Klarifikasi dataset MLBB_Kompetitif

Pada klasifikasi dataset disini bertujuan untuk menganalisis data apa yang akan input dan apa saja yang akan di klasifikasikan dalam proses klasifikasi tersebut contoh pada kasus ini di lakukan klasifikasi nama hero dan juga role dari game mobile legend. Dan pada pengklasifikasian nama hero dan role menunjukkan hasil bahwa akurasi yang didapatkan rendah dikarenakan pemilihan data yang di gunakan adalah imbalance karena jumlah fighter lebih banyak dari jumlah role lainnya sehingga keputusannya lebih besar ke role fighter.

lalu di ikuti dengan proses f1 score model dimana ini berfungsi sebagai ukuran keseimbangan antara precision dan recal pada model klasifikasi. Dengan kata lain, menilai kemampuan model untuk mengindentifikasi hasil positif secara akurat (presisi) sekaligus memastikan model tidak melewatkan hasil positif yang sebenarnya ada (recall).

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
# Install openpyxl dan pandas
!pip install openpyxl
!pip install pandas
# PURPOSE: to prevent any plot open in a new tab
%matplotlib inline
pd.pandas.set_option('display.max_column', None)
```

```
sumberdata= pd.read_csv('/content/MLBB_kompetitif.csv')
print(sumberdata.shape)
sumberdata.head() # menampilkan data
```

Visualisasi jumlah role (Utk mebuktikan imbalance data)
from matplotlib import pyplot as plt
import seaborn as sns
sumberdata.groupby('role').size().plot(kind='barh',
color=sns.palettes.mpl_palette('Dark2'))
plt.gca().spines[['top', 'right',]].set_visible(True)

Klasifikasi nama hero dan role

```
# Memprediksi label untuk data uji
y_pred = model.predict(x_test)
# Menghitung akurasi model
accuracy = accuracy_score(y_test, y_pred)
print("Accuracy:", accuracy)
datapersen = accuracy*100
print(f"Dalam Persen= {datapersen:.2f}")
# Membuat confusion matrix
conf_matrix = confusion_matrix(y_test, y_pred)
print("\nConfusion Matrix:")
print(conf_matrix)
```

Akurasi sebesar 38 persen, hal ini dikarenakan inbalnce data yang ada

```
import matplotlib.pyplot as plot
import seaborn as sns
# Membuat confusion matrix
conf_matrix = confusion_matrix(y_test, y_pred)
# Membuat visualisasi confusion matrix
plot.figure(figsize=(8, 6))
sns.heatmap(conf_matrix, annot=True, fmt='d', cmap='Blues',
xticklabels=model.classes_, yticklabels=model.classes_)
plot.xlabel('Predicted Labels')
plot.ylabel('True Labels')
plot.title('Confusion Matrix')
plot.show()
```

```
from sklearn.metrics import f1_score
f1 = f1_score(y_test, y_pred, average='weighted')
print(f" NILAI F1-score  {f1:.2f}")
```

Berdasarkan F1 score model yang dibangun memiliki kinerja buruk

NILAI PRESISI

```
from sklearn.metrics import precision_score
# Menghitung presisi
presisi = precision_score(y_test, y_pred, average='weighted')
print(f" NILAI presisi  {presisi:.2f}")
```

Hanya sekitar 15 persen dari data yang benar- benar memklasifikasikan

---

Mencari Recall

```
from sklearn.metrics import recall_score
# Menghitung recall
recall = recall_score(y_test, y_pred, average='weighted')
print(f" NILAI recall  {recall:.2f}")
```

***Sistem ***

---

keyboard_arrow_down

Regression (Linear Regression) [Fatih]

Regression Regression in machine learning berfungsi untuk memodelkan hubungan antara variabel terikat (dependent variable) dan satu atau lebih variabel bebas (independent variable). Dengan kata lain, regresi bertujuan untuk memprediksi nilai numerik kontinu berdasarkan fitur-fitur input.
pada kasus kali ini adalah menggunakan dataset yang berisi nama,role,win rate,pick rate,ban rate, dan release rate. Pada saat melakukan regression menggunkan teknik Regresi linear yaitu untuk memodelkan hubungan antara variabel dependen (biasanya disebut sebagai variabel target) dan satu atau lebih variabel independen (biasanya disebut sebagai fitur atau prediktor) dalam bentuk garis lurus. Dalam regresi linear, model matematis yang digunakan adalah suatu garis lurus, yang dinyatakan dalam rumus y = mx + c. Di sini, y adalah variabel dependen yang ingin diprediksi, x adalah variabel independen atau fitur, m adalah kemiringan (slope) garis, dan c adalah perpotongan garis dengan sumbu y (intercept).

keyboard_arrow_down

Install dan Import library

```
!pip install hvplot
```

```
import pandas as pd
import numpy as np
```

```
import matplotlib.pyplot as plt
import seaborn as sns
import hvplot.pandas

from sklearn.model_selection import train_test_split

from sklearn import metrics

from sklearn.linear_model import LinearRegression

%matplotlib inline
```

Pengenalan dataset dan eksplorasi data awal

```
#df =
pd.read_csv('/content/drive/MyDrive/Stechoq-ML/Kelompok/MLBB_kompetitif
.csv')
url =
'https://raw.githubusercontent.com/antonwijayacakra/MACHINE-Learning-St
eqhoc/master/Anton%20Wijaya_ML_%20Tugas%20Individu%20I/MLBB_kompetitif.
csv'
df = pd.read_csv(url)
```

```
df.head()
```

```
df.shape
```

```
df.info()
```

```
df.describe()
```

```
df.isna().sum()
```

```
from sklearn.preprocessing import LabelEncoder
import pandas as pd

label_encoder = LabelEncoder()
df['role_Encoded'] = label_encoder.fit_transform(df['role'])
```

df.drop(['role'], axis=1, inplace=True)

```
df.drop(labels='nama_hero', axis=1, inplace=True)
```

[ ] df.corr()

```
sns.heatmap(df.corr(), annot=True,cmap='Reds')
```

Split Train Test

```
X=df.drop('win_rate', axis=1)

y=df['win_rate']

print("X=",X.shape,"\ny=", y.shape)

X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.3, random_state=105)

X_test.shape
```

Linear Regression

```
model = LinearRegression()
model.fit(X_train, y_train)
```

Model Evaluation

```
model.coef_

pd.DataFrame(model.coef_, X.columns, columns=['Coedicients'])
```

Prediction

```
y_pred = model.predict(X_test)
```

Regression Evaluation Matrics

```
MAE= metrics.mean_absolute_error(y_test, y_pred)
MAPE = metrics.mean_absolute_percentage_error(y_test,y_pred)
MSE= metrics.mean_squared_error(y_test, y_pred)
RMSE= np.sqrt(MSE)
```

```
MAE
```

```
MAPE
```

```
MSE
```

```
RMSE
```

```
df['win_rate'].mean()
```

Clustering setiap role hero beserta namanya

Clustering disini digunakan untuk mengelompokkan data yang memiliki karakteristik atau atribut yang serupa. Dengan kata lain, clustering bertujuan untuk menemukan struktur atau pola tersembunyi dalam kumpulan data yang tidak berlabel. Tidak seperti klasifikasi yang menggunakan label yang sudah ada untuk mengelompokkan data, clustering tidak memerlukan label tersebut.

Pada Clustering ini digunakan algoritma K-Means Clustering adalah untuk memisahkan data menjadi sejumlah k kelompok atau klaster berdasarkan karakteristik yang mereka miliki. Cara kerja K-Means dimulai dengan menginisialisasi pusat klaster secara acak untuk setiap kelompok.
Dengan dataset yang tadi telah tersedia maka dilakukan Clustering dengan mengelompokkan hero menjadi 6 kelas yang sesuai dengan role hero tersebut. seperti contoh pada cluster 1 maka akan di isi dengan role hero fighter dan cluster 2 diisi dengan role hero support.
Hasil akhir yang di dapatkan setelah dilakukannya penggelompokan pada setiap hero dan role hero maka dihasilkan bahwa cluster hero yang memiliki role fighter memiliki jumlah lebih banyak sedangkan cluster hero yang memiliki role support memiliki jumlah yang paling sedikit dari cluster hero-hero lainnya.

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

```
from sklearn.cluster import KMeans
# Install openpyxl dan pandas
!pip install openpyxl
!pip install pandas
# PURPOSE: to prevent any plot open in a new tab
%matplotlib inline
pd.pandas.set_option('display.max_column', None)
sumberdata= pd.read_csv('/content/MLBB_kompetitif.csv')
print(sumberdata.shape)
sumberdata.head() # menampilkan data
```

Menampilkan nama hero sesuai dengan clusternya (rolenya)

```
data = pd.read_csv('/content/MLBB_kompetitif.csv')
X = data[['nama_hero', 'role']]
# Konversi role hero menjadi angka menggunakan one-hot encoding
X_encoded = pd.get_dummies(X)
# Jumlah cluster yang diinginkan adalah 6 kelas
kelas = 6
# Membuat model K-means
kmeans = KMeans(n_clusters=kelas, random_state=100)
# Melakukan clustering
kmeans.fit(X_encoded)
# Mendapatkan label cluster untuk setiap data
labels = kmeans.labels_
# Menambahkan label cluster ke dalam dataframe
data['cluster'] = labels
# Menampilkan hasil clustering
for cluster in range(kelas):
    heroes_in_cluster = data[data['cluster'] == cluster]['nama_hero']
    print(f"Cluster {cluster}: {heroes_in_cluster.values}")
```

Penjelasan tiap cluster

```
for cluster in range(kelas):
    heroes_in_cluster = data[data['cluster'] ==
cluster]['role'].unique()
    print(f"Cluster {cluster}: {heroes_in_cluster}")
```

VISUALISASI DATA

```
plt.figure(figsize=(10, 6))
sns.countplot(x='cluster', data=data, palette='viridis')
plt.xlabel('Cluster')
plt.ylabel('Jumlah Hero')
plt.title('Jumlah Hero dalam Setiap Cluster')
plt.show()
```

Cluster fighter memiliki jumlah hero paling banyak

Cluster support memiliki jumlah hero paling sedikit