PRAKTIKUM MACHINE LEARNING

UNIT 7 : BUILDING MODEL (BAGIAN I)

adult_klasifikasi.ipynb

Eka Praja Wiyata Mandala, S.Kom, M,Kom, CADS

Unit 7 : Building Model

Unit 7: Membangun Model

Tujuan: Membuat dan melatih model machine learning untuk memprediksi income berdasarkan dataset yang telah dipersiapkan.

Persiapan Data

```
from sklearn.model_selection import train_test_split
# Pisahkan fitur dan target
X = df.drop('income', axis=1)
y = df['income']

# Split data
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42, stratify=y)
```

Unit 7 : Building Model

Membangun dengan Decision Tree

```
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score, recall_score,precision_score,
classification_report, confusion_matrix

# Inisialisasi model
dt_model = DecisionTreeClassifier(random_state=42)

# Melatih model
dt_model.fit(X_train, y_train)

# Prediksi
y_pred_dt = dt_model.predict(X_test)
```

Unit 7 : Building Model


```python
# Evaluasi
print("Decision Tree Performance:")
print("Accuracy:", accuracy_score(y_test, y_pred_dt))
print("Precision:", precision_score(y_test,y_pred_dt, average='micro'))
print('Recall:', recall_score(y_test,y_pred_dt, average='micro'))
print("\nClassification Report:")
print(classification_report(y_test, y_pred_dt))
```

Kriteria:

- berapapun hasil **Accuracy**, **Precision**, **Recall**, selama nilainya di range yang sama (yaitu **0.771301138111350 4**), menunjukkan bahwa **preprocessing** yang kita gunakan **sudah benar dan sesuai**

Unit 7 : Building Model


```
sns.heatmap(confusion_matrix(y_test,y_pred_dt),annot=True,cmap='viridis', fmt='.
0f')
plt.xlabel('Predicted Values', fontdict={'size':14}, labelpad=10)
plt.ylabel('Actual Values', fontdict={'size':14}, labelpad=10)
plt.title('Confusion Matrix pada bagian testing untuk data asli')
plt.show()
```

komponen-komponennya:

Sumbu X: Predicted Values (Nilai Prediksi)

- 0: Model memprediksi income ≤50K
- 1: Model memprediksi income >50K

Sumbu Y: Actual Values (Nilai Sebenarnya)

- 0: Income sebenarnya ≤50K
- 1: Income sebenarnya >50K

Unit 7 : Building Model

Nilai dalam setiap sel:

- Kiri Atas (4147): True Negatives (TN) - Benar diprediksi ≤50K
- Kanan Atas (788): False Positives (FP) - Salah diprediksi >50K
- Kiri Bawah (699): False Negatives (FN) - Salah diprediksi ≤50K
- Kanan Bawah (868): True Positives (TP) - Benar diprediksi >50K

Interpretasi:

- Akurasi: Model benar dalam (4147 + 868) / (4147 + 788 + 699 + 868) = 77.1% kasus.
- Presisi untuk >50K: 868 / (788 + 868) = 52.4% dari prediksi >50K benar.
- Recall untuk >50K: 868 / (699 + 868) = 55.4% dari actual >50K terdeteksi.
- Model cenderung lebih baik dalam memprediksi income ≤50K (4147 benar vs 788 salah).
- Ada jumlah signifikan false positives (788) dan false negatives (699), menunjukkan ada ruang untuk peningkatan.

Unit 7 : Building Model

Kesimpulan:

- Model memiliki performa cukup baik dalam mengidentifikasi income ≤50K. Namun, model mengalami kesulitan dalam memprediksi income >50K dengan akurat.
- Ada keseimbangan yang cukup baik antara false positives dan false negatives, menunjukkan model tidak terlalu bias ke salah satu kelas.
- Untuk meningkatkan model, fokus mungkin perlu diberikan pada fitur-fitur yang lebih baik membedakan antara dua kategori income, terutama untuk kasus income >50K.

Unit 7 : Building Model

Feature Importance

```
feature_importance = pd.DataFrame({'feature': X.columns, 'importance': dt_model.
feature_importances_})
feature_importance = feature_importance.sort_values('importance',
ascending=False)

plt.figure(figsize=(10,6))
plt.bar(feature_importance['feature'][:18], feature_importance['importance']
[:18])
plt.title('Top 18 Feature Importance')
plt.xticks(rotation=45, ha='right')
plt.tight_layout()
plt.show()
```

Unit 7 : Building Model

Hyperparameter Tuning

Hyperparameter adalah parameter yang tidak dipelajari oleh model selama proses pelatihan, melainkan diatur sebelum proses pelatihan dimulai

Hyperparameter tuning adalah proses mencari nilai optimal dari hyperparameter suatu model machine learning untuk meningkatkan kinerja model tersebut.

Unit 7 : Building Model


```python
from sklearn.model_selection import GridSearchCV

# Definisikan parameter grid
param_grid = {
    'criterion': ['gini', 'entropy'],
    'max_depth': [5, 10, 15, 20],
    'min_samples_split': [2, 5, 10],
    'min_samples_leaf': [1, 2, 4]
}

# Inisialisasi GridSearchCV
grid_search = GridSearchCV(DecisionTreeClassifier(random_state=42), param_grid,
cv=5, scoring='accuracy')

# Lakukan pencarian
grid_search.fit(X_train, y_train)

# Tampilkan parameter terbaik
print("Best parameters:", grid_search.best_params_)
print("Best cross-validation score:", grid_search.best_score_)
```

Unit 7 : Building Model


```
# Gunakan model terbaik
best_dt_model = grid_search.best_estimator_

# Prediksi dengan model terbaik
y_pred_best_dt = best_dt_model.predict(X_test)
```

```
# Evaluasi model terbaik
print("\nBest Decision Tree Performance:")
print("Accuracy:", accuracy_score(y_test, y_pred_best_dt))
print("Precision:", precision_score(y_test,y_pred_best_dt, average='micro'))
print('Recall:', recall_score(y_test,y_pred_best_dt, average='micro'))
print("\nClassification Report:")
print(classification_report(y_test, y_pred_best_dt))
```

Unit 7 : Building Model


```
sns.heatmap(confusion_matrix(y_test,y_pred_best_dt),annot=True,cmap='viridis',
fmt='.0f')
plt.xlabel('Predicted Values', fontdict={'size':14}, labelpad=10)
plt.ylabel('Actual Values', fontdict={'size':14}, labelpad=10)
plt.title('Confusion Matrix pada bagian testing untuk data asli')
plt.show()
```

Unit 7 : Building Model

Analisis:

Komponen Confusion Matrix:

- True Negatives (TN): 4453 (kiri atas)
- False Positives (FP): 482 (kanan atas)
- False Negatives (FN): 631 (kiri bawah)
- True Positives (TP): 936 (kanan bawah)

Interpretasi:

- Akurasi: (4453 + 936) / (4453 + 482 + 631 + 936) = 82.9%
- Presisi untuk >50K: 936 / (482 + 936) = 66.0%
- Recall untuk >50K: 936 / (631 + 936) = 59.7%

Unit 7 : Building Model

Perbedaan dengan gambar sebelumnya:

- TN meningkat: 4453 vs 4147 sebelumnya
- FP menurun: 482 vs 788 sebelumnya
- FN sedikit menurun: 631 vs 699 sebelumnya
- TP meningkat: 936 vs 868 sebelumnya

Kesimpulan:

- Model ini memiliki performa yang lebih baik dibandingkan model sebelumnya.
- Akurasi meningkat dari 77.1% menjadi 82.9%.
- Presisi dan recall untuk kelas >50K juga meningkat.
- Model ini lebih baik dalam mengurangi false positives.

Unit 7 : Building Model


```
feature_importance = pd.DataFrame({'feature': X.columns, 'importance':
best_dt_model.feature_importances_})
feature_importance = feature_importance.sort_values('importance',
ascending=False)

plt.figure(figsize=(10,6))
plt.bar(feature_importance['feature'][:18], feature_importance['importance']
[:18])
plt.title('Top 18 Feature Importance')
plt.xticks(rotation=45, ha='right')
plt.tight_layout()
plt.show()
```

Unit 7 : Building Model

Simpan Model Terbaik

```
[ ]  import joblib

     # Simpan model terbaik
     joblib.dump(best_dt_model, f'{folder_name}/best_income_predictor_model.joblib')
     print("Model terbaik telah disimpan sebagai 'best_income_predictor_model.
     joblib'")
```