iness-case-study-scaler-clustering

December 25, 2024

[1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.impute import KNNImputer
from sklearn.preprocessing import OrdinalEncoder, StandardScaler
import warnings
warnings.filterwarnings('ignore')
```

[2]:
```
df = pd.read_csv("scaler_clustering.csv")
df.drop("Unnamed: 0",axis=1,inplace=True)
df.head()
```

[2]:

	company_hash	email_hash	orgyear	ctc	job_position	ctc_updated_year
0	atrgxnnt xzaxv	6de0a4417d18ab14334c3f43397fc13b30c35149d70c05…	2016.0	1100000	Other	2020.0
1	qtrxvzwt xzegwgbb rxbxnta	b0aaf1ac138b53cb6e039ba2c3d6604a250d02d5145c10…	2018.0	449999	FullStack Engineer	2019.0
2	ojzwnvwnxw vx	4860c670bcd48fb96c02a4b0ae3608ae6fdd98176112e9…	2015.0	2000000	Backend Engineer	2020.0
3	ngpgutaxv	effdede7a2e7c2af664c8a31d9346385016128d66bbc58…	2017.0	700000	Backend Engineer	2019.0
4	qxen sqghu	6ff54e709262f55cb999a1c1db8436cb2055d8f79ab520…	2017.0	1400000	FullStack Engineer	2019.0

[3]:
```
# data = pd.read_csv('https://d2beiqkhq929f0.cloudfront.net/public_assets/
  ↪assets/000/002/681/original/Scaler_Kmeans.csv')
# data.to_csv("Scaler K-means clustering.csv")
```

[4]:
```
# !pip install datahorse

# import datahorse
# df = datahorse.read("scaler_clustering.csv")
# df.drop("Unnamed: 0",axis=1,inplace=True)
# df.head()

# df.chat("what are the names of the columns")

# df.chat("How many unique job_position are there")

# df.chat("How many unique orgyear are there")

# df.chat("How many unique email_hash are there")

# df.chat("give at leaset 5 email_hash")
```

[5]:
```
df.shape
```

[5]: (205843, 6)

[6]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 205843 entries, 0 to 205842
Data columns (total 6 columns):
 #   Column            Non-Null Count   Dtype  
---  ------            --------------   -----  
 0   company_hash      205799 non-null  object 
 1   email_hash        205843 non-null  object 
 2   orgyear           205757 non-null  float64
 3   ctc               205843 non-null  int64  
 4   job_position      153279 non-null  object 
 5   ctc_updated_year  205843 non-null  float64
dtypes: float64(2), int64(1), object(3)
memory usage: 9.4+ MB
```

[7]:
```
df.duplicated().sum()
```

[7]: 34

[8]:
```
df_cleaned = df.drop_duplicates()
df_cleaned.duplicated().sum()
```

[8]: 0

[9]:
```
df_cleaned = df_cleaned.reset_index(drop=True)
df_cleaned
```

[9]:

```
                    company_hash  \
0                  atrgxnnt xzaxv
1       qtrxvzwt xzegwgbb rxbxnta
2                   ojzwnvwnxw vx
3                       ngpgutaxv
4                      qxen sqghu
…                             …
205804                  vuurt xzw
205805                  husqvawgb
205806                   vwwgrxnt
205807             zgn vuurxwvmrt
205808             bgqsvz onvzrtj

                                                email_hash  orgyear      ctc  \
0       6de0a4417d18ab14334c3f43397fc13b30c35149d70c05…   2016.0  1100000
1       b0aaf1ac138b53cb6e039ba2c3d6604a250d02d5145c10…   2018.0   449999
2       4860c670bcd48fb96c02a4b0ae3608ae6fdd98176112e9…   2015.0  2000000
3       effdede7a2e7c2af664c8a31d9346385016128d66bbc58…   2017.0   700000
4       6ff54e709262f55cb999a1c1db8436cb2055d8f79ab520…   2017.0  1400000
…                                                     …       …       …
205804  70027b728c8ee901fe979533ed94ffda97be08fc23f33b…   2008.0   220000
205805  7f7292ffad724ebbe9ca860f515245368d714c84705b42…   2017.0   500000
205806  cb25cc7304e9a24facda7f5567c7922ffc48e3d5d6018c…   2021.0   700000
205807  fb46a1a2752f5f652ce634f6178d0578ef6995ee59f6c8…   2019.0  5100000
205808  0bcfc1d05f2e8dc4147743a1313aa70a119b41b30d4a1f…   2014.0  1240000

             job_position  ctc_updated_year
0                   Other            2020.0
1       FullStack Engineer           2019.0
2         Backend Engineer           2020.0
3         Backend Engineer           2019.0
4       FullStack Engineer           2019.0
…                       …                …
205804                NaN            2019.0
205805                NaN            2020.0
205806                NaN            2021.0
205807                NaN            2019.0
205808                NaN            2016.0

[205809 rows x 6 columns]
```

[10]:
```
df_cleaned.to_csv("removed_outliers_df_cleaned.csv")
```

0.0.1 Apparently for single Anonymised Personal Identifiable Information (PII) id there exists multiple rows with same joining dates and company but different job positions, this couldn't be possible.

0.0.2 We will take the first row in case of duplicated PII ids.

[11]:
```
df_first_values = df_cleaned.drop_duplicates(subset=["email_hash"],␣
↪keep="first").reset_index(drop=True)
df_first_values
```

[11]:

	company_hash	email_hash	orgyear	ctc	job_position	ctc_updated_year
0	atrgxnnt xzaxv	6de0a4417d18ab14334c3f43397fc13b30c35149d70c05…	2016.0	1100000	Other	2020.0
1	qtrxvzwt xzegwgbb rxbxnta	b0aaf1ac138b53cb6e039ba2c3d6604a250d02d5145c10…	2018.0	449999	FullStack Engineer	2019.0
2	ojzwnvwnxw vx	4860c670bcd48fb96c02a4b0ae3608ae6fdd98176112e9…	2015.0	2000000	Backend Engineer	2020.0
3	ngpgutaxv	effdede7a2e7c2af664c8a31d9346385016128d66bbc58…	2017.0	700000	Backend Engineer	2019.0
4	qxen sqghu	6ff54e709262f55cb999a1c1db8436cb2055d8f79ab520…	2017.0	1400000	FullStack Engineer	2019.0
…	…	…	…	…	…	…
153438	mvqwrvjo	53442a1663ccfdbd473055fee4e4ac9f4bb398dc446242…	2011.0	2250000	NaN	2019.0
153439	husqvawgb	7f7292ffad724ebbe9ca860f515245368d714c84705b42…	2017.0	500000	NaN	2020.0
153440	vwwgrxnt	cb25cc7304e9a24facda7f5567c7922ffc48e3d5d6018c…	2021.0	700000	NaN	2021.0
153441	zgn vuurxwvmrt	fb46a1a2752f5f652ce634f6178d0578ef6995ee59f6c8…	2019.0	5100000	NaN	2019.0
153442	bgqsvz onvzrtj	0bcfc1d05f2e8dc4147743a1313aa70a119b41b30d4a1f…	2014.0	1240000	NaN	2016.0

```
[153443 rows x 6 columns]
```

[12]:
```
df_first_values = df_first_values[(df_first_values["orgyear"]>=2013) &␣
↪(df_first_values["orgyear"]<=2024)]
df_first_values
```

[12]:
```
                    company_hash  \
0                   atrgxnnt xzaxv
1         qtrxvzwt xzegwgbb rxbxnta
2                    ojzwnvwnxw vx
3                        ngpgutaxv
4                       qxen sqghu
…                              …
153437                         xgz
153439                   husqvawgb
153440                    vwwgrxnt
153441              zgn vuurxwvmrt
153442              bgqsvz onvzrtj

                                                email_hash  orgyear      ctc  \
0       6de0a4417d18ab14334c3f43397fc13b30c35149d70c05…   2016.0  1100000
1       b0aaf1ac138b53cb6e039ba2c3d6604a250d02d5145c10…   2018.0   449999
2       4860c670bcd48fb96c02a4b0ae3608ae6fdd98176112e9…   2015.0  2000000
3       effdede7a2e7c2af664c8a31d9346385016128d66bbc58…   2017.0   700000
4       6ff54e709262f55cb999a1c1db8436cb2055d8f79ab520…   2017.0  1400000
…                                                     …       …       …
153437  b8632b290be3b4b50c5ae979f3bf9a79ac805d172b1459…   2013.0  2280000
153439  7f7292ffad724ebbe9ca860f515245368d714c84705b42…   2017.0   500000
153440  cb25cc7304e9a24facda7f5567c7922ffc48e3d5d6018c…   2021.0   700000
153441  fb46a1a2752f5f652ce634f6178d0578ef6995ee59f6c8…   2019.0  5100000
153442  0bcfc1d05f2e8dc4147743a1313aa70a119b41b30d4a1f…   2014.0  1240000

              job_position  ctc_updated_year
0                    Other            2020.0
1        FullStack Engineer            2019.0
2          Backend Engineer            2020.0
3          Backend Engineer            2019.0
4        FullStack Engineer            2019.0
…                        …                 …
153437                 NaN            2019.0
153439                 NaN            2020.0
153440                 NaN            2021.0
153441                 NaN            2019.0
153442                 NaN            2016.0

[117999 rows x 6 columns]
```

[13]:
```
df_first_values["orgyear"] = df_first_values["orgyear"].astype("int64")
df_first_values["ctc_updated_year"] = df_first_values["ctc_updated_year"].
 ↪astype("int64")

df_first_values["years_of_Experience"] = df_first_values["ctc_updated_year"] -␣
 ↪df_first_values["orgyear"]
df_first_values
```

[13]:

	company_hash	email_hash	orgyear	ctc	job_position	ctc_updated_year	years_of_Experience
0	atrgxnnt xzaxv	6de0a4417d18ab14334c3f43397fc13b30c35149d70c05…	2016	1100000	Other	2020	4
1	qtrxvzwt xzegwgbb rxbxnta	b0aaf1ac138b53cb6e039ba2c3d6604a250d02d5145c10…	2018	449999	FullStack Engineer	2019	1
2	ojzwnvwnxw vx	4860c670bcd48fb96c02a4b0ae3608ae6fdd98176112e9…	2015	2000000	Backend Engineer	2020	5
3	ngpgutaxv	effdede7a2e7c2af664c8a31d9346385016128d66bbc58…	2017	700000	Backend Engineer	2019	2
4	qxen sqghu	6ff54e709262f55cb999a1c1db8436cb2055d8f79ab520…	2017	1400000	FullStack Engineer	2019	2
…	…	…	…	…	…	…	…
153437	xgz	b8632b290be3b4b50c5ae979f3bf9a79ac805d172b1459…	2013	2280000	NaN	2019	6
153439	husqvawgb	7f7292ffad724ebbe9ca860f515245368d714c84705b42…	2017	500000	NaN	2020	3
153440	vwwgrxnt	cb25cc7304e9a24facda7f5567c7922ffc48e3d5d6018c…	2021	700000	NaN	2021	0
153441	zgn vuurxwvmrt	fb46a1a2752f5f652ce634f6178d0578ef6995ee59f6c8…	2019	5100000	NaN	2019	0
153442	bgqsvz onvzrtj	0bcfc1d05f2e8dc4147743a1313aa70a119b41b30d4a1f…	2014	1240000	NaN	2016	2

```
[117999 rows x 7 columns]
```

[14]:
```
df_first_values = df_first_values[df_first_values["years_of_Experience"]>=0]
df_first_values
```

[14]:

	company_hash	email_hash	orgyear	ctc	job_position	ctc_updated_year	years_of_Experience
0	atrgxnnt xzaxv	6de0a4417d18ab14334c3f43397fc13b30c35149d70c05…	2016	1100000	Other	2020	4
1	qtrxvzwt xzegwgbb rxbxnta	b0aaf1ac138b53cb6e039ba2c3d6604a250d02d5145c10…	2018	449999	FullStack Engineer	2019	1
2	ojzwnvwnxw vx	4860c670bcd48fb96c02a4b0ae3608ae6fdd98176112e9…	2015	2000000	Backend Engineer	2020	5
3	ngpgutaxv	effdede7a2e7c2af664c8a31d9346385016128d66bbc58…	2017	700000	Backend Engineer	2019	2
4	qxen sqghu	6ff54e709262f55cb999a1c1db8436cb2055d8f79ab520…	2017	1400000	FullStack Engineer	2019	2
…	…	…	…	…	…	…	…
153437	xgz	b8632b290be3b4b50c5ae979f3bf9a79ac805d172b1459…	2013	2280000	NaN	2019	6
153439	husqvawgb	7f7292ffad724ebbe9ca860f515245368d714c84705b42…	2017	500000	NaN	2020	3
153440	vwwgrxnt	cb25cc7304e9a24facda7f5567c7922ffc48e3d5d6018c…	2021	700000	NaN	2021	0
153441	zgn vuurxwvmrt	fb46a1a2752f5f652ce634f6178d0578ef6995ee59f6c8…	2019	5100000	NaN	2019	0
153442	bgqsvz onvzrtj	0bcfc1d05f2e8dc4147743a1313aa70a119b41b30d4a1f…	2014	1240000	NaN	2016	2

```
[110847 rows x 7 columns]
```

[15]:
```
df_first_values = df_first_values[(df_first_values["ctc"]>=60000) &␣
↪(df_first_values["ctc"]<=20000000)]
df_first_values
```

[15]:
```
                    company_hash  \
0                  atrgxnnt xzaxv
1       qtrxvzwt xzegwgbb rxbxnta
2                   ojzwnvwnxw vx
3                       ngpgutaxv
4                      qxen sqghu
…                             …
153437                        xgz
153439                   husqvawgb
153440                    vwwgrxnt
153441            zgn vuurxwvmrt
153442            bgqsvz onvzrtj

                                                email_hash  orgyear      ctc  \
0       6de0a4417d18ab14334c3f43397fc13b30c35149d70c05…      2016  1100000
1       b0aaf1ac138b53cb6e039ba2c3d6604a250d02d5145c10…      2018   449999
2       4860c670bcd48fb96c02a4b0ae3608ae6fdd98176112e9…      2015  2000000
3       effdede7a2e7c2af664c8a31d9346385016128d66bbc58…      2017   700000
4       6ff54e709262f55cb999a1c1db8436cb2055d8f79ab520…      2017  1400000
…                                                     …       …       …
153437  b8632b290be3b4b50c5ae979f3bf9a79ac805d172b1459…      2013  2280000
153439  7f7292ffad724ebbe9ca860f515245368d714c84705b42…      2017   500000
153440  cb25cc7304e9a24facda7f5567c7922ffc48e3d5d6018c…      2021   700000
153441  fb46a1a2752f5f652ce634f6178d0578ef6995ee59f6c8…      2019  5100000
153442  0bcfc1d05f2e8dc4147743a1313aa70a119b41b30d4a1f…      2014  1240000

             job_position  ctc_updated_year  years_of_Experience
0                   Other              2020                    4
1       FullStack Engineer             2019                    1
2         Backend Engineer             2020                    5
3         Backend Engineer             2019                    2
4       FullStack Engineer             2019                    2
…                       …                …                    …
153437                NaN              2019                    6
153439                NaN              2020                    3
153440                NaN              2021                    0
153441                NaN              2019                    0
153442                NaN              2016                    2

[108277 rows x 7 columns]
```

[16]:
```
df_first_values["ctc"].min()
```

[16]:
```
60000
```

[17]:
```
df_first_values["ctc"].max()
```

[17]:
```
20000000
```

[18]:
```
df_first_values["job_position"] = df_first_values["job_position"].str.
↪replace("Other", "Unknown")
```

[19]:
```
df_first_values["job_position"].value_counts()
```

[19]:
```
job_position
Backend Engineer                    24557
FullStack Engineer                  13616
Unknown                             10345
Frontend Engineer                    5817
QA Engineer                          4071
                                    …
Application Developer (Frontend)        1
system engineer                         1
Software Developer II                   1
Graduate Software Developer             1
Android Application developer           1
Name: count, Length: 527, dtype: int64
```

[20]:
```
df_first_values["company_hash"].value_counts()
```

[20]:
```
company_hash
nvnv wgzohrnvzwj otqcxwto                   4503
xzegojo                                     2955
vbvkgz                                      1952
wgszxkvzn                                   1798
zgn vuurxwvmrt vwwghzn                      1671
                                            …
xzaxvz ouvwt qtotvqwy gqsvzxkvnxgz   xoqg      1
vbtqxntwy axsxnvr                           1
qtzx vrfvq                                  1
cgrvzjo                                     1
bvptbjnqxu td vbvkgz                        1
Name: count, Length: 26322, dtype: int64
```

[21]:
```
df_first_values.isna().sum()
```

[21]:
```
company_hash    24
email_hash       0
orgyear          0
ctc              0
```

```
job_position           25307
ctc_updated_year           0
years_of_Experience        0
dtype: int64
```

[22]:
```
df_first_values
```

[22]:
```
                    company_hash  \
0                  atrgxnnt xzaxv
1       qtrxvzwt xzegwgbb rxbxnta
2                   ojzwnvwnxw vx
3                       ngpgutaxv
4                      qxen sqghu
…                             …
153437                       xgz
153439                  husqvawgb
153440                   vwwgrxnt
153441             zgn vuurxwvmrt
153442             bgqsvz onvzrtj

                                                email_hash  orgyear      ctc  \
0       6de0a4417d18ab14334c3f43397fc13b30c35149d70c05…      2016  1100000
1       b0aaf1ac138b53cb6e039ba2c3d6604a250d02d5145c10…      2018   449999
2       4860c670bcd48fb96c02a4b0ae3608ae6fdd98176112e9…      2015  2000000
3       effdede7a2e7c2af664c8a31d9346385016128d66bbc58…      2017   700000
4       6ff54e709262f55cb999a1c1db8436cb2055d8f79ab520…      2017  1400000
…                                                     …       …       …
153437  b8632b290be3b4b50c5ae979f3bf9a79ac805d172b1459…      2013  2280000
153439  7f7292ffad724ebbe9ca860f515245368d714c84705b42…      2017   500000
153440  cb25cc7304e9a24facda7f5567c7922ffc48e3d5d6018c…      2021   700000
153441  fb46a1a2752f5f652ce634f6178d0578ef6995ee59f6c8…      2019  5100000
153442  0bcfc1d05f2e8dc4147743a1313aa70a119b41b30d4a1f…      2014  1240000

             job_position  ctc_updated_year  years_of_Experience
0                 Unknown              2020                    4
1       FullStack Engineer             2019                    1
2         Backend Engineer             2020                    5
3         Backend Engineer             2019                    2
4       FullStack Engineer             2019                    2
…                       …                …                    …
153437                NaN              2019                    6
153439                NaN              2020                    3
153440                NaN              2021                    0
153441                NaN              2019                    0
153442                NaN              2016                    2

[108277 rows x 7 columns]
```

[23]:
```
from sklearn.preprocessing import LabelEncoder
# Copy the DataFrame to avoid changing the original
df_imputed = df_first_values.copy()

# Initialize LabelEncoder
label_encoder = LabelEncoder()

# Apply LabelEncoder to job_position while preserving NaN values
df_imputed['job_position_encoded'] = df_imputed['job_position'].astype(str)  #␣
 ↪Convert to string to handle NaN
df_imputed.loc[df_imputed['job_position'].notnull(), 'job_position_encoded'] =␣
 ↪label_encoder.fit_transform(df_imputed['job_position'].dropna())

# Convert back to numeric (the NaN remains as is)
df_imputed['job_position_encoded'] = pd.
 ↪to_numeric(df_imputed['job_position_encoded'], errors='coerce')
```

[24]:
```
# Select relevant columns for KNN Imputer
impute_columns = ['job_position_encoded', 'orgyear', 'ctc', 'ctc_updated_year',␣
 ↪'years_of_Experience']

# Initialize the KNNImputer
knn_imputer = KNNImputer(n_neighbors=5)

# Perform KNN imputation
df_imputed[impute_columns] = knn_imputer.
 ↪fit_transform(df_imputed[impute_columns])

# Reverse the LabelEncoder transformation for the job_position column
df_imputed['job_position'] = label_encoder.
 ↪inverse_transform(df_imputed['job_position_encoded'].astype(int))
df_imputed
```

[24]:
```
                    company_hash  \
0                   atrgxnnt xzaxv
1       qtrxvzwt xzegwgbb rxbxnta
2                    ojzwnvwnxw vx
3                        ngpgutaxv
4                       qxen sqghu
…                              …
153437                        xgz
153439                   husqvawgb
153440                    vwwgrxnt
153441              zgn vuurxwvmrt
153442              bgqsvz onvzrtj

                                                email_hash  orgyear        ctc  \
```

```
0       6de0a4417d18ab14334c3f43397fc13b30c35149d70c05…   2016.0  1100000.0
1       b0aaf1ac138b53cb6e039ba2c3d6604a250d02d5145c10…   2018.0   449999.0
2       4860c670bcd48fb96c02a4b0ae3608ae6fdd98176112e9…   2015.0  2000000.0
3       effdede7a2e7c2af664c8a31d9346385016128d66bbc58…   2017.0   700000.0
4       6ff54e709262f55cb999a1c1db8436cb2055d8f79ab520…   2017.0  1400000.0
…                                                       …       …       …
153437  b8632b290be3b4b50c5ae979f3bf9a79ac805d172b1459…   2013.0  2280000.0
153439  7f7292ffad724ebbe9ca860f515245368d714c84705b42…   2017.0   500000.0
153440  cb25cc7304e9a24facda7f5567c7922ffc48e3d5d6018c…   2021.0   700000.0
153441  fb46a1a2752f5f652ce634f6178d0578ef6995ee59f6c8…   2019.0  5100000.0
153442  0bcfc1d05f2e8dc4147743a1313aa70a119b41b30d4a1f…   2014.0  1240000.0

                              job_position  ctc_updated_year  \
0                                  Unknown            2020.0
1                        FullStack Engineer            2019.0
2                          Backend Engineer            2020.0
3                          Backend Engineer            2019.0
4                        FullStack Engineer            2019.0
…                                        …                …
153437                   Senior QA Engineer            2019.0
153439  Lead Engineer - Software Development            2020.0
153440             Software Developer Intern            2021.0
153441                 Compliance operation             2019.0
153442           Back office executive Admin            2016.0

        years_of_Experience  job_position_encoded
0                       4.0                 500.0
1                       1.0                 164.0
2                       5.0                  86.0
3                       2.0                  86.0
4                       2.0                 164.0
…                       …                   …
153437                  6.0                 348.6
153439                  3.0                 196.2
153440                  0.0                 388.4
153441                  0.0                 109.4
153442                  2.0                  84.4

[108277 rows x 8 columns]
```

[25]:
```
label_encoder_company_hash = LabelEncoder()

# Encode 'company_hash'
df_imputed['company_hash_encoded'] = df_imputed['company_hash'].astype(str)
df_imputed.loc[df_imputed['company_hash'].notnull(), 'company_hash_encoded'] =␣
 ↪label_encoder_company_hash.fit_transform(df_imputed['company_hash'].dropna())
```

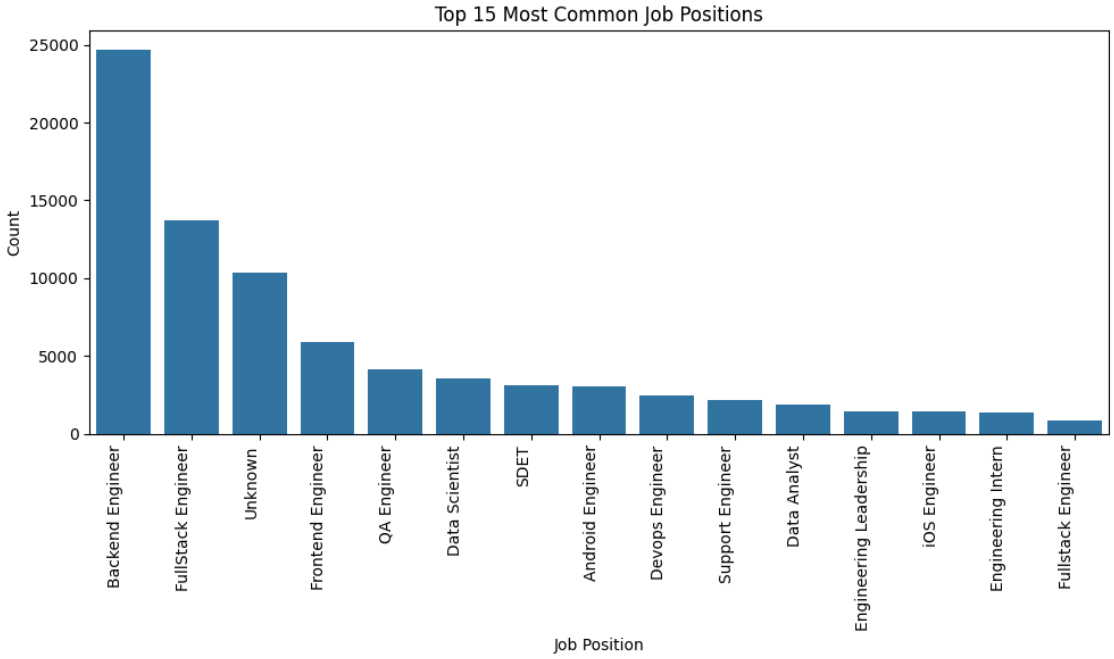
```
# Convert back to numeric, keeping NaN values as is
df_imputed['company_hash_encoded'] = pd.
 ↪to_numeric(df_imputed['company_hash_encoded'], errors='coerce')
```

[26]:
```
# Select relevant columns for KNN Imputer
impute_columns = ['job_position_encoded', 'company_hash_encoded', 'orgyear',␣
 ↪'ctc', 'ctc_updated_year', 'years_of_Experience']

# Initialize the KNNImputer
knn_imputer = KNNImputer(n_neighbors=5)

# Perform KNN imputation
df_imputed[impute_columns] = knn_imputer.
 ↪fit_transform(df_imputed[impute_columns])
```

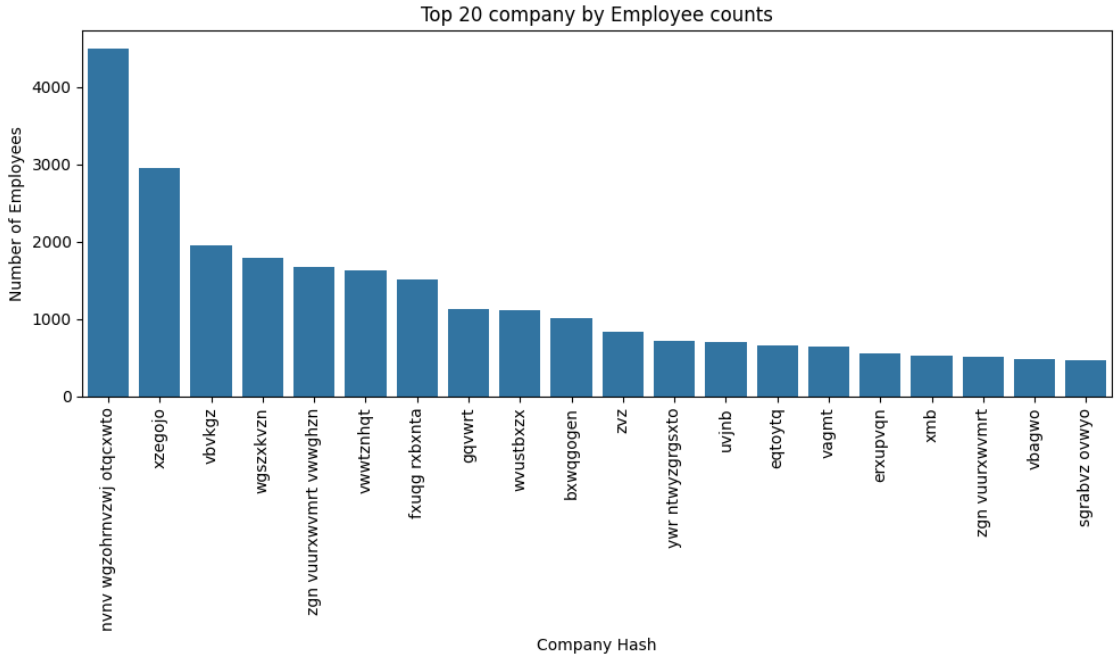
[27]:
```
# Decode 'company_hash'
df_imputed['company_hash'] = label_encoder_company_hash.
 ↪inverse_transform(df_imputed['company_hash_encoded'].astype(int))
df_imputed.isna().sum()
```

[27]:
```
company_hash            0
email_hash              0
orgyear                 0
ctc                     0
job_position            0
ctc_updated_year        0
years_of_Experience     0
job_position_encoded    0
company_hash_encoded    0
dtype: int64
```

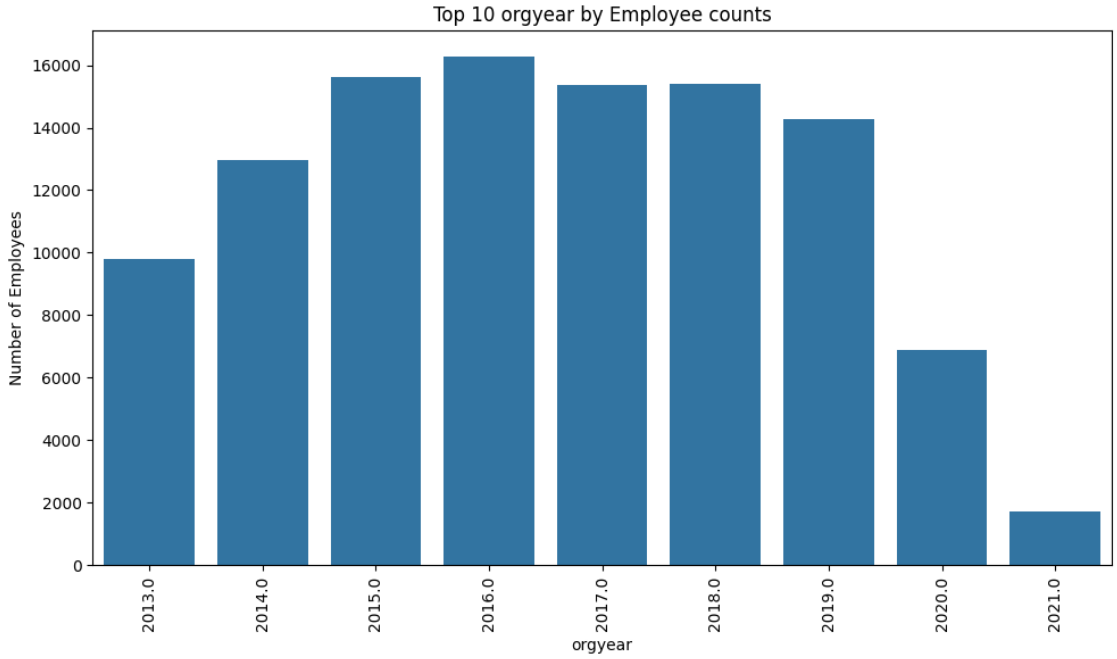
[28]:
```
top_10_jobs = df_imputed['job_position'].value_counts().nlargest(15)
plt.figure(figsize=(10, 6))
sns.barplot(x=top_10_jobs.index, y=top_10_jobs.values)
plt.xlabel('Job Position')
plt.ylabel('Count')
plt.title('Top 15 Most Common Job Positions')
plt.xticks(rotation=90, ha='right')
plt.tight_layout()
plt.show()
```



[29]:
```
top_20_company_counts = df_imputed['company_hash'].value_counts().nlargest(20)

plt.figure(figsize=(10, 6))
sns.barplot(x=top_20_company_counts.index, y=top_20_company_counts.values)
plt.xlabel('Company Hash')
plt.ylabel('Number of Employees')
plt.title('Top 20 company by Employee counts')
plt.xticks(rotation=90)
plt.tight_layout()
plt.show()
```



[30]:
```
counts_by_org_year = df_imputed['orgyear'].value_counts().nlargest(10)

plt.figure(figsize=(10, 6))
sns.barplot(x=counts_by_org_year.index, y=counts_by_org_year.values)
plt.xlabel('orgyear')
plt.ylabel('Number of Employees')
plt.title('Top 10 orgyear by Employee counts')
plt.xticks(rotation=90)
plt.tight_layout()
plt.show()
```



1 Manual Clustering

1.0.1 Manual Clustering on the basis of learner's company, job position and years of experience

1.0.2 Getting the 5 point summary of CTC (mean, median, max, min, count etc) on the basis of Company, Job Position, Years of Experience

[31]:
```
grouped = df_imputed.groupby(['company_hash', 'job_position',␣
 ↪'years_of_Experience'])
summary = grouped['ctc'].
 ↪agg(count='count',mean='mean',median='median',min='min',max='max').
 ↪reset_index()
summary.head()
```

[31]:

	company_hash	job_position	years_of_Experience	\
0	0	Data Analyst	0.0	
1	0000	Unknown	3.0	
2	01 ojztqsj	Android Engineer	3.0	
3	05mz exzytvrny uqxcvnt rxbxnta	Chief People Officer	2.0	
4	1	Quality Associate	3.0	

	count	mean	median	min	max
0	1	100000.0	100000.0	100000.0	100000.0
1	1	300000.0	300000.0	300000.0	300000.0

```
2      1   270000.0   270000.0   270000.0   270000.0
3      1  1100000.0  1100000.0  1100000.0  1100000.0
4      1   100000.0   100000.0   100000.0   100000.0
```

1.0.3 Merging the same with original dataset carefully and creating some flags showing learners with CTC greater than the Average of their Company's department having same Years of Experience - Call that flag designation with values [1,2,3]

[32]:
```
grouped = df_imputed.groupby(['company_hash', 'job_position',␣
 ↪'years_of_Experience'])['ctc'].mean().reset_index()
grouped = grouped.rename(columns={'ctc': 'avg_ctc'})

df_merged = pd.merge(df_imputed, grouped, on=['company_hash', 'job_position',␣
 ↪'years_of_Experience'], how='left')

# Create a flag 'designation' based on the condition:
# 1 if CTC > avg_ctc, 2 if CTC == avg_ctc, 3 if CTC < avg_ctc
df_merged['designation'] = df_merged.apply(lambda row: 1 if row['ctc'] >␣
 ↪row['avg_ctc'] else (2 if row['ctc'] == row['avg_ctc'] else 3), axis=1)
df_merged.head()
```

[32]:
```
               company_hash  \
0             atrgxnnt xzaxv
1  qtrxvzwt xzegwgbb rxbxnta
2              ojzwnvwnxw vx
3                  ngpgutaxv
4                 qxen sqghu

                                        email_hash  orgyear        ctc  \
0  6de0a4417d18ab14334c3f43397fc13b30c35149d70c05…   2016.0  1100000.0
1  b0aaf1ac138b53cb6e039ba2c3d6604a250d02d5145c10…   2018.0   449999.0
2  4860c670bcd48fb96c02a4b0ae3608ae6fdd98176112e9…   2015.0  2000000.0
3  effdede7a2e7c2af664c8a31d9346385016128d66bbc58…   2017.0   700000.0
4  6ff54e709262f55cb999a1c1db8436cb2055d8f79ab520…   2017.0  1400000.0

         job_position  ctc_updated_year  years_of_Experience  \
0             Unknown            2020.0                  4.0
1  FullStack Engineer            2019.0                  1.0
2    Backend Engineer            2020.0                  5.0
3    Backend Engineer            2019.0                  2.0
4  FullStack Engineer            2019.0                  2.0

   job_position_encoded  company_hash_encoded       avg_ctc  designation
0                 500.0                 683.0  1.085000e+06            1
1                 164.0               13936.0  4.599995e+05            3
2                  86.0               10957.0  2.000000e+06            2
```

```
3                  86.0                8558.0  1.298571e+06                3
4                 164.0               14287.0  1.000000e+06                1
```

1.0.4 Doing above analysis at Company & Job Position level. Name that flag Class with values [1,2,3]

[33]:
```
grouped1 = df_imputed.groupby(['company_hash', 'job_position'])['ctc'].mean().
 ↪reset_index()
grouped1 = grouped1.rename(columns={'ctc': 'avg_ctc'})

df_merged1 = pd.merge(df_imputed, grouped1, on=['company_hash',␣
 ↪'job_position'], how='left')

# Create a flag 'designation' based on the condition:
# 1 if CTC > avg_ctc, 2 if CTC == avg_ctc, 3 if CTC < avg_ctc
df_merged1['designation'] = df_merged1.apply(lambda row: 1 if row['ctc'] >␣
 ↪row['avg_ctc'] else (2 if row['ctc'] == row['avg_ctc'] else 3), axis=1)
df_merged1.head()
```

[33]:
```
                company_hash  \
0             atrgxnnt xzaxv
1  qtrxvzwt xzegwgbb rxbxnta
2              ojzwnvwnxw vx
3                  ngpgutaxv
4                 qxen sqghu

                                          email_hash  orgyear        ctc  \
0  6de0a4417d18ab14334c3f43397fc13b30c35149d70c05…    2016.0  1100000.0
1  b0aaf1ac138b53cb6e039ba2c3d6604a250d02d5145c10…    2018.0   449999.0
2  4860c670bcd48fb96c02a4b0ae3608ae6fdd98176112e9…    2015.0  2000000.0
3  effdede7a2e7c2af664c8a31d9346385016128d66bbc58…    2017.0   700000.0
4  6ff54e709262f55cb999a1c1db8436cb2055d8f79ab520…    2017.0  1400000.0

         job_position  ctc_updated_year  years_of_Experience  \
0             Unknown            2020.0                  4.0
1  FullStack Engineer            2019.0                  1.0
2    Backend Engineer            2020.0                  5.0
3    Backend Engineer            2019.0                  2.0
4  FullStack Engineer            2019.0                  2.0

   job_position_encoded  company_hash_encoded       avg_ctc  designation
0                 500.0                 683.0  1.085000e+06            1
1                 164.0               13936.0  7.683333e+05            3
2                  86.0               10957.0  2.000000e+06            2
3                  86.0                8558.0  1.543750e+06            3
4                 164.0               14287.0  8.466667e+05            1
```

1.0.5 Repeating the same analysis at the Company level. Name that flag Tier with values [1,2,3]

[34]:
```
grouped2 = df_imputed.groupby(['company_hash'])['ctc'].mean().reset_index()
grouped2 = grouped2.rename(columns={'ctc': 'avg_ctc'})

df_merged2 = pd.merge(df_imputed, grouped2, on=['company_hash'], how='left')

# Create a flag 'designation' based on the condition:
# 1 if CTC > avg_ctc, 2 if CTC == avg_ctc, 3 if CTC < avg_ctc
df_merged2['designation'] = df_merged2.apply(lambda row: 1 if row['ctc'] >␣
 ↪row['avg_ctc'] else (2 if row['ctc'] == row['avg_ctc'] else 3), axis=1)
df_merged2.head()
```

[34]:

	company_hash	email_hash	orgyear	ctc	job_position	ctc_updated_year	years_of_Experience	job_position_encoded	company_hash_encoded	avg_ctc	designation
0	atrgxnnt xzaxv	6de0a4417d18ab14334c3f43397fc13b30c35149d70c05…	2016.0	1100000.0	Unknown	2020.0	4.0	500.0	683.0	9.671429e+05	1
1	qtrxvzwt xzegwgbb rxbxnta	b0aaf1ac138b53cb6e039ba2c3d6604a250d02d5145c10…	2018.0	449999.0	FullStack Engineer	2019.0	1.0	164.0	13936.0	9.742621e+05	3
2	ojzwnvwnxw vx	4860c670bcd48fb96c02a4b0ae3608ae6fdd98176112e9…	2015.0	2000000.0	Backend Engineer	2020.0	5.0	86.0	10957.0	2.000000e+06	2
3	ngpgutaxv	effdede7a2e7c2af664c8a31d9346385016128d66bbc58…	2017.0	700000.0	Backend Engineer	2019.0	2.0	86.0	8558.0	1.522949e+06	3
4	qxen sqghu	6ff54e709262f55cb999a1c1db8436cb2055d8f79ab520…	2017.0	1400000.0	FullStack Engineer	2019.0	2.0	164.0	14287.0	8.480000e+05	1

1.0.6 Based on the manual clustering done so far, answering few questions like:Top 10 employees (earning more than most of the employees in the company) - Tier 1

[35]:
```
grouped = df_imputed.groupby(['company_hash', 'job_position',␣
 ↪'years_of_Experience'])['ctc'].agg(['mean', 'count']).reset_index()
grouped.head()
```

[35]:

	company_hash	job_position	years_of_Experience	\
0	0	Data Analyst	0.0	
1	0000	Unknown	3.0	
2	01 ojztqsj	Android Engineer	3.0	
3	05mz exzytvrny uqxcvnt rxbxnta	Chief People Officer	2.0	
4	1	Quality Associate	3.0	

	mean	count
0	100000.0	1
1	300000.0	1
2	270000.0	1
3	1100000.0	1
4	100000.0	1

[36]:
```
df_merged = pd.merge(df_imputed, grouped, on=['company_hash', 'job_position',␣
 ↪'years_of_Experience'], how='left')
df_merged['Class'] = df_merged.apply(lambda row: 1 if row['ctc'] > row['mean']␣
 ↪else 2 if row['ctc'] == row['mean'] else 3, axis=1)
df_merged.head()
```

[36]:

	company_hash	\
0	atrgxnnt xzaxv	
1	qtrxvzwt xzegwgbb rxbxnta	
2	ojzwnvwnxw vx	
3	ngpgutaxv	
4	qxen sqghu	

	email_hash	orgyear	ctc	\
0	6de0a4417d18ab14334c3f43397fc13b30c35149d70c05…	2016.0	1100000.0	
1	b0aaf1ac138b53cb6e039ba2c3d6604a250d02d5145c10…	2018.0	449999.0	
2	4860c670bcd48fb96c02a4b0ae3608ae6fdd98176112e9…	2015.0	2000000.0	
3	effdede7a2e7c2af664c8a31d9346385016128d66bbc58…	2017.0	700000.0	
4	6ff54e709262f55cb999a1c1db8436cb2055d8f79ab520…	2017.0	1400000.0	

	job_position	ctc_updated_year	years_of_Experience	\
0	Unknown	2020.0	4.0	
1	FullStack Engineer	2019.0	1.0	
2	Backend Engineer	2020.0	5.0	
3	Backend Engineer	2019.0	2.0	

```
4  FullStack Engineer            2019.0                  2.0

   job_position_encoded  company_hash_encoded          mean  count  Class
0                 500.0                 683.0  1.085000e+06      2      1
1                 164.0               13936.0  4.599995e+05      2      3
2                  86.0               10957.0  2.000000e+06      1      2
3                  86.0                8558.0  1.298571e+06      7      3
4                 164.0               14287.0  1.000000e+06      2      1
```

[37]:
```
df_merged = df_merged.sort_values(['company_hash', 'ctc'], ascending=[True,␣
 ↪False])

df_merged['Tier'] = df_merged.groupby('company_hash')['ctc'].
 ↪rank(method='first', ascending=False)
df_merged['Tier 1'] = df_merged['Tier'].apply(lambda x: 'Tier 1' if x <= 10␣
 ↪else '')
df_merged.head()
```

[37]:
```
                         company_hash  \
2451                                0
104559                           0000
45546                      01 ojztqsj
57764   05mz exzytvrny uqxcvnt rxbxnta
1851                                1

                                              email_hash  orgyear        ctc  \
2451    e80f7c9c26012bfdeca551e2b8642a93e45939d3d677c5…   2020.0   100000.0
104559  b3f3bb98cbca4b1ce5dfd5abb4e500ce6f6b66288a5202…   2017.0   300000.0
45546   819789ff4068fd5c8facf8a5074cdd2e1ff989c95ae02c…   2016.0   270000.0
57764   4702229ffb6968c87b16fc57e730769e554184e322e111…   2019.0  1100000.0
1851    8cc7aba49e96a0a80f7ed6c2ed79bc1d1e81171a28445c…   2017.0   100000.0

                job_position  ctc_updated_year  years_of_Experience  \
2451            Data Analyst            2020.0                  0.0
104559               Unknown            2020.0                  3.0
45546      Android Engineer            2019.0                  3.0
57764   Chief People Officer            2021.0                  2.0
1851       Quality Associate            2020.0                  3.0

        job_position_encoded  company_hash_encoded       mean  count  Class  \
2451                   121.6                   0.0   100000.0      1      2
104559                 500.0                   1.0   300000.0      1      2
45546                   19.0                   2.0   270000.0      1      2
57764                  101.6                   3.0  1100000.0      1      2
1851                   282.2                   4.0   100000.0      1      2

        Tier  Tier 1
```

```
2451     1.0  Tier 1
104559   1.0  Tier 1
45546    1.0  Tier 1
57764    1.0  Tier 1
1851     1.0  Tier 1
```

[38]:
```
tier_1_employees = df_merged[df_merged['Tier 1'] == 'Tier 1']
tier_1_employees
```

[38]:
```
                              company_hash  \
2451                                     0
104559                                0000
45546                          01 ojztqsj
57764      05mz exzytvrny uqxcvnt rxbxnta
1851                                     1
…                                      …
17525                          zyco xzaxv
106110  zyvzwt fgga qtztfvr eqvzwyxogq yi
3594                                    zz
44653         zzb ztdnstz vacxogqj ucn rna
67832                           zzgato

                                                email_hash  orgyear        ctc  \
2451    e80f7c9c26012bfdeca551e2b8642a93e45939d3d677c5…   2020.0   100000.0
104559  b3f3bb98cbca4b1ce5dfd5abb4e500ce6f6b66288a5202…   2017.0   300000.0
45546   819789ff4068fd5c8facf8a5074cdd2e1ff989c95ae02c…   2016.0   270000.0
57764   4702229ffb6968c87b16fc57e730769e554184e322e111…   2019.0  1100000.0
1851    8cc7aba49e96a0a80f7ed6c2ed79bc1d1e81171a28445c…   2017.0   100000.0
…                                                    …      …        …
17525   ed0a1202c31bdee343662f5517fc467fb8b96ccaf8e3eb…   2013.0   600000.0
106110  d1049cc22e5a8267cc61e61c1afa4753d554a9298b3744…   2015.0   900000.0
3594    7d4588453bc463b39db8c77ef0f856957fc42f5d54cae4…   2013.0  1370000.0
44653   ca8935e2314a1bac3947e60bbd2ee10524112898da29eb…   2017.0   600000.0
67832   d421e52125f8057c65fa554752be03b056221c8590ff26…   2014.0   130000.0

                job_position  ctc_updated_year  years_of_Experience  \
2451             Data Analyst            2020.0                  0.0
104559                Unknown            2020.0                  3.0
45546       Android Engineer            2019.0                  3.0
57764   Chief People Officer            2021.0                  2.0
1851       Quality Associate            2020.0                  3.0
…                        …                 …                    …
17525                 Unknown            2021.0                  8.0
106110        IoT Consultant            2017.0                  2.0
3594                  Unknown            2020.0                  7.0
44653      FullStack Engineer            2021.0                  4.0
67832     Credit risk manager            2017.0                  3.0
```

```
        job_position_encoded  company_hash_encoded       mean  count  Class  \
2451                   121.6                   0.0   100000.0      1      2
104559                 500.0                   1.0   300000.0      1      2
45546                   19.0                   2.0   270000.0      1      2
57764                  101.6                   3.0  1100000.0      1      2
1851                   282.2                   4.0   100000.0      1      2
…                      …                       …     …       …     …
17525                  500.0                26317.0   600000.0      1      2
106110                 186.2                26318.0   900000.0      1      2
3594                   500.0                26319.0  1370000.0      1      2
44653                  164.0                26320.0   600000.0      1      2
67832                  117.2                26321.0   130000.0      1      2

        Tier  Tier 1
2451     1.0  Tier 1
104559   1.0  Tier 1
45546    1.0  Tier 1
57764    1.0  Tier 1
1851     1.0  Tier 1
…        …    …
17525    1.0  Tier 1
106110   1.0  Tier 1
3594     1.0  Tier 1
44653    1.0  Tier 1
67832    1.0  Tier 1

[48942 rows x 14 columns]
```

1.0.7 Top 10 employees of data science in each company earning more than their peers - Class 1

[39]:
```
data_science_roles = ['Data Science Analyst','Data Scientist','Data Scientist␣
 ↪II','Associate␣Data Scientist','Senior Data Scientist']  # Add more roles if␣
 ↪necessary
data_science_df = df_merged[df_merged['job_position'].isin(data_science_roles)]

class_1_data_science = data_science_df[data_science_df['Class'] == 1]

class_1_data_science['Rank'] = class_1_data_science.
 ↪groupby('company_hash')['ctc'].rank(method='first', ascending=False)

top_10_class_1 = class_1_data_science[class_1_data_science['Rank'] <= 10]

top_10_class_1
```

[39]:

	company_hash	email_hash	\
89679	3p ntwyzgrgsxto	af617ba27ec944771314f1c2d739b8208d2b3337800f8f…	
62137	3rgi	a372713f7d18e6f03b5b469cbd1ddb8145c2688597c528…	
24867	adw ntwyzgrgsj	1f0a9f3b6a3de411b2ceece85ef7ef095278bf45496b0d…	
45446	adw ntwyzgrgsj	f77930e695dabbd49c7e2dc1d9cfe96c1d3b4808418c65…	
46685	adw ntwyzgrgsxto	d561093f7768b6db57a81e98f90897028dcf48881123d8…	
…	…	…	
34344	zxtrotz	751b1fb94f9054ecc14b44ebf91c3cbd92a47ea0194492…	
20069	zxtrotz	2182cd4f16b2a915d6c53f901f9911bb6b3b22caaaa0ae…	
64901	zxtrotz	163e546c8418ddc4b300471c1472044f582cd3be008da1…	
58314	zxtrotz	d5d7fa93cf62d046654e21716c7bdd613e5f559b47bc21…	
102507	zxztrtvuo	10d566c5fca40ffe1d133b79594d071880711ef480da9f…	

	orgyear	ctc	job_position	ctc_updated_year	\
89679	2018.0	1800000.0	Data Scientist	2019.0	
62137	2014.0	1710000.0	Data Scientist	2019.0	
24867	2015.0	1220000.0	Data Scientist	2019.0	
45446	2014.0	850000.0	Data Scientist	2019.0	
46685	2014.0	1760000.0	Data Scientist	2019.0	
…	…	…	…	…	
34344	2013.0	3000000.0	Data Scientist	2019.0	
20069	2017.0	750000.0	Data Scientist	2019.0	
64901	2015.0	700000.0	Data Scientist	2019.0	
58314	2017.0	650000.0	Data Scientist	2019.0	
102507	2017.0	1400000.0	Data Scientist	2019.0	

	years_of_Experience	job_position_encoded	company_hash_encoded	\
89679	1.0	125.0	109.0	
62137	5.0	125.0	111.0	
24867	4.0	125.0	226.0	
45446	5.0	125.0	226.0	
46685	5.0	125.0	234.0	
…	…	…	…	
34344	6.0	125.0	26269.0	
20069	2.0	125.0	26269.0	
64901	4.0	125.0	26269.0	
58314	2.0	125.0	26269.0	
102507	2.0	125.0	26314.0	

	mean	count	Class	Tier	Tier 1	Rank
89679	1.500000e+06	2	1	1.0	Tier 1	1.0
62137	1.155000e+06	2	1	2.0	Tier 1	1.0
24867	9.600000e+05	2	1	7.0	Tier 1	1.0
45446	8.250000e+05	2	1	19.0		2.0
46685	1.473333e+06	3	1	3.0	Tier 1	1.0
…	…	…	…	…	…	…
34344	2.045000e+06	2	1	1.0	Tier 1	1.0

```
20069   5.766667e+05      3      1  11.0           2.0
64901   6.666667e+05      3      1  13.0           3.0
58314   5.766667e+05      3      1  14.0           4.0
102507  1.325000e+06      2      1   7.0  Tier 1    1.0

[473 rows x 15 columns]
```

1.0.8 Bottom 10 employees of data science in each company earning less than their peers - Class 3

[40]:
```
data_science_roles = ['Data Science Analyst','Data Scientist','Data Scientist␣
↪II','Associate␣Data Scientist','Senior Data Scientist']  # Add more roles if␣
↪necessary
data_science_df = df_merged[df_merged['job_position'].isin(data_science_roles)]

# Filter further for employees in Class 3 (earning less than their peers)
class_3_data_science = data_science_df[data_science_df['Class'] == 3]

# Rank Data Science employees by 'ctc' within each company
class_3_data_science['Rank'] = class_3_data_science.
↪groupby('company_hash')['ctc'].rank(method='first', ascending=True)

# Filter to get the bottom 10 employees per company
bottom_10_class_3 = class_3_data_science[class_3_data_science['Rank'] <= 10]
bottom_10_class_3
```

[40]:
```
            company_hash                                      email_hash  \
62415    3p ntwyzgrgsxto  583a9600d8e74d5f3f6627da6b4ff3c466bf5cfee5ae9f…
76047               3rgi  24db964005796c656431df0b035768e8b9cee21f8cf425…
48626     adw ntwyzgrgsj  1443f6d836c4dc24a7a79f7f81702e6b684abdd22ea50e…
69862     adw ntwyzgrgsj  6ca1d030df1927f2c0904548a99dea059fd0aeaf39cb21…
101847  adw ntwyzgrgsxto  b7c81997457405be4a6c61fbe058f8de0e5b7929743499…
…                   …                                               …
36229           zxtrotz  2b7979bb62110fbb5e97f3f7f25d79f102536d3b84d538…
91572           zxtrotz  01717d934c1c75cacd31e29f8adcb5c109c627f7f26214…
104744          zxtrotz  af256544a43a5902d769859c21e05df919f05b490a227a…
7480            zxtrotz  8db7199e084be127053249086830cf3cda7f595e883a22…
33762         zxztrtvuo  f678c67bee8cad9370f6aaf4f4cc22ffd417fd753663c6…

        orgyear        ctc    job_position  ctc_updated_year  \
62415    2018.0  1200000.0  Data Scientist            2019.0
76047    2015.0   600000.0  Data Scientist            2020.0
48626    2014.0   800000.0  Data Scientist            2019.0
69862    2015.0   700000.0  Data Scientist            2019.0
101847   2015.0  1000000.0  Data Scientist            2019.0
…           …        …              …                 …
```

```
36229    2013.0  1090000.0  Data Scientist            2019.0
91572    2015.0   650000.0  Data Scientist            2019.0
104744   2015.0   650000.0  Data Scientist            2019.0
7480     2018.0   330000.0  Data Scientist            2020.0
33762    2019.0  1250000.0  Data Scientist            2021.0

        years_of_Experience  job_position_encoded  company_hash_encoded  \
62415                   1.0                 125.0                 109.0
76047                   5.0                 125.0                 111.0
48626                   5.0                 125.0                 226.0
69862                   4.0                 125.0                 226.0
101847                  4.0                 125.0                 234.0
…                       …                   …                     …
36229                   6.0                 125.0               26269.0
91572                   4.0                 125.0               26269.0
104744                  4.0                 125.0               26269.0
7480                    2.0                 125.0               26269.0
33762                   2.0                 125.0               26314.0

                mean  count  Class  Tier  Tier 1  Rank
62415   1.500000e+06      2      3   3.0  Tier 1   1.0
76047   1.155000e+06      2      3   8.0  Tier 1   1.0
48626   8.250000e+05      2      3  27.0           2.0
69862   9.600000e+05      2      3  38.0           1.0
101847  1.250000e+06      2      3  13.0           2.0
…                …    …     …    …      …   …
36229   2.045000e+06      2      3   6.0  Tier 1   4.0
91572   6.666667e+05      3      3  15.0           2.0
104744  6.666667e+05      3      3  16.0           3.0
7480    5.766667e+05      3      3  18.0           1.0
33762   1.325000e+06      2      3  10.0  Tier 1   1.0

[516 rows x 15 columns]
```

1.0.9 Bottom 10 employees (earning less than most of the employees in the company)- Tier 3

[41]:
```python
# Filter the DataFrame for Class 3 employees (those earning less than their␣
 ↪peers)
class_3_employees = df_merged[df_merged['Class'] == 3]

# Rank employees by 'ctc' within each company
class_3_employees['Rank'] = class_3_employees.groupby('company_hash')['ctc'].
 ↪rank(method='first', ascending=True)

# Filter to get the bottom 10 employees per company
bottom_10_employees = class_3_employees[class_3_employees['Rank'] <= 10]
```

```
# Create a 'Tier 3' designation
bottom_10_employees['Tier 3'] = bottom_10_employees['Rank'].apply(lambda x:␣
 ↪'Tier 3' if x <= 10 else '')

bottom_10_employees
```

[41]:

```
       company_hash                                        email_hash  \
60588         1 jtvq  26a4487b5be40da1e942bf3bc1189172727de0e1c74c5e…
92403            1bs  38dfe791fc911da418b67aa989a6aa7f00b8c680c6d4e1…
39196            1bs  c97fd1612080086b898e440529c86325ae8ddf2e9a0b60…
26348            1bs  08168bb1394cb9ebd0934c0021eba942cf915450259baf…
23810            1bs  13f9b6b7b59baa88d975991323dabdd66424550d633a88…
…                …                                                 …
7360       zxztrtvuo  01f98ed38bb3f6013cef8ebed0b0db568c586824b8e219…
44224      zxztrtvuo  f861d9f1bfee791938d90e9ad91069220eec8664b32fea…
56457      zxztrtvuo  d5b98d92628266c2037e78a367db956c140585fa027171…
90662      zxztrtvuo  0228801807a4911ebde807b5f88a273a51d92b25e6c160…
104441     zxztrtvuo  88534a8bc32c10480086fe9dac0375caeb489539561eee…

         orgyear        ctc        job_position  ctc_updated_year  \
60588     2018.0   660000.0    Backend Engineer            2019.0
92403     2015.0  2000000.0    Backend Engineer            2019.0
39196     2015.0  1800000.0    Backend Engineer            2019.0
26348     2017.0  1300000.0    Backend Engineer            2020.0
23810     2017.0  1000000.0    Backend Engineer            2021.0
…            …          …                   …                 …
7360      2019.0   500000.0   Frontend Engineer            2020.0
44224     2019.0   500000.0   Frontend Engineer            2020.0
56457     2019.0   500000.0    Backend Engineer            2021.0
90662     2019.0   500000.0  FullStack Engineer            2021.0
104441    2019.0   450000.0   Frontend Engineer            2020.0

         years_of_Experience  job_position_encoded  company_hash_encoded  \
60588                    1.0                  86.0                   6.0
92403                    4.0                  86.0                  25.0
39196                    4.0                  86.0                  25.0
26348                    3.0                  86.0                  25.0
23810                    4.0                  86.0                  25.0
…                        …                     …                     …
7360                     1.0                 161.0               26314.0
44224                    1.0                 161.0               26314.0
56457                    2.0                  86.0               26314.0
90662                    2.0                 164.0               26314.0
104441                   1.0                 161.0               26314.0

             mean  count  Class  Tier  Tier 1  Rank  Tier 3
```

```
60588   1180000.0      2      3   2.0  Tier 1   1.0  Tier 3
92403   2300000.0      6      3   5.0  Tier 1   8.0  Tier 3
39196   2300000.0      6      3   6.0  Tier 1   7.0  Tier 3
26348   1327500.0      4      3  16.0           6.0  Tier 3
23810   2300000.0      6      3  22.0           4.0  Tier 3
…           …    …    …   …      …   …      …
7360     514000.0      5      3  33.0           2.0  Tier 3
44224    514000.0      5      3  34.0           3.0  Tier 3
56457    876000.0      5      3  35.0           4.0  Tier 3
90662   1000000.0      2      3  36.0           5.0  Tier 3
104441   514000.0      5      3  42.0           1.0  Tier 3

[6301 rows x 16 columns]
```

1.0.10 Top 10 employees in each company - X department - having 5/6/7 years of experience earning more than their peers - Tier X

[42]:
```
# Filter the dataset for employees in "X department" with 5, 6, or 7 years of␣
 ↪experience
filtered_employees = df_merged[
    (df_merged['years_of_Experience'].isin([5, 6, 7]))]
filtered_employees
```

[42]:
```
                              company_hash  \
50255   12ny fgzatq qtotvqwy xzaxv ucn rna
14448                            1820axsxnvr
8885                                   1985
15180                                   1bs
103815                                  1bs
…                                       …
87097                             zxztrtvuo
72817                             zxztrtvuo
66627                             zxztrtvuo
923                           zxzvzxjv sqghu
3594                                    zz

                                                email_hash  orgyear        ctc  \
50255   735aeec9e89759154f3fa4f8f99e3da93b821669684298…   2013.0   650000.0
14448   7a64650f7c2c73dfe6b1a4c410eb64641e4dacd371bfda…   2014.0  1000000.0
8885    0e0b52e1fa76e607ca155769bbfb978c4dc91b7482df27…   2015.0  1100000.0
15180   a58fadbfbc00c007dfe6e5d5891f2dda013eb5cc66552a…   2014.0  1600000.0
103815  2b50861d0780b85284d70b0d8d284c6db631fc7462870f…   2014.0  1600000.0
…                                                          …        …
87097   1cd0a52ed52dae24d605d9cdc8536499c10ce62bfb070f…   2014.0  2250000.0
72817   3385dc93ba44f4f1cc237ef4f8e057dab2f693d8961b64…   2013.0  1800000.0
66627   4a5bc81d942bb281d18a27b6ceb622de65d07a13bc5ab9…   2014.0  1275000.0
923     6aeb0a4c55bd87b64c70a963b058a61c29b8654ac86fe9…   2015.0  1180000.0
```

```
3594    7d4588453bc463b39db8c77ef0f856957fc42f5d54cae4…   2013.0  1370000.0

                     job_position  ctc_updated_year  years_of_Experience  \
50255                        SDET            2019.0                  6.0
14448                          Na            2021.0                  7.0
8885    Linux System Administrator           2020.0                  5.0
15180          FullStack Engineer            2021.0                  7.0
103815                  Professor            2019.0                  5.0
…                              …                 …                    …
87097              Data Scientist            2021.0                  7.0
72817           Frontend Engineer            2019.0                  6.0
66627            Product Designer            2020.0                  6.0
923        Engineering Leadership            2020.0                  5.0
3594                      Unknown            2020.0                  7.0

        job_position_encoded  company_hash_encoded       mean  count  Class  \
50255                  311.0                  18.0   650000.0      1      2
14448                  227.4                  21.0  1000000.0      1      2
8885                   200.0                  23.0  1100000.0      1      2
15180                  164.0                  25.0  1600000.0      1      2
103815                 266.2                  25.0  1600000.0      1      2
…                          …                     …          …    …    …
87097                  125.0                26314.0  2250000.0      1      2
72817                  161.0                26314.0  1800000.0      1      2
66627                  256.2                26314.0  1275000.0      1      2
923                    147.0                26316.0  1180000.0      1      2
3594                   500.0                26319.0  1370000.0      1      2

        Tier  Tier 1
50255    1.0  Tier 1
14448    1.0  Tier 1
8885     1.0  Tier 1
15180    9.0  Tier 1
103815  12.0
…       …     …
87097    4.0  Tier 1
72817    5.0  Tier 1
66627    9.0  Tier 1
923      1.0  Tier 1
3594     1.0  Tier 1

[27693 rows x 14 columns]
```

[43]:
```
# Calculate the mean earnings ('ctc') for each company, job position, and years␣
 ↪of experience
grouped = filtered_employees.groupby(['company_hash', 'job_position',␣
 ↪'years_of_Experience'])['ctc'].agg(['mean']).reset_index()
```

```
grouped
```

[43]:
```
                            company_hash                job_position  \
0      12ny fgzatq qtotvqwy xzaxv ucn rna                        SDET
1                              1820axsxnvr                          Na
2                                     1985  Linux System Administrator
3                                      1bs         FullStack Engineer
4                                      1bs                   Professor
…                                      …                           …
20165                          zxztrtvuo           Frontend Engineer
20166                          zxztrtvuo           Frontend Engineer
20167                          zxztrtvuo            Product Designer
20168                       zxzvzxjv sqghu      Engineering Leadership
20169                                 zz                     Unknown

       years_of_Experience       mean
0                      6.0   650000.0
1                      7.0  1000000.0
2                      5.0  1100000.0
3                      7.0  1600000.0
4                      5.0  1600000.0
…                      …          …
20165                  6.0  1800000.0
20166                  7.0  2500000.0
20167                  6.0  1275000.0
20168                  5.0  1180000.0
20169                  7.0  1370000.0

[20170 rows x 4 columns]
```

[44]:
```
# Merge to get the mean ctc for comparison
filtered_employees = pd.merge(filtered_employees, grouped, on=['company_hash',␣
↪'job_position', 'years_of_Experience'], how='left')
filtered_employees
```

[44]:
```
                            company_hash  \
0      12ny fgzatq qtotvqwy xzaxv ucn rna
1                              1820axsxnvr
2                                     1985
3                                      1bs
4                                      1bs
…                                      …
27688                          zxztrtvuo
27689                          zxztrtvuo
27690                          zxztrtvuo
27691                       zxzvzxjv sqghu
27692                                 zz
```

```
                                                email_hash  orgyear        ctc  \
0      735aeec9e89759154f3fa4f8f99e3da93b821669684298…   2013.0   650000.0
1      7a64650f7c2c73dfe6b1a4c410eb64641e4dacd371bfda…   2014.0  1000000.0
2      0e0b52e1fa76e607ca155769bbfb978c4dc91b7482df27…   2015.0  1100000.0
3      a58fadbfbc00c007dfe6e5d5891f2dda013eb5cc66552a…   2014.0  1600000.0
4      2b50861d0780b85284d70b0d8d284c6db631fc7462870f…   2014.0  1600000.0
…                                                    …      …        …
27688  1cd0a52ed52dae24d605d9cdc8536499c10ce62bfb070f…   2014.0  2250000.0
27689  3385dc93ba44f4f1cc237ef4f8e057dab2f693d8961b64…   2013.0  1800000.0
27690  4a5bc81d942bb281d18a27b6ceb622de65d07a13bc5ab9…   2014.0  1275000.0
27691  6aeb0a4c55bd87b64c70a963b058a61c29b8654ac86fe9…   2015.0  1180000.0
27692  7d4588453bc463b39db8c77ef0f856957fc42f5d54cae4…   2013.0  1370000.0

                    job_position  ctc_updated_year  years_of_Experience  \
0                           SDET            2019.0                  6.0
1                             Na            2021.0                  7.0
2      Linux System Administrator            2020.0                  5.0
3              FullStack Engineer            2021.0                  7.0
4                       Professor            2019.0                  5.0
…                             …                 …                    …
27688             Data Scientist            2021.0                  7.0
27689          Frontend Engineer            2019.0                  6.0
27690           Product Designer            2020.0                  6.0
27691     Engineering Leadership            2020.0                  5.0
27692                    Unknown            2020.0                  7.0

       job_position_encoded  company_hash_encoded     mean_x  count  Class  \
0                     311.0                  18.0   650000.0      1      2
1                     227.4                  21.0  1000000.0      1      2
2                     200.0                  23.0  1100000.0      1      2
3                     164.0                  25.0  1600000.0      1      2
4                     266.2                  25.0  1600000.0      1      2
…                     …                      …     …     …    …
27688                 125.0                26314.0  2250000.0      1      2
27689                 161.0                26314.0  1800000.0      1      2
27690                 256.2                26314.0  1275000.0      1      2
27691                 147.0                26316.0  1180000.0      1      2
27692                 500.0                26319.0  1370000.0      1      2

       Tier  Tier 1     mean_y
0       1.0  Tier 1   650000.0
1       1.0  Tier 1  1000000.0
2       1.0  Tier 1  1100000.0
3       9.0  Tier 1  1600000.0
4      12.0          1600000.0
…      …     …         …
```

```
27688   4.0  Tier 1  2250000.0
27689   5.0  Tier 1  1800000.0
27690   9.0  Tier 1  1275000.0
27691   1.0  Tier 1  1180000.0
27692   1.0  Tier 1  1370000.0

[27693 rows x 15 columns]
```

[45]:
```
# Filter for employees earning more than their peers (greater than the mean␣
 ↪'ctc')
filtered_employees = filtered_employees[filtered_employees['ctc'] >␣
 ↪filtered_employees['mean_x']]

# Rank employees by 'ctc' within each company in descending order
filtered_employees['Rank'] = filtered_employees.groupby('company_hash')['ctc'].
 ↪rank(method='first', ascending=False)

# Select the top 10 employees per company
top_10_employees = filtered_employees[filtered_employees['Rank'] <= 10]

# Create a 'Tier X' designation
top_10_employees['Tier X'] = top_10_employees['Rank'].apply(lambda x: 'Tier X'␣
 ↪if x <= 10 else '')
top_10_employees
```

[45]:
```
                    company_hash  \
25                         247vx
55                          3rgi
70                 3x xzegntwy rna
124                 adw ntwyzgrgsj
128                 adw ntwyzgrgsj
…                              …
27644           zxxn ntwyzgrgsxto
27645           zxxn ntwyzgrgsxto
27646           zxxn ntwyzgrgsxto
27658  zxxn ntwyzgrgsxto rxbxnta
27683                   zxzlvwvqn

                                                email_hash  orgyear        ctc  \
25     5e63e173e7413b8ab790991bade2f4a814b897294b383c…   2013.0  1750000.0
55     a372713f7d18e6f03b5b469cbd1ddb8145c2688597c528…   2014.0  1710000.0
70     bc9299608cc9a5be077ce4e30802d09df016e2af03535b…   2014.0  2020000.0
124    cab00310b690ead0bd59fd160e9b160528c427c97bd038…   2014.0  4000000.0
128    ccd6080d08e8c6b5041569fb7fbd457fea0e2096d61b2e…   2014.0  1200000.0
…                                                    …        …          …
27644  b1011914447512b62a7c9974f92fdf4da9e0a9e19059b0…   2014.0   700000.0
27645  02a01571c7aef2bb2167d91653ef69a4c50368d1884b92…   2014.0   700000.0
```

```
27646  5b89e76138afba0d9ce0b07a29616ea6b48ca199b0cf75…   2014.0   700000.0
27658  e50925645389a6b6db07046634d7164717daa33dab0451…   2013.0  1100000.0
27683  6187e7fa4f28115a87f0b86930fe2d5df8d64078155d34…   2015.0  3600000.0

             job_position  ctc_updated_year  years_of_Experience  \
25       Backend Engineer            2019.0                  6.0
55         Data Scientist            2019.0                  5.0
70     FullStack Engineer            2021.0                  7.0
124               Unknown            2021.0                  7.0
128      Backend Engineer            2021.0                  7.0
…                      …                 …                    …
27644                SDET            2019.0                  5.0
27645         QA Engineer            2019.0                  5.0
27646         QA Engineer            2019.0                  5.0
27658  FullStack Engineer            2020.0                  7.0
27683    Backend Engineer            2020.0                  5.0

       job_position_encoded  company_hash_encoded        mean_x  count  Class  \
25                     86.0                  70.0  1.425000e+06      2      1
55                    125.0                 111.0  1.155000e+06      2      1
70                    164.0                 122.0  1.510000e+06      2      1
124                   500.0                 226.0  2.650000e+06      2      1
128                    86.0                 226.0  1.100000e+06      2      1
…                       …                    …           …    …    …
27644                 311.0               26296.0  6.666667e+05      3      1
27645                 278.0               26296.0  6.929999e+05     10      1
27646                 278.0               26296.0  6.929999e+05     10      1
27658                 164.0               26297.0  8.500000e+05      2      1
27683                  86.0               26310.0  2.550000e+06      2      1

       Tier  Tier 1        mean_y  Rank  Tier X
25      3.0  Tier 1  1.425000e+06   1.0  Tier X
55      2.0  Tier 1  1.155000e+06   1.0  Tier X
70      1.0  Tier 1  1.510000e+06   1.0  Tier X
124     2.0  Tier 1  2.650000e+06   1.0  Tier X
128     8.0  Tier 1  1.100000e+06   2.0  Tier X
…       …       …           …    …     …
27644  19.0          6.666667e+05   5.0  Tier X
27645  20.0          6.929999e+05   6.0  Tier X
27646  21.0          6.929999e+05   7.0  Tier X
27658  10.0  Tier 1  8.500000e+05   1.0  Tier X
27683   1.0  Tier 1  2.550000e+06   1.0  Tier X

[2339 rows x 17 columns]
```

1.0.11 Top 20 companies (based on their CTC)

[46]:
```
# Step 1: Calculate the average CTC for each company
company_avg_ctc = df_imputed.groupby('company_hash')['ctc'].mean().reset_index()

# Step 2: Sort companies by their average CTC in descending order
company_avg_ctc_sorted = company_avg_ctc.sort_values(by='ctc', ascending=False)

# Step 3: Select the top 10 companies
top_20_companies = company_avg_ctc_sorted.head(20)
top_20_companies
```

[46]:

	company_hash	ctc
1224	axvzvuuo ntwyzgrgsxto ucn rna	20000000.0
6584	jhnx atoxsztq yghot	20000000.0
15544	srgmvr tahwvnxgzvr otqcxwto	20000000.0
7611	mqgfz hzxctqoxnj	20000000.0
8541	ngjgnv bgngq wgqugqvnxgz	20000000.0
13726	qrgsxw ntwyzgrgsxto	20000000.0
7014	lghrtongfvnno	20000000.0
18845	vcvznt xzntqzvnxgzvr ntwyzgrgsj xzw	20000000.0
16948	twgbbtqwt onvqnhu	20000000.0
11347	opxrrovznv	19800000.0
13563	qgmgoyvonqv	19700000.0
21716	wo xzegwgbb	19200000.0
22581	wvzgnxw	18450000.0
24643	ygntr bxrrtzzxv qtstzwj rhwpzgf	18200000.0
3190	ctzatpxz ntwyzgrgsxto	18000000.0
16572	tong mqgvamvza uqxcvnt rxbxnta	18000000.0
11976	ovetnxuxz	17000000.0
26148	zwo oxzsvugqt	17000000.0
21383	wgxzmvot	16500000.0
18970	vfvvkat	16200000.0

1.0.12 Top 2 positions in every company (based on their CTC)

[47]:
```
grouped = df_imputed.groupby(['company_hash', 'job_position'])['ctc'].mean().
 ↪reset_index()

# Step 2: Rank job positions within each company based on their average CTC
grouped['Rank'] = grouped.groupby('company_hash')['ctc'].rank(method='first',␣
 ↪ascending=False)

# Step 3: Filter the top 2 positions in each company
top_positions = grouped[grouped['Rank'] <= 2]
top_positions
```

[47]:

	company_hash	job_position	ctc	\
0	0	Data Analyst	100000.0	
1	0000	Unknown	300000.0	
2	01 ojztqsj	Android Engineer	270000.0	
3	05mz exzytvrny uqxcvnt rxbxnta	Chief People Officer	1100000.0	
4	1	Quality Associate	100000.0	
…	…	…	…	
54092	zyco xzaxv	Unknown	600000.0	
54093	zyvzwt fgga qtztfvr eqvzwyxogq yi	IoT Consultant	900000.0	
54094	zz	Unknown	1370000.0	
54095	zzb ztdnstz vacxogqj ucn rna	FullStack Engineer	600000.0	
54096	zzgato	Credit risk manager	130000.0	

	Rank
0	1.0
1	1.0
2	1.0
3	1.0
4	1.0
…	…
54092	1.0
54093	1.0
54094	1.0
54095	1.0
54096	1.0

[32198 rows x 4 columns]

2 Data processing for Unsupervised clustering - Label encoding/ One- hot encoding, Standardization of data

[48]:
```
df_imputed.head()
```

[48]:

	company_hash	\
0	atrgxnnt xzaxv	
1	qtrxvzwt xzegwgbb rxbxnta	
2	ojzwnvwnxw vx	
3	ngpgutaxv	
4	qxen sqghu	

	email_hash	orgyear	ctc	\
0	6de0a4417d18ab14334c3f43397fc13b30c35149d70c05…	2016.0	1100000.0	
1	b0aaf1ac138b53cb6e039ba2c3d6604a250d02d5145c10…	2018.0	449999.0	
2	4860c670bcd48fb96c02a4b0ae3608ae6fdd98176112e9…	2015.0	2000000.0	
3	effdede7a2e7c2af664c8a31d9346385016128d66bbc58…	2017.0	700000.0	
4	6ff54e709262f55cb999a1c1db8436cb2055d8f79ab520…	2017.0	1400000.0	

```
         job_position  ctc_updated_year  years_of_Experience  \
0             Unknown            2020.0                  4.0
1  FullStack Engineer            2019.0                  1.0
2    Backend Engineer            2020.0                  5.0
3    Backend Engineer            2019.0                  2.0
4  FullStack Engineer            2019.0                  2.0

   job_position_encoded  company_hash_encoded
0                 500.0                 683.0
1                 164.0               13936.0
2                  86.0               10957.0
3                  86.0                8558.0
4                 164.0               14287.0
```

[49]:
```
from sklearn.preprocessing import LabelEncoder
email_hash_encoder = LabelEncoder()
df_imputed['email_hash_encoded'] = email_hash_encoder.
 ↪fit_transform(df_imputed['email_hash'])
# Frequency encoding for 'company_hash' and 'job_position'
df_imputed['company_hash_encoded'] = df_imputed['company_hash'].
 ↪map(df_imputed['company_hash'].value_counts())
df_imputed['job_position_encoded'] = df_imputed['job_position'].
 ↪map(df_imputed['job_position'].value_counts())
df_imputed.head()
```

[49]:
```
               company_hash  \
0            atrgxnnt xzaxv
1  qtrxvzwt xzegwgbb rxbxnta
2             ojzwnvwnxw vx
3                ngpgutaxv
4                qxen sqghu

                                       email_hash  orgyear        ctc  \
0  6de0a4417d18ab14334c3f43397fc13b30c35149d70c05…   2016.0  1100000.0
1  b0aaf1ac138b53cb6e039ba2c3d6604a250d02d5145c10…   2018.0   449999.0
2  4860c670bcd48fb96c02a4b0ae3608ae6fdd98176112e9…   2015.0  2000000.0
3  effdede7a2e7c2af664c8a31d9346385016128d66bbc58…   2017.0   700000.0
4  6ff54e709262f55cb999a1c1db8436cb2055d8f79ab520…   2017.0  1400000.0

         job_position  ctc_updated_year  years_of_Experience  \
0             Unknown            2020.0                  4.0
1  FullStack Engineer            2019.0                  1.0
2    Backend Engineer            2020.0                  5.0
3    Backend Engineer            2019.0                  2.0
4  FullStack Engineer            2019.0                  2.0
```

```
   job_position_encoded  company_hash_encoded  email_hash_encoded
0                 10352                     7               46299
1                 13751                   248               74714
2                 24667                     1               30424
3                 24667                    39              101563
4                 13751                     5               47172
```

[50]:
```
df = df_imputed[["email_hash_encoded","company_hash_encoded","orgyear","ctc","job_position_encod
df.head()
```

[50]:
```
   email_hash_encoded  company_hash_encoded  orgyear        ctc  \
0               46299                     7   2016.0  1100000.0
1               74714                   248   2018.0   449999.0
2               30424                     1   2015.0  2000000.0
3              101563                    39   2017.0   700000.0
4               47172                     5   2017.0  1400000.0

   job_position_encoded  ctc_updated_year  years_of_Experience
0                 10352            2020.0                  4.0
1                 13751            2019.0                  1.0
2                 24667            2020.0                  5.0
3                 24667            2019.0                  2.0
4                 13751            2019.0                  2.0
```

[51]:
```
df.shape
```

[51]:
```
(108277, 7)
```

[52]:
```
from sklearn.preprocessing import StandardScaler
Scaler = StandardScaler()
scaled_df = pd.DataFrame(Scaler.fit_transform(df), columns=df.columns)
scaled_df
```

[52]:
```
        email_hash_encoded  company_hash_encoded   orgyear       ctc  \
0                -0.250793             -0.486912 -0.236503 -0.038381
1                 0.658287             -0.258807  0.706469 -0.577364
2                -0.758681             -0.492590 -0.707988  0.707902
3                 1.517266             -0.456624  0.234983 -0.370062
4                -0.222863             -0.488805  0.234983  0.210380
…                      …                     …         …         …
108272            0.763768             -0.465142 -1.650959  0.940079
108273           -0.015293             -0.419711  0.234983 -0.535903
108274            1.015776             -0.403620  2.120925 -0.370062
108275            1.668785             -0.011773  1.177954  3.278432
108276           -1.566791             -0.260700 -1.179474  0.077707
```

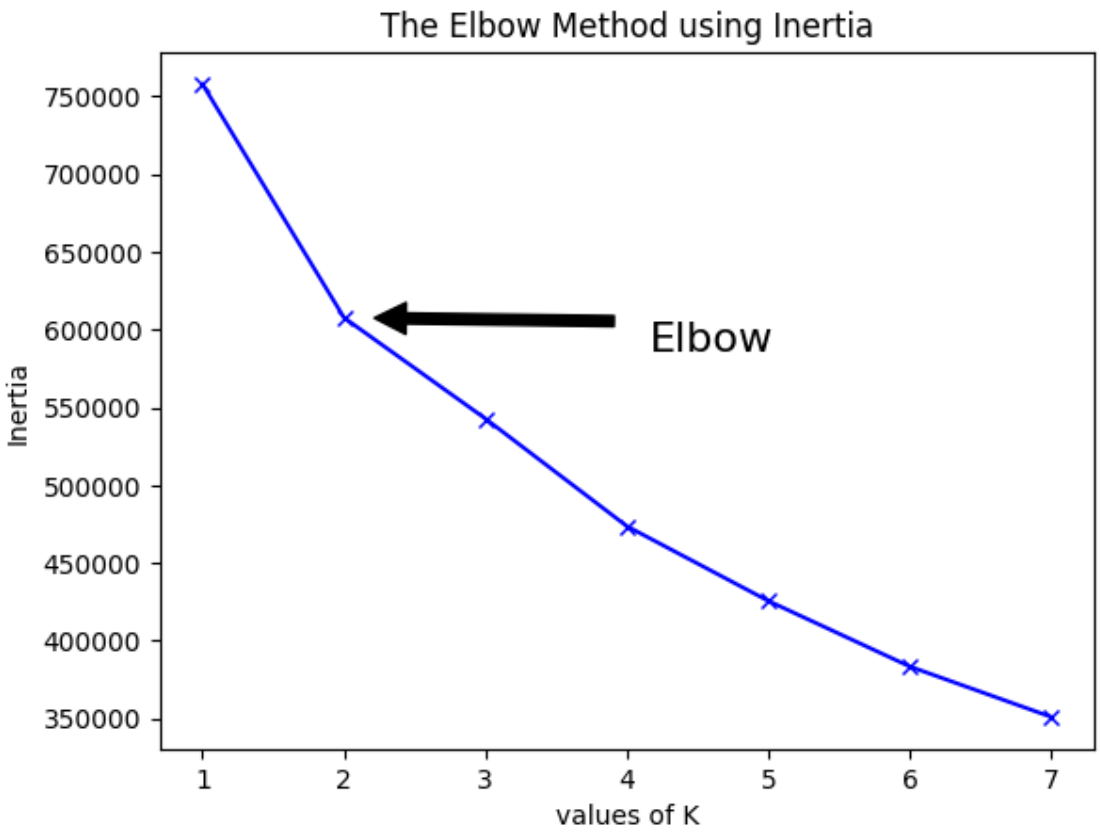
```
        job_position_encoded  ctc_updated_year  years_of_Experience
0                   0.102762          0.248502             0.399753
1                   0.463594         -0.574419            -1.092628
2                   1.622417          0.248502             0.897213
3                   1.622417         -0.574419            -0.595168
4                   0.463594         -0.574419            -0.595168
…                        …                 …                    …
108272             -0.987164         -0.574419             1.394673
108273             -0.976549          0.248502            -0.097708
108274             -0.991092          1.071424            -1.590088
108275             -0.978778         -0.574419            -1.590088
108276             -0.994171         -3.043184            -0.595168

[108277 rows x 7 columns]
```

[53]:
```
from sklearn.cluster import KMeans
inertia=[]
range_val = range(1,8)
for i in range_val:
  kmeans = KMeans(n_clusters=i)
  kmeans.fit_predict(pd.DataFrame(scaled_df))
  inertia.append(kmeans.inertia_)
```

[54]:
```
inertia
```

[54]:
```
[757938.9999999863,
 607816.275661611,
 542846.4838367489,
 473684.0362729593,
 425631.4482794325,
 383447.0990105066,
 351023.71455755475]
```

[55]:
```
plt.plot(range_val, inertia, "bx-")
plt.xlabel("values of K")
plt.ylabel("Inertia")
plt.title("The Elbow Method using Inertia")
plt.annotate('Elbow',
             xy=(2, inertia[1]),
             xytext=(0.55, 0.55),
             textcoords='figure fraction',
             fontsize=16,
             arrowprops=dict(facecolor='black', shrink=0.1)
            )
plt.show()
```



[56]:
```
kmeans_model = KMeans(n_clusters=2, init='k-means++')
kmeans_model.fit_predict(scaled_df)
labels = kmeans_model.labels_
np.unique(labels, return_counts=True)
```

[56]:
```
(array([0, 1], dtype=int32), array([54176, 54101]))
```

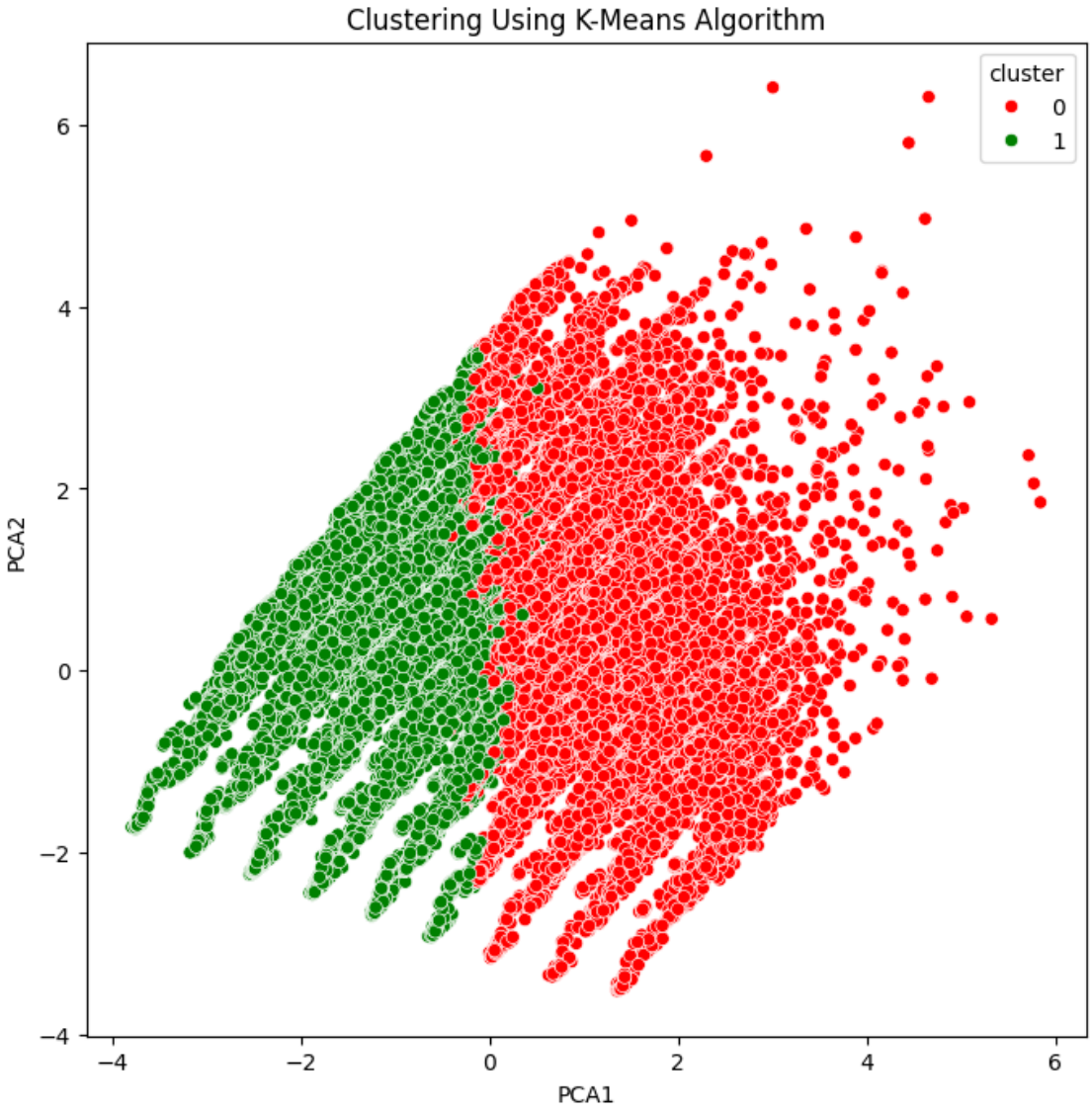
[57]:
```
from sklearn.decomposition import PCA
pca = PCA(n_components=2)
principal_components = pca.fit_transform(scaled_df)
pca_df = pd.DataFrame(data=principal_components, columns=["PCA1","PCA2"])
pca_df_kmeans = pd.concat([pca_df, pd.DataFrame({"cluster":labels})], axis=1)
pca_df_kmeans
```

[57]:

	PCA1	PCA2	cluster
0	0.476456	-0.174172	0
1	-1.077860	1.039799	1
2	1.453107	0.287751	0
3	-0.229559	1.296918	1
4	-0.208719	1.059587	1

```
...          ...       ...       ...
108272  2.306362 -0.271647         0
108273 -0.405669 -0.445034         1
108274 -2.736949 -0.323790         1
108275 -1.025663  1.453924         1
108276  0.972606  2.235457         0

[108277 rows x 3 columns]
```

[58]:
```
plt.figure(figsize=(8,8))
ax = sns.scatterplot(x="PCA1", y="PCA2", hue="cluster", data=pca_df_kmeans,␣
↪palette=["red","green"])
plt.title("Clustering Using K-Means Algorithm")
plt.show()
```



[59]:
```
from sklearn.metrics import silhouette_score
sil_coef_pca = silhouette_score(pca_df_kmeans.drop("cluster",axis=1),␣
 ↪pca_df_kmeans["cluster"])
sil_coef_pca
```

[59]:
```
0.3888895367191922
```

[60]:
```
sampled_df = scaled_df.sample(n=10000, random_state=42)
sampled_df
```

[60]:

	email_hash_encoded	company_hash_encoded	orgyear	ctc	\
51451	-0.019132	-0.411192	1.649440	0.044539	
33862	-0.948943	-0.288148	0.234983	-0.676868	
68411	0.870081	-0.056258	-0.236503	1.371264	
69378	-0.767351	-0.404567	0.234983	0.326468	
100363	0.619416	-0.485965	1.649440	0.127460	
…	…	…	…	…	
17037	-0.027898	-0.488805	1.177954	-0.618823	
61412	-0.773654	-0.333580	-0.236503	-0.079841	
61091	0.304573	0.167114	0.706469	0.417681	
77104	1.634136	-0.467982	-0.707988	8.560456	
15442	0.720833	-0.492590	0.234983	-0.320310	

	job_position_encoded	ctc_updated_year	years_of_Experience
51451	-0.966570	0.248502	-1.590088
33862	-0.991411	-0.574419	-0.595168
68411	0.102762	1.071424	0.897213
69378	-0.992154	-0.574419	-0.595168
100363	0.463594	1.071424	-1.092628
…	…	…	…
17037	0.463594	0.248502	-1.092628
61412	1.622417	-0.574419	-0.097708
61091	1.622417	0.248502	-0.595168
77104	0.463594	0.248502	0.897213
15442	-0.372190	0.248502	-0.097708

[10000 rows x 7 columns]

[67]:
```
from sklearn.cluster import AgglomerativeClustering
cluster = AgglomerativeClustering(n_clusters=2, metric="euclidean",␣
 ↪linkage="ward")
labels = cluster.fit_predict(sampled_df)
```

[68]:
```
np.unique(labels, return_counts=True)
```

[68]:
```
(array([0, 1]), array([4875, 5125]))
```

[69]:
```
pca = PCA(n_components=2)
principal_components = pca.fit_transform(sampled_df)
sampled_pca_df = pd.DataFrame(data=principal_components,␣
 ↪columns=["PCA1","PCA2"])
sampled_pca_df
```

[69]:

	PCA1	PCA2
0	-2.171669	0.331996
1	-0.606802	0.278263
2	0.842742	-0.872877
3	-0.370264	0.502962
4	-1.818485	0.108276
…	…	…
9995	-1.497520	0.555862
9996	0.460574	1.180000
9997	-0.658614	0.730243
9998	2.963731	1.315114
9999	-0.264909	-0.188341

[10000 rows x 2 columns]

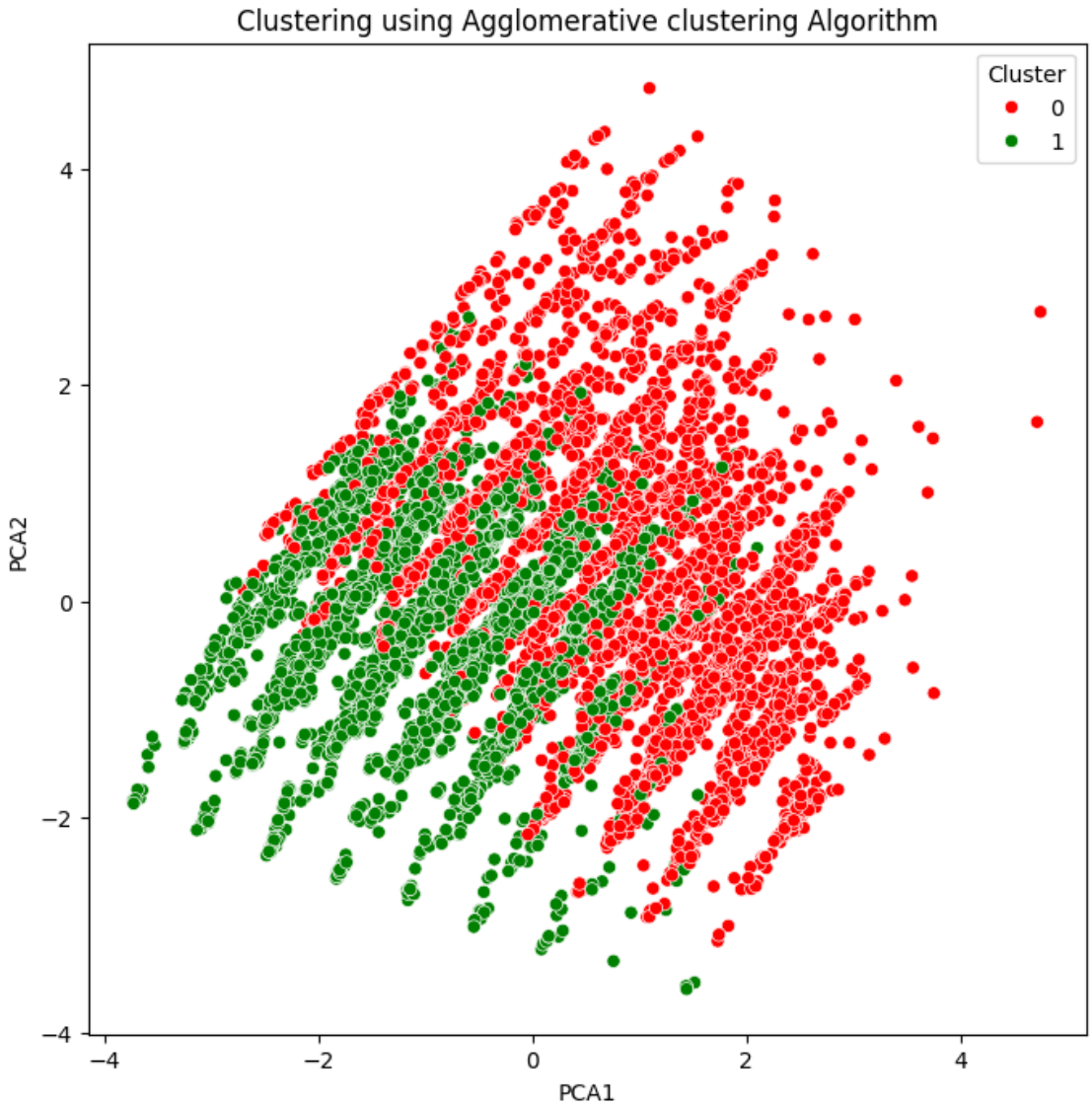
[70]:
```
sampled_pca_df_agg = pd.concat([sampled_pca_df,pd.DataFrame({"Cluster":cluster.
 ↪labels_})], axis=1)
sampled_pca_df_agg
```

[70]:

	PCA1	PCA2	Cluster
0	-2.171669	0.331996	1
1	-0.606802	0.278263	1
2	0.842742	-0.872877	0
3	-0.370264	0.502962	1
4	-1.818485	0.108276	1
…	…	…	…
9995	-1.497520	0.555862	0
9996	0.460574	1.180000	0
9997	-0.658614	0.730243	0
9998	2.963731	1.315114	0
9999	-0.264909	-0.188341	1

[10000 rows x 3 columns]

[71]:
```
plt.figure(figsize=(8,8))
ax = sns.scatterplot(x="PCA1", y="PCA2", hue="Cluster",␣
 ↪data=sampled_pca_df_agg, palette=["red", "green"])
plt.title("Clustering using Agglomerative clustering Algorithm")
plt.show()
```



[72]:
```
sil_coef_agg = silhouette_score(sampled_pca_df_agg.drop("Cluster",axis=1),␣
↪sampled_pca_df_agg["Cluster"])
sil_coef_agg
```

[72]: 0.20199218231397695

3 Insights

1. Top Paying job titles include Engineering Leadership, Backend Engineer, Product Manager, Program Manager, SDET, QA Engineer, Data Scientist, Android Engineer and FullStack Engineer.

2. Top paying companies include Cisco, Intel Technology India Pvt Ltd, Amazon, Walmart Labs, Symantec, Schneider Electric India, Morgan Stanley, Ericsson RD Bangalore and Samsung Electronics.
3. Among top paying companies, salary for these is getting lesser in recent years, Goldmaan Sachs, Tata Consultancy Services, Samsung Electronics, VMware, Dell, Dbs Bank, Hsbc software devlopement India and GE.
4. Among Top paying companies mean salary for these company is increasing every year, Amazon, Microsoft and Huawei Technologies 49.
5. Avg CTC seems to be decreasing with year.

4 Recommendations

1. Freshers who want to work on technical side should look for roles related to Backend Engineer, SDET, QA engineer, Dataa Scientist, Android Engineer,Full stack engineer to get good salaries as expirience increases.
2. Freshers who want best CTC should aim for companies like Cisco, Intel Technology India Pvt Ltd, Amazon, Walmart Labs, Symantec, Schneider Electric India, Morgan Stanley, Ericsson RD Bangalore and Samsung Electronics.

[ ]: