

```

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

train=pd.read_csv("C:\\Users\\deepi\\OneDrive\\Desktop\\
Summer_Anlytics\\Hackathon_Summer analytics\\
training_set_features.csv")
test=pd.read_csv("C:\\Users\\deepi\\OneDrive\\Desktop\\
Summer_Anlytics\\Hackathon_Summer analytics\\test_set_features.csv")
label=pd.read_csv("C:\\Users\\deepi\\OneDrive\\Desktop\\
Summer_Anlytics\\Hackathon_Summer analytics\\training_set_labels.csv")

```

```
train.head()
```

	respondent_id	xyz_concern	xyz_knowledge
behavioral_antiviral_meds \			
0	0	1.0	0.0
0.0			
1	1	3.0	2.0
0.0			
2	2	1.0	1.0
0.0			
3	3	1.0	1.0
0.0			
4	4	2.0	1.0
0.0			

	behavioral_avoidance	behavioral_face_mask
behavioral_wash_hands \		
0	0.0	0.0
0.0		
1	1.0	0.0
0.0		
2	1.0	0.0
0.0		
3	1.0	0.0
0.0		
4	1.0	0.0
0.0		

	behavioral_large_gatherings	behavioral_outside_home
0	0.0	1.0
0.0		
1	0.0	1.0
0.0		
2	0.0	0.0
0.0		
3	1.0	0.0
0.0		
4	1.0	0.0
0.0		

	behavioral_touch_face	...	income_poverty
marital_status \			
0	1.0	...	Below Poverty
0.0			Not

```

Married
1          1.0  ...          Below Poverty      Not
Married
2          0.0  ...  <= $75,000, Above Poverty      Not
Married
3          0.0  ...          Below Poverty      Not
Married
4          1.0  ...  <= $75,000, Above Poverty
Married

    rent_or_own    employment_status    hhs_geo_region
census_msa  \
0          Own    Not in Labor Force          oxchjgsf
Non-MSA
1          Rent          Employed          bhuqouqj    MSA, Not Principle
City
2          Own          Employed          qufhixun    MSA, Not Principle
City
3          Rent    Not in Labor Force          lrircsnp          MSA,
Principle City
4          Own          Employed          qufhixun    MSA, Not Principle
City

    household_adults    household_children    employment_industry  \
0          0.0          0.0          NaN
1          0.0          0.0          pxcmvdjn
2          2.0          0.0          rucpziiij
3          0.0          0.0          NaN
4          1.0          0.0          wxleyezf

    employment_occupation
0          NaN
1          xgwztkwe
2          xtkaffoo
3          NaN
4          emcorrxb

[5 rows x 36 columns]

label.head()

    respondent_id    xyz_vaccine    seasonal_vaccine
0          0          0          0
1          1          0          1
2          2          0          0
3          3          0          1
4          4          0          0

test.head()

```

respondent_id	xyz_concern	xyz_knowledge
behavioral_antiviral_meds \		
0 26707	2.0	2.0
0.0		
1 26708	1.0	1.0
0.0		
2 26709	2.0	2.0
0.0		
3 26710	1.0	1.0
0.0		
4 26711	3.0	1.0
1.0		
behavioral_avoidance	behavioral_face_mask	
behavioral_wash_hands \		
0 1.0	0.0	1.0
1 0.0	0.0	0.0
2 0.0	1.0	1.0
3 0.0	0.0	0.0
4 1.0	0.0	1.0
behavioral_large_gatherings	behavioral_outside_home \	
0 1.0	0.0	
1 0.0	0.0	
2 1.0	1.0	
3 0.0	0.0	
4 1.0	1.0	
behavioral_touch_face ...	income_poverty	
marital_status \		
0 1.0 ...	> \$75,000	Not
Married		
1 0.0 ...	Below Poverty	Not
Married		
2 1.0 ...	> \$75,000	
Married		
3 0.0 ...	<= \$75,000, Above Poverty	
Married		
4 1.0 ...	<= \$75,000, Above Poverty	Not
Married		
rent_or_own	employment_status	hhs_geo_region
census_msa \		
0 Rent	Employed	mlyzmhmf MSA, Not Principle
City		
1 Rent	Employed	bhuqouqj

```

Non-MSA
2          Own          Employed          lrircsnp
Non-MSA
3          Own  Not in Labor Force          lrircsnp  MSA, Not Principle
City
4          Own          Employed          lzgpxyit
Non-MSA

```

```

      household_adults  household_children  employment_industry \
0          1.0          0.0          atmlpfrs
1          3.0          0.0          atmlpfrs
2          1.0          0.0          nduyfdeo
3          1.0          0.0          NaN
4          0.0          1.0          fcxhlndr

```

```

      employment_occupation
0          hfxkjkmi
1          xqwwgdyp
2          pvmttkik
3          NaN
4          mxkfnird

```

```
[5 rows x 36 columns]
```

```
train=train.merge(label, on="respondent_id", how="right")
```

```
train.head()
```

```

      respondent_id  xyz_concern  xyz_knowledge
behavioral_antiviral_meds \
0          0          1.0          0.0
0.0
1          1          3.0          2.0
0.0
2          2          1.0          1.0
0.0
3          3          1.0          1.0
0.0
4          4          2.0          1.0
0.0

```

```

      behavioral_avoidance  behavioral_face_mask
behavioral_wash_hands \
0          0.0          0.0          0.0
1          1.0          0.0          1.0
2          1.0          0.0          0.0
3          1.0          0.0          1.0

```

4	1.0	0.0	1.0
---	-----	-----	-----

	behavioral_large_gatherings	behavioral_outside_home	\
0	0.0	1.0	
1	0.0	1.0	
2	0.0	0.0	
3	1.0	0.0	
4	1.0	0.0	

	behavioral_touch_face	...	rent_or_own	employment_status	\
0	1.0	...	Own	Not in Labor Force	
1	1.0	...	Rent	Employed	
2	0.0	...	Own	Employed	
3	0.0	...	Rent	Not in Labor Force	
4	1.0	...	Own	Employed	

	hhs_geo_region		census_msa	household_adults	\
0	oxchjgsf		Non-MSA	0.0	
1	bhuqouqj	MSA, Not Principle	City	0.0	
2	qufhixun	MSA, Not Principle	City	2.0	
3	lrircsnp	MSA, Principle	City	0.0	
4	qufhixun	MSA, Not Principle	City	1.0	

	household_children	employment_industry	employment_occupation	\
0	0.0	NaN	NaN	
1	0.0	pxcmvdjn	xgwztkwe	
2	0.0	rucpziij	xtkaffoo	
3	0.0	NaN	NaN	
4	0.0	wxleyezf	emcorrxb	

	xyz_vaccine	seasonal_vaccine
0	0	0
1	0	1
2	0	0
3	0	1
4	0	0

[5 rows x 38 columns]

train.isnull().sum()

respondent_id	0
xyz_concern	92
xyz_knowledge	116
behavioral_antiviral_meds	71
behavioral_avoidance	208
behavioral_face_mask	19
behavioral_wash_hands	42
behavioral_large_gatherings	87

behavioral_outside_home	82
behavioral_touch_face	128
doctor_recc_xyz	2160
doctor_recc_seasonal	2160
chronic_med_condition	971
child_under_6_months	820
health_worker	804
health_insurance	12274
opinion_xyz_vacc_effective	391
opinion_xyz_risk	388
opinion_xyz_sick_from_vacc	395
opinion_seas_vacc_effective	462
opinion_seas_risk	514
opinion_seas_sick_from_vacc	537
age_group	0
education	1407
race	0
sex	0
income_poverty	4423
marital_status	1408
rent_or_own	2042
employment_status	1463
hhs_geo_region	0
census_msa	0
household_adults	249
household_children	249
employment_industry	13330
employment_occupation	13470
xyz_vaccine	0
seasonal_vaccine	0

dtype: int64

*# replace columns with mode*

```
for column in train.columns:
    train[column] = train[column].fillna(train[column].mode()[0])
```

```
for column in test.columns:
    test[column] = test[column].fillna(test[column].mode()[0])
```

```
train.isnull().sum().sum(), test.isnull().sum().sum()
```

```
(0, 0)
```

*# Label encode obejct files*

```
from sklearn import preprocessing
```

```
le=preprocessing.LabelEncoder()
```

```
train.age_group=le.fit_transform(train.age_group)
```

```
train.education=le.fit_transform(train.education)
```

```

train.race=le.fit_transform(train.race)
train.sex=le.fit_transform(train.sex)
train.income_poverty=le.fit_transform(train.income_poverty)
train.marital_status=le.fit_transform(train.marital_status)
train.rent_or_own=le.fit_transform(train.rent_or_own)
train.employment_status=le.fit_transform(train.employment_status)
train.hhs_geo_region=le.fit_transform(train.hhs_geo_region)
train.census_msa=le.fit_transform(train.census_msa)
train.employment_industry=le.fit_transform(train.employment_industry)
train.employment_occupation=le.fit_transform(train.employment_occupati
on)

```

```

test.age_group=le.fit_transform(test.age_group)
test.education=le.fit_transform(test.education)
test.race=le.fit_transform(test.race)
test.sex=le.fit_transform(test.sex)
test.income_poverty=le.fit_transform(test.income_poverty)
test.marital_status=le.fit_transform(test.marital_status)
test.rent_or_own=le.fit_transform(test.rent_or_own)
test.employment_status=le.fit_transform(test.employment_status)
test.hhs_geo_region=le.fit_transform(test.hhs_geo_region)
test.census_msa=le.fit_transform(test.census_msa)
test.employment_industry=le.fit_transform(test.employment_industry)
test.employment_occupation=le.fit_transform(test.employment_occupation
)

```

```
train.head()
```

	respondent_id	xyz_concern	xyz_knowledge
behavioral_antiviral_meds \			
0	0	1.0	0.0
0.0			
1	1	3.0	2.0
0.0			
2	2	1.0	1.0
0.0			
3	3	1.0	1.0
0.0			
4	4	2.0	1.0
0.0			
behavioral_avoidance			
behavioral_wash_hands \			
0	0.0	0.0	0.0
1	1.0	0.0	1.0
2	1.0	0.0	0.0

3	1.0	0.0	1.0
4	1.0	0.0	1.0

	behavioral_large_gatherings	behavioral_outside_home \
0	0.0	1.0
1	0.0	1.0
2	0.0	0.0
3	1.0	0.0
4	1.0	0.0

	behavioral_touch_face	...	rent_or_own	employment_status
hhs_geo_region \				
0	1.0	...	0	1
8				
1	1.0	...	1	0
1				
2	0.0	...	0	0
9				
3	0.0	...	1	1
5				
4	1.0	...	0	0
9				

	census_msa	household_adults	household_children
employment_industry \			
0	2	0.0	0.0
4			
1	0	0.0	0.0
12			
2	0	2.0	0.0
14			
3	1	0.0	0.0
4			
4	0	1.0	0.0
18			

	employment_occupation	xyz_vaccine	seasonal_vaccine
0	21	0	0
1	19	0	1
2	21	0	0
3	21	0	1
4	5	0	0

[5 rows x 38 columns]

sns.distplot(train["xyz\_vaccine"])



```
C:\Users\deepi\AppData\Local\Temp\ipykernel_27068\685520833.py:1:
UserWarning:
```

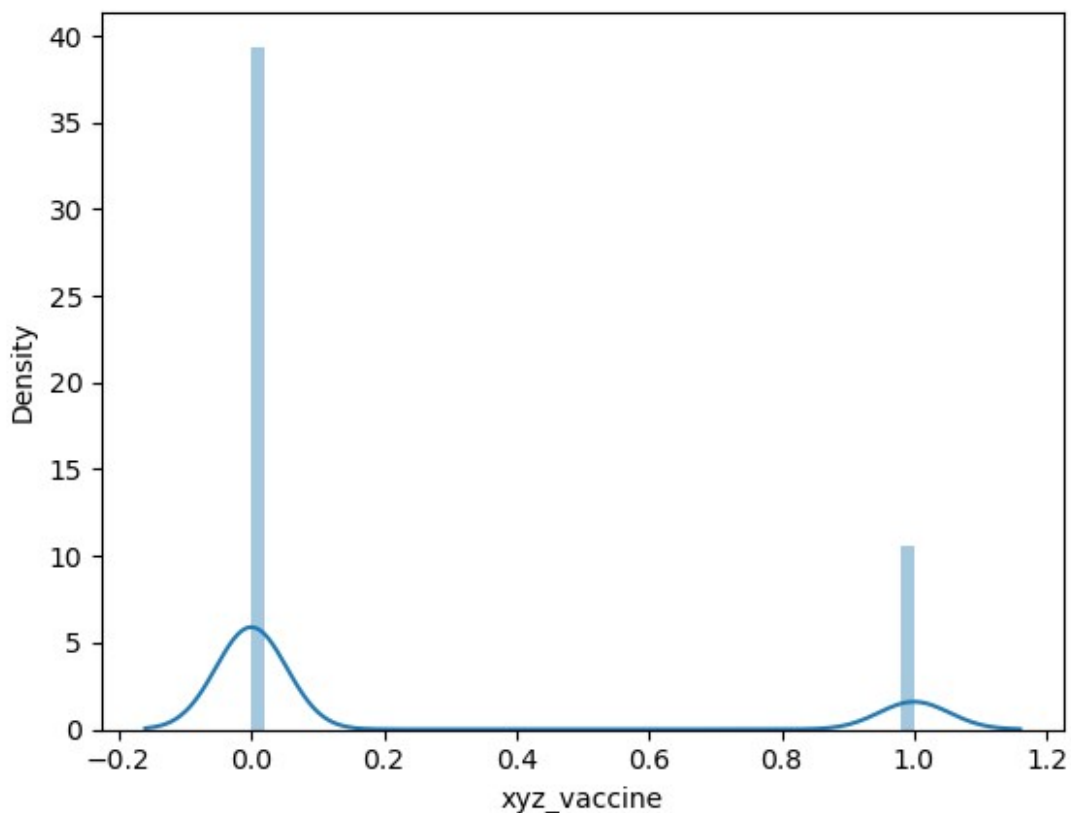
```
`distplot` is a deprecated function and will be removed in seaborn
v0.14.0.
```

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(train["xyz_vaccine"])
```

```
<Axes: xlabel='xyz_vaccine', ylabel='Density'>
```



```
sns.distplot(train["seasonal_vaccine"])
```

```
C:\Users\deepi\AppData\Local\Temp\ipykernel_27068\877207631.py:1:
UserWarning:
```

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

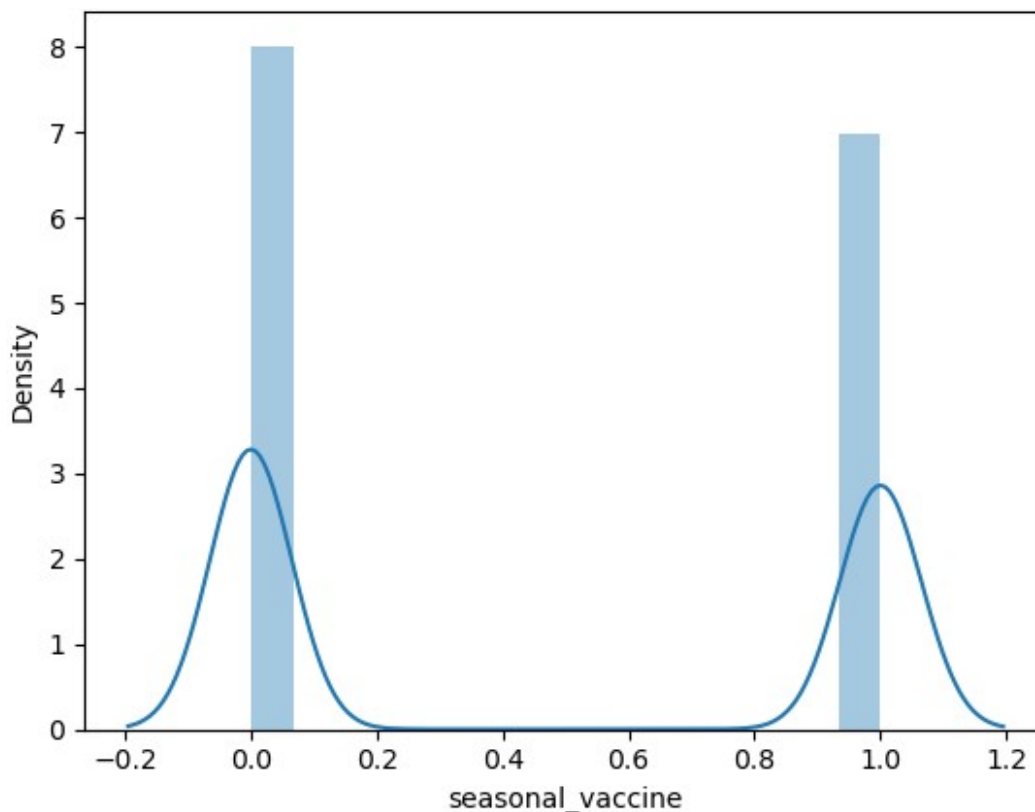
v0.14.0.

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(train["seasonal_vaccine"])
```

<Axes: xlabel='seasonal\_vaccine', ylabel='Density'>



```
# count values in train set
```

```
from collections import Counter  
print(Counter(train.xyz_vaccine))
```

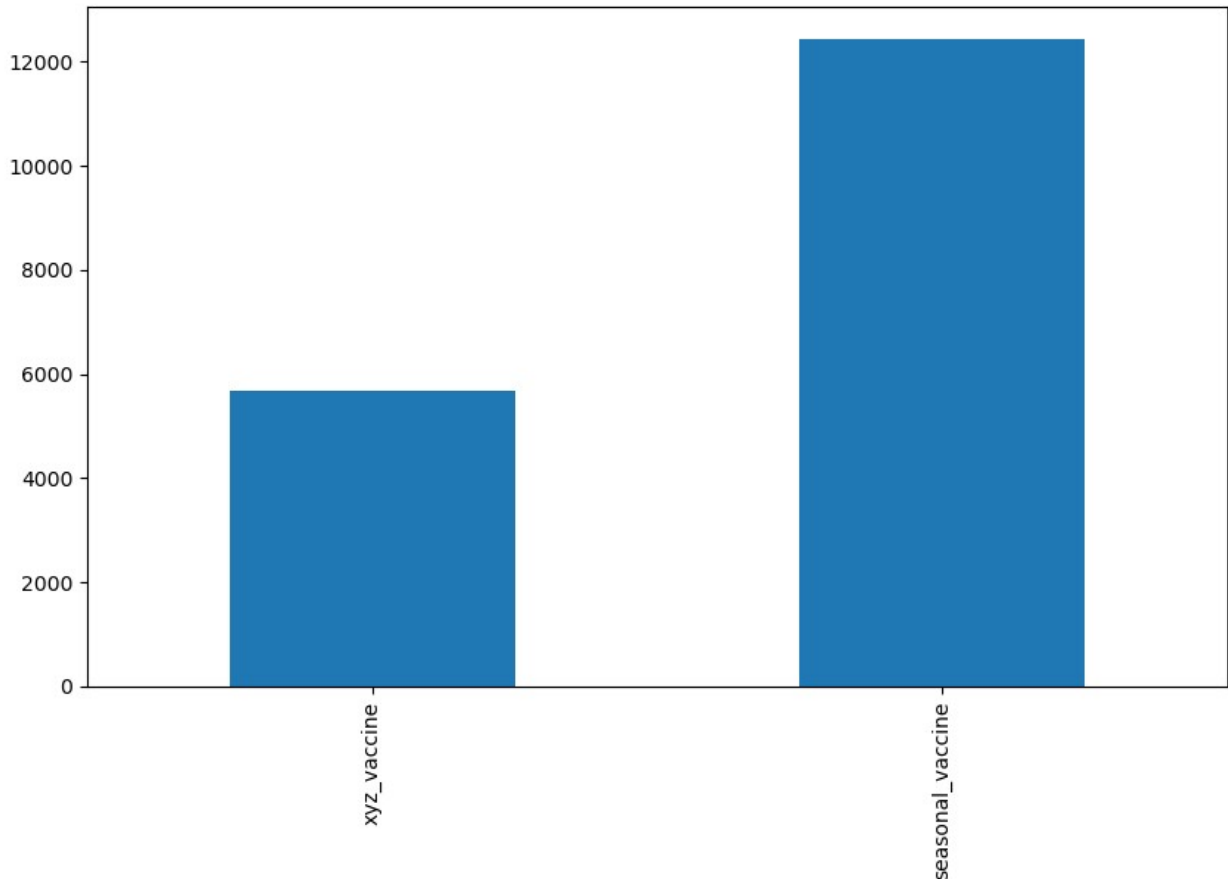
```
Counter({0: 21033, 1: 5674})
```

```
from collections import Counter  
print(Counter(train.seasonal_vaccine))
```

```
Counter({0: 14272, 1: 12435})
```

```
# converting binary column to category
target=['xyz_vaccine', "seasonal_vaccine"]
y_data=train[target]

# plot category data
plt.figure(figsize=(10,6))
y_data.sum(axis=0).plot.bar()
plt.show()
```



```
# define x, y x_test
y=train[target]
X=train.drop(["respondent_id", "xyz_vaccine", "seasonal_vaccine"],
axis=1)
X_test=test.drop(["respondent_id"], axis=1)

y.shape, X.shape, X_test.shape
((26707, 2), (26707, 35), (26708, 35))
```

```
# split training set for training and testing
from sklearn.model_selection import train_test_split

X_train, X_val, y_train, y_val=train_test_split(X, y, random_state=42,
test_size=0.2, stratify=y)

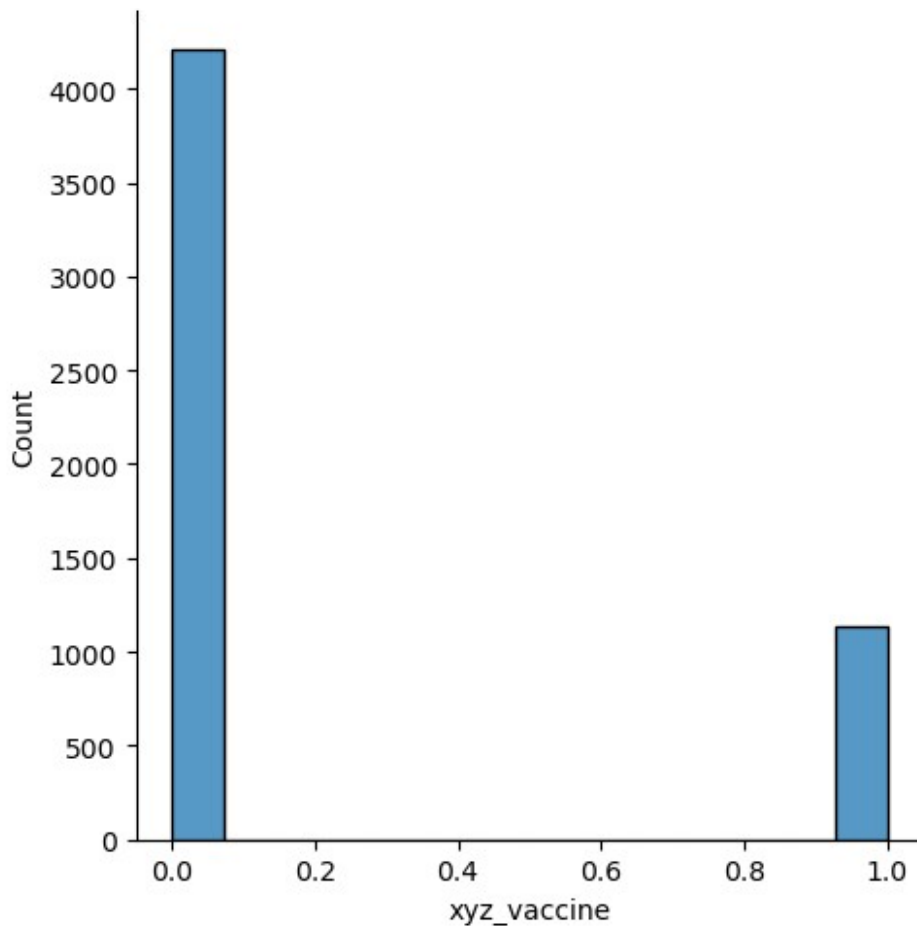
# shape of split
X_train.shape, X_val.shape, y_train.shape, y_val.shape
((21365, 35), (5342, 35), (21365, 2), (5342, 2))

from sklearn.preprocessing import StandardScaler, MinMaxScaler

# define the scaler
scaler=MinMaxScaler().fit(X_train)
X_train=scaler.transform(X_train)
X_test=scaler.transform(X_test)
X_val=scaler.transform(X_val)

sns.displot(y_val.xyz_vaccine)

<seaborn.axisgrid.FacetGrid at 0x1e2765d4200>
```



```
sns.distplot(y_val.seasonal_vaccine)
```

C:\Users\deepi\AppData\Local\Temp\ipykernel\_27068\3866735423.py:1:  
UserWarning:

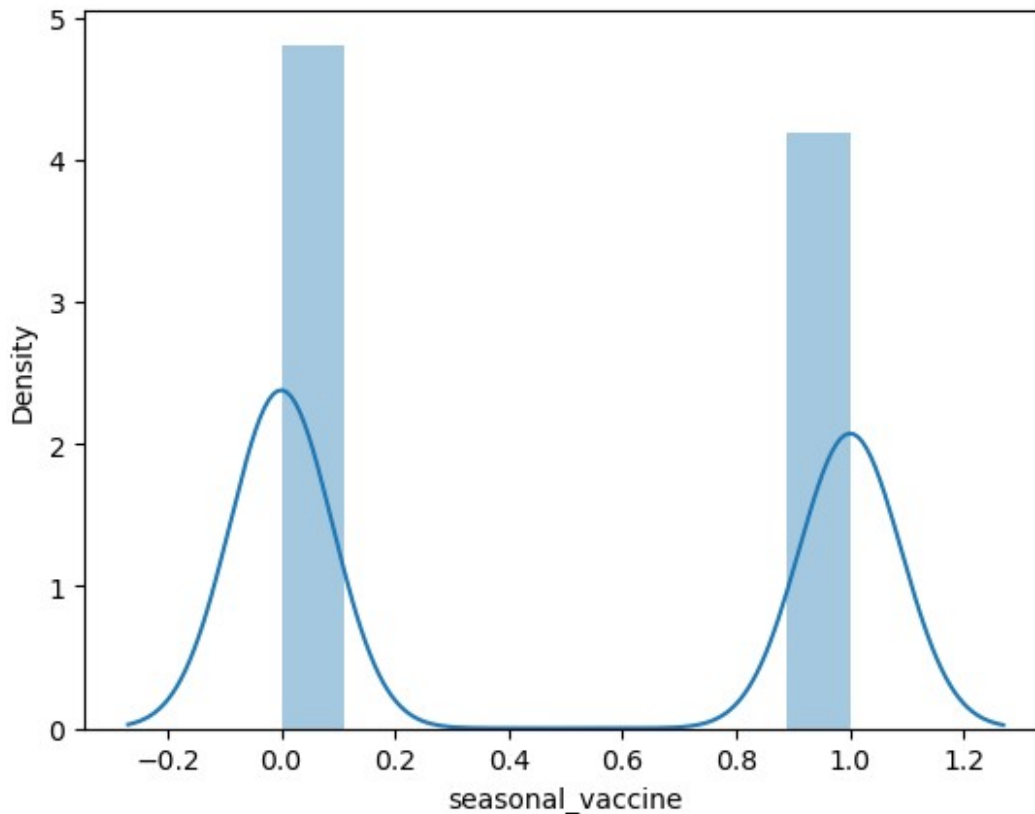
`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

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(y_val.seasonal_vaccine)
```

```
<Axes: xlabel='seasonal_vaccine', ylabel='Density'>
```



```
# define model and train the model

# MLPC

from sklearn.multioutput import MultiOutputClassifier
from sklearn.multiclass import OneVsRestClassifier
from sklearn.svm import SVC
from sklearn.neural_network import MLPClassifier

model=MLPClassifier(random_state=1, max_iter=300)
models=MultiOutputClassifier(model)

# fit the model

models.fit(X_train, y_train)
print(models.score(X_train, y_train))

C:\Users\deepi\anaconda3\Lib\site-packages\sklearn\neural_network\
_multilayer_perceptron.py:691: ConvergenceWarning: Stochastic
Optimizer: Maximum iterations (300) reached and the optimization
hasn't converged yet.
  warnings.warn(
C:\Users\deepi\anaconda3\Lib\site-packages\sklearn\neural_network\
_multilayer_perceptron.py:691: ConvergenceWarning: Stochastic
Optimizer: Maximum iterations (300) reached and the optimization
```

```
hasn't converged yet.  
warnings.warn(  

```

```
0.7390592089866604
```

```
y_pred=models.predict(X_val)  
print(models.score(X_val, y_val))  
y_pred
```

```
0.6426432047922127
```

```
array([[0, 0],  
       [1, 1],  
       [0, 0],  
       ...,  
       [0, 0],  
       [0, 0],  
       [1, 0]], dtype=int64)
```

```
from sklearn import metrics
```

```
my_metrics=metrics.classification_report(y_val, y_pred)  
print(my_metrics)
```

	precision	recall	f1-score	support
0	0.57	0.52	0.54	1135
1	0.76	0.71	0.73	2488
micro avg	0.70	0.65	0.67	3623
macro avg	0.66	0.61	0.64	3623
weighted avg	0.70	0.65	0.67	3623
samples avg	0.33	0.32	0.32	3623

```
C:\Users\deepi\anaconda3\Lib\site-packages\sklearn\metrics\  
_classification.py:1565: UndefinedMetricWarning: Precision is ill-  
defined and being set to 0.0 in samples with no predicted labels. Use  
'zero_division' parameter to control this behavior.
```

```
_warn_prf(average, modifier, f"{metric.capitalize()} is",  
len(result))
```

```
C:\Users\deepi\anaconda3\Lib\site-packages\sklearn\metrics\  
_classification.py:1565: UndefinedMetricWarning: Recall is ill-defined  
and being set to 0.0 in samples with no true labels. Use  
'zero_division' parameter to control this behavior.
```

```
_warn_prf(average, modifier, f"{metric.capitalize()} is",  
len(result))
```

```
C:\Users\deepi\anaconda3\Lib\site-packages\sklearn\metrics\  
_classification.py:1565: UndefinedMetricWarning: F-score is ill-  
defined and being set to 0.0 in samples with no true nor predicted  
labels. Use 'zero_division' parameter to control this behavior.
```

```
_warn_prf(average, modifier, f"{metric.capitalize()} is",
len(result))
```

```
from sklearn.metrics import roc_auc_score
```

```
print(roc_auc_score(y_val, y_pred))
```

```
0.730387080510952
```

```
df=pd.DataFrame({"Actual_xyz": y_val.xyz_vaccine, "Actual_seasonal":
y_val.seasonal_vaccine,
                 "xyz":y_pred[:, 0], "seasonal":y_pred[:, 0]})
```

```
df.head()
```

	Actual_xyz	Actual_seasonal	xyz	seasonal
22317	0	1	0	0
1215	0	1	1	1
10540	0	0	0	0
7407	0	0	0	0
14697	0	0	0	0

```
from sklearn.metrics import accuracy_score, confusion_matrix,
classification_report
```

```
# print(confusion_matrix(y_val, pred))
```

```
print(classification_report(y_val, y_pred))
```

```
print(accuracy_score(y_val, y_pred))
```

	precision	recall	f1-score	support
0	0.57	0.52	0.54	1135
1	0.76	0.71	0.73	2488
micro avg	0.70	0.65	0.67	3623
macro avg	0.66	0.61	0.64	3623
weighted avg	0.70	0.65	0.67	3623
samples avg	0.33	0.32	0.32	3623

```
0.6426432047922127
```

```
C:\Users\deepi\anaconda3\Lib\site-packages\sklearn\metrics\
_classification.py:1565: UndefinedMetricWarning: Precision is ill-
defined and being set to 0.0 in samples with no predicted labels. Use
`zero_division` parameter to control this behavior.
```

```
_warn_prf(average, modifier, f"{metric.capitalize()} is",
len(result))
```

```
C:\Users\deepi\anaconda3\Lib\site-packages\sklearn\metrics\
_classification.py:1565: UndefinedMetricWarning: Recall is ill-defined
and being set to 0.0 in samples with no true labels. Use
`zero_division` parameter to control this behavior.
```

```
_warn_prf(average, modifier, f"{metric.capitalize()} is",
len(result))
```



```
C:\Users\deepi\anaconda3\Lib\site-packages\sklearn\metrics\
_classification.py:1565: UndefinedMetricWarning: F-score is ill-
defined and being set to 0.0 in samples with no true nor predicted
labels. Use `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, f"{metric.capitalize()} is",
len(result))
```

```
# multilabel confusion matrix
```

```
from sklearn.metrics import multilabel_confusion_matrix
```

```
confusion_matrix=multilabel_confusion_matrix(y_val, y_pred)
confusion_matrix
```

```
array([[ [3769,  438],
         [ 550,  585]],
       [[2281,  573],
         [ 719, 1769]]], dtype=int64)
```

```
# make predictions
```

```
predict=models.predict(X_test)
predict
```

```
array([[0, 0],
       [0, 0],
       [1, 1],
       ...,
       [0, 0],
       [0, 0],
       [1, 1]], dtype=int64)
```

```
# predict probabilities
```

```
proba=models.predict_proba(X_test)
proba=np.asarray(proba)
proba
```

```
array([[ [0.9857386 , 0.0142614 ],
         [0.97018791, 0.02981209],
         [0.22934574, 0.77065426],
         ...,
         [0.86443072, 0.13556928],
         [0.99264437, 0.00735563],
         [0.05890341, 0.94109659]],
       [[ [0.80154575, 0.19845425],
         [0.99345953, 0.00654047],
         [0.11534483, 0.88465517],
         ...,
```

```

        [0.6403762 , 0.3596238 ],
        [0.69638709, 0.30361291],
        [0.09991701, 0.90008299]]])

# prepare submission
proba[0, :]

array([[0.9857386 , 0.0142614 ],
       [0.97018791, 0.02981209],
       [0.22934574, 0.77065426],
       ...,
       [0.86443072, 0.13556928],
       [0.99264437, 0.00735563],
       [0.05890341, 0.94109659]])

proba[0, :][:,1]    # xyz

array([0.0142614 , 0.02981209, 0.77065426, ..., 0.13556928,
       0.00735563,
       0.94109659])

proba[1, :][:, 1]    # seasonal

array([0.19845425, 0.00654047, 0.88465517, ..., 0.3596238 ,
       0.30361291,
       0.90008299])

df_vaccine=pd.DataFrame(proba[0, :][:,1], proba[1,:][:,1])
df_vaccine.reset_index(inplace=True)
df_vaccine.columns=["xyz_vaccine", "seasonal_vaccine"]
df_vaccine

   xyz_vaccine  seasonal_vaccine
0         0.198454             0.014261
1         0.006540             0.029812
2         0.884655             0.770654
3         0.915995             0.653639
4         0.614479             0.972902
...         ...
26703      0.562127             0.492162
26704      0.055067             0.066781
26705      0.359624             0.135569
26706      0.303613             0.007356
26707      0.900083             0.941097

[26708 rows x 2 columns]

submit=pd.DataFrame({"respondent_id":test.respondent_id,
                    "xyz_vaccine":df_vaccine.xyz_vaccine,
                    "seasonal_vaccine":df_vaccine.seasonal_vaccine})
submit

```

	respondent_id	xyz_vaccine	seasonal_vaccine
0	26707	0.198454	0.014261
1	26708	0.006540	0.029812
2	26709	0.884655	0.770654
3	26710	0.915995	0.653639
4	26711	0.614479	0.972902
...	...	...	...
26703	53410	0.562127	0.492162
26704	53411	0.055067	0.066781
26705	53412	0.359624	0.135569
26706	53413	0.303613	0.007356
26707	53414	0.900083	0.941097

[26708 rows x 3 columns]

```
submit.to_csv("submission.csv", index=False)
print("submission saved")
```

submission saved

*# read submission*

```
submission=pd.read_csv("submission.csv")
submission
```

	respondent_id	xyz_vaccine	seasonal_vaccine
0	26707	0.198454	0.014261
1	26708	0.006540	0.029812
2	26709	0.884655	0.770654
3	26710	0.915995	0.653639
4	26711	0.614479	0.972902
...	...	...	...
26703	53410	0.562127	0.492162
26704	53411	0.055067	0.066781
26705	53412	0.359624	0.135569
26706	53413	0.303613	0.007356
26707	53414	0.900083	0.941097

[26708 rows x 3 columns]