# 04_feature_engineering

May 26, 2024

## 1 Feature Engineering

### 1.1 Content

1. Data Imports
2. Convert Structure of Skills and Abilities
3. save data

[1]:
```
# imports
import json
from db import get_database
import pandas as pd
import numpy as np
```

## Data Imports

[2]:
```
# import authentication data
with open('./infos.json') as f:
    infos = json.load(f)
    onet = infos['onet']
    onetUsername = onet['username']
    onetPassword = onet['password']
    mongodb = infos['mongodb']
    mongoUusername = mongodb['username']
    mongoPpassword = mongodb['password']
    mongoUrl = mongodb['connectionString']
```

[3]:
```
# get data from mongodb and csv files
dbname = get_database()

collection = dbname["joined"]
fo_col = dbname["fo_joined"]

documents = collection.find()
fo = fo_col.find()

df = pd.DataFrame(list(documents))
fo_df = pd.DataFrame(list(fo))
skills_df = pd.read_csv('files/onet_skills.csv')
```

```
abilities_df = pd.read_csv('files/onet_abilities.csv')
```

[4]:
```
# Remove empty cols
print("Total documents: ", len(df))

# Drop columns where some values are null
df = df.dropna(how='all', axis=1)

# Drop rows where all some are null
df = df.dropna(how='all', axis=0)

# Replace any remaining NaN values with 0
df = df.fillna(0)

print("Total documents after dropping na: ", len(df))
```

```
Total documents:  846
Total documents after dropping na:  846
```

[5]:
```
# Remove columns that are not useful
df = df.
  ↪drop(['additional_information',"tasks","related_occupations","display","technology_skills",
  ↪axis=1)

# add field Berufshauptgruppen
df["Berufshauptgruppe"] = df["isco08"].str[0]

df.head(2)
```

[5]:
```
                                          occupation  \
0  {'code': '27-2011.00', 'title': 'Actors', 'tag…
1  {'code': '23-1021.00', 'title': 'Administrativ…

                                              skills  \
0  {'element': [{'id': '2.A.1.a', 'related': 'htt…
1  {'element': [{'id': '2.A.1.b', 'related': 'htt…

                                          abilities isco08       Name_de  \
0  {'element': [{'id': '1.A.1.a.3', 'related': 'h…   2655  Schauspieler
1  {'element': [{'id': '1.A.1.b.5', 'related': 'h…   2612       Richter

  Berufshauptgruppe
0                 2
1                 2
```

[6]:
```
# extract occupation column from the occupation field and convert to df (so we␣
  ↪can only save the name of the occupation)
occupation = pd.DataFrame(list(df["occupation"]))
```

```
# List of columns to drop
cols_to_drop =␣
 ↪["code","updated","sample_of_reported_job_titles","summary_resources","details_resources","
 ↪"tags"]

# Drop the columns
occupation = occupation.drop(cols_to_drop, axis=1)

occupation = occupation.iloc[:, 0]
df["occupation"] = occupation

df["occupation"].head(3)
```

[6]:
```
0                                                 Actors
1    Administrative Law Judges, Adjudicators, and H…
2    Aerospace Engineering and Operations Technolog…
Name: occupation, dtype: object
```

## Convert Structure of Skills and Abilities

[7]:
```
# Initialize all skill columns to 0
for _, skill in skills_df.iterrows():
    skilltext = ("s" + str(skill["skill_id"]))
    df[skilltext] = 0

# add Value to each skill
for index, row in df["skills"].items():
    row = pd.DataFrame(row['element'])

    # Merge the skills dataframe with the skills_df dataframe to get the new␣
 ↪skill id
    row = row.merge(skills_df, left_on='id', right_on='id', how='left')

    for _, skill_row in row.iterrows():
        dftext = ("s" + str(skill_row["skill_id"]))
        score_value = skill_row["score"]['value']
        df.loc[index, dftext] = score_value

df.head(2)
```

[7]:
```
                                          occupation  \
0                                             Actors
1  Administrative Law Judges, Adjudicators, and H…

                                              skills  \
0  {'element': [{'id': '2.A.1.a', 'related': 'htt…
1  {'element': [{'id': '2.A.1.b', 'related': 'htt…
```

```
                                              abilities isco08       Name_de  \
0  {'element': [{'id': '1.A.1.a.3', 'related': 'h…   2655  Schauspieler
1  {'element': [{'id': '1.A.1.b.5', 'related': 'h…   2612       Richter

  Berufshauptgruppe  s1  s2  s3  s4  …  s26  s27  s28  s29  s30  s31  s32  \
0                 2  72  72  69  69  …    0    0    0    0    0    0    0
1                 2  81  75  81  72  …    0    0   19   16   28    0   13

   s33  s34  s35
0    0    0    0
1    3    0    0

[2 rows x 41 columns]
```

[8]:
```
# Initialize all ability columns to 0
for _, ability in abilities_df.iterrows():
    abilitytext = ("a" + str(ability["ability_id"]))
    df[abilitytext] = 0

# add Value to each ability
for index, row in df["abilities"].items():
    row = pd.DataFrame(row['element'])

    # Merge the skills dataframe with the skills_df dataframe to get the new␣
 ↪skill id
    row = row.merge(abilities_df, left_on='id', right_on='id', how='left')

    for _, ability_row in row.iterrows():
        dftext = ("a" + str(ability_row["ability_id"]))
        score_value = ability_row["score"]['value']
        df.loc[index, dftext] = score_value

df.head(2)
```

[8]:
```
                                         occupation  \
0                                            Actors
1  Administrative Law Judges, Adjudicators, and H…

                                              skills  \
0  {'element': [{'id': '2.A.1.a', 'related': 'htt…
1  {'element': [{'id': '2.A.1.b', 'related': 'htt…

                                              abilities isco08       Name_de  \
0  {'element': [{'id': '1.A.1.a.3', 'related': 'h…   2655  Schauspieler
1  {'element': [{'id': '1.A.1.b.5', 'related': 'h…   2612       Richter
```

```
  Berufshauptgruppe  s1  s2  s3  s4  …  a43  a44  a45  a46  a47  a48  a49  \
0                 2  72  72  69  69  …    0    0    0    0    0    0    0
1                 2  81  75  81  72  …    0   31    0    0    0    0    0

   a50  a51  a52
0    0    0    0
1    0    0    0

[2 rows x 93 columns]
```

[10]:
```
# Remove columns that are not useful
df = df.drop(["skills","abilities","occupation"], axis=1)
df.head(2)
```

[10]:
```
  isco08       Name_de Berufshauptgruppe  s1  s2  s3  s4  s5  s6  s7  …  \
0   2655  Schauspieler                 2  72  72  69  69  50  50  50  …
1   2612       Richter                 2  81  75  81  72  81  66  56  …

   a43  a44  a45  a46  a47  a48  a49  a50  a51  a52
0    0    0    0    0    0    0    0    0    0    0
1    0   31    0    0    0    0    0    0    0    0

[2 rows x 90 columns]
```

[11]:
```
# Get columns that start with 'a' or 's'
cols = df.columns[df.columns.str.startswith(('a', 's'))]

# Divide these columns by 100
df[cols] = df[cols] / 100
```

[12]:
```
# merge dfs
fo_df = pd.merge(df, fo_df, left_on='isco08', right_on="isco08", how='left')
fo_df = fo_df.drop_duplicates(subset=['isco08'])
fo_df = fo_df.drop(columns=["_id"])
fo_df = fo_df.dropna()

df = df.drop_duplicates(subset=['isco08'])
```

## Save data

[13]:
```
fo_df.to_csv("files/fo_swiss.csv", index=False)
df.to_csv("files/switzerland_occupations.csv", index=False)
```