División aleatoria y Validación cruzada (Train/Test)

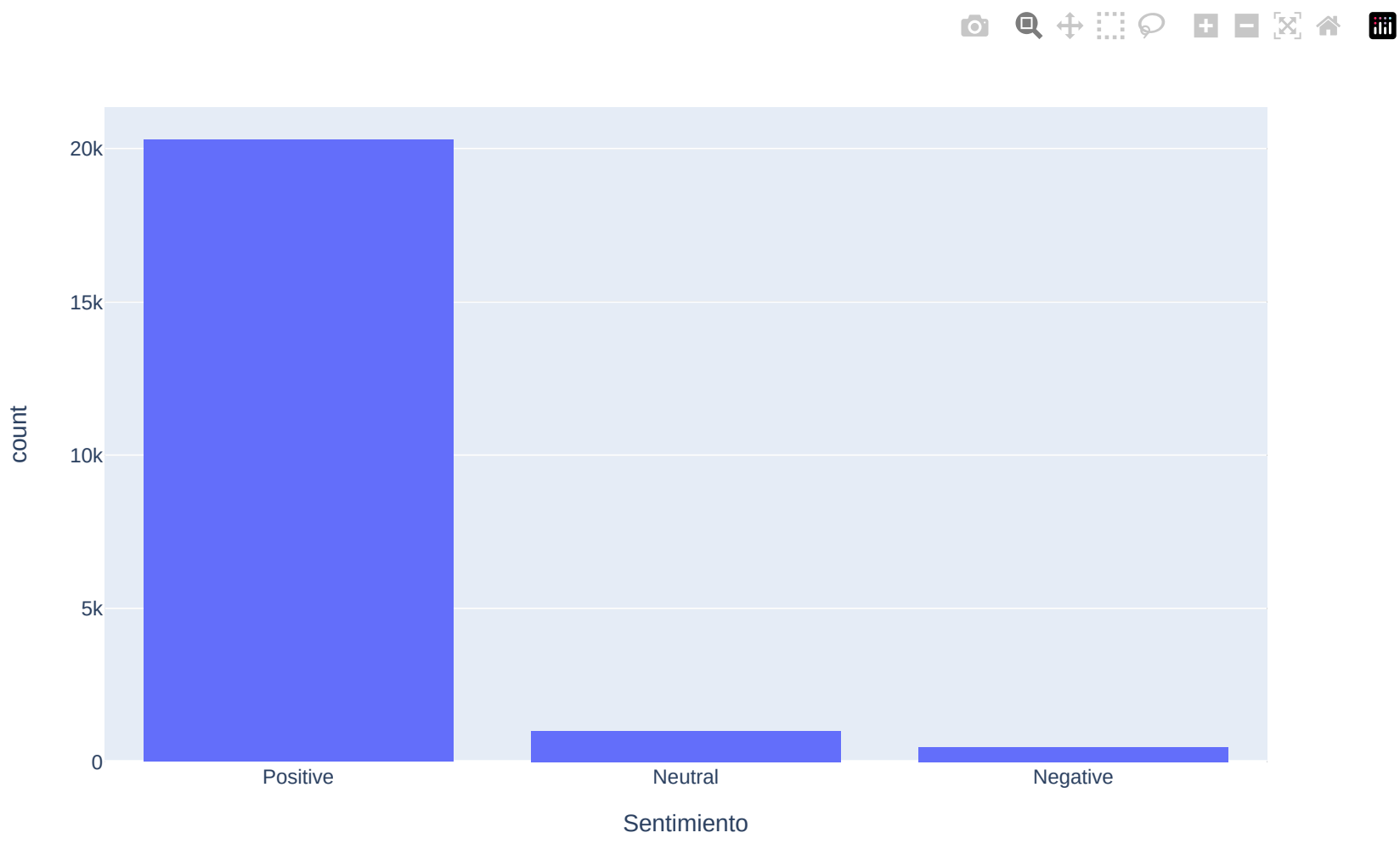
Índice

In [1]:
```
import plotly.express as px

from sklearn.model_selection import train_test_split, StratifiedShuffleSplit
from sklearn.ensemble import AdaBoostClassifier
from imblearn.over_sampling import SMOTE

import pickle
data_text_kbest = pickle.load(open("saved_feats/data_text_kbest", "rb"))
data_sentiment = pickle.load(open("saved_feats/data_sentiment", "rb"))
```

Splitter Function: train_test_split

In [2]:
```
X_train, X_test, y_train, y_test = train_test_split(
    data_text_kbest, data_sentiment, test_size=0.2, random_state=42)
```

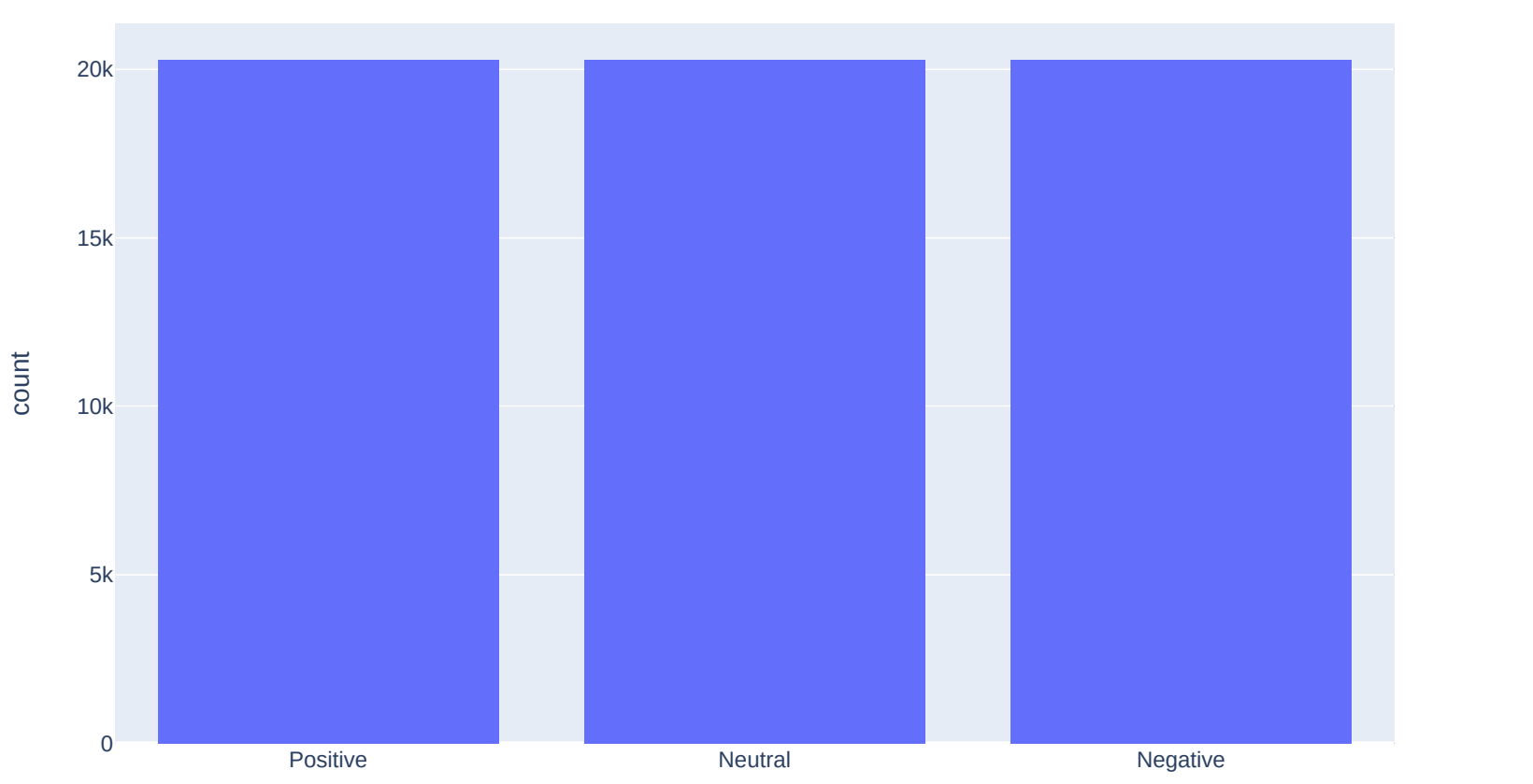
In [3]:
```
px.histogram(data_sentiment, x=y_train, labels={"x": "Sentimiento"})
```



In [4]:
```
oversample = SMOTE(random_state=42)
X_train_smt, y_train_smt = oversample.fit_resample(
    X_train, y_train)
```

In [5]:
```
adaboost = AdaBoostClassifier(n_estimators=100, random_state=42)

adaboost.fit(X_train_smt, y_train_smt)
predictedAda = adaboost.predict(X_test)
```

In [6]:
```
px.histogram(data_sentiment, x=y_train_smt, labels={"x": "Sentimiento"})
```



Splitter Class: StratifiedShuffleSplit

In [7]:
```
SSsplit = StratifiedShuffleSplit(n_splits=10, test_size=0.2, random_state=42)
for train_index, test_index in SSsplit.split(data_text_kbest,
                                             data_sentiment):
    ShuffleSplit_X_train, ShuffleSplit_X_test = data_text_kbest[
        train_index], data_text_kbest[test_index]
    ShuffleSplit_y_train, ShuffleSplit_y_test = data_sentiment[
        train_index], data_sentiment[test_index]
    # Balanceo SMOTE
    ShuffleSplit_X_train_smt, ShuffleSplit_y_train_smt = oversample.fit_resample(
        ShuffleSplit_X_train, ShuffleSplit_y_train)
    # Adaboost
    adaboost.fit(ShuffleSplit_X_train_smt, ShuffleSplit_y_train_smt)
    predictedAda = adaboost.predict(ShuffleSplit_X_test)
```

In [8]:
```
px.histogram(data_sentiment, x=ShuffleSplit_y_train_smt, labels={"x": "Sentimiento"})
```

Almacenamiento de features

In [9]:
```
pickle.dump(X_train_smt, open("saved_feats/X_train_smt", "wb"))
pickle.dump(X_test, open("saved_feats/X_test", "wb"))
pickle.dump(y_train_smt, open("saved_feats/y_train_smt", "wb"))
pickle.dump(y_test, open(
    "saved_feats/y_test", "wb"))

pickle.dump(ShuffleSplit_X_train_smt, open(
    "saved_feats/ShuffleSplit_X_train_smt", "wb"))
pickle.dump(ShuffleSplit_X_test, open("saved_feats/ShuffleSplit_X_test", "wb"))
pickle.dump(ShuffleSplit_y_train_smt, open(
    "saved_feats/ShuffleSplit_y_train_smt", "wb"))
pickle.dump(ShuffleSplit_y_test, open(
    "saved_feats/ShuffleSplit_y_test", "wb"))
```