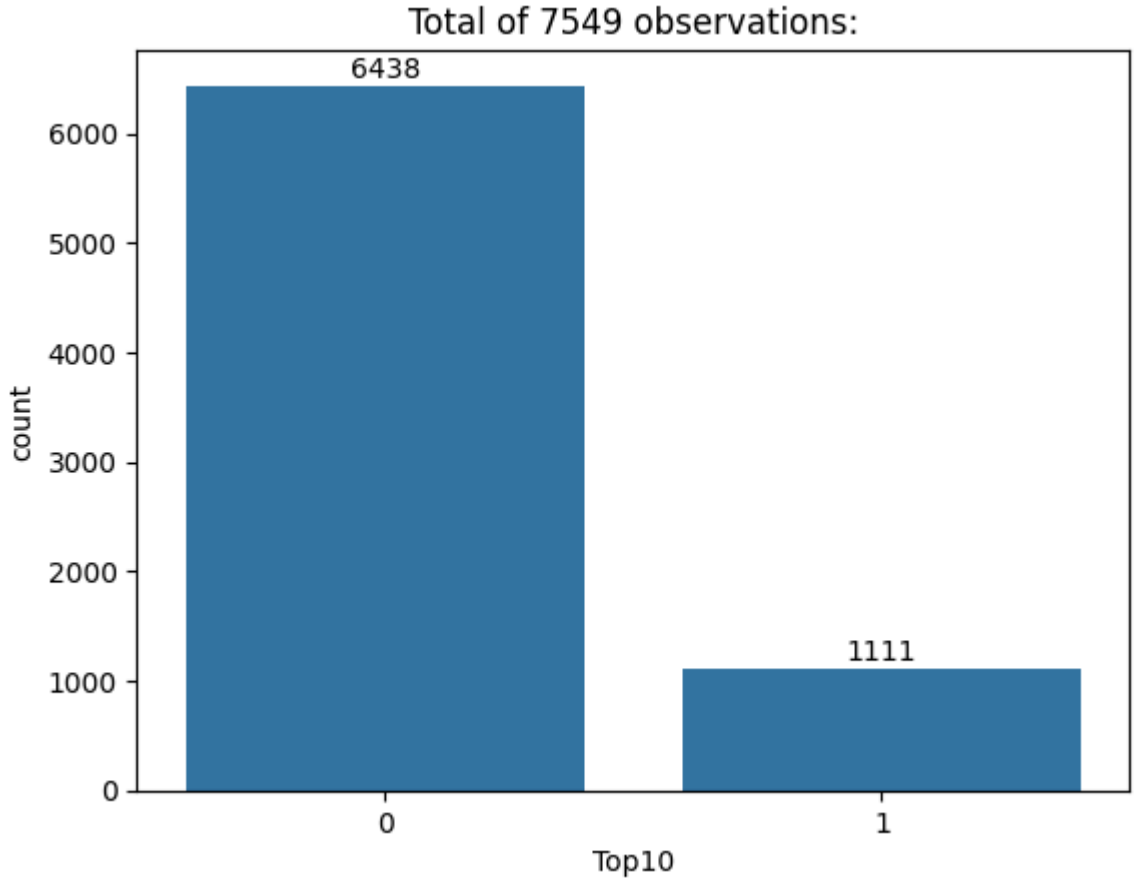
In [46]:
```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score

## CLEANING THE DATA
df = pd.read_csv("C:/Users/User1/mlprojekti/songs.csv", encoding='ISO-8859-1') #

timbres = [elem for elem in df.columns if "timbre" in elem or "confidence" in el

df.drop(timbres, axis=1, inplace = True) # Drop all columns names that contain "
df.drop(["pitch"], axis=1, inplace = True) # Drop column "pitch"


df1 = df.groupby("songID").filter(lambda x: (1 in x["Top10"].values) and (0 in x
# len(df1)) == 0. 0 songs have been on the top 10 chart in some year but have no
# This means that it is enough to simply remove duplicate songs

df = df.drop_duplicates(subset=["songID"], keep="first") # Remove duplicate song
df.drop(["artistname", "artistID", "year", "songID"], axis=1, inplace = True) #

df["songnamelength"] = df["songtitle"].str.len() # Create new column that states
df.drop(["songtitle"], axis=1, inplace = True) # Remove column for the song titl

df.dropna(axis=0, inplace=True) # Remove rows with NA values
```

In [30]:
```
#PLOTTING
ax = sns.countplot(x="Top10", data=df)
plt.title(f"Total of {len(df)} observations:")
for p in ax.patches:
    ax.annotate(f"{int(p.get_height())}", (p.get_x()+p.get_width()/2, p.get_heig
                fontsize=10, color='black', xytext=(0, 3),
                textcoords='offset points')
plt.show()
```



In [38]:
```
y = df["Top10"] # Label vector
X = df.drop(["Top10"], axis=1) # Feature vectors

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2, randc
```

In [41]:
```
clf = LogisticRegression()

clf.fit(X_train, y_train)

y_pred = clf.predict(X_test)

accuracy = accuracy_score(y_test, y_pred)

print(accuracy)
```

```
0.8622516556291391
```