Prototype 2

Settings

- Check the active interpreters: md (needs to be on the top), spark2, sh, shUser.
- File for prototype:

```
FILE='prep_household_data_60min_singleindex.csv'
```

Approach

- Download the data to local device.
- Manipulate the data with Pandas (Pycharm).
- Upload the data to the gateway.

Data Manipulation with Pandas

File data_preprocessing.py

```
import pandas as pd


input_path = "data_raw"
output_path = "data_prep"
# file = "household_data_1min_singleindex.csv"
# file = "household_data_15min_singleindex.csv"
file = "household_data_60min_singleindex.csv"
path_file = f"{input_path}/{file}"

df60_raw = pd.read_csv(path_file)

def get_df_prepared(df, num):
    ls_cols_additional = ["utc_timestamp", "cet_cest_timestamp", "interpolat
    df_tmp = df[ls_cols_additional + [col for col in df.columns if f"_reside
    df_tmp["unit"] = f"residential{num}"
    df_tmp.columns = df_tmp.columns.str.replace(f'DE\_KN\_residential{num}\_
    return df_tmp


def get_df_combined(df, max_num):
    df_appended = pd.DataFrame()
    for num in range(max_num):
        print(num)
        df_tmp = get_df_prepared(df, num=num)
        df_appended = pd.concat([df_appended, df_tmp], axis=0, ignore_index=
    return df_appended


df_prep = get_df_combined(df60_raw, max_num=5)
df_prep.to_csv(f"{output_path}/prep_{file}")

## Bring the data to the GW

Get the data from local to the gateway
```

Bring the data from the local device to the gateway:

```
andy@andy-legion:~/Documents/github/bdl03-2/data_prep$ scp -p prep_household
andy@andy-legion:~/Documents/github/bdl03-2/data_prep$ scp -p prep_household
andy@andy-legion:~/Documents/github/bdl03-2/data_prep$ scp -p prep_household
```

Move the data to the project folder

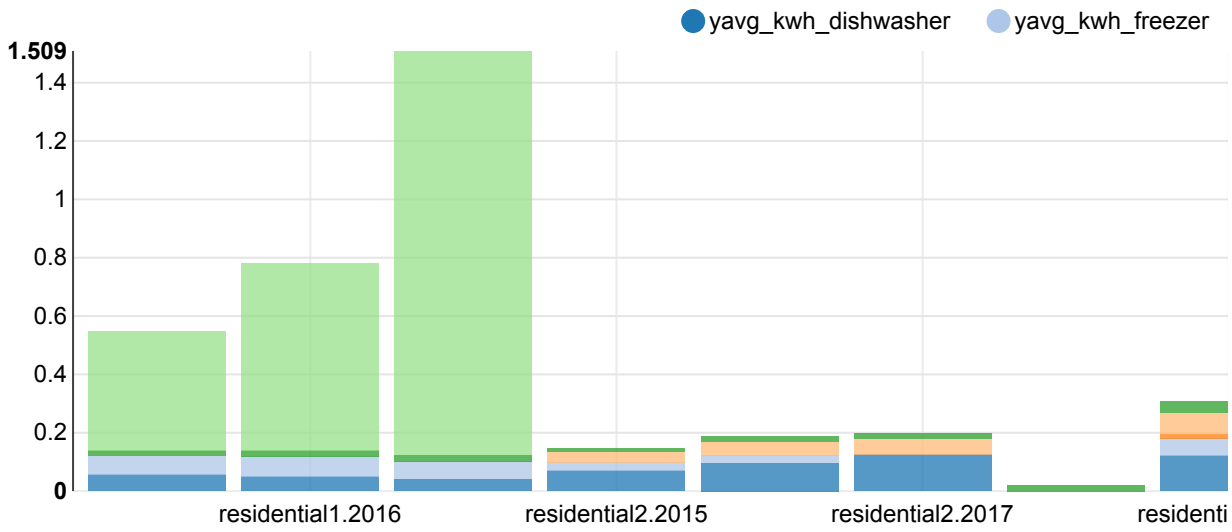
```
bd01@cl-hpsec1-50-gw-01-lx-ub18.lxd:~$ mv prep_household_data_1min_singleind
bd01@cl-hpsec1-50-gw-01-lx-ub18.lxd:~$ mv prep_household_data_15min_singlein
bd01@cl-hpsec1-50-gw-01-lx-ub18.lxd:~$ mv prep_household_data_60min_singlein
```

```
total 1.8G
-rw-rw-r-- 1 bd01 bd01 119M Feb  5 19:02 prep_household_data_15min_singleindex.csv
-rw-rw-r-- 1 bd01 bd01 1.6G Feb  5 19:00 prep_household_data_1min_singleindex.csv
-rw-rw-r-- 1 bd01 bd01  34M Feb  5 18:56 prep_household_data_60min_singleindex.csv
----hdfs----
Found 6 items
drwx------   - zeppelin hdfs          0 2021-02-06 06:00 /user/zeppelin/.Trash
drwxr-xr-x   - zeppelin hdfs          0 2021-02-06 13:38 /user/zeppelin/conf
drwxr-xr-x   - zeppelin hdfs          0 2021-02-06 12:57 /user/zeppelin/notebook
-rw-r--r--   1 zeppelin hdfs  124520554 2021-02-05 20:36 /user/zeppelin/prep_household_data_15min_singlein
dex.csv
-rw-r--r--   1 zeppelin hdfs 1712423587 2021-02-06 11:26 /user/zeppelin/prep_household_data_1min_singleind
ex.csv
-rw-r--r--   1 zeppelin hdfs   34626321 2021-02-05 20:35 /user/zeppelin/prep_household_data_60min_singlein
dex.csv
```

Load into a Spark dataframe

```
root
 |-- _c0: integer (nullable = true)
 |-- utc_timestamp: timestamp (nullable = true)
 |-- cet_cest_timestamp: string (nullable = true)
 |-- interpolated: string (nullable = true)
 |-- unit: string (nullable = true)
 |-- dishwasher: double (nullable = true)
 |-- freezer: double (nullable = true)
 |-- grid_import: double (nullable = true)
 |-- heat_pump: double (nullable = true)
 |-- pv: double (nullable = true)
 |-- washing_machine: double (nullable = true)
 |-- circulation_pump: double (nullable = true)
 |-- grid_export: double (nullable = true)
 |-- refrigerator: double (nullable = true)
 |-- ev: double (nullable = true)
 |-- timestamp: timestamp (nullable = true)
 |-- [illegible] (nullable = true)
```

Analysis: Which devices consume the majority of households energy (average kWh)?



First analysis shows a large variability among the largest energy consumer household devices. Whereas heat pumps consume by far the largest amount of households energy, electric vehicles probably follow second. The energy consumation of other factors such as dishwasher and freezer show huge variation across households. The usage of these devices depends on individual behaviour. Unfortunately, the data does not provide an indicator for household size or living area.

Learnings

Learnings regarding this project:

- Eventhough Zepellin is quite handy to use and is similiar to Jupyter, I really miss features such as autocompletion.
- Data manipulation with PySpark is challenging. PySpark might not even be the right chose. Therefore, I prefer Pandas to preprocess data.

Learnings regarding my Master Thesis:

- The average energy consumption for electrical devices are quite different across households. Unfortunately, the dataset does not provide household size or area information. For my thesis, I might need several datasets from different sources, providing households information, building details and energy consumption.