Laporan Proyek Machine Learning - Adib Ahmad Istiqlal

Domain Proyek

Indonesia sebagai negara berkembang memiliki banyak parameter yang berkaitan mengenai stabilitas perkonomian maupun keuangan. Parameter yang bekaitan tersebut salah satunya adalah pendapatan e-commerce. Pada penelitian yang dilakukan Rianty dan Rahayu dengan judul Pengaruh E-commerce Terhadap Pendapatan UMKM Yang Bermitra Gojek Dalam Masa Pandemi Covid-19, menyatakan bahwa e-commerce memiliki pendapatan yang baik bagi negara khususnya dari segi UMKM dengan peningkatan total transaksi hingga 5%. Peningkatan yang cukup tinggi, pada permasalahan ini penulis ingin mengetahui bagaimana pendapatan e-commerce dari parameter penggunaaan sistem dan lamanya membership terhadap pendapatan e-commerce dengan menggunakan model machine learning classic menggunakan *support vector regression* yang merupakan kembangan *support vector machine* yang diperkenalkan oleh Vapnik pada tahun 1992. Pada penelitian Support Vector Regression (SVR) Dalam Memprediksi Harga Minyak Kelapa Sawit di Indonesia dan Nilai Tukar Mata Uang EUR/USD yang dilakukan oleh Saadah, dkk pada tahun 2021 menghasilkan akurasi yang hampir mendekati 100% Terumata pada penggunaan kernel RBF.

Business Understanding

Pada pernyataan yang telah dijelaskan, sehingga masalah yang diangkat adalah

- Bagaimana pengaruh penggunaan sistem e-commerce dan lamanya membership user terhadapa pendapatan oleh e-commerce.
- Bagaimana akurasi pendapatan e-commerce dengan sistem e-commerce dan lamanya membership user menggunakan kernel linear.

Tujuan dari masalah yang diangkat adalah

- Mengetahui pengaruh korelasi terhadap parameter tersebut terhadap pendapatan oleh e-commer
- Mengetahui akurasi kernel linear terhadap prediksi pendapatan e-commerce dengan parameter penggunaan sistem e-commerce dan lamanya membership user

Solusi Statements Solusi yang dapat dilakukan

- Menggunakan korelasi dengan bantuan visualisasi heatmap dengan library seaborns
- Mengevaluasi hasil kernel linear *dengan mean squared error*
- Melakukan optimasi parameter kernel linear dengan parameter aslinya yaitu C, untuk meningkatkan hasil akurasi.

- ## Data Understanding

  Dataset yang digunakan pada penelitian ini adalah dataset pakaian secara online yang dapat dilakukan dari website atau app. Sumber dataset ini berasal dari Kaggle.com. Adapun kolom-kolom pada dataset ini, antara lain.
- E-mail : Alamat surat elektronik pengguna yang dapat digunakan sebagai ID.
- Address : Alamat tempat tinggal dari pengguna
- Avatar : Foto pengguna
- Avg. Session Length : Lamanya Session pengguna pada sistem yang tercatat
- Time on App : Lamanya penggunaan aplikasi perusahaan oleh pengguna

- Time on Website : Lamanya penggunaan aplikasi perusahaan oleh pengguna
- Length of Membership : Lamanya pengguna terdaftar
- Yearly Amount Spent : Pendapatan dari pengguna terhadap perusahaan.

Pada kolom diatas, **label** yang digunakan adalah kolom Yearly Amount Spent dan total dataset dari dataset ini berjumlah 500 baris.

Tahapan yang dilakukan untuk memahami data adalah.

- Teknik Visualisasi menggunakan matplotlib dan seaborn
- Statistik data menggunkan pandas

Data Preparation

Tahapan yang dilakukan

- **Melakukan EDA 1. Cek Null**

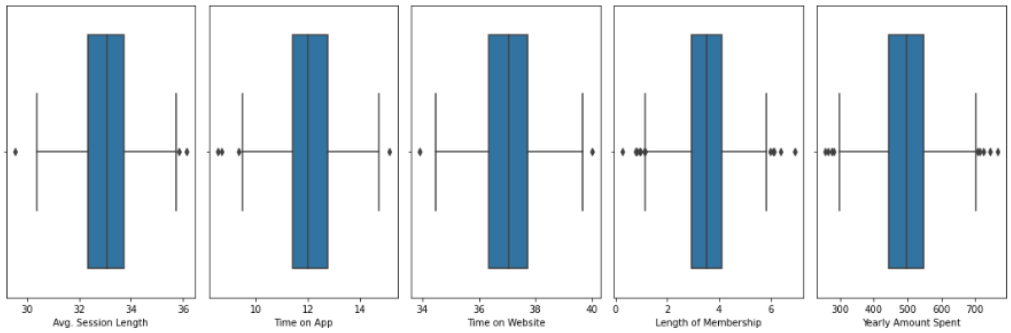
```
In [4]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 500 entries, 0 to 499
Data columns (total 8 columns):
 #   Column                Non-Null Count  Dtype  
---  ------                --------------  -----  
 0   Email                 500 non-null    object 
 1   Address               500 non-null    object 
 2   Avatar                500 non-null    object 
 3   Avg. Session Length   500 non-null    float64
 4   Time on App           500 non-null    float64
 5   Time on Website       500 non-null    float64
 6   Length of Membership  500 non-null    float64
 7   Yearly Amount Spent   500 non-null    float64
dtypes: float64(5), object(3)
memory usage: 31.4+ KB
```

Tidak terdapat data NaN dan Null

2. Cek Outlier

```
fig, axes = plt.subplots(1,5, figsize = (15,5))

for i, ax in zip(df.select_dtypes(include = 'number'), axes.flatten()):
    sns.boxplot(x = df[i], ax = ax)
    plt.tight_layout()
```



Handle Outlier

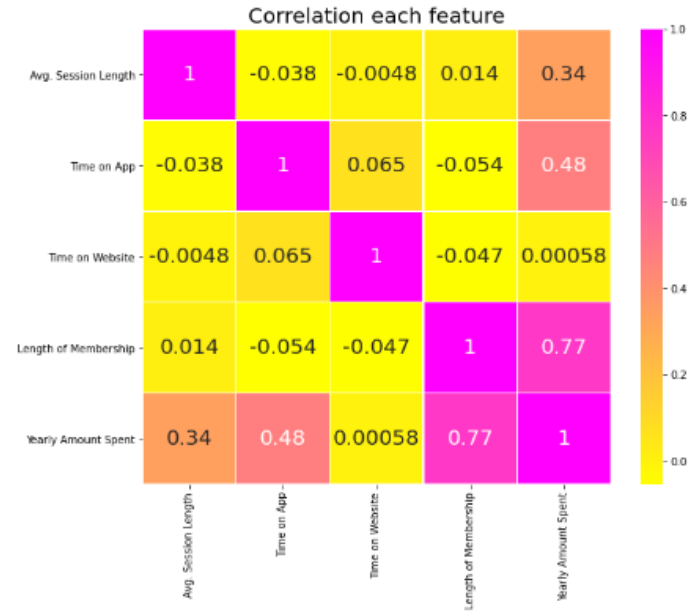
Disini saya tetap menggunakan outlier, meskipun data yang dimiliki sangat kecil. Saya tidak mengganti data pada nilai outliernya. Pada label yang

digunakan, saya akan memprediksi nilai pada kolom Yearly Amount Spent. Menghasilkan total dataset baru sebear 476 baris.

```
Q1 = df.quantile(0.25)
Q3 = df.quantile(0.75)

IQR = Q3 - Q1

df = df[~((df < (Q1 - 1.5 * IQR)) | (df > (Q3 + 1.5 * IQR))).any(axis = 1)]

df.shape
```

3. Korelasi parameter terhadap label

```
plt.figure(figsize=(10,8))
sns.heatmap(df.corr(), annot = True, annot_kws = {'fontsize':20}, cmap = 'spring_r', linewidth = 0.3)
plt.title('Correlation each feature', fontsize = 20)
plt.show()
```



- **Preprocessing 1. Reduction feature** Disini saya tidak menggunakan PCA dikarenakan tidak adanya korelasi yang tinggi antar fitur yang sama. Menurut perkiraan saya, Time On Website dengan Avg Session dapat dilakukan PCA. Namun dengan korelasi yang cukup rendah. Hal tersebut tidak perlu dilakukan dan yang saya gunakan hanyalah korelasi dengan rentang mendekati -1 dan +1

```
X = df[['Avg. Session Length', 'Time on App', 'Length of Membership']]
y = df['Yearly Amount Spent']
```

2. plit data (75%:25%) Untuk pembagian dataset, saya menggunakan 75% (Train) : 25% (Test) karena mengingat dataset yang kecil

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.25, random_state = 250)
print(f'Total of Dataset : {len(X)}')
print(f'Total of Train Dataset : {len(X_train)}')
print(f'Total of Test Dataset : {len(X_test)}')
```

3. Standardization Jenis standadization yang digunakan adalah StandardScaler milik sklearn.

```
kolom = ['Avg. Session Length', 'Time on App', 'Length of Membership']
scaler = StandardScaler()
scaler.fit(X_train[kolom])
X_train[kolom] = scaler.transform(X_train.loc[:, kolom])
X_train[kolom].head()
```

Modeling

Pada proses modeling, model yang digunakan SVR dikarenakan permasalahan regresi dengan jenis kernel linear. Pada tahapan ini terdapat dua tahapan, yaitu tanpa optimasi parameter dan menggunakan optimasi parameter dari kernel linear itu sendiri (nilai C) dengan rentang nilai 1-20. Pada penelitian yang dilakukan Noviana Pratiwi dan Yudi Setyawan berjudul ANALISIS AKURASI DARI PERBEDAAN FUNGSI KERNEL DAN COST PADA SUPPORT VECTOR MACHINE STUDI KASUS KLASIFIKASI CURAH HUJAN DI JAKARTA, menjelaskan bahwa parameter C merupakan parameter untuk mengontrol nilai error yang berpengaruh pada margin yang terbentuk. Tahapan yang dilakukan ialah:

1. Mengimport library SVR dari sklearn dan membuat variable yang berisi SVR

```
#Tanpa Optimasi
model = SVR(kernel = 'linear')
model.fit(X_train, y_train)
```

2. Mengimport library SVR dari sklearn dan membuat variable yang berisi SVR dan optimasi parameter C

```
#Menggunakan Optimasi C
for c in range(1, 21):
models = SVR(kernel = 'linear', C = c)
models.fit(X_train, y_train)
```

Adapun keunggulan dan kekurangan dari model SVR.

Keunggulan

- SVR mampu menghindari overfiting
- SVR efektif untuk menggeneralisasi sampel data yang sedikit
- SVR mampu melakukan penyelesaian norm error pada saat pinalti outlier selama fase pelatihan. Hal ini yang diketahui dengan kernel trick

Kekurangan

- kinerja SVR sangat bergantung terhadap parameter di dalamnya

Evaluation

Evaluasi yang digunakan pada hasil model ialah mean squared error. Alasan mengapa menggunakan metrik tersebut karena permasalahan yang diangkat mengenai regresi. Menurut Iwa Sungkawa dan Ries Tri Megasari pada penelitian PENERAPAN UKURAN KETEPATAN NILAI RAMALAN DATA DERET WAKTU DALAM SELEKSI MODEL PERAMALAN VOLUME PENJUALAN PT SATRIAMANDIRI CITRAMULIA menyatakan bahwa MSE merupakan salah satu model evaluasi terbaik pada masalah regresi. MSE sendiri bekerja melakukan perhitungan error antara nilai hasil prediksi dengan nilai sebesarnya. Berikut formula dari MSE.

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2$$

Hasil MSE yang didapatkan ialah 1. Tanpa optimasi

```
print('Mean Squared Error pada data train : {}'.format(mean_squared_error(y_pred = model.predict(X_train), y_true = y_train)))
```

```
Mean Squared Error pada data train : 109.02462710253819
```

```
print('Mean Squared Error pada data train : {}'.format(mean_squared_error(y_pred = model.predict(X_test), y_true = y_test)))
```

```
Mean Squared Error pada data train : 92.0247400277961
```

2. Menggunakan Optimasi Train

```
Mean Squared Error pada data train : 109.02 (C = 1)
Mean Squared Error pada data train : 104.96 (C = 2)
Mean Squared Error pada data train : 104.43 (C = 3)
Mean Squared Error pada data train : 103.53 (C = 4)
Mean Squared Error pada data train : 103.51 (C = 5)
Mean Squared Error pada data train : 103.43 (C = 6)
Mean Squared Error pada data train : 103.37 (C = 7)
Mean Squared Error pada data train : 103.24 (C = 8)
Mean Squared Error pada data train : 103.22 (C = 9)
Mean Squared Error pada data train : 103.21 (C = 10)
Mean Squared Error pada data train : 103.15 (C = 11)
Mean Squared Error pada data train : 103.08 (C = 12)
Mean Squared Error pada data train : 103.08 (C = 13)
Mean Squared Error pada data train : 103.07 (C = 14)
Mean Squared Error pada data train : 103.07 (C = 15)
Mean Squared Error pada data train : 103.07 (C = 16)
Mean Squared Error pada data train : 103.07 (C = 17)
Mean Squared Error pada data train : 103.07 (C = 18)
Mean Squared Error pada data train : 103.07 (C = 19)
Mean Squared Error pada data train : 103.07 (C = 20)
```

Test

```
Mean Squared Error pada data train : 92.02 (C = 1)
Mean Squared Error pada data train : 91.78 (C = 2)
Mean Squared Error pada data train : 91.72 (C = 3)
Mean Squared Error pada data train : 91.15 (C = 4)
Mean Squared Error pada data train : 91.15 (C = 5)
Mean Squared Error pada data train : 91.18 (C = 6)
Mean Squared Error pada data train : 90.9 (C = 7)
Mean Squared Error pada data train : 90.99 (C = 8)
Mean Squared Error pada data train : 90.88 (C = 9)
Mean Squared Error pada data train : 90.91 (C = 10)
Mean Squared Error pada data train : 91.03 (C = 11)
Mean Squared Error pada data train : 91.07 (C = 12)
Mean Squared Error pada data train : 91.07 (C = 13)
Mean Squared Error pada data train : 91.09 (C = 14)
Mean Squared Error pada data train : 91.09 (C = 15)
Mean Squared Error pada data train : 91.12 (C = 16)
Mean Squared Error pada data train : 91.12 (C = 17)
Mean Squared Error pada data train : 91.12 (C = 18)
Mean Squared Error pada data train : 91.12 (C = 19)
Mean Squared Error pada data train : 91.14 (C = 20)
```

3. Hasil prediksi dengan nilai

	Nilai Rill	Tanpa C	Dengan C
0	492.105052	496.108804	495.381434
1	347.776927	354.102847	349.938340
2	584.105885	579.166078	580.720424

Real

Pada hasil diatas dapat disimpulkan bahwa, kernel linear tanpa nilai C dan menggunakan nilai C hasil MSE tidak cukup berbeda jauh. Namun hasil prediksi yang didapatkan pada index ke-1 pada tanpa nilai C dan menggunakan nilai C mengalami perbedaan yang signifikan sekitar 7%. Hal ini menyatakan bahwa kernel linear dengan permasalahan regresi masih belum cukup baik dan dapat dilakukan percobaan kernel RBF seperti pada penelitian Support Vector Regression (SVR) Dalam Memprediksi Harga Minyak Kelapa Sawit di Indonesia dan Nilai Tukar Mata Uang EUR/USD yang dilakukan oleh Saadah, dkk