

Príloha A

```
import pandas as pd
from flask import Flask
from flask import render_template
import json
import requests
import time

data_path = './input/'
n_samples = 5760

def get_age_segment(age):
    if age < 18:
        return '18-'
    elif age <= 22:
        return '18-22'
    elif age <= 30:
        return '23-30'
    elif age <= 40:
        return '31-40'
    elif age <= 50:
        return '41-50'
    elif age <= 59:
        return '51-59'
    else:
        return '60+'

def get_location(GPS):
    gps=', '.join(str(x) for x in GPS)
    response = requests.get('https://maps.googleapis.com/maps/api/geocode/json?latlng='+gps+'&language=sk')
    d1 = response.json()
    for d2 in d1['results']:
        for d3 in d2['address_components']:
            for k,v in d3.items():
                if 'kraj' in v: return v

with open (data_path + '/geojson/sk_kraje.json') as data_file:
    kraje_json = json.load(data_file)

app = Flask(__name__)

@app.route("/")
def index():
    return render_template('index.html')

@app.route("/data")
def get_data():
    starttime=time.time()
    januar= pd.read_csv(data_path + 'm1.csv')
```

```

februar= pd.read_csv(data_path + 'm2.csv')
marec = pd.read_csv(data_path + 'm3.csv')
sur = pd.read_csv(data_path + 'suradnice.csv')
sur['latitude']=sur['latitude'].round(2)
sur['longitude']=sur['longitude'].round(2)

df = pd.merge(januar, februar, how = "left", on = "GEO1")
df = pd.merge(marec,sur,how = "left", on = "GEO1")
df = df[df['longitude'] != 0].sample(n=n_samples)

top_10_brands_en = {"Alcatel":"Alcatel", "samsung":"Samsung", "Sony":"Sony", "Apple":"Apple",
                    "Huawei":"Huawei", "Lenovo":"Lenovo", "HTC":"HTC",
                    "LG":"LG", "CAT":"CAT"}

df['device_en'] = df['Device'].apply(lambda Device: top_10_brands_en[Device]
                                     if (Device in top_10_brands_en) else 'Other')
df['age_segment'] = df['Age'].apply(lambda age: get_age_segment(age))
df['location'] = df.apply(lambda row: get_location([row['longitude'], row['latitude']]), axis=1)
drop_en = { int("1"): "Yes" }
df['drop'] = df['Drop'].apply(lambda Drop: drop_en[Drop]
                              if (Drop in drop_en) else 'No')
df['duration'] = df['Duration'].apply(lambda Duration : round(Duration))
df['count'] = df['Count'].apply(lambda Count : round(Count))
cols_to_keep = ["longitude", "latitude", 'device_en', 'age_segment', "Segment", 'location', 'drop', 'Network', 'duration', 'count']
df_clean = df[cols_to_keep].dropna()
print(df_clean)
print("Celkovy cas: ", time.time()-starttime, " pre dataset o velkosti: " ,n_samples)
return df_clean.to_json(orient='records')

if __name__ == "__main__":
    app.run(host='127.0.0.1', port=5000, debug=True)

```