

# Analyses Fabian Sanchez

Studentennr: 202187874

Groep: 2 VC

## Prerequisites

```
In [ ]: pip install pandasql
```

```
In [ ]: import pandas as pd
import numpy as np
from pandasql import sqldf
import matplotlib.pyplot as plt
from sklearn.metrics import root_mean_squared_error, mean_absolute_percentage_error

from sqlalchemy import create_engine
```

```
In [ ]: # verbinding
server = r"DESKTOP-1DRSSG0\SQLEXPRESS"
database = "DEP1_DWH"

engine = create_engine("mssql+pyodbc://@{}/{ }?driver=ODBC+Driver+17+for+SQL+Serv
```

## Gemakkelijk 2

We beschikken voor de dataset van 1300 residentiële gebruikers jammer genoeg niet over de overeenkomstige piekverbruiken. Daarom zullen we als benadering voor de maandpiek het volgende gebruiken, namelijk het maximum verbruik voor één kwartier per maand \* 4. We willen verifiëren of dat wel een goede benadering is.

Maak gebruik van de 2 bijhorende bestanden (van gebruiker 1301) om te controleren of deze aanpak wel een goede benadering is. Je moet de data van gebruiker 1301 niet toevoegen aan de DWH. Bepaal zelf welke metriek je hierbij best kan gebruiken. Denk daarbij aan de metrieken die aan bod komen in ML

## Preprocess piekvermogens

```
In [ ]: historiek_piekvermogen_1301 = pd.read_csv("../data/input/Sprint2/Historiek_piekv
historiek_piekvermogen_1301.drop(columns=["Van (tijdstip)", "Tot (tijdstip)", "M
historiek_piekvermogen_1301.head()
```

```
In [ ]: historiek_piekvermogen_1301['Volume'] = historiek_piekvermogen_1301['Volume'].st
historiek_piekvermogen_1301['Volume'] = historiek_piekvermogen_1301['Volume'].as
historiek_piekvermogen_1301["Van (datum)"] = pd.to_datetime(historiek_piekvermog
```

```
historiek_piekvermogen_1301.head()
```

```
In [ ]: historiek_piekvermogen_1301 = sqldf('''
        select Volume as MaandPiek,
               strftime('%Y-%m', "Van (datum)") as maand
        from historiek_piekvermogen_1301
        ''')

historiek_piekvermogen_1301.head(10)
```

```
In [ ]: historiek_piekvermogen_1301.head(20)
```

## Preprocess verbruikshistoriek

```
In [ ]: verbruikshistoriek_1301 = pd.read_csv("../data/input/Sprint2/Verbruikshistoriek_1301.csv")

verbruikshistoriek_1301.head()
```

```
In [ ]: verbruikshistoriek_1301["Van (datum)"] = pd.to_datetime(verbruikshistoriek_1301["Van (datum)"])

verbruikshistoriek_1301.head(20)
```

```
In [ ]: verbruikshistoriek_1301 = sqldf('''
        select max(Volume) as max_volume,
               strftime('%Y-%m', "Van (datum)") as maand
        from verbruikshistoriek_1301
        where Register like 'Afname%'
        group by maand
        order by maand
        ''')
```

```
In [ ]: verbruikshistoriek_1301.head(10)
```

```
In [ ]: # float
verbruikshistoriek_1301['max_volume'] = verbruikshistoriek_1301['max_volume'].astype(float)

verbruikshistoriek_1301['max_volume'] = verbruikshistoriek_1301['max_volume'].astype(float)
```

```
In [ ]: # maart 2025 had geen data

verbruikshistoriek_1301.drop(labels=[10], inplace=True)
```

```
In [ ]: verbruikshistoriek_1301.head(15)
```

## Merge

```
In [ ]: df_vergelijking = pd.merge(historiek_piekvermogen_1301, verbruikshistoriek_1301,
                                   on='Van (datum)', how='left')
df_vergelijking = df_vergelijking.rename(columns={'max_volume': 'MaxVolume', 'MaandPiek': 'Maand'})
df_vergelijking = df_vergelijking[['Maand', 'MaandPiek', 'MaxVolume']]
df_vergelijking.head(15)
```

## Visuele voorstelling

```
In [ ]: x = np.arange(len(df_vergelijking))

plt.figure(figsize=(15, 7))
plt.bar(x - 0.2, df_vergelijking['MaandPiek'], width=0.4, label='MaandPiek')
plt.bar(x + 0.2, df_vergelijking['MaxVolume'], width=0.4, label='MaxVolume')

plt.xticks(x, df_vergelijking['Maand'], rotation=20)
plt.legend()
plt.show()
```

## Statistische metriecken

```
In [ ]: waarnemingen = df_vergelijking['MaandPiek']
voorspellingen = df_vergelijking['MaxVolume']

rmse = root_mean_squared_error(waarnemingen, voorspellingen)
mape = mean_absolute_percentage_error(waarnemingen, voorspellingen)

print("RMSE:", rmse)
print("MAPE:", mape)
```

## Conclusie

Het is aan de hand van zowel de histogram als de RMSE en MAPE waarden duidelijk dat de voorspellingen van de piekvermogen zeer goed zijn voor gebruiker 1301. Dit zal waarschijnlijk geen toeval zijn, maar we dienen voorzichtig te zijn: de steekproef van 1 gebruiker is te klein om uitspraken te doen over de gehele populatie.

## Gemiddeld 2

Volgens Fluvius bedraagt de facturatiepiek voor een gemiddeld gezin 3.5 tot 4 kW en draait het verbruik rond 3500 kWh per jaar (<https://www.fluvius.be/nl/blog/capaciteitstarief/capaciteitstarief-nieuweberekening-nettarieven-piekvermogen>) In de onderstaande plot en de onderstaande tabel worden de netbeheerkosten voor 2023, 2024 en 2025 vergeleken. Begin met het maken van de plot en werk verder met de data nodig voor het maken van de plot om de onderstaande tabel te maken. In de tabel wordt de relatieve stijging tussen 2023 en 2025 berekend. We gaan bij de berekeningen uit van een gemiddelde klant met een facturatiepiek van 4kW en een jaarlijks verbruik van 3500 kWh. Bovendien gaan we ervan uit dat

- Fluvius Antwerpen = Fluvius (Antwerpen) • Fluvius Limburg = Fluvius (Limburg) • Fluvius Imewo = Fluvius (Imewo) • Intergem = Fluvius (Midden-Vlaanderen) • Iveka = Fluvius (Kempen) • Iverlek = Fluvius (Zenne-Dijle) • Gaselwest = Fluvius (West)

De andere netbeheerders worden genegeerd.

```
In [ ]: table_name = "FactNetworkCosts"
df_FactNetworkCosts = pd.read_sql(f"SELECT * FROM {table_name}", con=engine)
df_FactNetworkCosts.head()
```

```
In [ ]: df_FactNetworkCosts = sqldf(''
                                     select *
                                     from df_FactNetworkCosts
                                     where NetworkOperator not in ('P
                                     ''')
df_FactNetworkCosts.head()
```

```
In [ ]: df_FactNetworkCosts = df_FactNetworkCosts.replace({'Fluvius Antwerpen': 'Fluvius
df_FactNetworkCosts.head()
```

## Berekening jaarlijkse kost van een gemiddelde klant

Gemiddeld verbruik is 3500 kWh, gemiddelde facturatiepiek 4 kW.

Ik ging uit van een digitale meter. Ook bracht ik het prosumementarief in rekening met een installatie van 2.9 KVA.

Verbruikskosten zijn in cent uitgedrukt, dus wordt 4 -> 0,04 en 3500 -> 35 om naar euro te rekenen. Datamanagementkosten en Prosumementarief zijn per jaar, in euro.

```
In [ ]: df_FactNetworkCosts = sqldf(
    ''
    select DateKey, NetworkOperator,
    (CapacityTariff_DigitalMeter*0.04 +
     ConsumptionTariff_DigitalMeter_Normal*35 +
     ProsumerTariff*2.9 +
     DataManagementTariff_YearlyMonthlyReadMeters +
     DataManagementTariff_QuarterlyReadMeters)
    as NetCost
    from df_FactNetworkCosts
    order by NetworkOperator, DateKey
    ''
)
df_FactNetworkCosts.head(25)
```

```
In [ ]: df_FactNetworkCosts = sqldf(''
    select
        a.*,
        case
            when a.datekey = 20250101 then
                round(((a.netcost - b.netcost) / b.netcost * 100), 2) || '%'
            else null
        end as difference2023vs2025
    from
        df_FactNetworkCosts a
    left join
        (select networkoperator, netcost from df_FactNetworkCosts where datekey
        on
            a.networkoperator = b.networkoperator
    ''')
```

```
In [ ]: df_FactNetworkCosts.head(25)
```

## Grafische voorstelling

```
In [ ]: df_FactNetworkCosts['Year'] = df_FactNetworkCosts['DateKey'].astype(str).str[:4]

fig, ax = plt.subplots(figsize=(12, 8))

operators = df_FactNetworkCosts['NetworkOperator'].unique()
x_pos = np.arange(len(operators))
width = 0.25

colors = {'2023': 'green', '2024': 'orange', '2025': 'red'}

for i, year in enumerate(['2023', '2024', '2025']):
    data = df_FactNetworkCosts[df_FactNetworkCosts['Year'] == year]
    values = [data[data['NetworkOperator'] == op]['NetCost'].values[0] for op in operators]
    ax.bar(x_pos + (i-1)*width, values, width=width, color=colors[year], label=year)

ax.set_xticks(x_pos)
ax.set_xticklabels(operators, rotation=45, ha='right')
ax.set_ylabel('NetCost')
ax.legend(title='Year')

plt.show()
```