Laporan Analisis Pengaruh Feature Engineering Terhadap Performa 5 Model

Achmad Hadzami Setiawan (21/480222/PA/20851),
Anggit Ihsananto (21/477580/PA/20677),
Josiah Farrel Suwito (21/473370/PA/20381),
Muhammad Mahdi (21/473808/PA/20431)

I. Dataset

Dataset yang kami gunakan adalah dataset untuk kasus klasifikasi breast cancer, yang berasal dari kaggle [kaggle.com/datasets] atau di UCI Machine Learning [archive.ics.uci.edu]. Tujuan dari dataset ini adalah mengklasifikasikan sel kanker antara kelas benign (jinak) dan malignant (ganas).

II. Model

Pada eksperimen ini, kami menggunakan 5 jenis model classifier, yaitu Logistic Regression, SVM, KNN, Decision Tree, dan Naive Bayes.

III. Metode

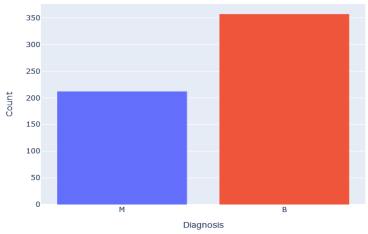
Metode yang kami gunakan untuk menganalisis pengaruh *feature engineering* pada dataset tersebut adalah dengan melakukan analisis data eksploratif terlebih dahulu untuk memahami dataset. Dataset tersebut kemudian dibagi menjadi 80% data training dan 20% data testing. Kemudian dilakukan preprocessing dan kami melakukan *feature engineering* pada data training saja. Setelah itu kami melakukan implementasi 5 model *classifier* pada data training yang telah diproses. Kemudian yang terakhir kami melakukan evaluasi terhadap hasil klasifikasi masing - masing model.

IV. Analisis Data Eksploratif

A. *Deskripsi Data*

Data kami terdiri 569 entri data dan 33 kolom. Kolom/atribut tersebut memiliki tipe data float kecuali pada fitur id yang bertipe int dan diagnosis yang bertipe object, serta satu kolom kosong yang hanya bernilai *N/A*. Sehingga dataset kami gunakan secara efektif hanya memiliki 30 fitur saja.

B. *Distribusi Data*



Data kami terdiri dari 357 *Benign* dan 212 *Malignant*.

C. *Implementasi Preprocessing*

1. Menangani Nilai Null

```
[10] # Cek apakah ada data yang null
     is_null = df.isnull().values.any()
     # Cek apakah ada data yang duplikat
     is_duplicated = df[df.duplicated()].shape[0] > 0
```

2. Menghapus Kolom yang Tidak Diperlukan

   Pada dataset kami terdapat 2 kolom yang tidak diperlukan yaitu *id* dan *Unnamed: 32*.

3. Menangani Tipe Data Non-Numerik

```
[15] df['diagnosis'] = (df['diagnosis'] == 'M').astype(int)
```

   Kami melakukan konversi tipe ke dalam bentuk integer karena hanya mencakup dua nilai yaitu 1 dan 0.

4. Data Original

```
[17] features = df.columns.tolist()
     features.remove("diagnosis")

     X_original = df[features]
     y_original = df["diagnosis"]
```

   Kami memisahkan fitur dengan label. Kemudian kami bagi menjadi data *training* dan data uji.

D. *Feature Engineering*

1. Normalisasi Min Max

```
scaler = MinMaxScaler()

X_min_max = pd.DataFrame(scaler.fit_transform(X_train),
                         columns=X_train.columns)
```

   Kami melakukan min-max scaling dengan memanfaatkan scaler dari scikit-learn yang kemudian kami ubah ke dalam bentuk dataframe.

2. Normalisasi Mean

```
mean = X_train.mean()
std = X_train.std()

X_mean = (X_train - mean) / std

X_mean.describe().T
```

3. Seleksi Fitur dengan *Mean Scaling*

   Sebelum melakukan seleksi fitur, kami melakukan *mean scaling*. Setelah itu, kami melakukan seleksi fitur dengan cara menggunakan nilai korelasi. Kami menggunakan nilai absolut dari korelasi dan hanya fitur yang memiliki nilai absolut lebih dari 0.2 yang kami gunakan.

4. *Principal component analysis (PCA)*

```
pca = PCA(n_components=7)
X_scaled = X_mean.copy()

X_pca = pca.fit_transform(X_scaled)
column_names = [f"PC-{i+1}" for i in range(X_pca.shape[1])]

X_pca = pd.DataFrame(X_pca, columns=column_names)

X_pca.head()
```

   Setelah melakukan percobaan dan kami mendapatkan jumlah komponen yang cukup optimal adalah berjumlah 7, cukup optimal disini artinya tidak terlalu banyak tetapi cukup representatif.

V. Pembahasan

Pada sebelum *training*, kami juga melakukan pemrosesan pada data uji. Setelah itu, kami membuat fungsi utilitas untuk memudahkan proses training dan evaluasi. Setelah melakukan *running,* didapatkan hasil sebagai berikut:

1. Pada model *Decision Tree*, hasil terbaik diperoleh dengan menggunakan metode PCA, dengan akurasi sebesar 0.956140 dan *F1 score* sebesar 0.942529
2. Pada model *Gaussian Naive Bayes*, hasil terbaik diperoleh dengan menggunakan data asli, dengan akurasi sebesar 0.973684 dan *F1 score* sebesar 0.963855.
3. Pada model *K-Nearest Neighbors* (KNN), hasil terbaik diperoleh dengan menggunakan metode feature selection (FS), dengan akurasi sebesar 0.964912 dan *F1 score* sebesar 0.953488. Metode *feature engineering* lainnya memberikan performa yang hampir sebanding.
4. Pada model *Logistic Regression*, hasil terbaik diperoleh dengan menggunakan metode normalisasi *mean*, dengan akurasi sebesar 0.982456 dan *F1 score* sebesar 0.97619. Metode *feature engineering* lainnya memberikan performa yang sedikit lebih rendah.
5. Pada model *Support Vector Machine Classifier* (SVMC), hasil terbaik diperoleh dengan menggunakan metode normalisasi *mean*, dengan akurasi sebesar 0.982456 dan *F1 score* sebesar 0.97619. Metode *feature engineering* lainnya memberikan performa yang sedikit lebih rendah.

	Accuracy	F1 Score
FS	0.9578946	0.9428488
Mean	0.9578946	0.9422728
Default	0.9543856	0.9372852
PCA	0.9403508	0.918734
Min-max	0.8999998	0.8750332

Tabel 1 Rata-rata nilai metrik *accuracy* dan *F1 Score dari kelima model*.

Tabel tersebut menunjukkan bahwa dalam dataset breast cancer ini, metode *feature selection* (FS) memiliki rata-rata akurasi dan *F1 score* yang paling tinggi dari metode lainnya. Metode normalisasi *mean*, *default* dan PCA memiliki performa yang hampir sebanding dengan FS, sedangkan metode min-max memiliki rata-rata yang sedikit lebih rendah dari metode lainnya.

VI. Kesimpulan

Dari keseluruhan analisis, dapat disimpulkan bahwa penggunaan *feature engineering* dapat mempengaruhi performa model. Metode yang paling optimal akan bergantung pada jenis model dan *dataset* yang digunakan. Pada *dataset breast cancer* ini, metode *feature selection* (FS) dan normalisasi *mean* memberikan hasil yang baik pada sebagian besar model yang dievaluasi.

Hal ini dapat terjadi karena metode FS dan metode normalisasi mean memiliki kelebihan, yaitu FS membantu mengurangi dimensi data dengan memilih *subset* fitur yang paling informatif sehingga model dapat fokus pada fitur yang memiliki dampak lebih besar terhadap klasifikasi. Sedangkan normalisasi mean membantu menghilangkan perbedaan skala antara fitur-fitur yang memiliki rentang nilai yang berbeda. Dalam banyak model *machine learning*, perbedaan skala dapat mempengaruhi bobot dan pengaruh relatif dari setiap fitur.

VII. Lampiran

Tautan Notebook: colab.research.google.com
Tautan Dataset: drive.google.com

VIII. Referensi

https://stats.oarc.ucla.edu/spss/seminars/efa-spss/