

Nama : Yoga Patangga Balapradhana
NIM : 1103194138

Penjelasan Regression Task

Data yang digunakan pada Regression Task ini adalah data RegresiUTSTelkom.csv yang disediakan oleh Teaching Assistant. Mula-mula file csv tersebut di-copy terlebih dahulu ke dalam Google Drive, untuk kemudian dapat dibaca dengan menggunakan baris perintah:

```
# Menghubungkan Google Colab dengan Google Drive
from google.colab import drive

# Mount Google Drive ke Colab
drive.mount('/content/drive')
file_path = '/content/drive/MyDrive/RegresiUTSTelkom.csv'
df = pd.read_csv(file_path)
print(df.head())
```

Setelah itu dilakukan eksplorasi data yang sudah dimuat tersebut menggunakan perintah:

```
print(df.shape)
print(df.info())
X = df.drop("2001", axis=1) # Features
y = df["2001"] # Target

(515344, 91)
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 515344 entries, 0 to 515343
Data columns (total 91 columns):
```

Sehingga dapat diketahui data tersebut memiliki 515.344 baris dan 91 kolom dengan tipe data integer pada kolom pertama (2001) dan tipe data float pada kolom sisanya. Kemudian setelah melihat data statistik dengan menggunakan perintah `df.describe()`, maka dapat disimpulkan kolom pertama adalah data tahun (dari tahun 1922 hingga tahun 2011) sehingga kolom tersebut dipilih sebagai target dan kolom sisanya sebagai features untuk diproses lebih lanjut.

Langkah selanjutnya adalah membagi dataset menjadi data latih dan data uji dengan persentase 70:30 menggunakan perintah `X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3)`. Setelah itu dilakukan seleksi fitur pada data latih sebanyak 3 langkah, yaitu 1) menghapus quasi-constant features, 2) menghapus fitur duplikat, dan 3) menghapus fitur yang memiliki korelasi tinggi.

Setelah dilakukan seleksi fitur, maka langkah selanjutnya adalah membuat model pipeline dengan perintah:

```
model_pipeline = Pipeline(steps=[
    ('scaler', StandardScaler()),
    ('regressor', LinearRegression())
])
```

Nama : Yoga Patangga Balapradhana
NIM : 1103194138

Pipeline ini nantinya akan menyesuaikan dengan model yang akan digunakan, misal baris di atas untuk Linear Regression. Jika model berikutnya akan menggunakan Decision Tree, maka baris (`'regressor', LinearRegression()`) dapat diganti dengan (`'regressor', DecisionTreeRegressor()`), demikian juga halnya dengan metode yang lainnya. Setelah itu, model dapat dilatih dan dilakukan prediksi pada data uji menggunakan perintah:

```
model_pipeline.fit(X_train, y_train)
y_pred = model_pipeline.predict(X_test)
```

Langkah terakhir adalah menggambar scatter plot untuk memvisualisasikan data aktual dan data prediksi dengan menggunakan matplotlib dan menghitung matriks evaluasi RMSE, MSE, dan R^2 dengan perintah:

```
rmse = root_mean_squared_error(y_test, y_pred)
mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)
print(f"RMSE: {rmse:.4f}")
print(f"MSE: {mse:.4f}")
print(f"RSquared: {r2:.4f}")
```

MSE atau Mean Squared Error adalah ukuran yang digunakan untuk mengevaluasi seberapa baik sebuah model regresi dalam memprediksi nilai target. MSE mengukur rata-rata kuadrat selisih antara nilai prediksi dan nilai aktual, sekaligus memberikan gambaran tentang seberapa besar kesalahan rata-rata model dalam memprediksi nilai. Nilai MSE yang lebih kecil menunjukkan bahwa model memiliki performa yang lebih baik dan kesalahan prediksi yang lebih kecil.

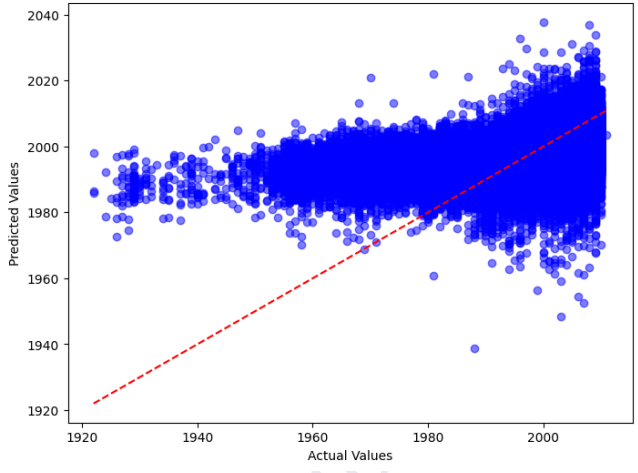
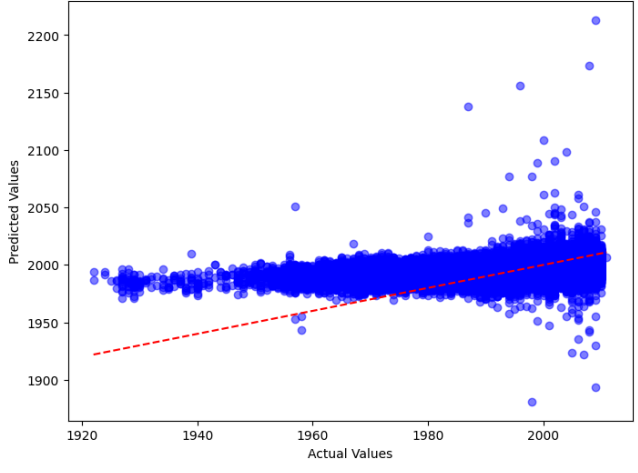
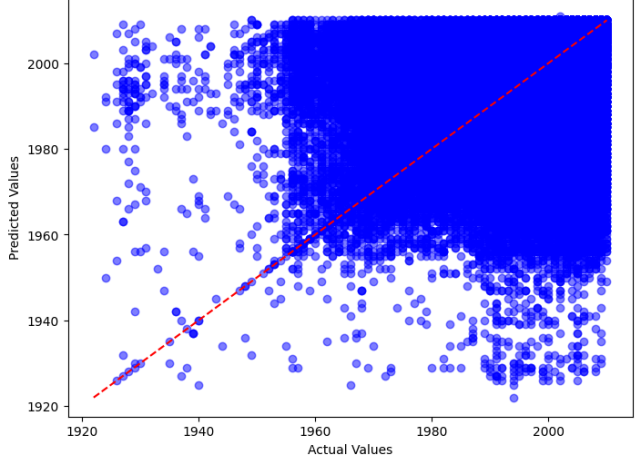
RMSE atau Root Mean Squared Error adalah akar kuadrat dari MSE, yang juga merupakan ukuran yang sering digunakan untuk menilai performa model regresi. RMSE memberikan ukuran kesalahan dalam unit yang sama dengan data asli, sehingga lebih mudah dipahami dibandingkan MSE. Semakin kecil nilai RMSE maka semakin baik model tersebut dalam memprediksi data.

R^2 atau disebut juga Koefisien Determinasi adalah ukuran statistik yang digunakan untuk menilai sejauh mana model regresi menjelaskan variabilitas dalam data. Dengan kata lain, R^2 mengukur seberapa baik model memprediksi nilai target dengan membandingkan variabilitas yang dapat dijelaskan oleh model terhadap variabilitas total data. Nilai R^2 yang lebih tinggi menunjukkan bahwa model lebih baik dalam menjelaskan variasi dalam data.

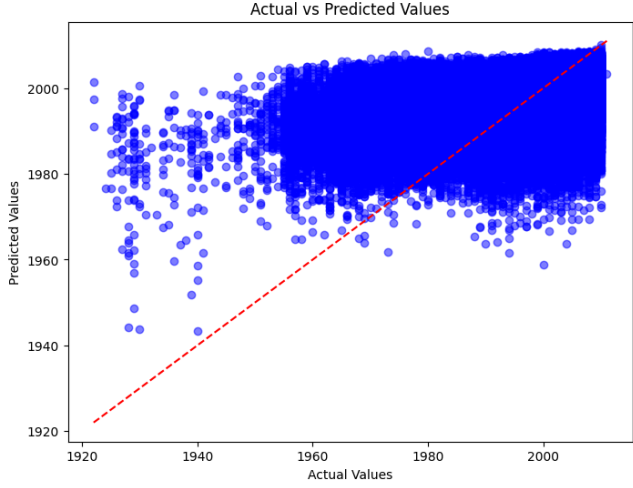
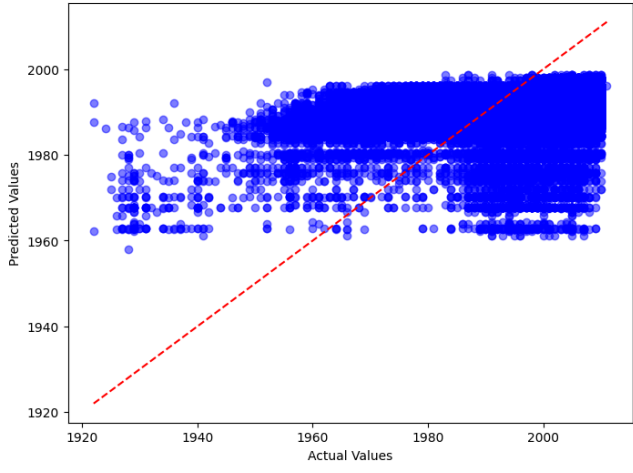
Model yang digunakan dalam Regression Task ini ada 5, yaitu Linear Regression, Polynomial Regression, Decision Tree Regressor, Bagging Regressor, dan AdaBoost Regressor. Perbandingan matriks evaluasi dari kelima model tersebut dapat dilihat pada Tabel 1 di bawah ini.

Nama : Yoga Patangga Balapradhana
NIM : 1103194138

Tabel 1. Perbandingan model berdasarkan matriks evaluasi (RMSE, MSE, R^2)

| Model | RMSE | MSE | R^2 | Visualisasi Actual vs Predicted Value |
|---------------|-------|--------|-------|--|
| Linear | 9,52 | 90,72 | 0,24 |  |
| Polynomial | 9,16 | 83,83 | 0,29 |  |
| Decision Tree | 13,34 | 178,04 | -0,5 |  |

Nama : Yoga Patangga Balapradhana
NIM : 1103194138

| | | | | |
|----------|-------|--------|-------|---|
| Bagging | 9,61 | 92,3 | 0,23 |  |
| AdaBoost | 12,64 | 159,84 | -0,34 |  |

Dari Tabel 1 di atas, dapat disimpulkan bahwa model Polynomial memberikan performa yang paling baik dengan nilai RMSE dan MSE yang terkecil, dan nilai R^2 terbesar dibandingkan keempat model yang lainnya. Selain itu, jika dilihat dari visualisasi actual vs predicted value, model Polynomial juga menunjukkan persebaran data yang relatif rapat dan paling mendekati nilai ideal (garis berwarna merah) dibandingkan keempat model lainnya.

Soal Analisis

1. Jika model linear regression atau decision tree mengalami underfitting pada dataset ini, strategi apa yang akan digunakan untuk meningkatkannya? Bandingkan setidaknya dua pendekatan berbeda (misal: transformasi fitur, penambahan features, atau perubahan model ke algoritma yang lebih kompleks), dan jelaskan bagaimana setiap solusi memengaruhi bias-variance tradeoff.

Penjelasan: Underfitting terjadi ketika model terlalu sederhana sehingga tidak dapat menangkap pola yang ada dalam data. Jika model linear regression atau decision tree mengalami underfitting pada dataset, ada beberapa strategi yang dapat digunakan untuk meningkatkan performanya.

Nama : Yoga Patangga Balapradhana
NIM : 1103194138

Cara pertama adalah dengan menambah fitur baru yang relevan, misalnya fitur yang lebih kompleks atau fitur yang dihasilkan dari fitur yang ada. Namun ada kalanya fitur yang ada mungkin tidak linear atau tidak cukup untuk menggambarkan hubungan yang ada antara input dan output. Dalam hal ini, dapat dilakukan transformasi fitur seperti logaritma, akar kuadrat, atau fitur polinomial sehingga dapat membantu model menangkap hubungan yang lebih kompleks. Misalnya, jika hubungan antara variabel input dan target adalah non-linear, kita bisa menggunakan fitur polinomial untuk mendekati hubungan tersebut.

Cara kedua adalah dengan mengganti model dengan algoritma yang lebih kompleks. Jika model linear regression atau decision tree sederhana masih mengalami underfitting, mengganti model ke algoritma yang lebih kompleks seperti Random Forest atau Gradient Boosting bisa menjadi salah satu solusi dikarenakan algoritma ini dapat menangkap hubungan non-linear yang lebih kompleks dalam data dan mengurangi risiko underfitting.

- 2. Selain MSE, jelaskan dua alternatif loss function untuk masalah regresi (misal: MAE, Huber loss) dan bandingkan keunggulan serta kelemahannya. Dalam skenario apa setiap loss function lebih cocok digunakan? (Contoh: data dengan outlier, distribusi target non-Gaussian, atau kebutuhan interpretasi model).**

Penjelasan: MAE atau Mean Absolute Error mengukur rata-rata dari nilai mutlak selisih antara nilai prediksi dan nilai aktual. Ini adalah salah satu loss function yang paling sederhana dan sering digunakan untuk masalah regresi. MAE memiliki keunggulan yaitu tidak terlalu sensitif terhadap outlier dikarenakan MAE menggunakan nilai absolut, yang berarti tidak ada pembesaran kesalahan seperti pada MSE. Namun, MAE memiliki kelemahan yaitu kurang stabil saat optimasi karena MAE menghasilkan gradien yang konstan untuk kesalahan prediksi yang dapat menyebabkan kesulitan dalam beberapa algoritma optimasi. Hal ini dapat menyebabkan model lebih sulit untuk mencapai nilai yang konvergen. Dengan keunggulan dan kelemahan ini, MAE lebih cocok digunakan untuk data dengan banyak nilai outlier, serta data dengan distribusi target yang tidak simetris atau tidak memiliki pola Gaussian.

Sedangkan Huber Loss adalah fungsi loss yang menggabungkan keuntungan dari MSE dan MAE. Huber Loss memberikan penalti kuadrat untuk kesalahan kecil (seperti MSE) dan penalti absolut untuk kesalahan besar (seperti MAE). Dengan kata lain, Huber Loss dapat menghindari penalti yang besar terhadap outlier seperti MSE, tetapi tetap sensitif terhadap kesalahan kecil seperti MAE.

- 3. Tanpa mengetahui nama fitur, metode apa yang dapat digunakan untuk mengukur pentingnya setiap fitur dalam model? Jelaskan prinsip teknikal di balik metode tersebut (misal: koefisien regresi, feature importance berdasarkan impurity reduction) serta keterbatasannya.**

Penjelasan: Untuk mengukur pentingnya setiap fitur dalam model tanpa mengetahui nama fitur, kita dapat menggunakan berbagai metode yang dapat menilai kontribusi setiap fitur terhadap performa model. Beberapa metode yang umum digunakan untuk menilai pentingnya fitur dalam model antara lain: Koefisien Regresi, L1 Regularization (Lasso), Feature Importance dengan Decision Tree dan Ensemble Methods, Permutation Feature Importance, dan SHAP (SHapley Additive exPlanations).

Nama : Yoga Patangga Balapradhana
NIM : 1103194138

Koefisien regresi dapat digunakan untuk mengukur pentingnya fitur, yaitu nilai koefisien yang lebih besar menunjukkan pengaruh yang lebih besar dari fitur tersebut terhadap prediksi. Namun, metode ini hanya dapat digunakan untuk model regresi linier atau model yang dapat diinterpretasikan dalam bentuk koefisien linier.

L1 regularization (Lasso) dapat digunakan pada regresi linier untuk melakukan seleksi fitur dengan cara menambahkan penalti terhadap jumlah nilai absolut dari koefisien regresi model. Dengan menambahkan penalti terhadap ukuran koefisien, Lasso mendorong koefisien untuk mendekati nol, dan beberapa koefisien dapat menjadi nol sepenuhnya, sehingga fitur yang tidak penting secara efektif dihapus. Lasso memiliki keterbatasan yaitu hanya dapat digunakan untuk model regresi linier, sehingga tidak cocok untuk model berbasis pohon atau non-linier.

Model Decision Tree dan algoritma Ensemble seperti Random Forest atau Gradient Boosting juga dapat mengukur pentingnya fitur berdasarkan pengurangan impurity. Pada prinsipnya, setiap keputusan dalam Decision Tree didasarkan pada pemilihan fitur terbaik yang memisahkan data dengan cara yang mengurangi ketidakmurnian (impurity). Impurity yang sering digunakan termasuk Gini impurity atau Entropy. Namun, model ini memiliki keterbatasan yaitu tidak dapat digunakan untuk model linier, karena ia hanya relevan untuk model berbasis pohon atau non-linier.

Permutation Feature Importance adalah metode model-agnostik yang mengukur seberapa pentingnya suatu fitur dengan mengacak (permutasi) nilai fitur tersebut dan menghitung penurunan dalam performa model. Jika permutasi suatu fitur menyebabkan penurunan performa yang signifikan, itu berarti fitur tersebut sangat penting. Sebaliknya, jika performa tidak banyak berubah setelah permutasi, fitur tersebut dianggap kurang penting. Metode ini memiliki keterbatasan yaitu memerlukan model yang sudah dilatih sebelumnya sehingga memakan waktu lebih lama tergantung pada ukuran dataset dan kompleksitas modelnya.

SHAP adalah metode yang didasarkan pada teori permainan dan memberikan penilaian yang lebih adil mengenai kontribusi setiap fitur dalam prediksi model. SHAP menghitung kontribusi marginal dari setiap fitur terhadap prediksi dengan cara membandingkan prediksi model saat fitur tersebut ada atau tidak ada dalam model. Metode ini memiliki keterbatasan yaitu bisa sangat mahal secara komputasi, terutama untuk model yang besar atau dataset yang sangat besar.

4. Bagaimana mendesain eksperimen untuk memilih hyperparameter optimal (misal: learning rate untuk SGDRegressor, max_depth untuk Decision Tree) pada dataset ini? Sertakan analisis tradeoff antara komputasi, stabilitas pelatihan, dan generalisasi model.

Penjelasan: Desain eksperimen untuk memilih hyperparameter optimal (misalnya, learning rate untuk SGDRegressor atau max_depth untuk Decision Tree) pada suatu dataset dapat dilakukan dengan menggunakan beberapa teknik pencarian seperti Grid Search, Random Search, atau Bayesian Optimization.

Metode Grid Search mencoba semua kombinasi dari hyperparameter yang telah ditentukan dalam grid pencarian, sehingga metode ini adalah pendekatan yang sangat sistematis dan menyeluruh. Metode ini dapat memberikan solusi yang sangat komprehensif untuk

menemukan kombinasi hyperparameter yang optimal dan cocok untuk ruang pencarian hyperparameter yang terbatas. Namun, jika ada banyak hyperparameter dan banyak nilai untuk masing-masing hyperparameter, pencarian grid dapat menjadi sangat lambat dan dapat terjadi overfitting pada set parameter tertentu.

Random Search memilih kombinasi hyperparameter secara acak dari ruang pencarian yang telah ditentukan, sehingga lebih efisien dibandingkan grid search terutama jika ada banyak parameter atau ruang pencarian yang besar. Namun, metode ini bisa jadi tidak memberikan solusi yang optimal karena tidak mencoba semua kombinasi.

Bayesian Optimization adalah metode yang lebih canggih untuk memilih hyperparameter dengan memanfaatkan probabilitas dan fungsi akuisisi. Metode ini sangat efisien dalam mencari hyperparameter optimal dengan sedikit percobaan sehingga cocok untuk ruang pencarian yang besar. Namun, pada praktiknya implementasinya lebih kompleks karena memerlukan perangkat tambahan dan tidak selalu tersedia secara langsung dalam semua library machine learning.

5. Jika menggunakan model linear regression dan residual plot menunjukkan pola non-linear serta heteroskedastisitas, langkah-langkah apa yang akan diambil? (contohnya: Transformasi data/ubah model yang akan dipakai/etc)

Penjelasan: Jika residual plot pada model linear regression menunjukkan pola non-linear serta heteroskedastisitas, artinya model linear yang digunakan tidak dapat menangkap hubungan non-linear antara fitur dan target serta terdapat variasi error yang tidak konsisten pada seluruh rentang prediksi. Untuk mengatasi hal ini, transformasi data yang umum dilakukan antara lain: Log Transformation, Square Root Transformation, atau Polynomial Transformation.

Selain itu, jika transformasi data tidak cukup untuk mengatasi masalah non-linear, kita mungkin perlu beralih ke model yang lebih kompleks dan dapat menangkap hubungan non-linear. Beberapa pilihan model non-linear yang bisa dicoba antara lain: Decision Tree Regressor, Random Forest Regressor, Support Vector Regression (SVR), atau Neural Networks.