```
from google.colab import files
uploaded = files.upload()
```

Choose Files large_twitch_features.csv

- **large_twitch_features.csv**(text/csv) - 7870814 bytes, last modified: 4/9/2023 - 100% done

```
Saving large_twitch_features.csv to large_twitch_features.csv
```

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

```
df = pd.read_csv('large_twitch_features.csv')
```

```
df.head(7)
```

	views	mature	life_time(hr)	created_at	updated_at	numeric_id	dead_account	language	affiliate
0	7879	1	969	2/16/2016	10/12/2018	0	0	EN	1
1	500	0	2699	5/19/2011	10/8/2018	1	0	EN	0
2	382502	1	3149	2/27/2010	10/12/2018	2	0	EN	1
3	386	0	1344	1/26/2015	10/1/2018	3	0	EN	0
4	2486	0	1784	11/22/2013	10/11/2018	4	0	EN	0
5	4987	1	1288	4/3/2015	10/12/2018	5	0	EN	1
6	234	0	358	9/14/2017	9/7/2018	6	0	EN	0

```
df.tail(7)
```

	views	mature	life_time(hr)	created_at	updated_at	numeric_id	dead_account	language	affi
168107	10752	0	2241	8/19/2012	10/8/2018	168107	0	EN	
168108	10057	1	851	6/13/2016	10/12/2018	168108	0	EN	
168109	4965	0	810	7/20/2016	10/8/2018	168109	0	EN	
168110	4128	1	2080	1/31/2013	10/12/2018	168110	0	EN	
168111	3545	0	1797	11/8/2013	10/10/2018	168111	0	EN	
168112	892736	1	2135	12/7/2012	10/12/2018	168112	0	EN	
168113	791	0	2005	1/22/2013	7/20/2018	168113	0	EN	

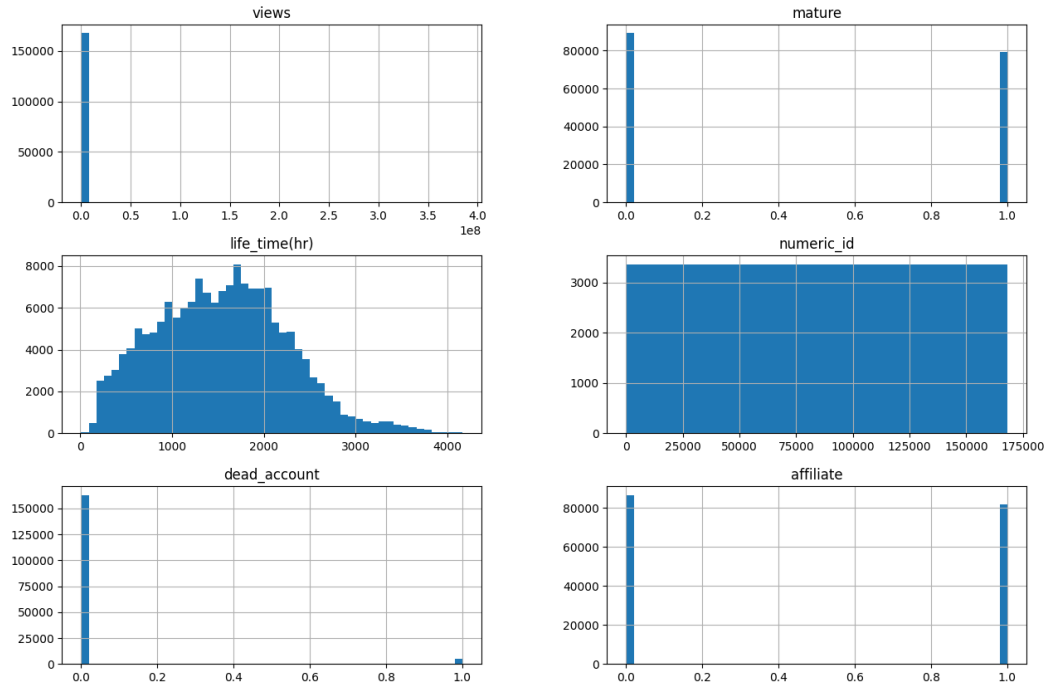
```
df.shape
```

```
(168114, 9)
```

```
df.isnull().sum()
```

```
views           0
mature          0
life_time(hr)   0
created_at      0
updated_at      0
numeric_id      0
dead_account    0
language        0
affiliate       0
dtype: int64
```

```
df.hist(bins=50, figsize=(15,10))
plt.show()
```




```
df.corr()
```

```
<ipython-input-9-2f6f6606aa2c>:1: FutureWarning: The default value of numeric_only in DataFrame.corr is
  df.corr()
```

	views	mature	life_time(hr)	numeric_id	dead_account	affiliate
views	1.000000	-0.021052	0.053711	0.001807	-0.009929	-0.048112
mature	-0.021052	1.000000	0.018040	-0.002002	-0.067051	0.179874
life_time(hr)	0.053711	0.018040	1.000000	-0.003837	-0.022837	-0.122796
numeric_id	0.001807	-0.002002	-0.003837	1.000000	0.000865	0.001466
dead_account	-0.009929	-0.067051	-0.022837	0.000865	1.000000	-0.167585
affiliate	-0.048112	0.179874	-0.122796	0.001466	-0.167585	1.000000

```
df = df.drop(['created_at','updated_at','language','dead_account'],axis=1)
```

```
df
```

	views	mature	life_time(hr)	numeric_id	affiliate
0	7879	1	969	0	1
1	500	0	2699	1	0
2	382502	1	3149	2	1

```
x = df.iloc[:,2:]
x
```

	life_time(hr)	numeric_id	affiliate
0	969	0	1
1	2699	1	0
2	3149	2	1
3	1344	3	0
4	1784	4	0
...	...	...	...
168109	810	168109	0
168110	2080	168110	0
168111	1797	168111	1
168112	2135	168112	0
168113	2005	168113	0

168114 rows × 3 columns

```
x.insert(0,column="views",value=df['views'].values)
```

```
x
```

	views	life_time(hr)	numeric_id	affiliate
0	7879	969	0	1
1	500	2699	1	0
2	382502	3149	2	1
3	386	1344	3	0
4	2486	1784	4	0
...	...	...	...	...
168109	4965	810	168109	0
168110	4128	2080	168110	0
168111	3545	1797	168111	1
168112	892736	2135	168112	0
168113	791	2005	168113	0

168114 rows × 4 columns

```
y = df.iloc[:,1:2]
y
```

	mature
0	1
1	0
2	1
3	0

```
from sklearn.preprocessing import LabelEncoder, StandardScaler
standard_scaler = StandardScaler()
label_encoder = LabelEncoder()
y = label_encoder.fit_transform(y)
x = standard_scaler.fit_transform(x)
```

```
/usr/local/lib/python3.9/dist-packages/sklearn/preprocessing/_label.py:116: DataConversionWarning: A column-vector y was passed when a 1
  y = column_or_1d(y, warn=True)
```

168113 0

```
x
```

```
array([[-0.0542023 , -0.80035002, -1.7320405 ,  1.03033241],
       [-0.05642081,  1.61685712, -1.7320199 , -0.97056056],
       [ 0.0584287 ,  2.24561043, -1.73199929,  1.03033241],
       ...,
       [-0.05550532,  0.35655606,  1.73199929,  1.03033241],
       [ 0.21183136,  0.82881965,  1.7320199 , -0.97056056],
       [-0.05633332,  0.64717981,  1.7320405 , -0.97056056]])
```

```
y
```

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

```
from sklearn.model_selection import train_test_split
xtrain,xtest,ytrain,ytest = train_test_split(x,y)
```

```
from sklearn.svm import SVC
svc = SVC(kernel='rbf')
```

```
svc.fit(xtrain, ytrain)
```

```
SVC()
```

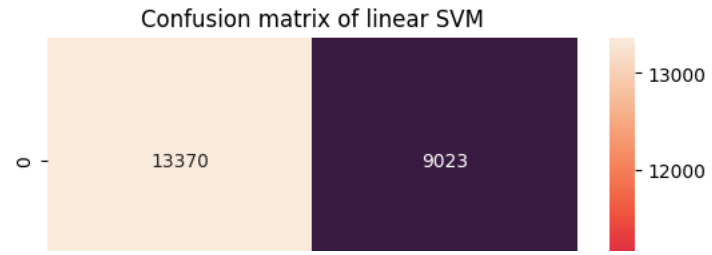
```
y_pred = svc.predict(xtest)
```

```
y_pred
```

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

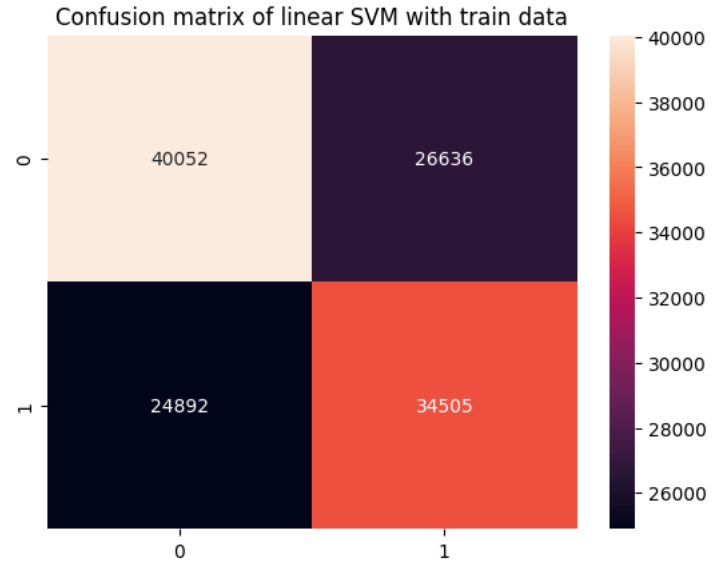
```
from sklearn.metrics import classification_report, confusion_matrix
import seaborn as sns

cm = confusion_matrix(ytest,y_pred)
sns.heatmap(cm, annot=True, fmt='d').set_title('Confusion matrix of linear SVM') # fmt='d' formats the numbers as digits, which means integer

print(classification_report(ytest,y_pred))
```

```
                precision    recall  f1-score   support

             0       0.62      0.60      0.61     22393
             1       0.56      0.58      0.57     19636

      accuracy                           0.59     42029
     macro avg       0.59      0.59      0.59     42029
  weighted avg       0.59      0.59      0.59     42029
```




```
y_pred_train = svc.predict(xtrain)

cm_train = confusion_matrix(ytrain,y_pred_train)
sns.heatmap(cm_train, annot=True, fmt='d').set_title('Confusion matrix of linear SVM with train data')

print(classification_report(ytrain,y_pred_train))
```

```
                precision    recall  f1-score   support

             0       0.62      0.60      0.61     66688
             1       0.56      0.58      0.57     59397

      accuracy                           0.59    126085
     macro avg       0.59      0.59      0.59    126085
  weighted avg       0.59      0.59      0.59    126085
```




```
from sklearn.metrics import accuracy_score
classifier_prediction = svc.predict(xtest)
print(accuracy_score(ytest,classifier_prediction))
```

```
0.588831521092579
```

✓ 4m 9s completed at 12:13 PM