"""
CS230 Spring 2022 - Final Project Interactive Data-Explorer
Author: Ash Apsangi
Date: 5/11/2022
Description: Web application using streamlit UI that displays various charts plotted
from given restaurant dataset

"""

```
import streamlit as st
from streamlit_option_menu import option_menu
import plotly.express as px
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from fuzzywuzzy import process, fuzz
from plotly.offline import init_notebook_mode, iplot
import os


st.set_option('deprecation.showPyplotGlobalUse', False)


st.set_page_config(page_title="CS230 Spring 2022 - Aishwarya Apsangi",layout='wide')


def create_sidebar():
    with st.sidebar:
        choose = option_menu("Ash's Data Explorer",
                             ["Home Page", "Data Collected", "Latitude Data", "Fast Food Restaurants
Map", "Pie Chart", "Bar Chart","Choropleth Map","Iterate"],
                             icons=['activity','table', 'bar-chart', 'bar-chart', 'activity','pie-chart ',
'bar-chart','activity','activity'],
                             menu_icon="app-indicator", default_index=0,
                             styles={
                                 "container": {"padding": "5!important", "background-color": "#fafafa"},
                                 "icon": {"color": "orange", "font-size": "25px"},
                                 "nav-link": {"font-size": "16px", "text-align": "left", "margin": "0px",
                                              "--hover-color": "#eee"},
                                 "nav-link-selected": {"background-color": "#02ab21"},
                             }
```

```
                    )


    data = pd.read_csv('./dataset/FastFoodRestaurants.csv')
    data = data[["address", "city", "country", "latitude", "longitude", "name", "postalCode",
"province"]]
    data.head()
    # Now sort list with unique names
    sorted(data.name.unique())
    data['lowername'] = data['name'].apply(lambda x: x.lower().strip())
    unique_names = sorted(data.lowername.unique())
    # unique_names

    restaurants_counts_by_name = data.lowername.value_counts()
    restaurants_counts_by_name = restaurants_counts_by_name[restaurants_counts_by_name
> 250]
    # restaurants_counts_by_name
    restaurants_list_by_name_counts = list(restaurants_counts_by_name.index)
    # restaurants_list_by_name_counts

    if choose == "Home Page":
        st.markdown(""" <style> .font {
        font-size:50px ; font-family: 'Cooper Black'; color: #87023E;}
        </style> """, unsafe_allow_html=True)
        st.markdown('<p class="font">Fast Food Restaurants Across America</p>',
unsafe_allow_html=True)
        st.image("istockphoto.jpg")
        st.text("")
        st.text("")
        st.subheader("Fast food occupies an outsize place in American culture. The grease runs
through our national veins. But the food itself — the White Castle sliders, the KFC buckets, the
Whoppers and Baconators and Egg McMuffins — is only part of the story.")



    if choose == "Data Collected":
        st.markdown(""" <style> .font {
        font-size:40px ; font-family: 'Cooper Black'; color: #001075;}
        </style> """, unsafe_allow_html=True)
        st.markdown('<p class="font">DATA TABLE ALTERATIONS:</p>',
unsafe_allow_html=True)
        original_title = '<p style="font-family:serif; color:#d33682; ">Famished and in a hurry?
Discover the best fast food restaurants in the US, ace on-the-go joints for burgers, fries and
more based on the table below.</p>'
```

```
    st.markdown(original_title, unsafe_allow_html=True)
    df = pd.read_csv("dataset/Datafiniti_Fast_Food_Restaurants.csv")
    st.write('### Full Dataset', data)
    selected_indices = st.multiselect('Select rows:', data.index)
    selected_rows = data.loc[selected_indices]
    st.write('### Selected Rows', selected_rows)




if choose == "Latitude Data":
    st.markdown(""" <style> .font {
    font-size:30px ; font-family: 'Cooper Black'; color: #001075;}
    </style> """, unsafe_allow_html=True)
    st.markdown('<p class="font">Skew Based On Coordinates</p>', unsafe_allow_html=True)
    latitude = st.slider('Please enter the latitude', 0, 130, 25)
    st.write(f"The latitude is {latitude}")
    @st.cache

    def load_latitude_data(nrows):
        data = pd.read_csv('FastFoodRestaurants.csv',nrows=nrows)
        return data
    names_data = load_latitude_data(10000)

    #WeeklyDemand Data
    st.subheader('Latitude Data Of Fast Food Restaurants in the US')
    st.bar_chart(names_data['latitude'])


    def load_data(nrows):
        data = pd.read_csv('Datafiniti_Fast_Food_Restaurants.csv', nrows=nrows)
        return data
    names_data = load_data(10000)
    df = pd.DataFrame(names_data[:200], columns = ['latitude','longitude'])
    df.hist()
    st.pyplot()
    st.text(" ")
    st.text(" ")
    st.subheader('HISTOGRAM VIEW:')
    st.line_chart(df)
    chart_data = pd.DataFrame(names_data[:100], columns=['latitude', 'longitude'])
    st.area_chart(chart_data)
```

```
    if choose == "Fast Food Restaurants Map":
        df = pd.read_csv("dataset/Datafiniti_Fast_Food_Restaurants.csv")
        fig = px.scatter_geo(df,lat = 'latitude',lon = 'longitude', hover_name = "name")
        fig.update_layout(title = 'Fast Food Restaurants in the US Map', title_x = 0.5)
        fig.show()


    if choose == "Pie Chart":
        with st.container():
                st.markdown(""" <style> .font {
                font-size:40px ; font-family: 'Cooper Black'; color: #001075;}
                </style> """, unsafe_allow_html=True)
                st.markdown('<p class="font">Fast Food Restaurants Top Counts By Name</p>',
unsafe_allow_html=True)
                data = pd.read_csv('dataset/Datafiniti_Fast_Food_Restaurants.csv')
                data = data[["address", "city", "country", "latitude", "longitude", "name", "postalCode",
"province"]]
                sorted(data.name.unique())
                data['lowername'] = data['name'].apply(lambda x: x.lower().strip())
                unique_names = sorted(data.lowername.unique())
                restaurants_counts = data.lowername.value_counts()
                restaurants_counts = restaurants_counts[restaurants_counts > 250]
                #restaurants_counts
                restaurants_list = list(restaurants_counts.index)
                #restaurants_list
                fig, ax = plt.subplots(figsize=(30, 30))
                ax.pie(restaurants_counts, labels=restaurants_list, autopct="%1.1f%%",
textprops={'fontsize': 70})
                ax.axis("equal")
                st.pyplot(fig)
                name = {'taco bell':'red', 'mcdonalds':'green','burger king':'purple','jack in the
box':'pink','wendys':'green','subway':'grey','arbys':'yellow'}
                labels = list(name.keys())
                handles = [plt.Rectangle((0,0),1,1, color=name[label]) for label in labels]
                plt.legend(handles, labels)



    if choose == "Bar Chart":
        with st.container():
                CountLegend = ['Count']
                st.markdown(""" <style> .font {
```

```
            font-size:40px ; font-family: 'Cooper Black'; color: #001075;}
            </style> """, unsafe_allow_html=True)
            st.markdown('<p class="font">Fast Food Restaurants Top Counts By Province</p>',
unsafe_allow_html=True)
            data = pd.read_csv('./dataset/FastFoodRestaurants.csv')
            data = data[["address", "city", "country", "latitude", "longitude", "name", "postalCode",
"province"]]
            restaurants_counts_by_province = data.province.value_counts()
            restaurants_counts_by_province =
restaurants_counts_by_province[restaurants_counts_by_province > 250]
            # restaurants_counts
            restaurants_list_by_province_count = list(restaurants_counts_by_province.index)
            # restaurants_list

            fig = plt.figure(figsize=(5, 5))
            ax = fig.add_axes([0, 0, 1, 1])
            CountLegend = ['Count']
            ax.bar(restaurants_list_by_province_count, restaurants_counts_by_province)
            plt.xlabel('Province', fontsize=16)
            plt.ylabel('Count', fontsize=16)
            plt.legend(CountLegend, loc=1)
            st.pyplot(fig)
            st.markdown(""" <style> .font {
            font-size:50px ; font-family: 'Cooper Black'; color: #001075;}
            </style> """, unsafe_allow_html=True)
            st.markdown('<p class="font">Fast Food Restaurants Top by Count Brand-wise</p>',
unsafe_allow_html=True)

            fig = plt.figure(figsize=(5, 5))
            ax = fig.add_axes([0, 0, 1, 1])
            ax.bar(restaurants_list_by_name_counts, restaurants_counts_by_name)
            plt.xlabel('Brand', fontsize=16)
            plt.ylabel('Count', fontsize=16)
            plt.legend(CountLegend, loc=1)
            st.pyplot(fig)

    if choose == "Choropleth Map":
        state_vals = data['province'].value_counts().index.tolist()
        state_counts = data['province'].value_counts()

        data= [dict(type='choropleth',
                    locations = state_vals, # Spatial coordinates
                    z = state_counts, # Data to be color-coded
                    locationmode = 'USA-states', # set of locations match entries in `locations`
```

```
                colorscale = 'Reds',
                marker_line_color = 'grey',
                colorbar_title = "No. of Fast Fast Restaurants"
            )]
        layout = dict(title = 'Sorting the Restaurants by Color, State, and Shade',
                  geo = dict(scope='usa'))
        iplot(dict(data=data, layout=layout))




    if choose == "Iterate":
        with st.container():
            st.title('Iterate using for loop')
            similar_name_list = list()
            for restuarant in restaurants_list_by_name_counts:
                query = data['lowername'].unique()
                results = process.extract(restuarant, query, limit=9, scorer=fuzz.token_sort_ratio)
                similar_name_list.append(results)
            similar_name_list



def print_on_streamlit():
    with st.container():
        create_sidebar()



if __name__ == '__main__':
    print_on_streamlit()
```