Capital Bikeshare: Anlayse und Prognose der Ausleihvorgänge

Ziel

Es sollen die Ausleihvoränge aus den Jahren 2015-2017 analysiert werden.

1. Auffälligkeiten: Explorative Datenanalyse und Visualisierung
2. Prognose der Ausleihvorgänge nach Wahl (eigenem Ermessen) insgesamt oder pro Station und pro Tag oder pro Stunde

Herunterladen und Parsen der Orignaldaten (raw)

Hinweis: Die Notebooks sind so aufgebaut, dass sie zu einer Verarbeitungs-Pipeline gehören und in der Reihenfolge der Nummern (Prefixe) ausgeführt sollten, da spätere Notebooks (die mit einer größeren Anfangsnummer) Daten aus den vorherigen Notebooks verwenden. Nur Notebooks mit ganzen *10*er-Nummern gehören zur eigentlichen Verarbeitungs-Pipeline.

In [1]:

```
# Die Trip-Daten liegen in einem s3-Bucket - Verwendung einer speziellen Bibliothek (bo
to3)
# Sowie eine Bibliotheken (wget und requests) für das Herunteladen und Speichern von Da
teien per www.
# Installieren falls nicht verfügbar!
!pip install boto3
!pip install wget
!pip install requests
```

```
Collecting boto3
  Downloading boto3-1.14.50-py2.py3-none-any.whl (129 kB)
Collecting jmespath<1.0.0,>=0.7.1
  Downloading jmespath-0.10.0-py2.py3-none-any.whl (24 kB)
Collecting s3transfer<0.4.0,>=0.3.0
  Downloading s3transfer-0.3.3-py2.py3-none-any.whl (69 kB)
Collecting botocore<1.18.0,>=1.17.50
  Downloading botocore-1.17.50-py2.py3-none-any.whl (6.6 MB)
Requirement already satisfied: python-dateutil<3.0.0,>=2.1 in c:\users\sim
on\anaconda3\lib\site-packages (from botocore<1.18.0,>=1.17.50->boto3) (2.
8.1)
Collecting docutils<0.16,>=0.10
  Downloading docutils-0.15.2-py3-none-any.whl (547 kB)
Requirement already satisfied: urllib3<1.26,>=1.20; python_version != "3.
4" in c:\users\simon\anaconda3\lib\site-packages (from botocore<1.18.0,>=
1.17.50->boto3) (1.25.9)
Requirement already satisfied: six>=1.5 in c:\users\simon\anaconda3\lib\si
te-packages (from python-dateutil<3.0.0,>=2.1->botocore<1.18.0,>=1.17.50->
boto3) (1.15.0)
Installing collected packages: jmespath, docutils, botocore, s3transfer, b
oto3
  Attempting uninstall: docutils
    Found existing installation: docutils 0.16
    Uninstalling docutils-0.16:
      Successfully uninstalled docutils-0.16
Successfully installed boto3-1.14.50 botocore-1.17.50 docutils-0.15.2 jmes
path-0.10.0 s3transfer-0.3.3
Collecting wget
  Downloading wget-3.2.zip (10 kB)
Building wheels for collected packages: wget
  Building wheel for wget (setup.py): started
  Building wheel for wget (setup.py): finished with status 'done'
  Created wheel for wget: filename=wget-3.2-py3-none-any.whl size=9686 sha
256=d4a99d35f4ac8c96603190f5fbdeb7de801eb02e16eee67dca13db10b96bd541
  Stored in directory: c:\users\simon\appdata\local\pip\cache\wheels\bd\a8
\c3\3cf2c14a1837a4e04bd98631724e81f33f462d86a1d895fae0
Successfully built wget
Installing collected packages: wget
Successfully installed wget-3.2
Requirement already satisfied: requests in c:\users\simon\anaconda3\lib\si
te-packages (2.24.0)
Requirement already satisfied: idna<3,>=2.5 in c:\users\simon\anaconda3\li
b\site-packages (from requests) (2.10)
Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1 in
c:\users\simon\anaconda3\lib\site-packages (from requests) (1.25.9)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\simon\anacon
da3\lib\site-packages (from requests) (2020.6.20)
Requirement already satisfied: chardet<4,>=3.0.2 in c:\users\simon\anacond
a3\lib\site-packages (from requests) (3.0.4)
```

In [2]:

```
import os
import re
import glob
import wget
import json
import requests
import datetime
import pandas as pd
import boto3
from botocore import UNSIGNED
from botocore.config import Config
from zipfile import ZipFile
```

In [3]:

```
pd.__version__
```

Out[3]:

```
'1.0.5'
```

Herunterladen der Daten

Es wurden manuell Trip-Daten aus den Jahren 2015, 2016 und 2017 heruntergeladen!

In [4]:

```
RAW_DATA_PATH = '../data/raw/'
TRIP_DATA_PATH = RAW_DATA_PATH + 'tripdata/'
TRIP_ZIP_FILE_SUFFIX = 'tripdata.zip'
TRIP_DATA_CSV_FILE_PATTERN = '^[0-9]{4}.*tripdata\.csv$'
PARSED_DATA_PATH = '../data/'
RAW_TRIPS_FILE = 'trips_raw.pkl'
RAW_WEATHER_FILE = 'weather_raw.pkl'
ALT_WEATHER_FILE = 'weather_alt.pkl'
WEATHER_DATA_PATH = RAW_DATA_PATH + 'weather/'
```

In [5]:

```
URL_TEMPLATE_WEATHER_DATA = 'https://api.meteostat.net/v1/history/hourly?station=72405&
start={}&end={}&time_zone=Europe/Berlin&time_format=Y-m-d%20H:i&key=xPVZEykm'

URL_ALT_WEATHER_DATA = 'https://open.meteostat.net/hourly/72405.csv.gz'
COLS_ALT_WEATHER_DATA = ['date', 'hour', 'temperature', 'dewpoint',
                     'precipitation', 'precipitation_3', 'precipitation_6',
                     'snowdepth', 'windspeed', 'peakgust', 'winddirection', 'humidity',
'pressure', 'condition']

WEATHER_COLS_TO_DROP = ['precipitation_3', 'precipitation_6', 'snowdepth', 'peakgust',
'condition']
```

In [6]:

```
# Mapping auf praktische und einheitliche Namen der Merkmale
# Leerzeichen in Namen vermeiden, einheitliche Kleinschreibung wird hier verwendet

TRIP_COLS_NAME_MAP = {
        'Duration': 'duration',
        'Start date': 'start_ts',
        'End date': 'end_ts',
        'Start station number': 'start_station_id',
        'Start station': 'start_station_name',
        'End station number': 'end_station_id',
        'End station': 'end_station_name',
        'Bike number': 'bike_number',
        'Member': 'member'
    }
```

In [7]:

```
TRIP_DATA_PARSE_DATES = [1,2]
```

In [8]:

```
BUCKET_NAME = 'capitalbikeshare-data'
```

In [9]:

```
# Setze auf None, um alle verfügbaren Dateien herunterzuladen
TRIP_FILES_TO_LOAD = {
    '2015-capitalbikeshare-tripdata.zip',
    '2016-capitalbikeshare-tripdata.zip',
    '2017-capitalbikeshare-tripdata.zip'
}
```

In [ ]:

```

```

In [10]:

```
# Lade alle (noch nicht geladenen) Trip-Data-Dateien aus dem s2 Bucket (BUCKET_NAME sie
he oben)
def load_trip_data(delta_only=True, target_path=TRIP_DATA_PATH, trip_files_to_load=TRIP
_FILES_TO_LOAD):
    # create target path if it does not exist
    if not os.path.exists(target_path):
        print('Creating dir', target_path, '...')
        os.makedirs(target_path)

    # init s3 bucket access
    s3 = boto3.resource('s3', config=Config(signature_version=UNSIGNED))
    bucket = s3.Bucket(BUCKET_NAME)


    # load files
    ignored = 0
    skipped = 0
    downloaded = 0
    for obj in bucket.objects.all():
        source_file = obj.key
        target_file = target_path+source_file

        if (source_file.endswith(TRIP_ZIP_FILE_SUFFIX)):

            # process file only if no file_list specified or source file in file list
            if (trip_files_to_load is None or source_file in trip_files_to_load):
                if (os.path.exists(target_file)):
                    #print('Skipping existing file', source_file, '...')
                    skipped += 1
                else:
                    print('Downloading', source_file, '...')
                    bucket.download_file(source_file, target_file)
                    downloaded += 1

            else:
                ignored += 1

    print('tripdata-files ignored:', ignored)
    print('tripdata-files skipped:', skipped)
    print('tripdata-files downloaded:', downloaded)
```

In [ ]:

```

```

In [15]:

```
# Lade die in den Trip-Data-Files enthaltenen csv-Dateien (teilweise mehrere) in eine L
iste aus DataFrames
# konkateniere alle DataFrames zu einem und verwendete einheitliche Spaltennamen ohne L
eerhzeichen
def read_raw_trip_data(target_path=TRIP_DATA_PATH, trip_files_to_load=TRIP_FILES_TO_LOA
D, check_content=True):
    load_trip_data(target_path=target_path, trip_files_to_load=trip_files_to_load)
    df_list = []
    schema_index = 0
    # walk through all zip files
    for file in sorted(glob.glob(target_path+'*'+TRIP_ZIP_FILE_SUFFIX)):
        print('Unzipping', file)
        # a zip file may contain multiple csv files
        with ZipFile(file) as zfile:
            for name in zfile.namelist():
                if re.match(TRIP_DATA_CSV_FILE_PATTERN, name):
                    print('Reading file', name , '...')

                    df_trips = pd.read_csv(zfile.open(name), parse_dates=TRIP_DATA_PARS
E_DATES)

                    df_trips.rename(columns=TRIP_COLS_NAME_MAP, inplace=True)

                    if check_content:
                        check_trip_data(df_trips)

                    df_list.append(df_trips)


                else:
                    print('Skipping file', name, '!')
    # return list of trip-DataFrames (raw format)
    return df_list
```

In [16]:

```
def check_trip_data(df):
    print('trips:', df.shape[0],
          '\tmin start date:', df['start_ts'].min(),
          '\tmax start date: ', df['start_ts'].max())
```

In [ ]:

```

```

In [17]:

```
def get_trip_data(target_path=TRIP_DATA_PATH, trip_files_to_load=TRIP_FILES_TO_LOAD):
    df_list = read_raw_trip_data(target_path=target_path, trip_files_to_load=trip_files
_to_load)
    print('Concatenating', len(df_list), 'dataframes ...')
    # concat all individual dataframes
    df_trips = pd.concat(df_list)
    print('Total trips combined:', df_trips.shape[0])
    print('Done.')
    return df_trips
```

In [ ]:

```
```

In [ ]:

```
df_trips_raw = get_trip_data()
```

```
Creating dir ../data/raw/tripdata/ ...
Downloading 2015-capitalbikeshare-tripdata.zip ...
Downloading 2016-capitalbikeshare-tripdata.zip ...
Downloading 2017-capitalbikeshare-tripdata.zip ...
tripdata-files ignored: 36
tripdata-files skipped: 0
tripdata-files downloaded: 3
Unzipping ../data/raw/tripdata\2015-capitalbikeshare-tripdata.zip
Reading file 2015Q1-capitalbikeshare-tripdata.csv ...
trips: 423719   min start date: 2015-01-01 00:02:44     max start date:  2
015-03-31 23:59:52
Reading file 2015Q2-capitalbikeshare-tripdata.csv ...
trips: 999818   min start date: 2015-04-01 00:02:23     max start date:  2
015-06-30 23:58:37
Reading file 2015Q3-capitalbikeshare-tripdata.csv ...
trips: 1056366  min start date: 2015-07-01 00:00:25     max start date:  2
015-09-30 23:57:53
Reading file 2015Q4-capitalbikeshare-tripdata.csv ...
trips: 706003   min start date: 2015-10-01 00:01:30     max start date:  2
015-12-31 23:57:57
Unzipping ../data/raw/tripdata\2016-capitalbikeshare-tripdata.zip
Reading file 2016Q1-capitalbikeshare-tripdata.csv ...
trips: 552399   min start date: 2016-01-01 00:06:58     max start date:  2
016-03-31 23:59:42
Reading file 2016Q2-capitalbikeshare-tripdata.csv ...
trips: 942333   min start date: 2016-04-01 00:00:30     max start date:  2
016-06-30 23:59:50
Reading file 2016Q3-capitalbikeshare-tripdata.csv ...
```

In [ ]:

```
```

In [19]:

```
df_trips_raw.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 10277677 entries, 0 to 815263
Data columns (total 9 columns):
 #   Column              Dtype         
---  ------              -----         
 0   duration            int64         
 1   start_ts            datetime64[ns]
 2   end_ts              datetime64[ns]
 3   start_station_id    int64         
 4   start_station_name  object        
 5   end_station_id      int64         
 6   end_station_name    object        
 7   bike_number         object        
 8   Member type         object        
dtypes: datetime64[ns](2), int64(3), object(4)
memory usage: 784.1+ MB
```

In [20]:

```
df_trips_raw.to_pickle(PARSED_DATA_PATH+RAW_TRIPS_FILE)
```

In [ ]:

```

```

In [ ]:

```

```

In [21]:

```
def load_weather_data_year(year, replace=False, target_path=WEATHER_DATA_PATH, url_temp
late=URL_TEMPLATE_WEATHER_DATA):

    # create target path if it does not exist
    if not os.path.exists(target_path):
        print('Creating dir', target_path, '...')
        os.makedirs(target_path)

    target_file = target_path + 'weather_' + str(year) + '.json'

    if (replace & os.path.exists(target_file)):
        os.remove(target_file)

    if (not os.path.exists(target_file)):
        url = url_template.format(str(year)+'-01-01', str(year)+'-12-31')
        print('Downloading',  url, '...')
        weather_json = requests.get(url).json()
        with open(target_path+'weather_'+str(year)+'.json', 'w') as outfile:
            json.dump(weather_json, outfile)
    else:
        print('File ', os.path.basename(target_file), 'already downloaded!')
```

In [ ]:

```

```

In [22]:

```
def load_weather_data(year_start, year_end, replace=False, target_path=WEATHER_DATA_PAT
H, url_template=URL_TEMPLATE_WEATHER_DATA):
    for year in range(year_start, year_end+1):
        load_weather_data_year(year, replace=replace, target_path=target_path, url_temp
late=url_template)
```

In [23]:

```
def parse_weather_data(file):
    data = json.load(open(file))
    df = pd.DataFrame(data['data'])
    df.drop(WEATHER_COLS_TO_DROP+['time'], axis=1, inplace=True)
    df.rename(columns={'time_local': 'time_ts'}, inplace=True)
    return df
```

In [24]:

```
def get_weather_data_for_trips(df_trips):

    target_path = WEATHER_DATA_PATH

    load_weather_data(
        year_start=df_trips['start_ts'].min().year,
        year_end=df_trips['start_ts'].max().year,
        target_path=target_path)

    df_list = []
    # walk through all zip files
    for file in sorted(glob.glob(target_path+'weather*.json')):
        df_list.append(parse_weather_data(file))
        print('parsed', file, 'with', df_list[-1].shape[0], 'rows')


    df = pd.concat(df_list)

    df['time_ts'] = pd.to_datetime(df['time_ts'])

    return df
```

In [25]:

```
df_weather = get_weather_data_for_trips(df_trips_raw)
```

```
Creating dir ../data/raw/weather/ ...
Downloading https://api.meteostat.net/v1/history/hourly?station=72405&star
t=2015-01-01&end=2015-12-31&time_zone=Europe/Berlin&time_format=Y-m-d%20H:
i&key=xPVZEykm ...
```

```
---------------------------------------------------------------------------
-
OSError                                   Traceback (most recent call las
t)
C:\Datenbanken\Anaconda3\lib\site-packages\urllib3\connectionpool.py in ur
lopen(self, method, url, body, headers, retries, redirect, assert_same_hos
t, timeout, pool_timeout, release_conn, chunked, body_pos, **response_kw)
    666             if is_new_proxy_conn:
--> 667                 self._prepare_proxy(conn)
    668

C:\Datenbanken\Anaconda3\lib\site-packages\urllib3\connectionpool.py in _p
repare_proxy(self, conn)
    929         conn.set_tunnel(self._proxy_host, self.port, self.proxy_he
aders)
--> 930         conn.connect()
    931

C:\Datenbanken\Anaconda3\lib\site-packages\urllib3\connection.py in connec
t(self)
    315             # self._tunnel_host below.
--> 316             self._tunnel()
    317             # Mark this connection as not reusable

C:\Datenbanken\Anaconda3\lib\http\client.py in _tunnel(self)
    899             self.close()
--> 900             raise OSError("Tunnel connection failed: %d %s" % (cod
e,
    901                                                                mes
sage.strip()))

OSError: Tunnel connection failed: 407 Proxy Authentication Required

During handling of the above exception, another exception occurred:

MaxRetryError                             Traceback (most recent call las
t)
C:\Datenbanken\Anaconda3\lib\site-packages\requests\adapters.py in send(se
lf, request, stream, timeout, verify, cert, proxies)
    438             if not chunked:
--> 439                 resp = conn.urlopen(
    440                     method=request.method,

C:\Datenbanken\Anaconda3\lib\site-packages\urllib3\connectionpool.py in ur
lopen(self, method, url, body, headers, retries, redirect, assert_same_hos
t, timeout, pool_timeout, release_conn, chunked, body_pos, **response_kw)
    723
--> 724             retries = retries.increment(
    725                 method, url, error=e, _pool=self, _stacktrace=sys.
exc_info()[2]

C:\Datenbanken\Anaconda3\lib\site-packages\urllib3\util\retry.py in increm
ent(self, method, url, response, error, _pool, _stacktrace)
    438         if new_retry.is_exhausted():
--> 439             raise MaxRetryError(_pool, url, error or ResponseError
(cause))
    440

MaxRetryError: HTTPSConnectionPool(host='api.meteostat.net', port=443): Ma
x retries exceeded with url: /v1/history/hourly?station=72405&start=2015-0
1-01&end=2015-12-31&time_zone=Europe/Berlin&time_format=Y-m-d%20H:i&key=xP
```

```
VZEykm (Caused by ProxyError('Cannot connect to proxy.', OSError('Tunnel c
onnection failed: 407 Proxy Authentication Required')))

During handling of the above exception, another exception occurred:

ProxyError                                Traceback (most recent call las
t)
<ipython-input-25-b6ecb832eb3d> in <module>
----> 1 df_weather = get_weather_data_for_trips(df_trips_raw)

<ipython-input-24-c13de5b3a4e5> in get_weather_data_for_trips(df_trips)
      3     target_path = WEATHER_DATA_PATH
      4
----> 5     load_weather_data(
      6         year_start=df_trips['start_ts'].min().year,
      7         year_end=df_trips['start_ts'].max().year,

<ipython-input-22-fb5e244486ae> in load_weather_data(year_start, year_end,
replace, target_path, url_template)
      1 def load_weather_data(year_start, year_end, replace=False, target_
path=WEATHER_DATA_PATH, url_template=URL_TEMPLATE_WEATHER_DATA):
      2     for year in range(year_start, year_end+1):
----> 3         load_weather_data_year(year, replace=replace, target_path=
target_path, url_template=url_template)

<ipython-input-21-b0d6be098e8b> in load_weather_data_year(year, replace, t
arget_path, url_template)
     14         url = url_template.format(str(year)+'-01-01', str(year)+'-
12-31')
     15         print('Downloading',  url, '...')
---> 16         weather_json = requests.get(url).json()
     17         with open(target_path+'weather_'+str(year)+'.json', 'w') a
s outfile:
     18             json.dump(weather_json, outfile)

C:\Datenbanken\Anaconda3\lib\site-packages\requests\api.py in get(url, par
ams, **kwargs)
     74
     75     kwargs.setdefault('allow_redirects', True)
---> 76     return request('get', url, params=params, **kwargs)
     77
     78

C:\Datenbanken\Anaconda3\lib\site-packages\requests\api.py in request(meth
od, url, **kwargs)
     59     # cases, and look like a memory leak in others.
     60     with sessions.Session() as session:
---> 61         return session.request(method=method, url=url, **kwargs)
     62
     63

C:\Datenbanken\Anaconda3\lib\site-packages\requests\sessions.py in request
(self, method, url, params, data, headers, cookies, files, auth, timeout,
 allow_redirects, proxies, hooks, stream, verify, cert, json)
    528         }
    529         send_kwargs.update(settings)
--> 530         resp = self.send(prep, **send_kwargs)
    531
    532         return resp

C:\Datenbanken\Anaconda3\lib\site-packages\requests\sessions.py in send(se
```

```
lf, request, **kwargs)
    641
    642         # Send the request
--> 643         r = adapter.send(request, **kwargs)
    644
    645         # Total elapsed time of the request (approximately)

C:\Datenbanken\Anaconda3\lib\site-packages\requests\adapters.py in send(se
lf, request, stream, timeout, verify, cert, proxies)
    508
    509             if isinstance(e.reason, _ProxyError):
--> 510                 raise ProxyError(e, request=request)
    511
    512             if isinstance(e.reason, _SSLError):

ProxyError: HTTPSConnectionPool(host='api.meteostat.net', port=443): Max r
etries exceeded with url: /v1/history/hourly?station=72405&start=2015-01-0
1&end=2015-12-31&time_zone=Europe/Berlin&time_format=Y-m-d%20H:i&key=xPVZE
ykm (Caused by ProxyError('Cannot connect to proxy.', OSError('Tunnel conn
ection failed: 407 Proxy Authentication Required')))
```

In [22]:

```
df_weather.head()
```

Out[22]:

	time_ts	temperature	dewpoint	humidity	precipitation	windspeed	winddirection	pressu
0	2015-01-01 00:00:00	2.2	-14.0	29.0	0.0	5.4	220.0	1026
1	2015-01-01 01:00:00	1.1	-12.3	36.0	NaN	7.6	210.0	1026
2	2015-01-01 02:00:00	1.1	-11.0	40.0	0.0	5.4	230.0	1026
3	2015-01-01 03:00:00	0.6	-11.8	39.0	0.0	5.4	250.0	1025
4	2015-01-01 04:00:00	0.6	-11.2	41.0	0.0	9.4	170.0	1025

In [23]:

```
df_weather.to_pickle(PARSED_DATA_PATH+RAW_WEATHER_FILE)
```

In [24]:

```
df_weather.head()
```

Out[24]:

	time_ts	temperature	dewpoint	humidity	precipitation	windspeed	winddirection	pressu
0	2015-01-01 00:00:00	2.2	-14.0	29.0	0.0	5.4	220.0	1026
1	2015-01-01 01:00:00	1.1	-12.3	36.0	NaN	7.6	210.0	1026
2	2015-01-01 02:00:00	1.1	-11.0	40.0	0.0	5.4	230.0	1026
3	2015-01-01 03:00:00	0.6	-11.8	39.0	0.0	5.4	250.0	1025
4	2015-01-01 04:00:00	0.6	-11.2	41.0	0.0	9.4	170.0	1025

In [ ]:

```
```

In [28]:

```
def get_alt_weather_data_for_trips(df_trips):
    df = pd.read_csv(URL_ALT_WEATHER_DATA, names=COLS_ALT_WEATHER_DATA, parse_dates=[0
])
    df = df[(df.date >= df_trips['start_ts'].min()) & (df.date <= df_trips['start_ts'].
max())]
    df.drop(WEATHER_COLS_TO_DROP, axis=1, inplace=True)
    df['hour'] = df['hour'].str[0:2].astype('int')
    return df.reset_index(drop=True)
```

In [29]:

```
df_weather_alt = get_alt_weather_data_for_trips(df_trips_raw)
```

```
---------------------------------------------------------------------------
-
OSError                                   Traceback (most recent call las
t)
C:\Datenbanken\Anaconda3\lib\urllib\request.py in do_open(self, http_clas
s, req, **http_conn_args)
   1349             try:
-> 1350                 h.request(req.get_method(), req.selector, req.dat
a, headers,
   1351                           encode_chunked=req.has_header('Transfer-
encoding'))

C:\Datenbanken\Anaconda3\lib\http\client.py in request(self, method, url,
 body, headers, encode_chunked)
   1239         """Send a complete request to the server."""
-> 1240         self._send_request(method, url, body, headers, encode_chun
ked)
   1241

C:\Datenbanken\Anaconda3\lib\http\client.py in _send_request(self, method,
url, body, headers, encode_chunked)
   1285             body = _encode(body, 'body')
-> 1286         self.endheaders(body, encode_chunked=encode_chunked)
   1287

C:\Datenbanken\Anaconda3\lib\http\client.py in endheaders(self, message_bo
dy, encode_chunked)
   1234             raise CannotSendHeader()
-> 1235         self._send_output(message_body, encode_chunked=encode_chun
ked)
   1236

C:\Datenbanken\Anaconda3\lib\http\client.py in _send_output(self, message_
body, encode_chunked)
   1005         del self._buffer[:]
-> 1006         self.send(msg)
   1007

C:\Datenbanken\Anaconda3\lib\http\client.py in send(self, data)
    945             if self.auto_open:
--> 946                 self.connect()
    947             else:

C:\Datenbanken\Anaconda3\lib\http\client.py in connect(self)
   1401
-> 1402             super().connect()
   1403

C:\Datenbanken\Anaconda3\lib\http\client.py in connect(self)
    921         if self._tunnel_host:
--> 922             self._tunnel()
    923

C:\Datenbanken\Anaconda3\lib\http\client.py in _tunnel(self)
    899             self.close()
--> 900             raise OSError("Tunnel connection failed: %d %s" % (cod
e,
    901                                                                mes
sage.strip()))

OSError: Tunnel connection failed: 407 Proxy Authentication Required
```

```
During handling of the above exception, another exception occurred:

URLError                                  Traceback (most recent call las
t)
<ipython-input-29-7de41a5f03ce> in <module>
----> 1 df_weather_alt = get_alt_weather_data_for_trips(df_trips_raw)

<ipython-input-28-bae58425e874> in get_alt_weather_data_for_trips(df_trip
s)
      1 def get_alt_weather_data_for_trips(df_trips):
----> 2     df = pd.read_csv(URL_ALT_WEATHER_DATA, names=COLS_ALT_WEATHER_
DATA, parse_dates=[0])
      3     df = df[(df.date >= df_trips['start_ts'].min()) & (df.date <=
 df_trips['start_ts'].max())]
      4     df.drop(WEATHER_COLS_TO_DROP, axis=1, inplace=True)
      5     df['hour'] = df['hour'].str[0:2].astype('int')

C:\Datenbanken\Anaconda3\lib\site-packages\pandas\io\parsers.py in parser_
f(filepath_or_buffer, sep, delimiter, header, names, index_col, usecols, s
queeze, prefix, mangle_dupe_cols, dtype, engine, converters, true_values,
 false_values, skipinitialspace, skiprows, skipfooter, nrows, na_values, k
eep_default_na, na_filter, verbose, skip_blank_lines, parse_dates, infer_d
atetime_format, keep_date_col, date_parser, dayfirst, cache_dates, iterato
r, chunksize, compression, thousands, decimal, lineterminator, quotechar,
 quoting, doublequote, escapechar, comment, encoding, dialect, error_bad_l
ines, warn_bad_lines, delim_whitespace, low_memory, memory_map, float_prec
ision)
    674         )
    675
--> 676         return _read(filepath_or_buffer, kwds)
    677
    678     parser_f.__name__ = name

C:\Datenbanken\Anaconda3\lib\site-packages\pandas\io\parsers.py in _read(f
ilepath_or_buffer, kwds)
    428     # though mypy handling of conditional imports is difficult.
    429     # See https://github.com/python/mypy/issues/1297
--> 430     fp_or_buf, _, compression, should_close = get_filepath_or_buff
er(
    431         filepath_or_buffer, encoding, compression
    432     )

C:\Datenbanken\Anaconda3\lib\site-packages\pandas\io\common.py in get_file
path_or_buffer(filepath_or_buffer, encoding, compression, mode)
    170
    171     if isinstance(filepath_or_buffer, str) and is_url(filepath_or_
buffer):
--> 172         req = urlopen(filepath_or_buffer)
    173         content_encoding = req.headers.get("Content-Encoding", Non
e)
    174         if content_encoding == "gzip":

C:\Datenbanken\Anaconda3\lib\site-packages\pandas\io\common.py in urlopen
(*args, **kwargs)
    139     import urllib.request
    140
--> 141     return urllib.request.urlopen(*args, **kwargs)
    142
    143
```

```
C:\Datenbanken\Anaconda3\lib\urllib\request.py in urlopen(url, data, timeo
ut, cafile, capath, cadefault, context)
    220     else:
    221         opener = _opener
--> 222     return opener.open(url, data, timeout)
    223
    224 def install_opener(opener):

C:\Datenbanken\Anaconda3\lib\urllib\request.py in open(self, fullurl, dat
a, timeout)
    523
    524         sys.audit('urllib.Request', req.full_url, req.data, req.he
aders, req.get_method())
--> 525         response = self._open(req, data)
    526
    527         # post-process response

C:\Datenbanken\Anaconda3\lib\urllib\request.py in _open(self, req, data)
    540
    541         protocol = req.type
--> 542         result = self._call_chain(self.handle_open, protocol, prot
ocol +
    543                                   '_open', req)
    544         if result:

C:\Datenbanken\Anaconda3\lib\urllib\request.py in _call_chain(self, chain,
kind, meth_name, *args)
    500         for handler in handlers:
    501             func = getattr(handler, meth_name)
--> 502             result = func(*args)
    503             if result is not None:
    504                 return result

C:\Datenbanken\Anaconda3\lib\urllib\request.py in https_open(self, req)
   1391
   1392         def https_open(self, req):
-> 1393             return self.do_open(http.client.HTTPSConnection, req,
   1394                 context=self._context, check_hostname=self._check_
hostname)
   1395

C:\Datenbanken\Anaconda3\lib\urllib\request.py in do_open(self, http_clas
s, req, **http_conn_args)
   1351                           encode_chunked=req.has_header('Transfer-
encoding'))
   1352             except OSError as err: # timeout error
-> 1353                 raise URLError(err)
   1354             r = h.getresponse()
   1355         except:

URLError: <urlopen error Tunnel connection failed: 407 Proxy Authenticatio
n Required>
```

In [27]:

```
df_weather_alt.to_pickle(PARSED_DATA_PATH+ALT_WEATHER_FILE)
```

In [28]:

```
df_weather_alt.head()
```

Out[28]:

	date	hour	temperature	dewpoint	precipitation	windspeed	winddirection	humidity	pre
0	2015-01-02	0	5.0	-6.6	0.0	16.6	200.0	43.0	
1	2015-01-02	1	4.4	-6.8	0.0	16.6	200.0	44.0	
2	2015-01-02	2	3.3	-6.1	0.0	14.8	200.0	50.0	
3	2015-01-02	3	4.4	-5.7	0.0	14.8	210.0	48.0	
4	2015-01-02	4	4.4	-5.7	0.0	9.4	220.0	48.0	

In [ ]:

```

```

In [ ]:

```

```