import numpy as np *# linear algebra*

import pandas as pd *# data processing, CSV file I/O (e.g. pd.read\_csv)*

import plotly.express as px

import plotly.graph\_objs as go

from plotly.subplots import make\_subplots

import nltk

from nltk.corpus import stopwords

import tensorflow as tf

from tensorflow.keras.optimizers import Adam

from tensorflow.keras.callbacks import ModelCheckpoint

from sklearn.model\_selection import train\_test\_split

from transformers import AutoTokenizer, TFAutoModelForSequenceClassification

import os

for dirname, \_, filenames **in** os.walk('/kaggle/input'):

for filename **in** filenames:

print(os.path.join(dirname, filename))

nltk.download('stopwords')

fake\_news\_path = "/kaggle/input/fake-and-real-news-dataset/Fake.csv"

real\_news\_path = "/kaggle/input/fake-and-real-news-dataset/True.csv"

In [3]:

fake\_news = pd.read\_csv(fake\_news\_path)

real\_news = pd.read\_csv(real\_news\_path)

In [4]:

linkcode

fake\_news.head(3)

real\_news.head(3)

real = real\_news.copy()

fake = fake\_news.copy()

In [7]:

real['Label'] = 'Real'

fake['Label'] = 'Fake'

In [8]:

news = pd.concat([real, fake], axis=0, ignore\_index=True)

news.reset\_index()

news.head()

print(f"Samples available: **{**news.shape[0]**}\n**#features of dataset: **{**news.shape[1]**}**")

news\_ds = news.sample(1000).drop(['title', 'date', 'subject'], axis=1)

news\_ds.head(3)

CLASS\_NAMES = ['Fake', 'Real']

class\_mapper = {

'Fake':0,

'Real':1

}

In [12]:

linkcode

news\_ds['Label'] = news\_ds['Label'].map(class\_mapper)

news\_ds.head(3)

class\_dist = px.histogram(data\_frame=news,

y='Label',

color='Label',

title='Fake vs Real news Original dataset',

text\_auto=True)

class\_dist.update\_layout(showlegend=False)

class\_dist.show()