

```
In [1]: import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from sklearn.linear_model import LogisticRegression
from sklearn.ensemble import RandomForestClassifier, GradientBoostingClassifier
from sklearn.metrics import classification_report, roc_auc_score
```

```
In [2]: df = pd.read_csv("C:\\Users\\damma\\Downloads\\Customer Churn Prediction\\Churn_Modelling.csv")
```

```
In [3]: df.head()
```

Out[3]:

	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMem
0	1	15634602	Hargrave	619	France	Female	42	2	0.00	1	1	
1	2	15647311	Hill	608	Spain	Female	41	1	83807.86	1	0	
2	3	15619304	Onio	502	France	Female	42	8	159660.80	3	1	
3	4	15701354	Boni	699	France	Female	39	1	0.00	2	0	
4	5	15737888	Mitchell	850	Spain	Female	43	2	125510.82	1	1	

```
In [4]: print("Shape:", df.shape)
print("\nMissing values:\n", df.isnull().sum())
print("\nChurn distribution:\n", df['Exited'].value_counts())
```

Shape: (10000, 14)

Missing values:

RowNumber	0
CustomerId	0
Surname	0
CreditScore	0
Geography	0
Gender	0
Age	0
Tenure	0
Balance	0
NumOfProducts	0
HasCrCard	0
IsActiveMember	0
EstimatedSalary	0
Exited	0

dtype: int64

Churn distribution:

Exited	
0	7963
1	2037

Name: count, dtype: int64

```
In [5]: plt.figure(figsize=(5,4))
sns.countplot(x='Exited', data=df, palette='coolwarm')
plt.title("Churn vs Non-Churn Customers")
plt.xticks([0,1], ['Stayed', 'Churned'])
plt.show()
```

c:\Users\damma\anaconda4\envs\tf\lib\site-packages\seaborn_oldcore.py:1498: FutureWarning: is_categorical_dtype is deprecated and will be removed in a future version. Use isinstance(dtype, CategoricalDtype) instead
if pd.api.types.is_categorical_dtype(vector):
c:\Users\damma\anaconda4\envs\tf\lib\site-packages\seaborn_oldcore.py:1498: FutureWarning: is_categorical_dtype is deprecated and will be removed in a future version. Use isinstance(dtype, CategoricalDtype) instead
if pd.api.types.is_categorical_dtype(vector):
c:\Users\damma\anaconda4\envs\tf\lib\site-packages\seaborn_oldcore.py:1498: FutureWarning: is_categorical_dtype is deprecated and will be removed in a future version. Use isinstance(dtype, CategoricalDtype) instead
if pd.api.types.is_categorical_dtype(vector):


```
In [24]: plt.figure(figsize=(6,4))
sns.distplot(df[df['Exited']==0]['CreditScore'], label='Stayed', hist=True, kde=True)
sns.distplot(df[df['Exited']==1]['CreditScore'], label='Churned', hist=True, kde=True)
plt.title("Credit Score Distribution by Churn")
plt.legend()
plt.show()
```

C:\Users\damma\AppData\Local\Temp\ipykernel_102580\2469896775.py:2: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see <https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751>

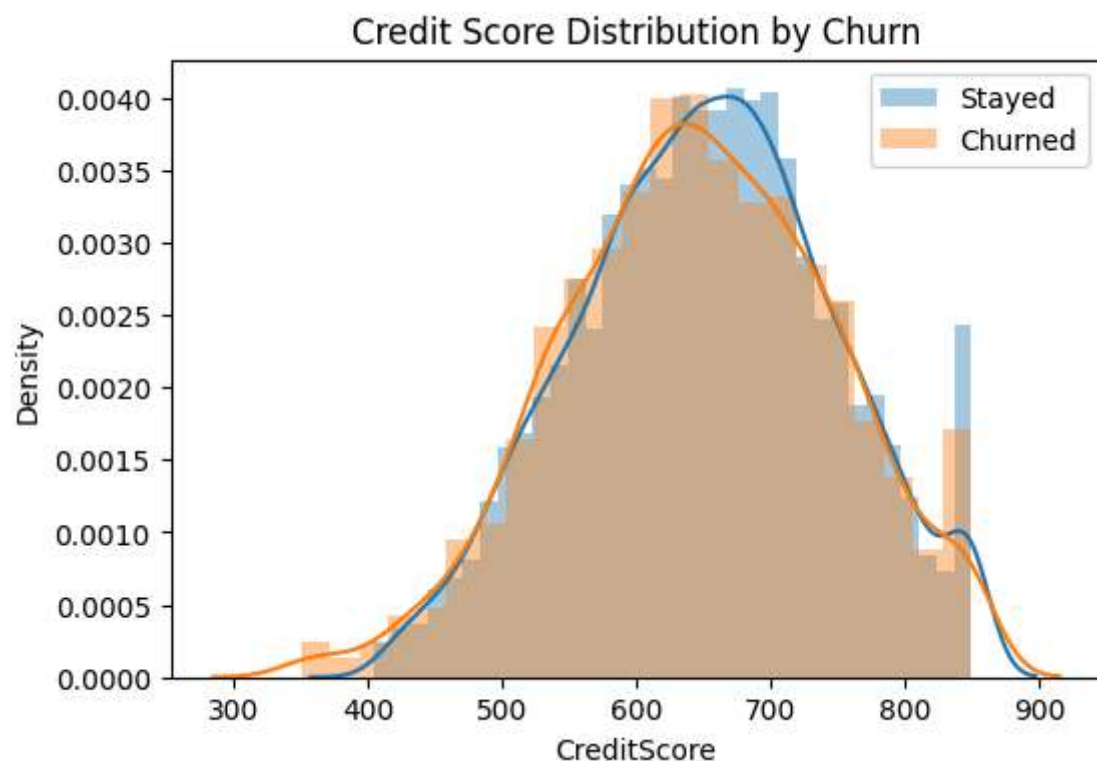
```
sns.distplot(df[df['Exited']==0]['CreditScore'], label='Stayed', hist=True, kde=True)
c:\Users\damma\anaconda4\envs\tf\lib\site-packages\seaborn\_oldcore.py:1498: FutureWarning: is_categorical_dtype is de
precated and will be removed in a future version. Use isinstance(dtype, CategoricalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
c:\Users\damma\anaconda4\envs\tf\lib\site-packages\seaborn\_oldcore.py:1119: FutureWarning: use_inf_as_na option is de
precated and will be removed in a future version. Convert inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
C:\Users\damma\AppData\Local\Temp\ipykernel_102580\2469896775.py:3: UserWarning:
```

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).

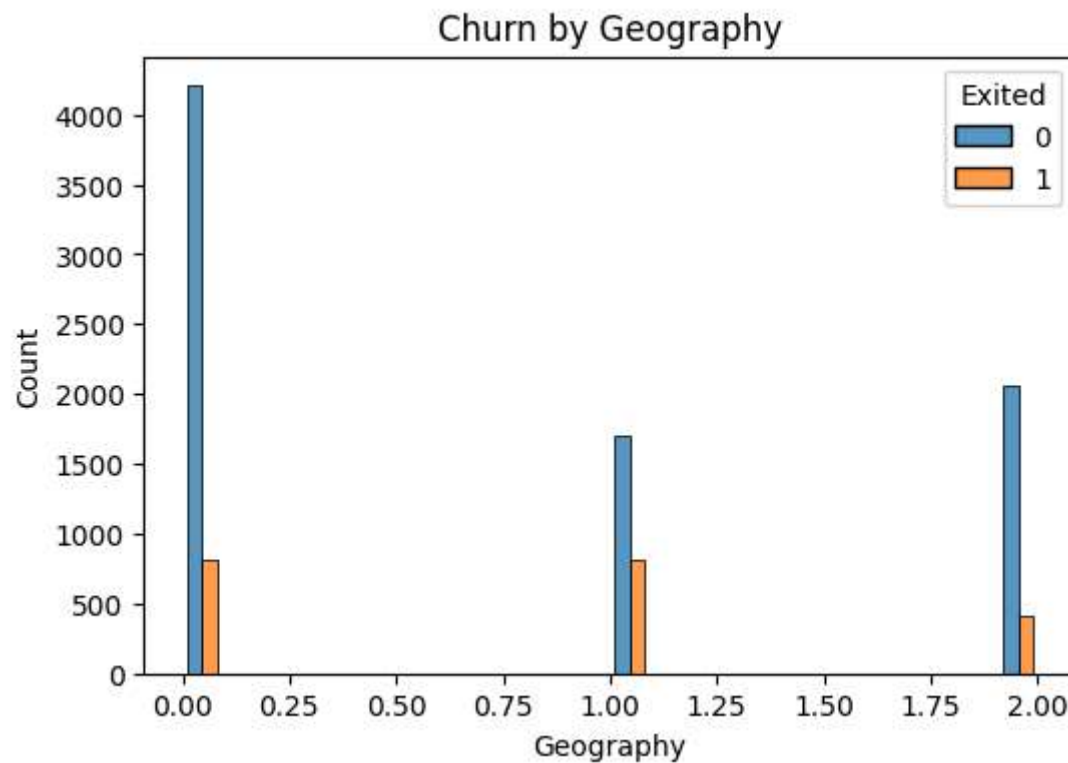
For a guide to updating your code to use the new functions, please see <https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751>

```
sns.distplot(df[df['Exited']==1]['CreditScore'], label='Churned', hist=True, kde=True)
c:\Users\damma\anaconda4\envs\tf\lib\site-packages\seaborn\_oldcore.py:1498: FutureWarning: is_categorical_dtype is de
precated and will be removed in a future version. Use isinstance(dtype, CategoricalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
c:\Users\damma\anaconda4\envs\tf\lib\site-packages\seaborn\_oldcore.py:1119: FutureWarning: use_inf_as_na option is de
precated and will be removed in a future version. Convert inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
```



```
In [23]: plt.figure(figsize=(6,4))
sns.histplot(data=df, x='Geography', hue='Exited', multiple='dodge', shrink=0.8)
plt.title("Churn by Geography")
plt.show()
```

```
c:\Users\damma\anaconda4\envs\tf\lib\site-packages\seaborn\_oldcore.py:1498: FutureWarning: is_categorical_dtype is de
precated and will be removed in a future version. Use isinstance(dtype, CategoricalDtype) instead
    if pd.api.types.is_categorical_dtype(vector):
c:\Users\damma\anaconda4\envs\tf\lib\site-packages\seaborn\_oldcore.py:1498: FutureWarning: is_categorical_dtype is de
precated and will be removed in a future version. Use isinstance(dtype, CategoricalDtype) instead
    if pd.api.types.is_categorical_dtype(vector):
c:\Users\damma\anaconda4\envs\tf\lib\site-packages\seaborn\_oldcore.py:1498: FutureWarning: is_categorical_dtype is de
precated and will be removed in a future version. Use isinstance(dtype, CategoricalDtype) instead
    if pd.api.types.is_categorical_dtype(vector):
c:\Users\damma\anaconda4\envs\tf\lib\site-packages\seaborn\_oldcore.py:1498: FutureWarning: is_categorical_dtype is de
precated and will be removed in a future version. Use isinstance(dtype, CategoricalDtype) instead
    if pd.api.types.is_categorical_dtype(vector):
c:\Users\damma\anaconda4\envs\tf\lib\site-packages\seaborn\_oldcore.py:1119: FutureWarning: use_inf_as_na option is de
precated and will be removed in a future version. Convert inf values to NaN before operating instead.
    with pd.option_context('mode.use_inf_as_na', True):
```



```
In [6]: drop_cols = ['RowNumber', 'CustomerId', 'Surname']
df = df.drop(columns=drop_cols)
```

```
In [7]: label_enc = LabelEncoder()
for col in df.select_dtypes(include=['object']).columns:
    df[col] = label_enc.fit_transform(df[col])
```

```
In [8]: X = df.drop(columns=['Exited'])
y = df['Exited']
```

```
In [9]: X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42, stratify=y)
```

```
In [10]: log_reg = LogisticRegression(max_iter=1000)
log_reg.fit(X_train, y_train)
y_pred_lr = log_reg.predict(X_test)
print("\nLogistic Regression:")
print(classification_report(y_test, y_pred_lr))
print("ROC-AUC:", roc_auc_score(y_test, log_reg.predict_proba(X_test)[:, 1]))
```

Logistic Regression:

	precision	recall	f1-score	support
0	0.80	1.00	0.89	1593
1	0.00	0.00	0.00	407
accuracy			0.80	2000
macro avg	0.40	0.50	0.44	2000
weighted avg	0.63	0.80	0.71	2000

ROC-AUC: 0.554111893094944


```
c:\Users\damma\anaconda4\envs\tf\lib\site-packages\sklearn\metrics\_classification.py:1469: UndefinedMetricWarning: Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` parameter to control this behavior.
_warn_prf(average, modifier, msg_start, len(result))
c:\Users\damma\anaconda4\envs\tf\lib\site-packages\sklearn\metrics\_classification.py:1469: UndefinedMetricWarning: Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` parameter to control this behavior.
_warn_prf(average, modifier, msg_start, len(result))
c:\Users\damma\anaconda4\envs\tf\lib\site-packages\sklearn\metrics\_classification.py:1469: UndefinedMetricWarning: Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` parameter to control this behavior.
_warn_prf(average, modifier, msg_start, len(result))
```

```
In [11]: rf = RandomForestClassifier(n_estimators=100, random_state=42)
rf.fit(X_train, y_train)
y_pred_rf = rf.predict(X_test)
print("\nRandom Forest:")
print(classification_report(y_test, y_pred_rf))
print("ROC-AUC:", roc_auc_score(y_test, rf.predict_proba(X_test)[: , 1]))
```

```
Random Forest:
              precision    recall  f1-score   support

    0           0.88        0.97        0.92        1593
    1           0.78        0.46        0.58         407

 accuracy          0.86
 macro avg         0.83
 weighted avg      0.86
```

```
ROC-AUC: 0.846416524382626
```

```
In [12]: gb = GradientBoostingClassifier(random_state=42)
gb.fit(X_train, y_train)
y_pred_gb = gb.predict(X_test)
print("\nGradient Boosting:")
print(classification_report(y_test, y_pred_gb))
print("ROC-AUC:", roc_auc_score(y_test, gb.predict_proba(X_test)[: , 1]))
```

```
Gradient Boosting:
              precision    recall  f1-score   support

    0           0.88        0.97        0.92        1593
    1           0.79        0.48        0.59         407

 accuracy          0.87
 macro avg         0.83
 weighted avg      0.86
```

```
ROC-AUC: 0.867301045267147
```

```
In [26]: # Simple churn check

age = 40
balance = 60000
is_active = True

if balance < 5000 or not is_active:
    print("Likely to Churn")
else:
    print("Likely to Stay")
```

```
Likely to Stay
```