**Supported Findings.ipynb**

**# # Data cleaning and EDA**

pip install plotly

pip install notebook-as-pdf

import pandas as pd

import glob

import matplotlib

import plotly.express as px

import matplotlib.pyplot as plt

import warnings

warnings.filterwarnings("ignore")

import seaborn as sns

%matplotlib inline

#Links to datasets

url\_incidents = "https://github.com/NissyM/CrowdDoing/raw/main/California\_Fire\_Incidents.csv"

path="https://github.com/NissyM/CrowdDoing/raw/main/"

list\_files = []

#file\_list = glob.glob(path + "/\*.csv")

df\_incidents = pd.read\_csv(url\_incidents)

for i in glob.glob("climate\_data\_\*"):

list\_files.append(pd.read\_csv(i))

csv\_merged = pd.concat(list\_files, ignore\_index=True)

csv\_merged.to\_csv('ClimateData.csv', index=False)

#print(csv\_merged.tail())

# change the datatype of date field to Date

csv\_merged['Date'] = pd.to\_datetime(csv\_merged['Date'])

#print("Output:")

#print(result)

#Append the data and create a climate dataset with averaged values grouped by Counties

averaged\_csv=csv\_merged.groupby(["County", csv\_merged["Date"].dt.strftime('%Y')])['Temperature','Wind'].mean().reset\_index()

#csv\_merged.groupby(csv\_merged['County','Date'].dt.strftime('%Y'))['Temperature'].mean().sort\_values()

print(averaged\_csv.head(12))

df2 = df\_incidents.rename(columns={'Counties': 'County','ArchiveYear': 'Date'})

# datatype changed back to int so as to match with the Forestfires dataset

averaged\_csv['Date']=averaged\_csv['Date'].astype(int)

# Merge both datasets and select required columns using outer join on Year and County

df3 = pd.merge(averaged\_csv, df2[['AcresBurned','AdminUnit','Date','County','Name','Started']], on=['County','Date'], how='outer')

# Drop rows with missing columns after join that are missing for columns from Forestfires dataset

rslt\_df= df3[df3['Date'] > 2012].dropna()

#Include a calculated column of count of occurences of forest fires in each county

rslt\_df['count'] = rslt\_df.groupby('County')['County'].transform('count')

#len(rslt\_df['County'].unique()) ==> 54 counties

# Ploting Total WildFire Occurences over time

plt.figure(figsize=(10, 5))

sns.countplot(rslt\_df['Date'])

plt.title('Total WildFire Occurences over time')

plt.xlabel('Years')

plt.ylabel('No. of Occurences')

plt.show()

# Ploting County wise Occurences of Forest Fires each year

fig=px.histogram(rslt\_df, x='Date',color="County", barmode='group',labels={"Date": "Years", "count": "Occurences of ForestFires"}

).update\_layout(

title={"text": "County wise Occurences of Forest Fires each year", "x": 0.5}, yaxis\_title="Occurences")

fig.show()

# Ploting Trends

import matplotlib.pyplot as plt

import plotly.express as px

%matplotlib inline

#matplotlib.style.use('fivethirtyeight')

#df[df['date'] == 6]['location'].value\_counts()

top\_10 = rslt\_df[rslt\_df['Date'] == 2017]['County'].value\_counts()[:50]

#print(top\_10.head(10))

#print(top\_10.tail(10))

top\_10.plot(kind='bar',figsize=(15,8),color="orange")

plt.title('Counties with Wildfire Occurences in year 2017')

plt.show()

test\_df=rslt\_df[rslt\_df['County'].isin(['Riverside','Kern','San Luis Obispo','Yolo','San Mateo'])]

#print(test\_df.head(10))

dftest = test\_df.groupby(['County','Date']).agg({'Temperature': 'mean', 'Wind': 'mean'}).reset\_index()

print(dftest.head(10))

px.line(data\_frame = dftest, x = 'Date', y = 'Temperature',color = 'County',labels={"Date": "Years", "Temperature": "Temperature (°F)"},title='Temperature Trend over the years')

px.line(data\_frame = dftest, x = 'Date', y = 'Wind',color = 'County',labels={"Date": "Years", "Wind": "Wind Speed (°F)"},title='Wind Speed Trend over the years')

# Ploting Graphs of Temperature and Wind Speed

import plotly.express as px

Rs\_df = rslt\_df[(rslt\_df['Date'] == 2017) & rslt\_df['County'].isin(['Riverside','Kern','San Luis Obispo','Yolo','San Mateo'])]

df1 = Rs\_df.groupby("County").agg({'Temperature': 'mean',

'Wind': 'mean'}).reset\_index()

#test=Rs\_df.groupby("County")['County'].value\_counts()['Temperature','Wind','Precipitation','AcresBurned'].mean().reset\_index()

print(df1)

fig = px.bar(df1,x="County", y="Temperature", title="Temperature in the Top and Low WildFire Occurence County in 2017",color='County');

fig.show()

wind\_fig=px.bar(df1,x="County", y="Wind", title="Wind Speed in the Top and Low WildFire Occurence County in 2017",color='County');

wind\_fig.show()

#test.plot.bar(x="County", y="['County'].value\_counts()", rot=70, title="AcresBurned in the Top 3 WildFire Occurence County in 2017");

# Ploting Correlation

import numpy as np

pearsoncorr = rslt\_df.corr(method='pearson')

print(pearsoncorr)

sns.heatmap(pearsoncorr,

xticklabels=pearsoncorr.columns,

yticklabels=pearsoncorr.columns,

cmap='RdBu\_r',

annot=True,

linewidth=0.5)

# Ploting Number of Acres Burned in Thousands by Year Between 2013-2019

rslt\_df.plot(kind='scatter', x='AcresBurned', y='Date', figsize=(13,8), s=5)

plt.xlabel('Number of Acres Burned in Thousands')

plt.ylabel('Year of Fire Incident')

plt.title('Number of Acres Burned in Thousands by Year Between 2013-2019');

# Ploting Acres Burned by County Between 2013-2019

rslt\_df[['County', 'AcresBurned']].groupby('County').sum().head(10).plot(kind='bar', figsize=(12,7))

plt.xlabel('County Name')

plt.ylabel('Acres Burned in Thousands')

plt.title('Acres Burned by County Between 2013-2019');