Progress TA

1972011 Juan Kenny Filemon

Menggunakan Outlier

```
df = df.drop(columns=['State', 'RaceEthnicityCategory'])
```

Drop state dan ras dari dataframe

```
# Import label encoder
from sklearn import preprocessing
# label_encoder object knows how to understand word labels.
label_encoder = preprocessing.LabelEncoder()
# Encode labels in column 'Sex'.
df['Sex']= label_encoder.fit_transform(df['Sex'])
print(df.head())
```

Label encoding untuk kolom sex

```
binary = {'Yes' : 1, 'No' : 0}

df['PhysicalActivities'] = df['PhysicalActivities'].replace(binary)
df['HadHeartAttack'] = df['HadHeartAttack'].replace(binary)
df['HadAngina'] = df['HadAngina'].replace(binary)
df['HadStroke'] = df['HadStroke'].replace(binary)
df['HadAsthma'] = df['HadAsthma'].replace(binary)
df['HadSkinCancer'] = df['HadSkinCancer'].replace(binary)
df['HadCOPD'] = df['HadCOPD'].replace(binary)
df['HadDepressiveDisorder'] = df['HadDepressiveDisorder'].replace(binary)
df['HadKidneyDisease'] = df['HadKidneyDisease'].replace(binary)
df['HadArthritis'] = df['HadArthritis'].replace(binary)
df['DeafOrHardOfHearing'] = df['DeafOrHardOfHearing'].replace(binary)
df['BlindOrVisionDifficulty'] = df['BlindOrVisionDifficulty'].replace(binary)
df['DifficultyConcentrating'] = df['DifficultyConcentrating'].replace(binary)
df['DifficultyWalking'] = df['DifficultyWalking'].replace(binary)
df['DifficultyDressingBathing'] = df['DifficultyDressingBathing'].replace(binary)
df['DifficultyErrands'] = df['DifficultyErrands'].replace(binary)
df['ChestScan'] = df['ChestScan'].replace(binary)
df['AlcoholDrinkers'] = df['AlcoholDrinkers'].replace(binary)
df['HIVTesting'] = df['HIVTesting'].replace(binary)
df['FluVaxLast12'] = df['FluVaxLast12'].replace(binary)
df['PneumoVaxEver'] = df['PneumoVaxEver'].replace(binary)
df['HighRiskLastYear'] = df['HighRiskLastYear'].replace(binary)
```

Proses mapping untuk kolom binary

ealthDays	LastCheckupTime	PhysicalActivities	SleepHours	RemovedTeeth	HadHeartAttack	HadAngina	HadStroke
0.0	Within past year (anytime less than 12 months ...	1	9.0	None of them	0	0	0
0.0	Within past year (anytime less than 12 months ...	1	6.0	None of them	0	0	0
0.0	Within past year (anytime less than 12 months ...	0	8.0	6 or more, but not all	0	0	0
0.0	Within past year (anytime less than 12 months ...	1	9.0	None of them	0	0	0
15.0	Within past year (anytime less than 12 months ...	1	5.0	1 to 5	0	0	0

Hasil setelah mapping untuk kolom binary

```
mappingGen = {'Excellent' : 4, 'Very good' : 3, 'Good' : 2, 'Fair' : 1, 'Poor' : 0}
df['GeneralHealth'] = df['GeneralHealth'].replace(mappingGen)

mappingLast = {'Within past year (anytime less than 12 months ago)' : 3, 'Within past 2 years (1 year but less than 2 years ago)' : 2, 'Within past 5 yea
df['LastCheckupTime'] = df['LastCheckupTime'].replace(mappingLast)

mappingRemoved = {'None of them' : 3, '1 to 5' : 2, '6 or more, but not all' : 1, 'All' : 0}
df['RemovedTeeth'] = df['RemovedTeeth'].replace(mappingRemoved)

mappingDiabet = {'No' : 3, 'No, pre-diabetes or borderline diabetes' : 2, 'Yes, but only during pregnancy (female)' : 1, 'Yes' : 0}
df['HadDiabetes'] = df['HadDiabetes'].replace(mappingDiabet)

mappingSmoker = {'Never smoked' : 3, 'Former smoker' : 2, 'Current smoker - now smokes some days' : 1, 'Current smoker - now smokes every day' : 0}
df['SmokerStatus'] = df['SmokerStatus'].replace(mappingSmoker)

mappingECigar = {'Never used e-cigarettes in my entire life' : 3, 'Not at all (right now)' : 2, 'Use them some days' : 1, 'Use them every day' : 0}
df['ECigaretteUsage'] = df['ECigaretteUsage'].replace(mappingECigar)

mappingAge = {'Age 80 or older' : 12, 'Age 75 to 79' : 11, 'Age 70 to 74' : 10, 'Age 65 to 69' : 9, 'Age 60 to 64' : 8, 'Age 55 to 59' : 7, 'Age 50 to 54
df['AgeCategory'] = df['AgeCategory'].replace(mappingAge)

mappingTdap = {'Yes, received Tdap' : 3, 'Yes, received tetanus shot but not sure what type' : 2, 'Yes, received tetanus shot, but not Tdap' : 1, 'No, di
df['TetanusLast10Tdap'] = df['TetanusLast10Tdap'].replace(mappingTdap)

mappingCov = {'No' : 2, 'Tested positive using home test without a health professional' : 1, 'Yes' : 0}
df['CovidPos'] = df['CovidPos'].replace(mappingCov)
```

Proses mapping untuk kolom kategorikal

```
df
```

	Sex	GeneralHealth	PhysicalHealthDays	MentalHealthDays	LastCheckupTime	PhysicalActivities	SleepHours	RemovedTeeth	HadHeartAttack	HadAngina	Had
0	0	3	4.0	0.0	3	1	9.0	3	0	0	
1	1	3	0.0	0.0	3	1	6.0	3	0	0	
2	1	3	0.0	0.0	3	0	8.0	1	0	0	
3	0	1	5.0	0.0	3	1	9.0	3	0	0	
4	0	2	3.0	15.0	3	1	5.0	2	0	0	
...	...	...	...	...	...	...	...	...	...	...	
246017	1	3	0.0	0.0	2	1	6.0	3	0	0	
246018	0	1	0.0	7.0	3	1	7.0	3	0	0	
246019	1	2	0.0	15.0	3	1	7.0	2	0	0	
246020	0	4	2.0	2.0	3	1	7.0	3	0	0	
246021	1	3	0.0	0.0	3	0	5.0	3	1	0	

246022 rows × 38 columns

Hasil setelah mapping untuk kolom kategorikal

```
X = df[['Sex', 'GeneralHealth', 'PhysicalHealthDays', 'MentalHealthDays', 'LastCheckupTime', 'PhysicalActivities', 'SleepHours', 'RemovedTeeth', 'HadAngi
y = df[['HadHeartAttack']]

#membagi data training dan data testing
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2, random_state = 0)
```

Split data untuk training dan testing, labelnya yaitu HadHeartAttack

20% data untuk testing dan training 80%

```python
#mengaktifkan package untuk klasifikasi KNN
from sklearn.neighbors import KNeighborsClassifier

#mengaktifkan fungsi klasifikasi untuk knn
knn = KNeighborsClassifier (n_neighbors=2)
```

```python
#memasukkan data training pada fungsi klasifikasi untuk KNN
knn.fit(X_train, y_train)
```

```
C:\Users\Juan\AppData\Local\Programs\Python\Python312\Lib\site-packages\sklearn\neighbors\_classification.py:238: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Please change the shape of y to (n_samples,), for example using ravel().
  return self._fit(X, y)
```

```
KNeighborsClassifier
KNeighborsClassifier(n_neighbors=2)
```

```python
#menentukan prediksi
y_pred = knn.predict (X_test)
y_pred
```

```
array([0, 0, 0, ..., 0, 0, 0], dtype=int64)
```

```python
#menentukan probabilitas prediksi
knn.predict_proba(X_test)
```

```
array([[1., 0.],
       [1., 0.],
       [1., 0.],
       ...,
       [1., 0.],
       [1., 0.],
       [1., 0.]])
```

```python
#import package untuk melihat keakuratan data prediksi dengan data aktual
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix

accuracy = accuracy_score(y_test, y_pred)
print(accuracy)

#menampilkan matriks hasil prediksi
print(confusion_matrix(y_test, y_pred))
```

```
0.9419977644548319
[[46297   166]
 [ 2688    54]]
```

KNearest Neighbor:
Akurasi = 0.94
Confusion Matrix =
TN 46129 FP 334
FN 2426 TP 316

```python
#ketepatan hasil prediksi
print(classification_report(y_test, y_pred))
```

```
              precision    recall  f1-score   support

           0       0.95      1.00      0.97     46463
           1       0.25      0.02      0.04      2742

    accuracy                           0.94     49205
   macro avg       0.60      0.51      0.50     49205
weighted avg       0.91      0.94      0.92     49205
```

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

dt = DecisionTreeClassifier(criterion='gini', max_depth=1, random_state=0)
dt.fit(X_train, y_train)
```

```
DecisionTreeClassifier(max_depth=1, random_state=0)
```

```python
y_pred = dt.predict(X_test)

accuracy = accuracy_score(y_test, y_pred)
print('Accuracy: ', accuracy)

print('Confusion Matrix:')
print(confusion_matrix(y_test, y_pred))

print('Classification Report:')
print(classification_report(y_test, y_pred))
```

```
Accuracy:  0.9442739558987908
Confusion Matrix:
[[46463     0]
 [ 2742     0]]
Classification Report:
              precision    recall  f1-score   support

           0       0.94      1.00      0.97     46463
           1       0.00      0.00      0.00      2742

    accuracy                           0.94     49205
   macro avg       0.47      0.50      0.49     49205
weighted avg       0.89      0.94      0.92     49205
```

Decision Tree:
Akurasi = 0.94
Confusion Matrix =
TN 46463 FP 0
FN 2742 TP 0

```python
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix

rf = RandomForestClassifier(n_estimators=100, criterion='gini', random_state=0)
rf.fit(X_train, y_train)
```

```
C:\Users\Juan\AppData\Local\Programs\Python\Python312\Lib\site-packages\sklearn\base.py:1351: Data
1d array was expected. Please change the shape of y to (n_samples,), for example using ravel().
  return fit_method(estimator, *args, **kwargs)
```

RandomForestClassifier

RandomForestClassifier(random_state=0)

```python
y_pred = rf.predict(X_test)

accuracy = accuracy_score(y_test, y_pred)
print('Accuracy: ', accuracy)

print('Confusion Matrix: ')
print(confusion_matrix(y_test, y_pred))

print('Classification Report: ')
print(classification_report(y_test, y_pred))
```

```
Accuracy:  0.9476475967889442
Confusion Matrix:
[[46157   306]
 [ 2270   472]]
Classification Report:
              precision    recall  f1-score   support

           0       0.95      0.99      0.97     46463
           1       0.61      0.17      0.27      2742

    accuracy                           0.95     49205
   macro avg       0.78      0.58      0.62     49205
weighted avg       0.93      0.95      0.93     49205
```

Random Forest :
Akurasi = 0.95
Confusion Matrix =
TN 46171 FP 292
FN 2299 TP 443

Selanjutnya untuk Hyperparameter Tuning, dan F1 Score, balanced accuracy jika data imbalanced.

Gambar Diagram Kurva ROC, Setelah itu tahap tanpa outlier, dan Lanjut menggunakan Flask untuk pemindahan model kedalam website