

Import Dataframe

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

from google.colab import drive
drive.mount('/content/drive')

Mounted at /content/drive
```

Load and Explore Data

Earthquakes Data

```
eq=pd.read_csv('/content/drive/My Drive/Data_science/Project/earthquake_data.csv')
eq
```

	title	magnitude	date_time	cdi	mmi	alert	tsunami	sig	net	nst	dmin
0	M 7.0 - 18 km SW of Malango, Solomon Islands	7.0	22-11-2022 02:03	8	7	green	1	768	us	117	0.509
1	M 6.9 - 204 km SW of Bengkulu, Indonesia	6.9	18-11-2022 13:37	4	4	green	0	735	us	99	2.229
2	M 7.0 -	7.0	12-11-2022 07:09	3	3	green	1	755	us	147	3.125
3	M 7.3 - 205 km ESE of Neiafu, Tonga	7.3	11-11-2022 10:48	5	5	green	1	833	us	149	1.865
4	M 6.6 -	6.6	09-11-2022 10:14	0	2	green	1	670	us	131	4.998
...
777	M 7.7 - 28 km SSW of Puerto El Triunfo	7.7	13-01-2001 17:22	0	8	NaN	0	912	us	427	0.000

Next steps:

 [View recommended plots](#)

```
eq.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 782 entries, 0 to 781
Data columns (total 19 columns):
#   Column      Non-Null Count  Dtype
---  ---
0   title       782 non-null    object
1   magnitude   782 non-null    float64
2   date_time   782 non-null    object
3   cdi         782 non-null    int64
4   mmi         782 non-null    int64
5   alert       415 non-null    object
6   tsunami     782 non-null    int64
7   sig         782 non-null    int64
8   net         782 non-null    object
9   nst         782 non-null    int64
10  dmin        782 non-null    float64
11  gap         782 non-null    float64
12  magType     782 non-null    object
13  depth       782 non-null    float64
14  latitude    782 non-null    float64
15  longitude   782 non-null    float64
16  location    777 non-null    object
```

```

17 continent    206 non-null    object
18 country      484 non-null    object
dtypes: float64(6), int64(5), object(8)
memory usage: 116.2+ KB

```

```
eq.shape
```

```
(782, 19)
```

✓ Preprocess Data

แปลงข้อมูลจากข้อความเป็นตัวเลข

```
from sklearn.preprocessing import LabelEncoder
```

```

le_alert=LabelEncoder()
le_net=LabelEncoder()
le_magType=LabelEncoder()
le_location=LabelEncoder()
le_continent=LabelEncoder()
le_country=LabelEncoder()

```

#แปลงข้อมูล

```

alert_en=le_alert.fit_transform(eq['alert'])
net_en=le_net.fit_transform(eq['net'])
magType_en=le_magType.fit_transform(eq['magType'])
location_en=le_location.fit_transform(eq['location'])
continent_en=le_continent.fit_transform(eq['continent'])
country_en=le_country.fit_transform(eq['country'])

```

#ใส่ข้อมูลเข้าไปใหม่ในตาราง

```

eq['alert_en']=alert_en
eq['net_en']=net_en
eq['magType_en']=magType_en
eq['location_en']=location_en
eq['continent_en']=continent_en
eq['country_en']=country_en

```

```
eq.info()
```

```

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 782 entries, 0 to 781
Data columns (total 25 columns):
 #   Column                Non-Null Count  Dtype  
---  --
 0   title                 782 non-null   object  
 1   magnitude             782 non-null   float64 
 2   date_time             782 non-null   object  
 3   cdi                   782 non-null   int64   
 4   mmi                   782 non-null   int64   
 5   alert                 415 non-null   object  
 6   tsunami              782 non-null   int64   
 7   sig                   782 non-null   int64   
 8   net                   782 non-null   object  
 9   nst                   782 non-null   int64   
10   dmin                  782 non-null   float64 
11   gap                   782 non-null   float64 
12   magType               782 non-null   object  
13   depth                 782 non-null   float64 
14   latitude              782 non-null   float64 
15   longitude             782 non-null   float64 
16   location              777 non-null   object  
17   continent             206 non-null   object  
18   country               484 non-null   object  
19   alert_en              782 non-null   int64   
20   net_en                782 non-null   int64   
21   magType_en           782 non-null   int64   
22   location_en           782 non-null   int64   
23   continent_en          782 non-null   int64   
24   country_en            782 non-null   int64   
dtypes: float64(6), int64(11), object(8)
memory usage: 152.9+ KB

```

✓ Feature Importance

```
import pandas as pd
from sklearn.ensemble import RandomForestClassifier
# แบ่งข้อมูลเป็น features และ target
X = eq.drop(['title', 'date_time', 'alert', 'net', 'magType', 'location', 'continent', 'country', 'tsunami'], axis=1)
y = eq['tsunami']
```

```
# สร้างและฝึกโมเดล RandomForestClassifier
rf_classifier = RandomForestClassifier(random_state=42)
rf_classifier.fit(X, y)
```

```
RandomForestClassifier
RandomForestClassifier(random_state=42)
```

```
# หาค่าความสำคัญของคุณสมบัตินี้
feature_importances = rf_classifier.feature_importances_
```

```
# สร้าง DataFrame เพื่อแสดงค่าความสำคัญของคุณสมบัตินี้
feature_importance_df = pd.DataFrame({
    'Feature': X.columns,
    'Importance': feature_importances
})
```

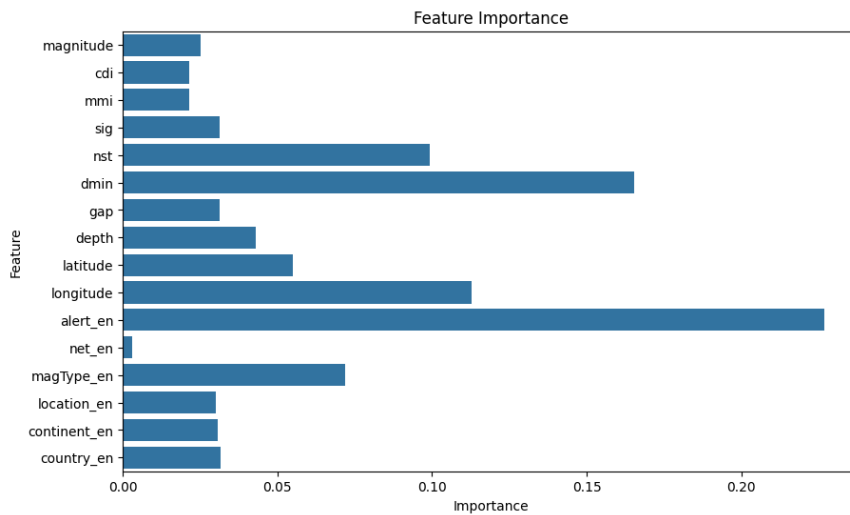
```
feature_importance_df
```

	Feature	Importance
0	magnitude	0.025141
1	cdi	0.021389
2	mmi	0.021324
3	sig	0.031382
4	nst	0.099289
5	dmin	0.165238
6	gap	0.031250
7	depth	0.042967
8	latitude	0.054857
9	longitude	0.112903
10	alert_en	0.226955
11	net_en	0.003097
12	magType_en	0.071947
13	location_en	0.030145
14	continent_en	0.030622
15	country_en	0.031494

Next steps: [View recommended plots](#)

```
import seaborn as sns
```

```
# แสดงค่าความสำคัญของคุณสมบัตินี้เป็นกราฟ
plt.figure(figsize=(10, 6))
sns.barplot(x='Importance', y='Feature', data=feature_importance_df)
plt.xlabel('Importance')
plt.ylabel('Feature')
plt.title('Feature Importance')
plt.show()
```



```
# เลือกเฉพาะ 5 คุณสมบัติที่มีค่า Importance มากที่สุด
top_5_features = feature_importance_df.nlargest(5, 'Importance')
```

```
# แสดงชื่อคุณสมบัติที่เลือก
print("Selected features:")
print(top_5_features)
```

```
Selected features:
   Feature  Importance
10  alert_en    0.226955
5    dmin      0.165238
9  longitude    0.112903
4    nst       0.099289
12 magType_en    0.071947
```

```
# เลือกเฉพาะคุณสมบัติที่สำคัญเพื่อสร้าง X
X = eq[top_5_features['Feature']]
# ใช้ 'tsunami' เป็นตัวแปรตาม (target variable)
y = eq['tsunami']
```

▼ train_test_split

```
from sklearn.model_selection import train_test_split
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import accuracy_score, confusion_matrix
import matplotlib.pyplot as plt
import seaborn as sns
# แบ่งข้อมูลเป็นชุดฝึกและชุดทดสอบ
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

```
# Train Model
knn_model = KNeighborsClassifier(n_neighbors=3)
knn_model.fit(X_train, y_train)
```

```
▼ KNeighborsClassifier
KNeighborsClassifier(n_neighbors=3)
```

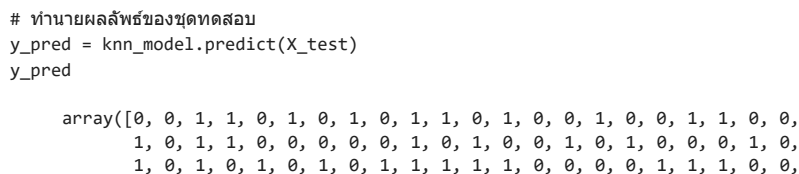
▼ Cross Validation

```
# ใช้ Cross Validation เพื่อประเมินความแม่นยำของโมเดลโดยใช้ K-fold Cross Validation ที่ k=5
from sklearn.model_selection import cross_val_score
scores = cross_val_score(knn_model, X, y, cv=5)
mean_score = scores.mean()
```

- Model KNeighborsClassifier

- ▼ Evaluate Model

Accuracy: 0.89171974522293



```
0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 1, 1, 1, 1, 0, 1, 1, 0, 0, 0, 0,
0, 1, 0, 1, 1, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0,
0, 0, 0, 0, 1, 1, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 1, 1,
0, 1, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 0, 0, 0, 0, 1, 0, 0, 1, 1, 1,
1, 1, 1])
```

```
from sklearn.metrics import precision_score, recall_score, f1_score
```

```
# คำนวณ Recall
```

```
recall = recall_score(y_test, y_pred)
recall
```

```
0.8939393939393939
```

```
# คำนวณ F1-Score
```

```
f1 = f1_score(y_test, y_pred)
f1
```

```
0.8740740740740741
```

```
# คำนวณค่า Precision จากการทำนายของโมเดล
```

```
Precision = precision_score(y_test, y_pred)
Precision
```

```
0.855072463768116
```

```
# กำหนดค่าคะแนนในรูปแบบของ List
```

```
scores = [Precision, recall, f1]
labels = ['Precision', 'Recall', 'F1-Score']
```

```
import numpy as np
```

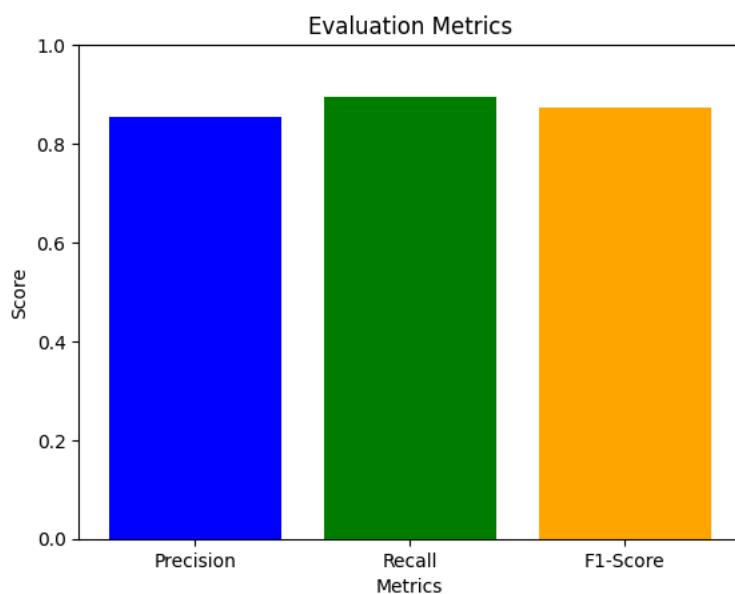
```
import matplotlib.pyplot as plt
```

```
# ค่า Precision, Recall, และ F1-Score
```

```
scores = [Precision, recall, f1]
labels = ['Precision', 'Recall', 'F1-Score']
```

```
# พล็อตกราฟ
```

```
plt.bar(labels, scores, color=['blue', 'green', 'orange'])
plt.xlabel('Metrics')
plt.ylabel('Score')
plt.title('Evaluation Metrics')
plt.ylim([0, 1]) # ตั้งค่าเพื่อให้แกน y อยู่ระหว่าง 0 ถึง 1
plt.show()
```



บันทึกโมเดลเป็นไฟล์

```
import joblib
```

```
# บันทึกโมเดลเป็นไฟล์
```

```
joblib.dump(knn_model, 'knn_model.pkl')
```

```
['knn_model.pkl']
```

```
# โหลดโมเดลจากไฟล์
```

```
knn_model = joblib.load('knn_model.pkl')
```

```
# ใช้โมเดลในการทำนาย
```

```
new_data = [[4.5, 3.6, 0.8, 2.5, 5.1]] # ข้อมูลที่ต้องการทำนาย
```

```
prediction = knn_model.predict(new_data)
```

```
print("Prediction:", prediction)
```

```
Prediction: [0]
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/base.py:439: UserWarning: X does not have valid feature names, but KNeighborsClassifier  
warnings.warn(
```

