarkan *feature selection* tersebut didapatkan bahwa *price_range*, *market_price_brand*, *market_price_category*, dan *market_price_city* merupakan

empat *feature* atau atribut yang sangat mempengaruhi harga jual produk berdasarkan *dataset* hasil *scraping*.

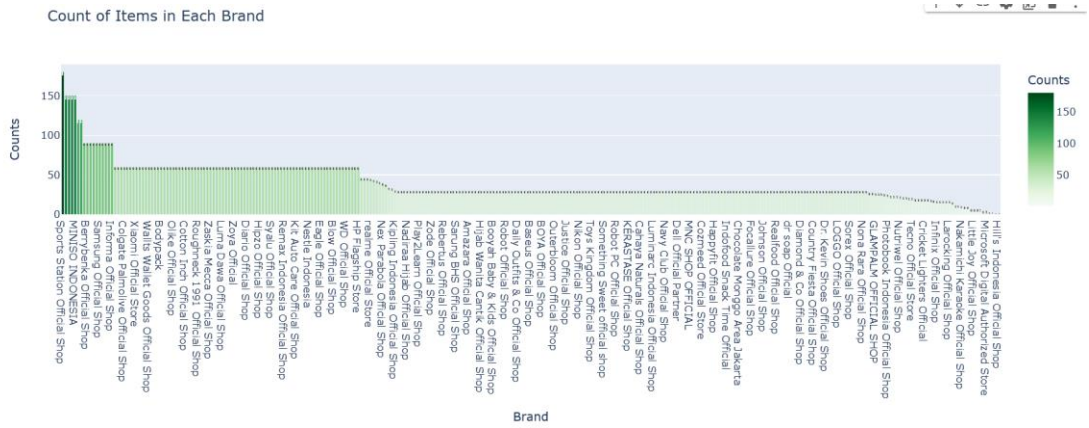
4.1.3 Visualization

Visualisasi adalah konversi data ke dalam format visual untuk menampilkan suatu informasi sehingga karakteristik dari data dapat dianalisis atau dilaporkan. Visualisasi data adalah salah satu teknik paling baik dan menarik untuk eksplorasi data. Berikut visualisasi dari *dataset* yang digunakan.

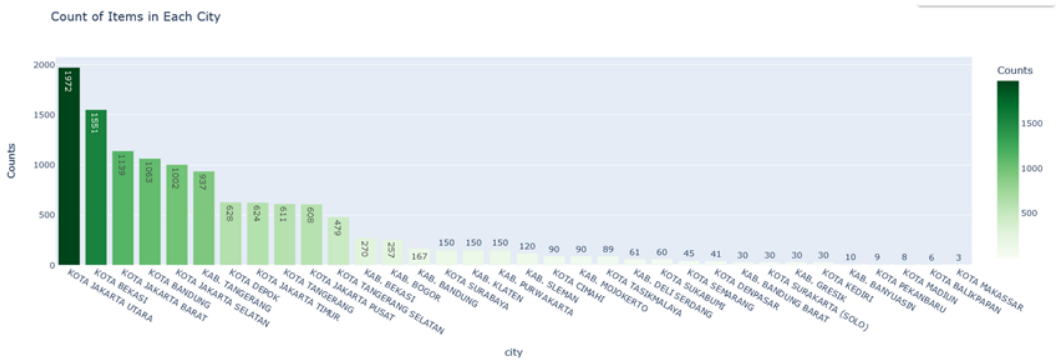
A. Visualisasi Data Count



Gambar 4.32. Visualisasi Data Count Category



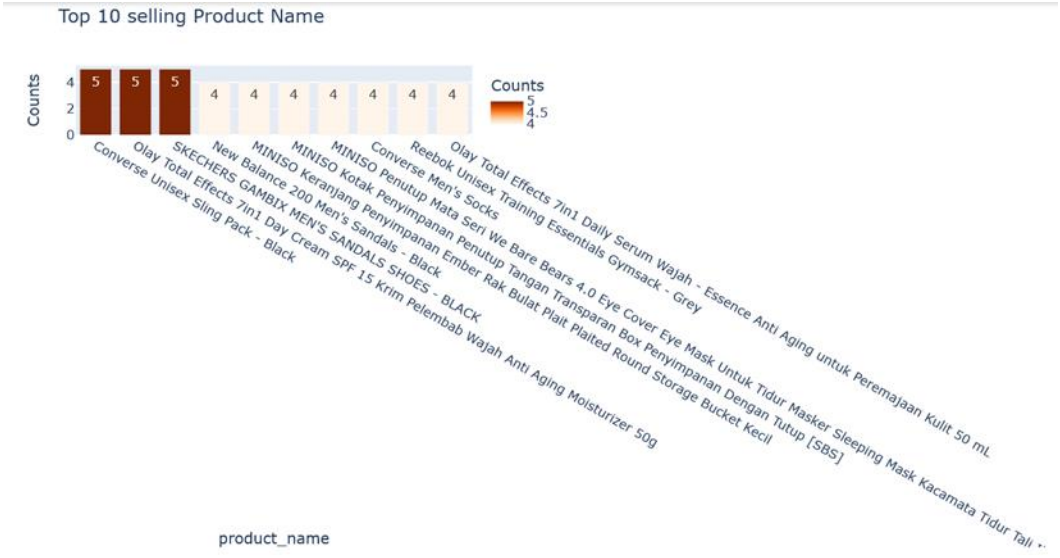
Gambar 4.33. Visualisasi Data Count Brand



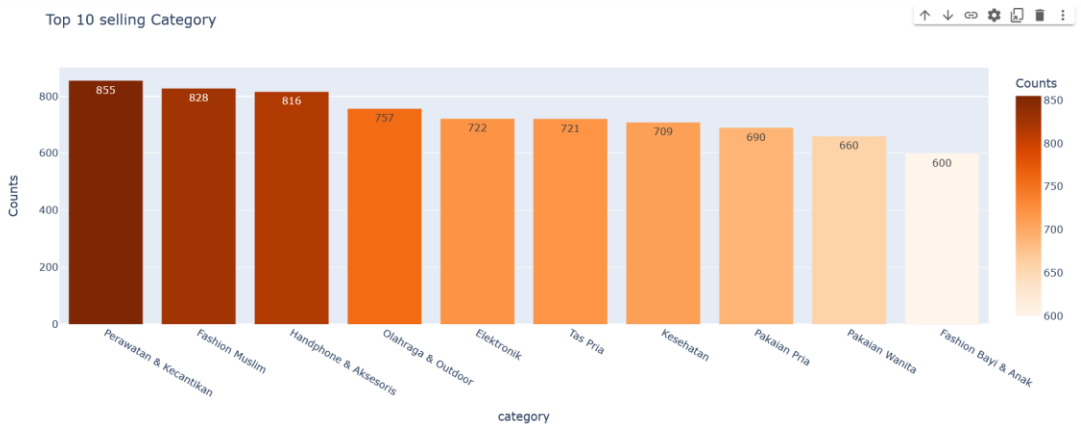
Gambar 4.34. Visualisasi Data Count City

Gambar 4.35. Visualisasi Data Count Price

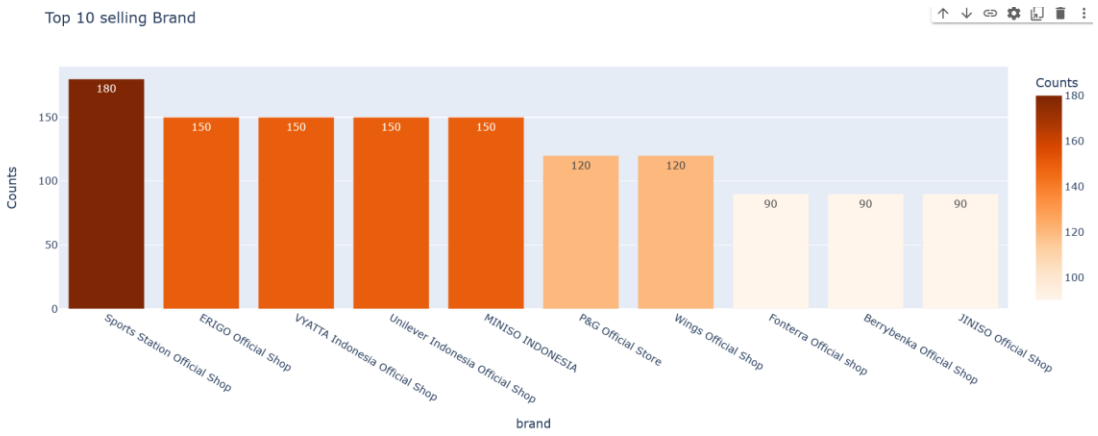
B. Visualisasi Top 10



Gambar 4.36. Visualisasi Top 10 Product Name



Gambar 4.37. Visualisasi Top 10 Category



Gambar 4.38. Visualisasi Top 10 Brand

4.1.4 Modeling

Pada *dataset* baru hasil *scraping* dilakukan modeling terutama modeling dengan menggunakan model algoritma random forest sesuai. Berdasarkan *train score*, *test score*, dan MAPE pada *dataset* baru diperoleh hasil kurang lebih sama dengan menggunakan *dataset sample* yang ditunjukan pada Gambar 4.39.

	Model	Train Score	Test Score	MAPE
0	Random Forest	0.992196	0.947182	0.192451
1	Linear Regression	0.862535	0.866333	0.495587
2	Decision Tree	1.000000	0.916585	0.209146
3	K-Neighbors	0.930719	0.886643	0.283216
4	Lasso	0.862517	0.866465	0.494040
5	Ridge	0.862535	0.866339	0.495321

Gambar 4.39. *train score*, *test score*, dan MAPE

Model random forest termasuk ke salah satu model dengan MAPE rendah, yakni sebesar 0,19. *Tuning hyperparameter* dilakukan terhadap model random forest dengan menggunakan *randomized search dan grid search* yang ditunjukan pada Gambar 4.40 dan 4.41.

```
print(rf_random.best_estimator_)
```

```
RandomForestRegressor(max_depth=30, max_features='sqrt', min_samples_leaf=2,
                      min_samples_split=10, n_estimators=50)
```

```
# training score
print('Training Score: ')
print(rf_random.score(X_train, y_train))
```

```
Training Score:
0.9664072152029405
```

```
y_pred = rf_random.predict(X_test)
```

```
# testing score
print('Testing Score: ')
print(rf_random.score(X_test, y_test))
```

```
Testing Score:
0.9427269628614144
```

```
mean_absolute_percentage_error(y_true=y_test, y_pred=y_pred)
```

```
0.22741436869023982
```

Gambar 4.40. Hasil *tuning hyperparameter* dengan *randomized search*

```
print(grid_results.best_estimator_)
```

```
RandomForestRegressor(max_depth=16, n_estimators=256)
```

```
# Extract the best decision forest
best_clf = grid_results.best_estimator_
y_pred = best_clf.predict(X_test)
```

```
# training score
print('Training Score: ')
print(best_clf.score(X_train, y_train))
```

```
Training Score:
0.9901736584081354
```

```
# testing score
print('Testing Score: ')
print(best_clf.score(X_test, y_test))
```

```
Testing Score:
0.9481211916185184
```

```
mean_absolute_percentage_error(y_true=y_test, y_pred=y_pred)
```

```
0.19074868527986982
```

Gambar 4.40. Hasil *tuning hyperparameter* dengan *grid search*

4.1.5 Evaluation

Setelah dilakukan *tuning hyperparameter* baik dengan *randomized search* maupun *grid search*, diperoleh hasil model mendapatkan tingkat akurasi yang lebih baik dengan menggunakan *grid search*, yakni MAPE sebesar 0,19 atau 19%. Berdasarkan hasil tersebut, model random forest cocok untuk dataset baru dibandingkan model algoritma lainnya dengan menggunakan *tuning hyperparameter* dan setelah dilakukan tahapan *preprocessing data* yang cukup panjang.

BAB V

PENUTUP

5.1 Kesimpulan

Hal-hal yang mempengaruhi harga jual suatu produk bisa berbeda-beda antara marketplace satu dengan marketplace lainnya. Pada Shopee Indonesia, harga jual suatu produk khususnya produk Shopee Mall Indonesia sangat dipengaruhi oleh *price_range*, *market_price_brand, market_price_category*, dan *market_price_city*.

Pada penelitian dilakukan percobaan perbandingan beberapa algoritma baik pada *dataset sample* maupun *dataset* hasil *scraping*. Berdasarkan percobaan tersebut didapatkan bahwa metode atau model Random Forest memiliki tingkat *error* terendah dan skor *training* serta *testing* tertinggi dibandingkan metode algoritma lainnya.

Pada percoobaan awal menggunakan *dataset* Big Basket memperoleh tingkat *error* menggunakan metode *random forest* terhadap *dataset sample* sebesar 0,084 sedangkan terhadap *dataset* hasil *scraping* sebesar 0,19 setelah dilakukan *tuning hyperparameter* dengan menggunakan *GridSearchCV*.

5.2 Saran dan Diskusi

Penelitian ini masih terdapat banyak kekurangan sehingga diharapkan penelitian selanjutnya dapat memperbaharui dan melengkapi penelitian ini. Berikut beberapa saran untuk penelitian selanjutnya.

1. Melakukan *scraping product* menggunakan metode selain BeautifulSoup dan mengambil *features* atau atribut yang lebih lengkap.
2. Melakukan *preprocessing data* yang lebih baik
3. Menggunakan metode atau model algoritma *deep learning.*

DAFTAR PUSTAKA

APJI (2022) 'Hasil Survey Profil Internet Indonesia 2022', Apji.or.Od, (June). Available at: apji.or.id.

Kemkominfo (2019) 'Kemkominfo: Pertumbuhan e-Commerce Indonesia Capai 78 persen', Available at: https://www.kominfo.go.id/content/detail/16770/kemkominfo-pertumbuhan-e-commerce-indonesiacapai-78-persen/0/sorotan_media. Diakses pada tanggal 7 September 2022.

Patras, P. T. (2018). "Analisis Perhitungan Harga Pokok Produksi dan Penentuan Harga Jual (Studi Kasus pada Weerstand.co).

Ahmad, S., Singh, P. and Sagar, A. K. (2018) 'A Survey on Big Data Analytics', *Proceedings - IEEE 2018 International Conference on Advances in Computing, Communication Control and Networking, ICACCCN 2018*, 4, pp. 256–260. doi: 10.1109/ICACCCN.2018.8748774.

Henri Slat, A., Harga Pokok, A. and Henri Slat Fakultas Ekonomi Jurusan Akuntansi Universitas Sam ratulangi Manado, A. (2013) 'Analisis Harga Pokok Produk Dengan Metode Full Costing Dan Penentuan Harga Jual', *110 Jurnal EMBA*, 1(3), pp. 110–117.

Sari, M. E. L. (2019) 'PENGARUH VIRAL MARKETING DAN BRAND AMBASSADOR TERHADAP KEPUTUSAN PEMBELIAN MELALUI APLIKASI SHOPEE (Studi Kasus pada Mahasiswa Manajemen Bisnis Syari'ah IAIN Kudus) SKRIPSI', *Rabit : Jurnal Teknologi dan Sistem Informasi Univrab*, 1(1), p. 2019. Available at: http://www.ghbook.ir/index.php?name= فرهنگ و رسانه های نوین&option=com_dbook&task=readonline&book_id=13650&page=73&chkhashk=ED9C9491B4&Itemid=218&lang=fa&tmpl=component%0Ahttp://www.albayan.ae%0Ahttps://scholar.google.co.id/scholar?hl=en&q=APLIKASI+PENGENA.

Overview, C. and Objectives, L. (2013) 'Big Basket', pp. 4–6.

Ferdyandi, M., Setiawan, N. Y. and Bachtiar, F. A. (2022) 'Prediksi Potensi Penjualan Makanan Beku berdasarkan Ulasan Pengguna Shopee menggunakan Metode Decision Tree Algoritma C4 . 5 dan Random Forest (Studi Kasus Dapur Lilis)', 6(2), pp. 588–596.

Wandani, A. (2021) 'Sentimen Analisis Pengguna Twitter pada Event Flash Sale Menggunakan Algoritma K-NN, Random Forest, dan Naive Bayes', Jurnal Sains Komputer & Informatika (J-SAKTI, 5(2), pp. 651–665.

F. Apriliza, A. Oktavyani, and D. al Kaazhim, “Perbandingan Metode Linear Regression dan Exponential Smoothing Dalam Peramalan Penerimaan Mahasiswa Baru,” Jurnal Riset Komputer), vol. 9, no. 3, pp. 2407–389, 2022, doi: 10.30865/jurikom.v9i3.4300.

R. Latifah, E. Setia Wulandari, and dan Priadhana Edi Kreshna, “Model Decision Tree untuk Prediksi Jadwal Kerja menggunakan Scikit-Learn,” 2019.

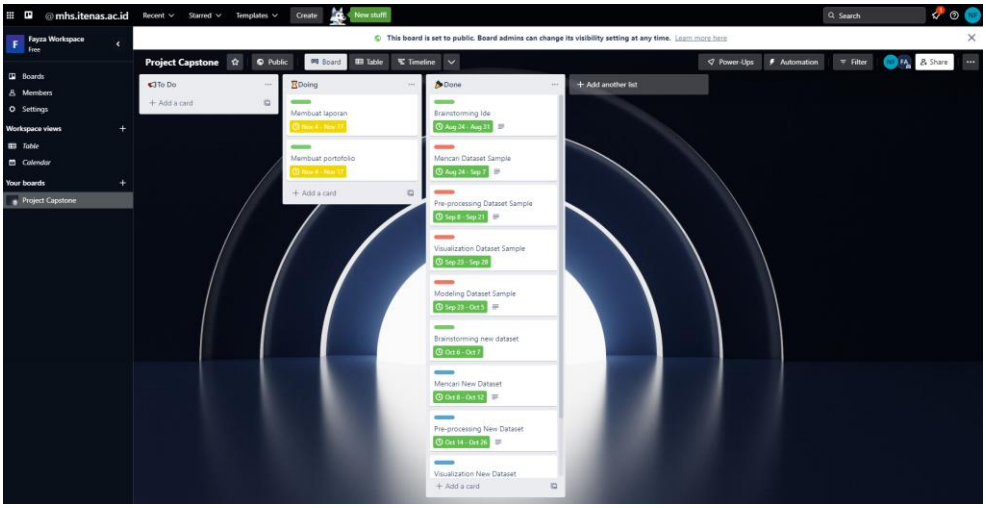
J. Tanuwijaya and S. Hansun, “LQ45 stock index prediction using k-nearest neighbors regression,” International Journal of Recent Technology and Engineering, vol. 8, no. 3, pp. 2388–2391, Sep. 2019, doi: 10.35940/ijrte.C4663.098319.

Y. A. Mait, D. Tineke Salaki, H. A. H. Komalig, and K. Kunci, “Kajian Model Prediksi Metode Least Absolute Shrinkage and Selection Operator (LASSO) pada Data Mengandung Multikolinearitas LASSO Metode kuadrat terkecil Multikolinearitas.” [Online]. Available: https://ejournal.unsrat.ac.id/index.php/decartesian

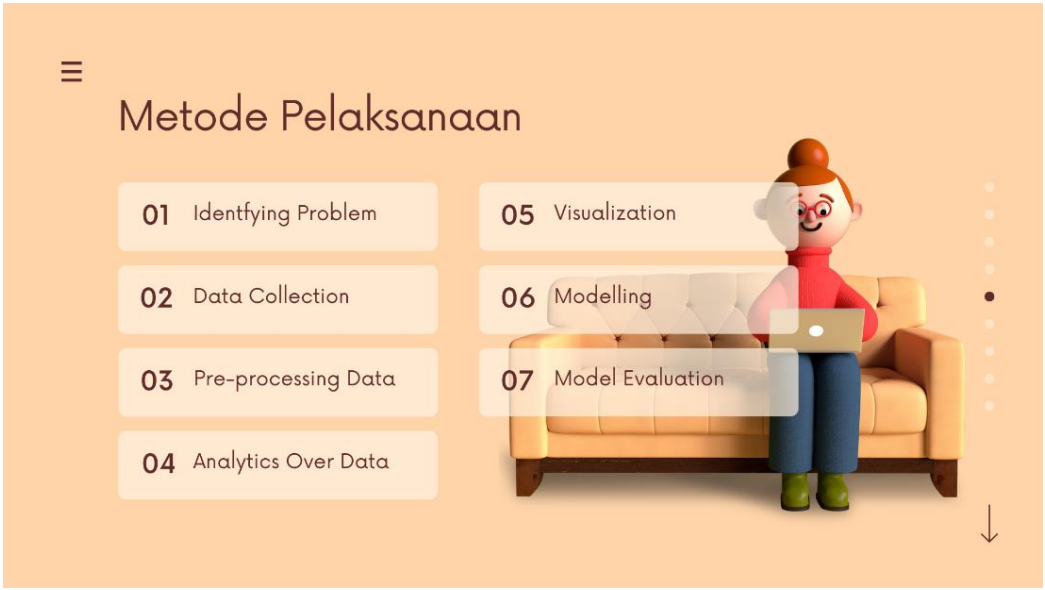
R. Apriandi, M. Bagus Insan, F. Rizmawan, H. As Haq, and D. Dwi Priyono, “PERANCANGAN APLIKASI PREDIKSI HARGA EMAS, PERAK, DOLAR, MENGGUNAKAN ALGORITMA REGRESSION BERBASIS WEB”, [Online]. Available: https://ejournal.stmikgici.ac.id/

LAMPIRAN

Lampiran 1. Trello



Lampiran 2. Rancangan Portofolio




Bisa.ai
Kampus Merdeka
Project Capstone AI-Hacker
Big Data Analytics Harga Jual pada Shopee
Bagian 1: Pre-processing Data
Created by Naufal Mufid F.


Bisa.ai
Kampus Merdeka
Project Capstone AI-Hacker
Big Data Analytics Harga Jual pada Shopee
Bagian 2: Visualization
Created by Dzakiyyah Al Kaazhim


Bisa.ai
Kampus Merdeka
Project Capstone AI-Hacker
Big Data Analytics Harga Jual pada Shopee
Bagian 3: Modeling
Created by Fayza Apriliza


Bisa.ai
Kampus Merdeka
Project Capstone AI-Hacker
Big Data Analytics Harga Jual pada Shopee
Bagian tambahan: Scrapping Dataset
Created by Fayza Apriliza