Data Science Salaries Prediction Model

June 30, 2023

1 Data Analysis Salaries Analysis and Classification

The aim of this project is to perform a complete Data Analysis on the dataset concerning the salary of Data Engineers from various companies and countries, and deploy a ML model that can predict the size of the company that the engineer works for

The dataset is taken from Kaggle and you can download It from here Click here to download dataset

Data Science Job Salaries Dataset contains 11 columns, each are:

```
work_year: The year the salary was paid.
experience_level: The experience level in the job during the year
employment_type: The type of employment for the role
job_title: The role worked in during the year.
salary: The total gross salary amount paid.
salary_currency: The currency of the salary paid as an ISO 4217 currency code.
salaryinusd: The salary in USD
employee_residence: Employee's primary country of residence in during the work year as an ISO 3
remote_ratio: The overall amount of work done remotely
company_location: The country of the employer's main office or contracting branch
company_size: The median number of people that worked for the company during the year
```

1.1 Data Import and Understanding

Importing the libraries used for this project

[4]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import warnings

warnings.filterwarnings('ignore')
```

Importing the dataset into our project

[5]:
```
df = pd.read_csv('ds_salaries.csv')

df.head()
```

[5]:
```
   work_year experience_level employment_type                 job_title  \
0       2023               SE              FT  Principal Data Scientist
1       2023               MI              CT             ML Engineer
2       2023               MI              CT             ML Engineer
3       2023               SE              FT           Data Scientist
4       2023               SE              FT           Data Scientist

   salary salary_currency  salary_in_usd employee_residence  remote_ratio  \
0   80000             EUR          85847                 ES           100
1   30000             USD          30000                 US           100
2   25500             USD          25500                 US           100
3  175000             USD         175000                 CA           100
4  120000             USD         120000                 CA           100

  company_location company_size
0               ES            L
1               US            S
2               US            S
3               CA            M
4               CA            M
```

1.2 Exploratory Data Analysis

We're going to extract some insights from the data like the average, max, min and plot the salary in terms of other features

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

[6]:
```
         work_year        salary  salary_in_usd  remote_ratio
count  3755.000000  3.755000e+03    3755.000000   3755.000000
mean   2022.373635  1.906956e+05  137570.389880     46.271638
std       0.691448  6.716765e+05   63055.625278     48.589050
min    2020.000000  6.000000e+03    5132.000000      0.000000
25%    2022.000000  1.000000e+05   95000.000000      0.000000
50%    2022.000000  1.380000e+05  135000.000000      0.000000
75%    2023.000000  1.800000e+05  175000.000000    100.000000
max    2023.000000  3.040000e+07  450000.000000    100.000000
```

Cleaning the dataset by dropping the rows that has NaN

[7]:
```
df.dropna(inplace=True)
df
```

[7]:
```
      work_year experience_level employment_type                 job_title  \
0          2023               SE              FT  Principal Data Scientist
1          2023               MI              CT             ML Engineer
2          2023               MI              CT             ML Engineer
3          2023               SE              FT           Data Scientist
```

```
4          2023               SE              FT            Data Scientist
…           …                 …               …                  …
3750       2020               SE              FT            Data Scientist
3751       2021               MI              FT  Principal Data Scientist
3752       2020               EN              FT            Data Scientist
3753       2020               EN              CT     Business Data Analyst
3754       2021               SE              FT      Data Science Manager

       salary salary_currency  salary_in_usd employee_residence  remote_ratio  \
0       80000             EUR          85847                 ES           100
1       30000             USD          30000                 US           100
2       25500             USD          25500                 US           100
3      175000             USD         175000                 CA           100
4      120000             USD         120000                 CA           100
…       …               …             …                 …            …
3750   412000             USD         412000                 US           100
3751   151000             USD         151000                 US           100
3752   105000             USD         105000                 US           100
3753   100000             USD         100000                 US           100
3754  7000000             INR          94665                 IN            50

     company_location company_size
0                  ES            L
1                  US            S
2                  US            S
3                  CA            M
4                  CA            M
…                 …             …
3750               US            L
3751               US            L
3752               US            S
3753               US            L
3754               IN            L

[3755 rows x 11 columns]
```

1.2.1 Salary vs Year

Grouping the data showing the mean of the salary with Job Title in Years

[8]:
```
salary_jobtitle_pivot_table = pd.
↪pivot_table(df,values='salary_in_usd',index='work_year',columns='job_title',aggfunc='mean')
salary_jobtitle_pivot_table
```

[8]:
```
job_title  3D Computer Vision Researcher  AI Developer  AI Programmer  \
work_year
2020                                 NaN           NaN            NaN
```

```
2021                            12704.5           NaN            NaN
2022                            30000.0    193768.000        40000.0
2023                                NaN    115252.875        70000.0

job_title  AI Scientist  Analytics Engineer  Applied Data Scientist  \
work_year
2020            45896.0                 NaN                     NaN
2021            25410.6                 NaN            82137.500000
2022           140815.0       137969.807018           160800.000000
2023           231232.5       170210.652174            56329.333333

job_title  Applied Machine Learning Engineer  \
work_year
2020                                     NaN
2021                                     NaN
2022                                     NaN
2023                                99875.5

job_title  Applied Machine Learning Scientist  Applied Scientist  \
work_year
2020                                    NaN               NaN
2021                             230700.000               NaN
2022                              89888.875      188311.111111
2023                              66461.500      191143.500000

job_title  Autonomous Vehicle Technician  …  Principal Data Engineer  \
work_year                                 …
2020                                 NaN  …                     NaN
2021                             45555.0  …                192500.0
2022                                 NaN  …                     NaN
2023                              7000.0  …                     NaN

job_title  Principal Data Scientist  Principal Machine Learning Engineer  \
work_year
2020                       148261.0                                   NaN
2021                       239152.4                                   NaN
2022                       155499.0                              190000.0
2023                        85847.0                                   NaN

job_title  Product Data Analyst  Product Data Scientist  Research Engineer  \
work_year
2020                    13036.0                     NaN                NaN
2021                        NaN                     NaN                NaN
2022                   120000.0                  8000.0         207870.000
2023                    16414.0                     NaN         156114.375

job_title  Research Scientist  Software Data Engineer  Staff Data Analyst  \
```

```
work_year
2020            246000.000000                    NaN             15000.0
2021             83003.600000                    NaN                 NaN
2022            142188.733333                    NaN                 NaN
2023            177539.945455                62510.0                 NaN

job_title  Staff Data Scientist
work_year
2020                        NaN
2021                   105000.0
2022                        NaN
2023                        NaN

[4 rows x 93 columns]
```

Cleaning the pivot table from the NaN

[9]:
```
salary_jobtitle_pivot_table.replace(np.nan,0,inplace=True)
salary_jobtitle_pivot_table
```

[9]:
```
job_title  3D Computer Vision Researcher  AI Developer  AI Programmer  \
work_year
2020                                 0.0         0.000            0.0
2021                             12704.5         0.000            0.0
2022                             30000.0    193768.000        40000.0
2023                                 0.0    115252.875        70000.0

job_title  AI Scientist  Analytics Engineer  Applied Data Scientist  \
work_year
2020            45896.0            0.000000                0.000000
2021            25410.6            0.000000            82137.500000
2022           140815.0       137969.807018           160800.000000
2023           231232.5       170210.652174            56329.333333

job_title  Applied Machine Learning Engineer  \
work_year
2020                                     0.0
2021                                     0.0
2022                                     0.0
2023                                 99875.5

job_title  Applied Machine Learning Scientist  Applied Scientist  \
work_year
2020                                    0.000           0.000000
2021                               230700.000           0.000000
2022                                89888.875      188311.111111
2023                                66461.500      191143.500000
```

```
job_title  Autonomous Vehicle Technician  …  Principal Data Engineer  \
work_year                                 …
2020                                 0.0  …                      0.0
2021                             45555.0  …                 192500.0
2022                                 0.0  …                      0.0
2023                              7000.0  …                      0.0

job_title  Principal Data Scientist  Principal Machine Learning Engineer  \
work_year
2020                      148261.0                                  0.0
2021                      239152.4                                  0.0
2022                      155499.0                             190000.0
2023                       85847.0                                  0.0

job_title  Product Data Analyst  Product Data Scientist  Research Engineer  \
work_year
2020                    13036.0                     0.0              0.000
2021                        0.0                     0.0              0.000
2022                   120000.0                  8000.0         207870.000
2023                    16414.0                     0.0         156114.375

job_title  Research Scientist  Software Data Engineer  Staff Data Analyst  \
work_year
2020             246000.000000                     0.0             15000.0
2021              83003.600000                     0.0                 0.0
2022             142188.733333                     0.0                 0.0
2023             177539.945455                 62510.0                 0.0

job_title  Staff Data Scientist
work_year
2020                        0.0
2021                   105000.0
2022                        0.0
2023                        0.0

[4 rows x 93 columns]
```

[10]:
```python
import matplotlib.pyplot as plt

# Set the figure size
plt.figure(figsize=(10, 10))

# Plot the pivot table as a bar chart
salary_jobtitle_pivot_table.plot(kind='bar', stacked=True)

# Set the labels and title
plt.xlabel('Year')
```

```
plt.ylabel('Average Salary (USD)')
plt.title('Average Salary by Year and Job Title')
plt.legend(loc='upper right', bbox_to_anchor=(2, 1))

# Show the plot
plt.show()
```

```
<Figure size 1000x1000 with 0 Axes>
```


Average Salary by Year and Job Title
1e6
Average Salary (USD)
0
1
2
3
4
5
6
7
8
2020
2021
2022
2023
Year
3D Computer Vision Researcher
AI Developer
AI Programmer
AI Scientist
Analytics Engineer
Applied Data Scientist
Applied Machine Learning Engineer
Applied Machine Learning Scientist
Applied Scientist
Autonomous Vehicle Technician
Azure Data Engineer
BI Analyst
BI Data Analyst
BI Data Engineer
BI Developer
Big Data Architect
Big Data Engineer
Business Data Analyst
Business Intelligence Engineer
Cloud Data Architect
Cloud Data Engineer
Cloud Database Engineer
Compliance Data Analyst
Computer Vision Engineer
Computer Vision Software Engineer
Data Analyst
Data Analytics Consultant
Data Analytics Engineer
Data Analytics Lead
Data Analytics Manager
Data Analytics Specialist
Data Architect
Data DevOps Engineer
Data Engineer
Data Infrastructure Engineer
Data Lead
Data Management Specialist
Data Manager
Data Modeler
Data Operations Analyst
Data Operations Engineer
Data Quality Analyst
Data Science Consultant
Data Science Engineer
Data Science Lead
Data Science Manager
Data Science Tech Lead
Data Scientist
Data Scientist Lead
Data Specialist
Data Strategist
Deep Learning Engineer
Deep Learning Researcher
Director of Data Science
ETL Developer
ETL Engineer
Finance Data Analyst
Financial Data Analyst
Head of Data
Head of Data Science
Head of Machine Learning
Insight Analyst
Lead Data Analyst
Lead Data Engineer
Lead Data Scientist
Lead Machine Learning Engineer
ML Engineer
MLOps Engineer
Machine Learning Developer
Machine Learning Engineer
Machine Learning Infrastructure Engineer
Machine Learning Manager
Machine Learning Research Engineer
Machine Learning Researcher
Machine Learning Scientist
Machine Learning Software Engineer
Manager Data Management
Marketing Data Analyst
Marketing Data Engineer
NLP Engineer
Power BI Developer
Principal Data Analyst
Principal Data Architect
Principal Data Engineer
Principal Data Scientist
Principal Machine Learning Engineer
Product Data Analyst
Product Data Scientist
Research Engineer
Research Scientist
Software Data Engineer
Staff Data Analyst
Staff Data Scientist

1.2.2 Salary vs Experience level

Creating the pivot table

[11]:
```
salary_experience_pivot_table = pd.
↪pivot_table(df,values='salary_in_usd',index='experience_level',columns='job_title',aggfunc=
salary_experience_pivot_table.replace(np.nan,0,inplace=True)
salary_experience_pivot_table
```

[11]:
```
job_title         3D Computer Vision Researcher   AI Developer  AI Programmer  \
experience_level
EN                                      35000.0  130884.500000        55000.0
EX                                          0.0       0.000000            0.0
MI                                       5409.0  137510.000000            0.0
SE                                      10000.0  147666.666667            0.0

job_title          AI Scientist  Analytics Engineer  Applied Data Scientist  \
experience_level
EN                  52781.285714       130000.000000            66679.000000
EX                 200000.000000       175125.000000                0.000000
MI                 117726.200000       102480.230769            77977.000000
SE                 201278.000000       158404.024691           208439.333333

job_title         Applied Machine Learning Engineer  \
experience_level
EN                                              0.0
EX                                              0.0
MI                                          99875.5
SE                                              0.0

job_title         Applied Machine Learning Scientist  Applied Scientist  \
experience_level
EN                                         36696.00      167356.666667
EX                                             0.00           0.000000
MI                                        135820.50           0.000000
SE                                        106279.75      192907.692308

job_title         Autonomous Vehicle Technician  …  Principal Data Engineer  \
experience_level                                 …
EN                                       7000.0  …                      0.0
EX                                          0.0  …                      0.0
MI                                      45555.0  …                      0.0
SE                                          0.0  …                 192500.0

job_title         Principal Data Scientist  \
```

```
experience_level
EN                              0.000000
EX                         416000.000000
MI                         151000.000000
SE                         169728.166667

job_title         Principal Machine Learning Engineer  Product Data Analyst  \
experience_level
EN                                                0.0              100000.0
EX                                                0.0                   0.0
MI                                                0.0               45621.5
SE                                           190000.0                   0.0

job_title         Product Data Scientist  Research Engineer  \
experience_level
EN                                   0.0      130000.000000
EX                                   0.0           0.000000
MI                                   0.0      178000.000000
SE                                8000.0      174773.181818

job_title         Research Scientist  Software Data Engineer  \
experience_level
EN                    118280.888889                     0.0
EX                     84053.000000                     0.0
MI                    141575.086957                 75020.0
SE                    179892.979592                 50000.0

job_title         Staff Data Analyst  Staff Data Scientist
experience_level
EN                               0.0                   0.0
EX                           15000.0                   0.0
MI                               0.0                   0.0
SE                               0.0              105000.0

[4 rows x 93 columns]
```

[12]:
```python
import matplotlib.pyplot as plt

# Set the figure size
plt.figure(figsize=(10, 10))

# Plot the pivot table as a bar chart
salary_experience_pivot_table.plot(kind='bar', stacked=True)

# Set the labels and title
plt.xlabel('Experience Level')
plt.ylabel('Average Salary (USD)')
```

```
plt.title('Average Salary by Experience Level and Job Title')
plt.legend(loc='upper right', bbox_to_anchor=(2, 1))

# Show the plot
plt.show()
```

```
<Figure size 1000x1000 with 0 Axes>
```


Average Salary by Experience Level and Job Title
1e7
1.0
0.8
0.6
0.4
0.2
0.0
Average Salary (USD)
EN
EX
MI
SE
Experience Level
3D Computer Vision Researcher
AI Developer
AI Programmer
AI Scientist
Analytics Engineer
Applied Data Scientist
Applied Machine Learning Engineer
Applied Machine Learning Scientist
Applied Scientist
Autonomous Vehicle Technician
Azure Data Engineer
BI Analyst
BI Data Analyst
BI Data Engineer
BI Developer
Big Data Architect
Big Data Engineer
Business Data Analyst
Business Intelligence Engineer
Cloud Data Architect
Cloud Data Engineer
Cloud Database Engineer
Compliance Data Analyst
Computer Vision Engineer
Computer Vision Software Engineer
Data Analyst
Data Analytics Consultant
Data Analytics Engineer
Data Analytics Lead
Data Analytics Manager
Data Analytics Specialist
Data Architect
Data DevOps Engineer
Data Engineer
Data Infrastructure Engineer
Data Lead
Data Management Specialist
Data Manager
Data Modeler
Data Operations Analyst
Data Operations Engineer
Data Quality Analyst
Data Science Consultant
Data Science Engineer
Data Science Lead
Data Science Manager
Data Science Tech Lead
Data Scientist
Data Scientist Lead
Data Specialist
Data Strategist
Deep Learning Engineer
Deep Learning Researcher
Director of Data Science
ETL Developer
ETL Engineer
Finance Data Analyst
Financial Data Analyst
Head of Data
Head of Data Science
Head of Machine Learning
Insight Analyst
Lead Data Analyst
Lead Data Engineer
Lead Data Scientist
Lead Machine Learning Engineer
ML Engineer
MLOps Engineer
Machine Learning Developer
Machine Learning Engineer
Machine Learning Infrastructure Engineer
Machine Learning Manager
Machine Learning Research Engineer
Machine Learning Researcher
Machine Learning Scientist
Machine Learning Software Engineer
Manager Data Management
Marketing Data Analyst
Marketing Data Engineer
NLP Engineer
Power BI Developer
Principal Data Analyst
Principal Data Architect
Principal Data Engineer
Principal Data Scientist
Principal Machine Learning Engineer
Product Data Analyst
Product Data Scientist
Research Engineer
Research Scientist
Software Data Engineer
Staff Data Analyst
Staff Data Scientist

1.2.3 Salary vs Employment type

[13]:
```
salary_employment_type_pivot_table = pd.
 ↪pivot_table(df,values='salary_in_usd',index='employment_type',columns='job_title',aggfunc='
salary_employment_type_pivot_table.replace(np.nan,0,inplace=True)
salary_employment_type_pivot_table
```

[13]:
```
job_title        3D Computer Vision Researcher   AI Developer  AI Programmer  \
employment_type
CT                                    0.000000       0.000000            0.0
FL                                    0.000000       0.000000            0.0
FT                                26666.666667  136666.090909        55000.0
PT                                 5409.000000       0.000000            0.0

job_title         AI Scientist  Analytics Engineer  Applied Data Scientist  \
employment_type
CT                    0.000000         7500.000000                     0.0
FL                    0.000000            0.000000                     0.0
FT                124138.142857       153788.911765                113726.3
PT                 12000.000000            0.000000                     0.0

job_title        Applied Machine Learning Engineer  \
employment_type
CT                                             0.0
FL                                             0.0
FT                                         99875.5
PT                                             0.0

job_title        Applied Machine Learning Scientist  Applied Scientist  \
employment_type
CT                                          30469.0           0.000000
FL                                          30523.0           0.000000
FT                                         125244.2      190264.482759
PT                                             0.0           0.000000

job_title        Autonomous Vehicle Technician  …  Principal Data Engineer  \
employment_type                                 …
CT                                         0.0  …                     0.0
FL                                     45555.0  …                     0.0
FT                                      7000.0  …                192500.0
PT                                         0.0  …                     0.0

job_title        Principal Data Scientist  \
employment_type
```

```
CT                          416000.000000
FL                               0.000000
FT                          167052.714286
PT                               0.000000

job_title        Principal Machine Learning Engineer  Product Data Analyst  \
employment_type
CT                                               0.0                   0.0
FL                                               0.0                   0.0
FT                                          190000.0               56497.2
PT                                               0.0                   0.0

job_title        Product Data Scientist  Research Engineer  \
employment_type
CT                                  0.0           0.000000
FL                                  0.0           0.000000
FT                               8000.0      163108.378378
PT                                  0.0           0.000000

job_title        Research Scientist  Software Data Engineer  \
employment_type
CT                         0.000000                     0.0
FL                         0.000000                 50000.0
FT                    161214.195122                 75020.0
PT                         0.000000                     0.0

job_title        Staff Data Analyst  Staff Data Scientist
employment_type
CT                              0.0              105000.0
FL                              0.0                   0.0
FT                          15000.0                   0.0
PT                              0.0                   0.0

[4 rows x 93 columns]
```

[14]:
```python
import matplotlib.pyplot as plt

# Set the figure size
plt.figure(figsize=(10, 10))

# Plot the pivot table as a bar chart
salary_employment_type_pivot_table.plot(kind='bar', stacked=True)

# Set the labels and title
plt.xlabel('Emplyment type')
plt.ylabel('Average Salary (USD)')
plt.title('Average Salary by Employment Type and Job Title')
```

```
plt.legend(loc='upper right', bbox_to_anchor=(2, 1))

# Show the plot
plt.show()
```

```
<Figure size 1000x1000 with 0 Axes>
```


Average Salary by Employment Type and Job Title
1e7
1.0
0.8
0.6
0.4
0.2
0.0
Average Salary (USD)
CT
FL
FT
PT
Emplyment type
3D Computer Vision Researcher
AI Developer
AI Programmer
AI Scientist
Analytics Engineer
Applied Data Scientist
Applied Machine Learning Engineer
Applied Machine Learning Scientist
Applied Scientist
Autonomous Vehicle Technician
Azure Data Engineer
BI Analyst
BI Data Analyst
BI Data Engineer
BI Developer
Big Data Architect
Big Data Engineer
Business Data Analyst
Business Intelligence Engineer
Cloud Data Architect
Cloud Data Engineer
Cloud Database Engineer
Compliance Data Analyst
Computer Vision Engineer
Computer Vision Software Engineer
Data Analyst
Data Analytics Consultant
Data Analytics Engineer
Data Analytics Lead
Data Analytics Manager
Data Analytics Specialist
Data Architect
Data DevOps Engineer
Data Engineer
Data Infrastructure Engineer
Data Lead
Data Management Specialist
Data Manager
Data Modeler
Data Operations Analyst
Data Operations Engineer
Data Quality Analyst
Data Science Consultant
Data Science Engineer
Data Science Lead
Data Science Manager
Data Science Tech Lead
Data Scientist
Data Scientist Lead
Data Specialist
Data Strategist
Deep Learning Engineer
Deep Learning Researcher
Director of Data Science
ETL Developer
ETL Engineer
Finance Data Analyst
Financial Data Analyst
Head of Data
Head of Data Science
Head of Machine Learning
Insight Analyst
Lead Data Analyst
Lead Data Engineer
Lead Data Scientist
Lead Machine Learning Engineer
ML Engineer
MLOps Engineer
Machine Learning Developer
Machine Learning Engineer
Machine Learning Infrastructure Engineer
Machine Learning Manager
Machine Learning Research Engineer
Machine Learning Researcher
Machine Learning Scientist
Machine Learning Software Engineer
Manager Data Management
Marketing Data Analyst
Marketing Data Engineer
NLP Engineer
Power BI Developer
Principal Data Analyst
Principal Data Architect
Principal Data Engineer
Principal Data Scientist
Principal Machine Learning Engineer
Product Data Analyst
Product Data Scientist
Research Engineer
Research Scientist
Software Data Engineer
Staff Data Analyst
Staff Data Scientist

1.2.4 Salary vs Employee residence

[15]:
```
salary_employee_residence_pivot_table = pd.
↪pivot_table(df,values='salary_in_usd',index='employee_residence',columns='job_title')
salary_employee_residence_pivot_table.replace(np.nan,0,inplace=True)
salary_employee_residence_pivot_table
```

[15]:
```
job_title           3D Computer Vision Researcher  AI Developer  \
employee_residence
AE                                            0.0           0.0
AM                                            0.0           0.0
AR                                            0.0           0.0
AS                                        20000.0           0.0
AT                                            0.0           0.0
…                                               …             …
TR                                            0.0           0.0
UA                                            0.0       84000.0
US                                        50000.0      200000.0
UZ                                            0.0           0.0
VN                                            0.0           0.0

job_title           AI Programmer  AI Scientist  Analytics Engineer  \
employee_residence
AE                            0.0           0.0            0.000000
AM                            0.0           0.0            0.000000
AR                            0.0           0.0        48000.000000
AS                            0.0           0.0            0.000000
AT                            0.0           0.0            0.000000
…                               …             …                   …
TR                            0.0           0.0            0.000000
UA                            0.0           0.0            0.000000
US                            0.0      142500.0       160244.395604
UZ                            0.0           0.0            0.000000
VN                            0.0           0.0            0.000000

job_title           Applied Data Scientist  Applied Machine Learning Engineer  \
employee_residence
AE                                     0.0                                0.0
AM                                     0.0                                0.0
AR                                     0.0                                0.0
AS                                     0.0                                0.0
AT                                 50000.0                                0.0
…                                    …                                  …
TR                                     0.0                                0.0
```

```
UA                                    0.0                                  0.0
US                               238000.0                             130000.0
UZ                                    0.0                                  0.0
VN                                    0.0                                  0.0

job_title           Applied Machine Learning Scientist  Applied Scientist  \
employee_residence
AE                                                 0.0           0.000000
AM                                                 0.0           0.000000
AR                                                 0.0           0.000000
AS                                                 0.0           0.000000
AT                                                 0.0           0.000000
…                                                    …                  …
TR                                                 0.0           0.000000
UA                                                 0.0           0.000000
US                                            188800.0      190264.482759
UZ                                                 0.0           0.000000
VN                                             38400.0           0.000000

job_title           Autonomous Vehicle Technician  …  \
employee_residence                                 …
AE                                            0.0  …
AM                                            0.0  …
AR                                            0.0  …
AS                                        45555.0  …
AT                                            0.0  …
…                                               …  …
TR                                            0.0  …
UA                                            0.0  …
US                                            0.0  …
UZ                                            0.0  …
VN                                            0.0  …

job_title           Principal Data Engineer  Principal Data Scientist  \
employee_residence
AE                                      0.0                       0.0
AM                                      0.0                       0.0
AR                                      0.0                       0.0
AS                                      0.0                       0.0
AT                                      0.0                       0.0
…                                         …                         …
TR                                      0.0                       0.0
UA                                      0.0                       0.0
US                                 192500.0                  255500.0
UZ                                      0.0                       0.0
VN                                      0.0                       0.0
```

```
job_title           Principal Machine Learning Engineer  Product Data Analyst  \
employee_residence
AE                                                  0.0                   0.0
AM                                                  0.0                   0.0
AR                                                  0.0                   0.0
AS                                                  0.0                   0.0
AT                                                  0.0                   0.0
…                                                   …                     …
TR                                                  0.0                   0.0
UA                                                  0.0                   0.0
US                                             190000.0                120000.0
UZ                                                  0.0                   0.0
VN                                                  0.0                   0.0

job_title           Product Data Scientist  Research Engineer  \
employee_residence
AE                                     0.0           0.000000
AM                                     0.0           0.000000
AR                                     0.0           0.000000
AS                                     0.0           0.000000
AT                                     0.0           0.000000
…                                      …             …
TR                                     0.0           0.000000
UA                                     0.0           0.000000
US                                     0.0      168156.580645
UZ                                     0.0           0.000000
VN                                     0.0           0.000000

job_title           Research Scientist  Software Data Engineer  \
employee_residence
AE                            0.000000                     0.0
AM                            0.000000                     0.0
AR                            0.000000                     0.0
AS                            0.000000                     0.0
AT                        61989.000000                     0.0
…                                 …                       …
TR                            0.000000                     0.0
UA                            0.000000                     0.0
US                       181411.929825                     0.0
UZ                            0.000000                     0.0
VN                            0.000000                     0.0

job_title           Staff Data Analyst  Staff Data Scientist
employee_residence
AE                                 0.0                   0.0
AM                                 0.0                   0.0
AR                                 0.0                   0.0
```

```
AS                                  0.0                   0.0
AT                                  0.0                   0.0
…                                   …                     …
TR                                  0.0                   0.0
UA                                  0.0                   0.0
US                                  0.0              105000.0
UZ                                  0.0                   0.0
VN                                  0.0                   0.0

[78 rows x 93 columns]
```

[16]:
```python
import matplotlib.pyplot as plt

# Set the figure size
plt.figure(figsize=(10, 10))

# Plot the pivot table as a bar chart
salary_employee_residence_pivot_table.plot(kind='bar', stacked=True)

# Set the labels and title
plt.xlabel('Employee Residence')
plt.ylabel('Average Salary (USD)')
plt.title('Average Salary by Employee Residence and Job Title')
plt.legend(loc='upper right', bbox_to_anchor=(2, 1))

# Show the plot
plt.show()
```

```
<Figure size 1000x1000 with 0 Axes>
```


Average Salary by Employee Residence and Job Title
1e7
1.0
0.8
0.6
0.4
0.2
0.0
Average Salary (USD)
Employee Residence
3D Computer Vision Researcher
AI Developer
AI Programmer
AI Scientist
Analytics Engineer
Applied Data Scientist
Applied Machine Learning Engineer
Applied Machine Learning Scientist
Applied Scientist
Autonomous Vehicle Technician
Azure Data Engineer
BI Analyst
BI Data Analyst
BI Data Engineer
BI Developer
Big Data Architect
Big Data Engineer
Business Data Analyst
Business Intelligence Engineer
Cloud Data Architect
Cloud Data Engineer
Cloud Database Engineer
Compliance Data Analyst
Computer Vision Engineer
Computer Vision Software Engineer
Data Analyst
Data Analytics Consultant
Data Analytics Engineer
Data Analytics Lead
Data Analytics Manager
Data Analytics Specialist
Data Architect
Data DevOps Engineer
Data Engineer
Data Infrastructure Engineer
Data Lead
Data Management Specialist
Data Manager
Data Modeler
Data Operations Analyst
Data Operations Engineer
Data Quality Analyst
Data Science Consultant
Data Science Engineer
Data Science Lead
Data Science Manager
Data Science Tech Lead
Data Scientist
Data Scientist Lead
Data Specialist
Data Strategist
Deep Learning Engineer
Deep Learning Researcher
Director of Data Science
ETL Developer
ETL Engineer
Finance Data Analyst
Financial Data Analyst
Head of Data
Head of Data Science
Head of Machine Learning
Insight Analyst
Lead Data Analyst
Lead Data Engineer
Lead Data Scientist
Lead Machine Learning Engineer
ML Engineer
MLOps Engineer
Machine Learning Developer
Machine Learning Engineer
Machine Learning Infrastructure Engineer
Machine Learning Manager
Machine Learning Research Engineer
Machine Learning Researcher
Machine Learning Scientist
Machine Learning Software Engineer
Manager Data Management
Marketing Data Analyst
Marketing Data Engineer
NLP Engineer
Power BI Developer
Principal Data Analyst
Principal Data Architect
Principal Data Engineer
Principal Data Scientist
Principal Machine Learning Engineer
Product Data Analyst
Product Data Scientist
Research Engineer
Research Scientist
Software Data Engineer
Staff Data Analyst
Staff Data Scientist

1.2.5 Salary vs Remote ratio

[17]:
```
salary_remote_ratio_pivot_table = pd.
 ↪pivot_table(df,values='salary_in_usd',index='remote_ratio',columns='job_title')
salary_remote_ratio_pivot_table.replace(np.nan,0,inplace=True)
salary_remote_ratio_pivot_table
```

[17]:
```
job_title     3D Computer Vision Researcher   AI Developer  AI Programmer  \
remote_ratio
0                                 20000.0   98118.166667        70000.0
50                                 7704.5  166666.666667            0.0
100                               50000.0  207309.000000        40000.0

job_title      AI Scientist  Analytics Engineer  Applied Data Scientist  \
remote_ratio
0             191278.000000       160663.369565           110037.000000
50             94842.333333        68750.000000            52119.000000
100            90357.300000       148471.890909           131855.428571

job_title     Applied Machine Learning Engineer  \
remote_ratio
0                                    130000.0
50                                        0.0
100                                   69751.0

job_title     Applied Machine Learning Scientist  Applied Scientist  \
remote_ratio
0                                 108000.000000      191116.875000
50                                212974.333333           0.000000
100                                70813.875000      189215.384615

job_title     Autonomous Vehicle Technician  …  Principal Data Engineer  \
remote_ratio                                 …
0                                   7000.0  …                      0.0
50                                 45555.0  …                      0.0
100                                    0.0  …                 192500.0

job_title     Principal Data Scientist  Principal Machine Learning Engineer  \
remote_ratio
0                       220000.000000                                  0.0
50                           0.000000                                  0.0
100                     195052.714286                             190000.0

job_title     Product Data Analyst  Product Data Scientist  Research Engineer  \
```

```
remote_ratio
0                          20000.0                    0.0      173395.133333
50                             0.0                    0.0           0.000000
100                        65621.5                 8000.0      119022.285714

job_title     Research Scientist  Software Data Engineer  Staff Data Analyst  \
remote_ratio
0                 174970.777778                     0.0             15000.0
50                 97190.000000                 50000.0                 0.0
100               158944.235294                 75020.0                 0.0

job_title     Staff Data Scientist
remote_ratio
0                              0.0
50                             0.0
100                       105000.0

[3 rows x 93 columns]
```

[18]:
```python
import matplotlib.pyplot as plt

# Set the figure size
plt.figure(figsize=(10, 10))

# Plot the pivot table as a bar chart
salary_remote_ratio_pivot_table.plot(kind='bar', stacked=True)

# Set the labels and title
plt.xlabel('Remote Ratio')
plt.ylabel('Average Salary (USD)')
plt.title('Average Salary by Remote ratio and Job Title')
plt.legend(loc='upper right', bbox_to_anchor=(2, 1))

# Show the plot
plt.show()
```

```
<Figure size 1000x1000 with 0 Axes>
```


Average Salary by Remote ratio and Job Title
1e6
Average Salary (USD)
0
2
4
6
8
0
50
100
Remote Ratio
3D Computer Vision Researcher
AI Developer
AI Programmer
AI Scientist
Analytics Engineer
Applied Data Scientist
Applied Machine Learning Engineer
Applied Machine Learning Scientist
Applied Scientist
Autonomous Vehicle Technician
Azure Data Engineer
BI Analyst
BI Data Analyst
BI Data Engineer
BI Developer
Big Data Architect
Big Data Engineer
Business Data Analyst
Business Intelligence Engineer
Cloud Data Architect
Cloud Data Engineer
Cloud Database Engineer
Compliance Data Analyst
Computer Vision Engineer
Computer Vision Software Engineer
Data Analyst
Data Analytics Consultant
Data Analytics Engineer
Data Analytics Lead
Data Analytics Manager
Data Analytics Specialist
Data Architect
Data DevOps Engineer
Data Engineer
Data Infrastructure Engineer
Data Lead
Data Management Specialist
Data Manager
Data Modeler
Data Operations Analyst
Data Operations Engineer
Data Quality Analyst
Data Science Consultant
Data Science Engineer
Data Science Lead
Data Science Manager
Data Science Tech Lead
Data Scientist
Data Scientist Lead
Data Specialist
Data Strategist
Deep Learning Engineer
Deep Learning Researcher
Director of Data Science
ETL Developer
ETL Engineer
Finance Data Analyst
Financial Data Analyst
Head of Data
Head of Data Science
Head of Machine Learning
Insight Analyst
Lead Data Analyst
Lead Data Engineer
Lead Data Scientist
Lead Machine Learning Engineer
ML Engineer
MLOps Engineer
Machine Learning Developer
Machine Learning Engineer
Machine Learning Infrastructure Engineer
Machine Learning Manager
Machine Learning Research Engineer
Machine Learning Researcher
Machine Learning Scientist
Machine Learning Software Engineer
Manager Data Management
Marketing Data Analyst
Marketing Data Engineer
NLP Engineer
Power BI Developer
Principal Data Analyst
Principal Data Architect
Principal Data Engineer
Principal Data Scientist
Principal Machine Learning Engineer
Product Data Analyst
Product Data Scientist
Research Engineer
Research Scientist
Software Data Engineer
Staff Data Analyst
Staff Data Scientist

1.2.6 Salary vs Company location

[19]:
```
salary_company_location_pivot_table = pd.
 ↪pivot_table(df,index='company_location',values='salary_in_usd',columns='job_title')
salary_company_location_pivot_table.replace(np.nan,0,inplace=True)
salary_company_location_pivot_table
```

[19]:
```
job_title         3D Computer Vision Researcher  AI Developer  AI Programmer  \
company_location
AE                                          0.0           0.0            0.0
AL                                      10000.0           0.0            0.0
AM                                          0.0           0.0            0.0
AR                                          0.0           0.0            0.0
AS                                      20000.0           0.0            0.0
…                                           …             …              …
TH                                          0.0           0.0            0.0
TR                                          0.0           0.0            0.0
UA                                          0.0       84000.0            0.0
US                                          0.0      200000.0            0.0
VN                                          0.0           0.0            0.0

job_title          AI Scientist  Analytics Engineer  Applied Data Scientist  \
company_location
AE                     0.000000            0.000000                     0.0
AL                     0.000000            0.000000                     0.0
AM                     0.000000            0.000000                     0.0
AR                     0.000000            0.000000                     0.0
AS                 18053.000000            0.000000                     0.0
…                           …                   …                       …
TH                     0.000000            0.000000                     0.0
TR                     0.000000            0.000000                     0.0
UA                     0.000000            0.000000                     0.0
US                113428.571429       159024.347826                238000.0
VN                     0.000000            0.000000                     0.0

job_title         Applied Machine Learning Engineer  \
company_location
AE                                              0.0
AL                                              0.0
AM                                              0.0
AR                                              0.0
AS                                              0.0
…                                               …
TH                                              0.0
```

```
TR                                          0.0
UA                                          0.0
US                                     130000.0
VN                                          0.0

job_title         Applied Machine Learning Scientist  Applied Scientist  \
company_location
AE                                             0.0           0.000000
AL                                             0.0           0.000000
AM                                             0.0           0.000000
AR                                             0.0           0.000000
AS                                             0.0           0.000000
…                                              …              …
TH                                             0.0           0.000000
TR                                             0.0           0.000000
UA                                             0.0           0.000000
US                                        141550.0      190264.482759
VN                                             0.0           0.000000

job_title         Autonomous Vehicle Technician  …  Principal Data Engineer  \
company_location                                 …
AE                                          0.0  …                      0.0
AL                                          0.0  …                      0.0
AM                                          0.0  …                      0.0
AR                                          0.0  …                      0.0
AS                                          0.0  …                      0.0
…                                           …  …                       …
TH                                          0.0  …                      0.0
TR                                          0.0  …                      0.0
UA                                          0.0  …                      0.0
US                                          0.0  …                 192500.0
VN                                          0.0  …                      0.0

job_title         Principal Data Scientist  \
company_location
AE                                     0.0
AL                                     0.0
AM                                     0.0
AR                                     0.0
AS                                     0.0
…                                      …
TH                                     0.0
TR                                     0.0
UA                                     0.0
US                                255500.0
VN                                     0.0
```

```
job_title         Principal Machine Learning Engineer  Product Data Analyst  \
company_location
AE                                                0.0                   0.0
AL                                                0.0                   0.0
AM                                                0.0                   0.0
AR                                                0.0                   0.0
AS                                                0.0                   0.0
…                                                 …                     …
TH                                                0.0                   0.0
TR                                                0.0                   0.0
UA                                                0.0                   0.0
US                                           190000.0              120000.0
VN                                                0.0                   0.0

job_title         Product Data Scientist  Research Engineer  \
company_location
AE                                   0.0           0.000000
AL                                   0.0           0.000000
AM                                   0.0           0.000000
AR                                   0.0           0.000000
AS                                   0.0           0.000000
…                                    …             …
TH                                   0.0           0.000000
TR                                   0.0           0.000000
UA                                   0.0           0.000000
US                                   0.0      168156.580645
VN                                   0.0           0.000000

job_title         Research Scientist  Software Data Engineer  \
company_location
AE                          0.000000                     0.0
AL                          0.000000                     0.0
AM                          0.000000                     0.0
AR                          0.000000                     0.0
AS                          0.000000                     0.0
…                           …                            …
TH                          0.000000                     0.0
TR                          0.000000                     0.0
UA                          0.000000                     0.0
US                     179146.206897                     0.0
VN                          0.000000                     0.0

job_title         Staff Data Analyst  Staff Data Scientist
company_location
AE                               0.0                   0.0
AL                               0.0                   0.0
AM                               0.0                   0.0
```

```
AR                                  0.0                    0.0
AS                                  0.0                    0.0
…                                   …                      …
TH                                  0.0                    0.0
TR                                  0.0                    0.0
UA                                  0.0                    0.0
US                                  0.0               105000.0
VN                                  0.0                    0.0

[72 rows x 93 columns]
```

[20]:
```python
import matplotlib.pyplot as plt

# Set the figure size
plt.figure(figsize=(15, 15))

# Plot the pivot table as a bar chart
salary_company_location_pivot_table.plot(kind='bar', stacked=True)

# Set the labels and title
plt.xlabel('Company Location')
plt.ylabel('Average Salary (USD)')
plt.title('Average Salary by Company Loction and Job Title')
plt.legend(loc='upper right', bbox_to_anchor=(2, 1))

# Show the plot
plt.show()
```

```
<Figure size 1500x1500 with 0 Axes>
```


Average Salary by Company Loction and Job Title
1e7
1.0
0.8
0.6
0.4
0.2
0.0
Average Salary (USD)
Company Location
3D Computer Vision Researcher
AI Developer
AI Programmer
AI Scientist
Analytics Engineer
Applied Data Scientist
Applied Machine Learning Engineer
Applied Machine Learning Scientist
Applied Scientist
Autonomous Vehicle Technician
Azure Data Engineer
BI Analyst
BI Data Analyst
BI Data Engineer
BI Developer
Big Data Architect
Big Data Engineer
Business Data Analyst
Business Intelligence Engineer
Cloud Data Architect
Cloud Data Engineer
Cloud Database Engineer
Compliance Data Analyst
Computer Vision Engineer
Computer Vision Software Engineer
Data Analyst
Data Analytics Consultant
Data Analytics Engineer
Data Analytics Lead
Data Analytics Manager
Data Analytics Specialist
Data Architect
Data DevOps Engineer
Data Engineer
Data Infrastructure Engineer
Data Lead
Data Management Specialist
Data Manager
Data Modeler
Data Operations Analyst
Data Operations Engineer
Data Quality Analyst
Data Science Consultant
Data Science Engineer
Data Science Lead
Data Science Manager
Data Science Tech Lead
Data Scientist
Data Scientist Lead
Data Specialist
Data Strategist
Deep Learning Engineer
Deep Learning Researcher
Director of Data Science
ETL Developer
ETL Engineer
Finance Data Analyst
Financial Data Analyst
Head of Data
Head of Data Science
Head of Machine Learning
Insight Analyst
Lead Data Analyst
Lead Data Engineer
Lead Data Scientist
Lead Machine Learning Engineer
ML Engineer
MLOps Engineer
Machine Learning Developer
Machine Learning Engineer
Machine Learning Infrastructure Engineer
Machine Learning Manager
Machine Learning Research Engineer
Machine Learning Researcher
Machine Learning Scientist
Machine Learning Software Engineer
Manager Data Management
Marketing Data Analyst
Marketing Data Engineer
NLP Engineer
Power BI Developer
Principal Data Analyst
Principal Data Architect
Principal Data Engineer
Principal Data Scientist
Principal Machine Learning Engineer
Product Data Analyst
Product Data Scientist
Research Engineer
Research Scientist
Software Data Engineer
Staff Data Analyst
Staff Data Scientist

1.2.7 Salary vs Company size

[21]:
```
salary_company_size_pivot_table = pd.
↪pivot_table(df,values='salary_in_usd',index='company_size',columns='job_title')
salary_company_size_pivot_table.replace(np.nan,0,inplace=True)
salary_company_size_pivot_table
```

[21]:
```
job_title     3D Computer Vision Researcher   AI Developer  AI Programmer  \
company_size
L                                       0.0  257309.000000        70000.0
M                                   12704.5   97900.833333        40000.0
S                                   30000.0  133768.000000            0.0

job_title      AI Scientist  Analytics Engineer  Applied Data Scientist  \
company_size
L              173744.166667       130000.000000           149824.166667
M               82704.000000       153623.455446            52772.666667
S               61189.800000        48000.000000            80000.000000

job_title     Applied Machine Learning Engineer  \
company_size
L                                           0.0
M                                      130000.0
S                                       69751.0

job_title     Applied Machine Learning Scientist  Applied Scientist  \
company_size
L                                 154623.800000      191024.666667
M                                  84965.333333      189450.000000
S                                  30523.000000           0.000000

job_title     Autonomous Vehicle Technician  …  Principal Data Engineer  \
company_size                                 …
L                                       0.0  …                 185000.0
M                                   45555.0  …                 200000.0
S                                    7000.0  …                      0.0

job_title     Principal Data Scientist  Principal Machine Learning Engineer  \
company_size
L                           172961.75                                190000.0
M                           159174.00                                    0.0
S                           416000.00                                    0.0

job_title     Product Data Analyst  Product Data Scientist  Research Engineer  \
```

```
company_size
L                          11243.0                   8000.0           0.000000
M                         120000.0                      0.0      166910.114286
S                          20000.0                      0.0       96578.000000

job_title     Research Scientist  Software Data Engineer  Staff Data Analyst  \
company_size
L                 101745.800000                 75020.0                 0.0
M                 180579.460317                 50000.0             15000.0
S                  79217.750000                     0.0                 0.0

job_title     Staff Data Scientist
company_size
L                              0.0
M                         105000.0
S                              0.0

[3 rows x 93 columns]
```

[22]:
```
import matplotlib.pyplot as plt

# Set the figure size
plt.figure(figsize=(10, 10))

# Plot the pivot table as a bar chart
salary_company_size_pivot_table.plot(kind='bar', stacked=True)

# Set the labels and title
plt.xlabel('Company Size')
plt.ylabel('Average Salary (USD)')
plt.title('Average Salary by Company Size and Job Title')
plt.legend(loc='upper right', bbox_to_anchor=(2, 1))

# Show the plot
plt.show()
```

```
<Figure size 1000x1000 with 0 Axes>
```


Average Salary by Company Size and Job Title
1e6
Average Salary (USD)
Company Size
L
M
S
3D Computer Vision Researcher
AI Developer
AI Programmer
AI Scientist
Analytics Engineer
Applied Data Scientist
Applied Machine Learning Engineer
Applied Machine Learning Scientist
Applied Scientist
Autonomous Vehicle Technician
Azure Data Engineer
BI Analyst
BI Data Analyst
BI Data Engineer
BI Developer
Big Data Architect
Big Data Engineer
Business Data Analyst
Business Intelligence Engineer
Cloud Data Architect
Cloud Data Engineer
Cloud Database Engineer
Compliance Data Analyst
Computer Vision Engineer
Computer Vision Software Engineer
Data Analyst
Data Analytics Consultant
Data Analytics Engineer
Data Analytics Lead
Data Analytics Manager
Data Analytics Specialist
Data Architect
Data DevOps Engineer
Data Engineer
Data Infrastructure Engineer
Data Lead
Data Management Specialist
Data Manager
Data Modeler
Data Operations Analyst
Data Operations Engineer
Data Quality Analyst
Data Science Consultant
Data Science Engineer
Data Science Lead
Data Science Manager
Data Science Tech Lead
Data Scientist
Data Scientist Lead
Data Specialist
Data Strategist
Deep Learning Engineer
Deep Learning Researcher
Director of Data Science
ETL Developer
ETL Engineer
Finance Data Analyst
Financial Data Analyst
Head of Data
Head of Data Science
Head of Machine Learning
Insight Analyst
Lead Data Analyst
Lead Data Engineer
Lead Data Scientist
Lead Machine Learning Engineer
ML Engineer
MLOps Engineer
Machine Learning Developer
Machine Learning Engineer
Machine Learning Infrastructure Engineer
Machine Learning Manager
Machine Learning Research Engineer
Machine Learning Researcher
Machine Learning Scientist
Machine Learning Software Engineer
Manager Data Management
Marketing Data Analyst
Marketing Data Engineer
NLP Engineer
Power BI Developer
Principal Data Analyst
Principal Data Architect
Principal Data Engineer
Principal Data Scientist
Principal Machine Learning Engineer
Product Data Analyst
Product Data Scientist
Research Engineer
Research Scientist
Software Data Engineer
Staff Data Analyst
Staff Data Scientist

1.3 Machine Learning

1.4 Classification

In this section, we're going to build a model that predicts the size of the company that the employee belongs to We're going to test 4 classification algorithms and compare the accuracy of each of them, and then deploy the model to be used on out-of-sample data

Data Pre-processing

[23]:
```
from sklearn import preprocessing
```

[24]:
```
features = df.columns
features = list(features)

features.remove('salary')
features.remove('salary_currency')
features.remove('company_size')


X = df[features].values

X
```

[24]:
```
array([[2023, 'SE', 'FT', …, 'ES', 100, 'ES'],
       [2023, 'MI', 'CT', …, 'US', 100, 'US'],
       [2023, 'MI', 'CT', …, 'US', 100, 'US'],
       …,
       [2020, 'EN', 'FT', …, 'US', 100, 'US'],
       [2020, 'EN', 'CT', …, 'US', 100, 'US'],
       [2021, 'SE', 'FT', …, 'IN', 50, 'IN']], dtype=object)
```

[25]:
```
features
```

[25]:
```
['work_year',
 'experience_level',
 'employment_type',
 'job_title',
 'salary_in_usd',
 'employee_residence',
 'remote_ratio',
 'company_location']
```

[26]:
```
Y = df['company_size'].values
Y
```

[26]:
```
array(['L', 'S', 'S', …, 'S', 'L', 'L'], dtype=object)
```

[27]:
```
X[:,1]
```

[27]:
```
array(['SE', 'MI', 'MI', …, 'EN', 'EN', 'SE'], dtype=object)
```

As you might see that there are some features that have categorical values, which is not convenient for an algorithm such as KNN, for that we can label them and convert them to int or floats

[28]:
```
le = preprocessing.LabelEncoder()

for (index, feature) in enumerate(features):
    available_values = df[feature].unique()
    column_type = df[feature].dtype
    print(available_values)
    print(feature)
    print(index)

    # Checking if the type of the column is object
    if column_type == 'object':
        le.fit(available_values)
        # Perform label encoding or any other desired operations
        X[:,index] = le.transform(X[:,index])

X
```

```
[2023 2022 2020 2021]
work_year
0
['SE' 'MI' 'EN' 'EX']
experience_level
1
['FT' 'CT' 'FL' 'PT']
employment_type
2
['Principal Data Scientist' 'ML Engineer' 'Data Scientist'
 'Applied Scientist' 'Data Analyst' 'Data Modeler' 'Research Engineer'
 'Analytics Engineer' 'Business Intelligence Engineer'
 'Machine Learning Engineer' 'Data Strategist' 'Data Engineer'
 'Computer Vision Engineer' 'Data Quality Analyst'
 'Compliance Data Analyst' 'Data Architect'
 'Applied Machine Learning Engineer' 'AI Developer' 'Research Scientist'
 'Data Analytics Manager' 'Business Data Analyst' 'Applied Data Scientist'
 'Staff Data Analyst' 'ETL Engineer' 'Data DevOps Engineer' 'Head of Data'
 'Data Science Manager' 'Data Manager' 'Machine Learning Researcher'
 'Big Data Engineer' 'Data Specialist' 'Lead Data Analyst'
 'BI Data Engineer' 'Director of Data Science'
 'Machine Learning Scientist' 'MLOps Engineer' 'AI Scientist']
```

```
 'Autonomous Vehicle Technician' 'Applied Machine Learning Scientist'
 'Lead Data Scientist' 'Cloud Database Engineer' 'Financial Data Analyst'
 'Data Infrastructure Engineer' 'Software Data Engineer' 'AI Programmer'
 'Data Operations Engineer' 'BI Developer' 'Data Science Lead'
 'Deep Learning Researcher' 'BI Analyst' 'Data Science Consultant'
 'Data Analytics Specialist' 'Machine Learning Infrastructure Engineer'
 'BI Data Analyst' 'Head of Data Science' 'Insight Analyst'
 'Deep Learning Engineer' 'Machine Learning Software Engineer'
 'Big Data Architect' 'Product Data Analyst'
 'Computer Vision Software Engineer' 'Azure Data Engineer'
 'Marketing Data Engineer' 'Data Analytics Lead' 'Data Lead'
 'Data Science Engineer' 'Machine Learning Research Engineer'
 'NLP Engineer' 'Manager Data Management' 'Machine Learning Developer'
 '3D Computer Vision Researcher' 'Principal Machine Learning Engineer'
 'Data Analytics Engineer' 'Data Analytics Consultant'
 'Data Management Specialist' 'Data Science Tech Lead'
 'Data Scientist Lead' 'Cloud Data Engineer' 'Data Operations Analyst'
 'Marketing Data Analyst' 'Power BI Developer' 'Product Data Scientist'
 'Principal Data Architect' 'Machine Learning Manager'
 'Lead Machine Learning Engineer' 'ETL Developer' 'Cloud Data Architect'
 'Lead Data Engineer' 'Head of Machine Learning' 'Principal Data Analyst'
 'Principal Data Engineer' 'Staff Data Scientist' 'Finance Data Analyst']
job_title
3
[ 85847  30000  25500 …  28369 412000  94665]
salary_in_usd
4
['ES' 'US' 'CA' 'DE' 'GB' 'NG' 'IN' 'HK' 'PT' 'NL' 'CH' 'CF' 'FR' 'AU'
 'FI' 'UA' 'IE' 'IL' 'GH' 'AT' 'CO' 'SG' 'SE' 'SI' 'MX' 'UZ' 'BR' 'TH'
 'HR' 'PL' 'KW' 'VN' 'CY' 'AR' 'AM' 'BA' 'KE' 'GR' 'MK' 'LV' 'RO' 'PK'
 'IT' 'MA' 'LT' 'BE' 'AS' 'IR' 'HU' 'SK' 'CN' 'CZ' 'CR' 'TR' 'CL' 'PR'
 'DK' 'BO' 'PH' 'DO' 'EG' 'ID' 'AE' 'MY' 'JP' 'EE' 'HN' 'TN' 'RU' 'DZ'
 'IQ' 'BG' 'JE' 'RS' 'NZ' 'MD' 'LU' 'MT']
employee_residence
5
[100   0  50]
remote_ratio
6
['ES' 'US' 'CA' 'DE' 'GB' 'NG' 'IN' 'HK' 'NL' 'CH' 'CF' 'FR' 'FI' 'UA'
 'IE' 'IL' 'GH' 'CO' 'SG' 'AU' 'SE' 'SI' 'MX' 'BR' 'PT' 'RU' 'TH' 'HR'
 'VN' 'EE' 'AM' 'BA' 'KE' 'GR' 'MK' 'LV' 'RO' 'PK' 'IT' 'MA' 'PL' 'AL'
 'AR' 'LT' 'AS' 'CR' 'IR' 'BS' 'HU' 'AT' 'SK' 'CZ' 'TR' 'PR' 'DK' 'BO'
 'PH' 'BE' 'ID' 'EG' 'AE' 'LU' 'MY' 'HN' 'JP' 'DZ' 'IQ' 'CN' 'NZ' 'CL'
 'MD' 'MT']
company_location
7
```

[28]:
```
array([[2023, 3, 2, …, 26, 100, 25],
       [2023, 2, 0, …, 75, 100, 70],
       [2023, 2, 0, …, 75, 100, 70],
       …,
       [2020, 0, 2, …, 75, 100, 70],
       [2020, 0, 0, …, 75, 100, 70],
       [2021, 3, 2, …, 39, 50, 38]], dtype=object)
```

1.4.1 Normalize the data

Now, we're going to normalize the data to equalize the features scales and improve the convergence towards an optmized solution

[29]:
```
X = preprocessing.StandardScaler().fit(X).transform(X.astype(float))

X
```

[29]:
```
array([[  0.90599446,   0.58573566,   0.02592668, …,  -2.08756459,
          1.10591825,  -2.12181783],
       [  0.90599446,  -0.51784558, -14.95171653, …,   0.46018613,
          1.10591825,   0.45062463],
       [  0.90599446,  -0.51784558, -14.95171653, …,   0.46018613,
          1.10591825,   0.45062463],
       …,
       [ -3.43330297,  -2.72500806,   0.02592668, …,   0.46018613,
          1.10591825,   0.45062463],
       [ -3.43330297,  -2.72500806, -14.95171653, …,   0.46018613,
          1.10591825,   0.45062463],
       [ -1.9868705 ,   0.58573566,   0.02592668, …,  -1.41163073,
          0.07674278,  -1.37866779]])
```

1.4.2 Train/Test split

We won't train our model on all of the data we have, as It will give as a specialized model that will not give good and accurate results. For that, we gonna split our dataset into 2 parts (Train Data and Test Data)

[30]:
```
from sklearn.model_selection import train_test_split

train_x, test_x, train_y, test_y = train_test_split(X,Y,test_size=0.
 ↪2,random_state=4)

train_x.shape
```

[30]:
```
(3004, 8)
```

1.4.3 K-nearest neighbours

The first algorithm that we will try is the KNN

[31]:
```python
accuracies = []
```

Training

[32]:
```python
from sklearn.neighbors import KNeighborsClassifier

knc = KNeighborsClassifier(n_neighbors=4)

knc.fit(train_x,train_y)
```

[32]:
```
KNeighborsClassifier(n_neighbors=4)
```

Prediction

[33]:
```python
yhat = knc.predict(test_x)

yhat
```

[33]:
```
array(['M', 'M', 'M', 'M', 'L', 'M', 'M', 'M', 'M', 'M', 'L', 'M', 'L',
       'M', 'M', 'M', 'M', 'M', 'M', 'M', 'S', 'M', 'M', 'M', 'M', 'L',
       'M', 'M', 'M', 'M', 'M', 'L', 'L', 'M', 'M', 'M', 'M', 'M', 'M',
       'M', 'M', 'L', 'M', 'M', 'M', 'L', 'M', 'M', 'M', 'M', 'M', 'M',
       'M', 'L', 'L', 'M', 'M', 'M', 'M', 'M', 'M', 'L', 'M', 'M', 'S',
       'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M',
       'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M',
       'M', 'M', 'M', 'M', 'M', 'M', 'L', 'L', 'M', 'M', 'M', 'M', 'L',
       'M', 'M', 'M', 'M', 'M', 'M', 'M', 'L', 'L', 'M', 'M', 'M', 'M',
       'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'L', 'M', 'L', 'M',
       'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'L', 'M', 'M', 'M',
       'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'L', 'M',
       'M', 'L', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M',
       'M', 'M', 'M', 'M', 'M', 'M', 'M', 'L', 'M', 'L', 'M', 'M', 'M',
       'M', 'M', 'M', 'M', 'M', 'M', 'M', 'L', 'M', 'M', 'L', 'M', 'M',
       'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M',
       'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M',
       'M', 'L', 'M', 'M', 'L', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M',
       'M', 'M', 'M', 'L', 'L', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M',
       'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'L', 'M',
       'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'S', 'L', 'M', 'M',
       'M', 'M', 'L', 'M', 'M', 'M', 'M', 'L', 'L', 'M', 'M', 'M', 'M',
       'M', 'M', 'M', 'M', 'M', 'M', 'M', 'L', 'L', 'M', 'M', 'M', 'M',
       'M', 'M', 'M', 'M', 'L', 'L', 'M', 'M', 'M', 'M', 'L', 'M', 'M',
       'L', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'L', 'M', 'M', 'M', 'M',
       'M', 'M', 'M', 'M', 'M', 'M', 'M', 'L', 'M', 'M', 'M', 'M', 'M',
       'M', 'L', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M',
       'M', 'M', 'M', 'L', 'L', 'M', 'M', 'L', 'M', 'M', 'S', 'M', 'M',
       'M', 'M', 'M', 'M', 'L', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M',
       'M', 'M', 'M', 'S', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M',
       'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M',
```

```
'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'L', 'M', 'M',
'M', 'L', 'M', 'M', 'M', 'M', 'S', 'M', 'M', 'M', 'M', 'L', 'M',
'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M',
'M', 'M', 'M', 'L', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'L', 'M',
'M', 'M', 'M', 'M', 'M', 'M', 'L', 'M', 'L', 'M', 'M', 'M', 'M',
'M', 'L', 'L', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M',
'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'L', 'M', 'M', 'M',
'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M',
'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M',
'M', 'M', 'M', 'S', 'M', 'M', 'M', 'M', 'M', 'M', 'L', 'M', 'M',
'L', 'M', 'M', 'M', 'M', 'M', 'M', 'L', 'L', 'M', 'L', 'M', 'M',
'M', 'M', 'M', 'L', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M',
'M', 'M', 'L', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M',
'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'L', 'M', 'M', 'M',
'M', 'M', 'M', 'M', 'L', 'M', 'M', 'M', 'M', 'M', 'L', 'L', 'M',
'M', 'M', 'L', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'L', 'M', 'M',
'M', 'L', 'M', 'L', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M',
'M', 'M', 'M', 'M', 'L', 'M', 'M', 'M', 'M', 'L', 'M', 'M', 'M',
'M', 'M', 'M', 'L', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'S',
'M', 'M', 'M', 'M', 'M', 'L', 'M', 'M', 'M', 'M', 'M', 'M', 'M',
'M', 'M', 'M', 'M', 'L', 'M', 'M', 'S', 'M', 'L', 'M', 'M', 'M',
'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'L', 'L', 'M', 'L', 'M',
'L', 'M', 'M', 'S', 'L', 'M', 'M', 'M', 'M', 'L', 'M', 'M', 'M',
'M', 'M', 'L', 'M', 'M', 'M', 'M', 'M', 'L', 'L', 'M', 'M', 'M',
'S', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M',
'M', 'L', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'L', 'M', 'M',
'M', 'M', 'M', 'L', 'M', 'M', 'L', 'M', 'M', 'M'], dtype=object)
```

Model evaluation

[34]:
```
from sklearn.metrics import accuracy_score

print(f"The train set accuracy is {accuracy_score(train_y,knc.
 ↪predict(train_x))}")
print(f"The accuracy of test set is {accuracy_score(yhat,test_y)}")
```

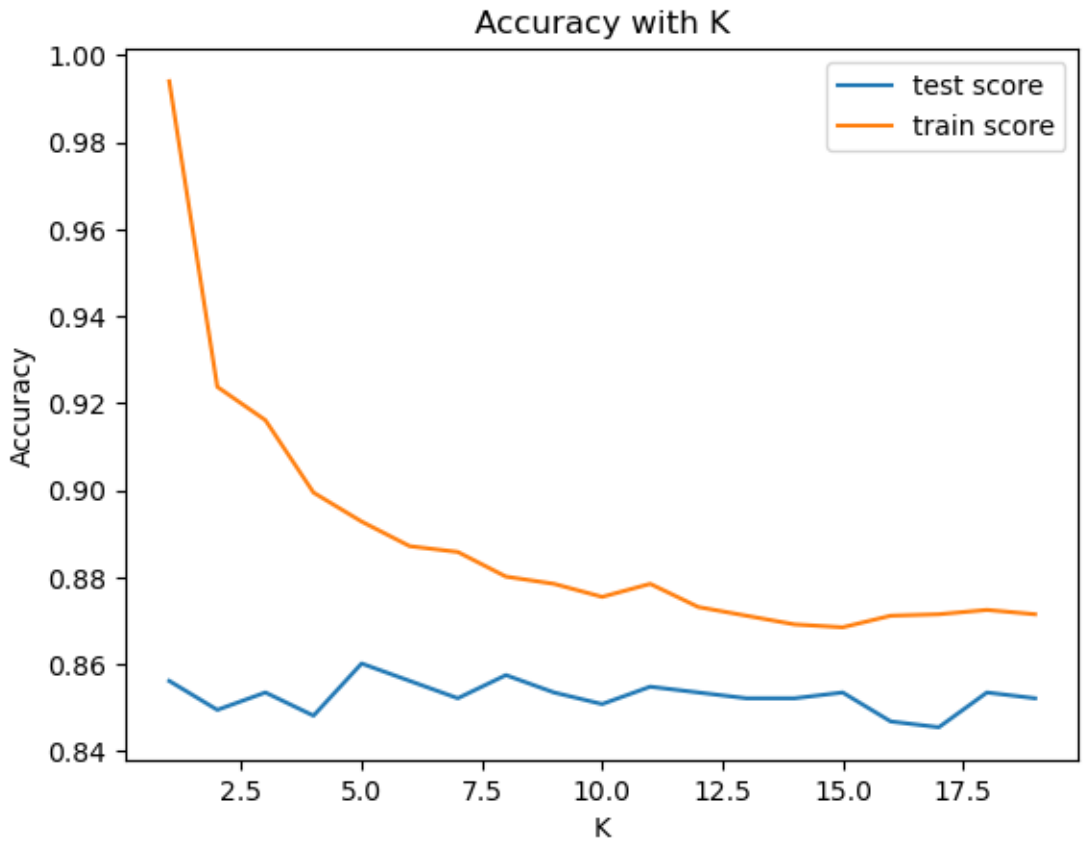
```
The train set accuracy is 0.8994673768308922
The accuracy of test set is 0.848202396804261
```

That was only an example of 4 neighbours, let us now try neighours from 1 to 10

[35]:
```
test_scores = []
train_scores = []

Ks = 20

for i in range(1,Ks):

    knc = KNeighborsClassifier(n_neighbors=i)
```

```
    knc.fit(train_x,train_y)
    yhat = knc.predict(test_x)

    test_score = accuracy_score(yhat,test_y)
    train_score = accuracy_score(train_y,knc.predict(train_x))

    test_scores.append(test_score)
    train_scores.append(train_score)

plt.plot(range(1,Ks),test_scores)
plt.plot(range(1,Ks),train_scores)
plt.xlabel("K")
plt.ylabel("Accuracy")
plt.title("Accuracy with K")
plt.legend(['test score','train score'])
```

[35]: <matplotlib.legend.Legend at 0x7f06f885a590>



[36]:
```
print(f"The best accuracy in our dataset is {max(test_scores)} with␣
 ↪K={test_scores.index(max(test_scores))}")
accuracies.append(max(test_scores))
```

```
The best accuracy in our dataset is 0.8601864181091877 with K=4
```

So the K with the highest accuracy is **K=4** with an accuracy of **0.860186418091877**

Let us now, evaluate our model using Jaccard Index

[37]:
```
from sklearn.metrics import jaccard_score

print(f"The Jaccard score for this model is␣
 ↪{jaccard_score(test_y,yhat,average='micro')}")
```

```
The Jaccard score for this model is 0.7424593967517401
```

As we have a multi-class classification, we need to set-up an average which can be of {None, micro, macro and weighted} We chose micro, because It calculates the total true positives, false positives, true negatives and false negative across all classes

Let us now try and plot the confusion matrix of our model to see the precision of our model with each and every class

[38]:
```
from sklearn.metrics import classification_report, confusion_matrix
import itertools
def plot_confusion_matrix(cm, classes,
                          normalize=False,
                          title='Confusion matrix',
                          cmap=plt.cm.Blues):
    """
    This function prints and plots the confusion matrix.
    Normalization can be applied by setting `normalize=True`.
    """
    if normalize:
        cm = cm.astype('float') / cm.sum(axis=1)[:, np.newaxis]
        print("Normalized confusion matrix")
    else:
        print('Confusion matrix, without normalization')

    print(cm)

    plt.imshow(cm, interpolation='nearest', cmap=cmap)
    plt.title(title)
    plt.colorbar()
    tick_marks = np.arange(len(classes))
    plt.xticks(tick_marks, classes, rotation=45)
    plt.yticks(tick_marks, classes)

    fmt = '.2f' if normalize else 'd'
```

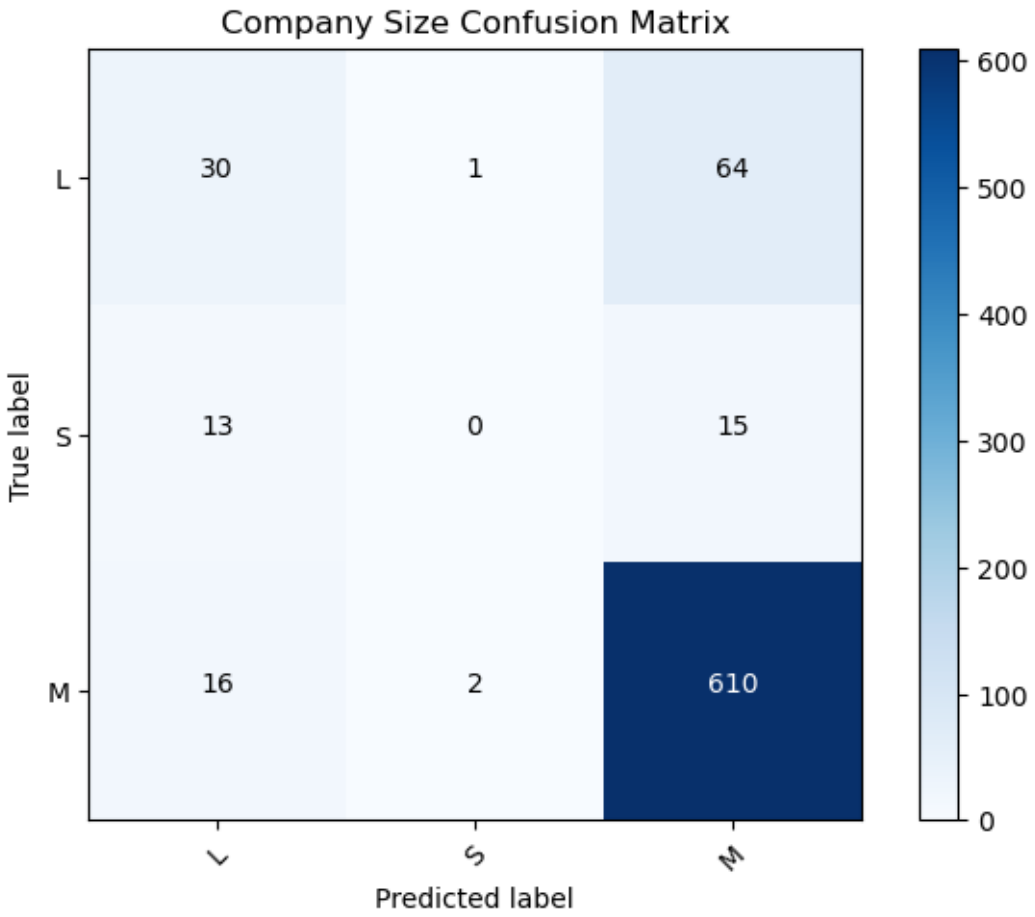
```
    thresh = cm.max() / 2.
    for i, j in itertools.product(range(cm.shape[0]), range(cm.shape[1])):
        plt.text(j, i, format(cm[i, j], fmt),
                 horizontalalignment="center",
                 color="white" if cm[i, j] > thresh else "black")

    plt.tight_layout()
    plt.ylabel('True label')
    plt.xlabel('Predicted label')

cnf_matrix = confusion_matrix(test_y,yhat,labels = df['company_size'].unique())
print(cnf_matrix)
```

```
[[ 30   1  64]
 [ 13   0  15]
 [ 16   2 610]]
```

Plotting the confusion matrix

[39]:
```
plot_confusion_matrix(cnf_matrix,df['company_size'].unique(),title="Company␣
↪Size Confusion Matrix")
```

```
Confusion matrix, without normalization
[[ 30   1  64]
 [ 13   0  15]
 [ 16   2 610]]
```



Seeing this plot, we conclude that : **Large** companies, he got 64 of them as medium, 1 as Small and 30 as Large, which is not that accurate **Small** companies, he got 15 of them as medium, 0 as Small and 13 as Large, which is not that accurate **Medium** companies, he got 610 of them as medium, 2 as Small and 16 as Large, which is really accurate

We can explain this with the amount of Data for each category, as we have so much data for medium sized companies and not that much for Small and Large companies

We can also see the classification report to give an eye of the f1_score, precision and recall

[40]:
```
print(classification_report(test_y,yhat))
```

```
              precision    recall  f1-score   support

           L       0.51      0.32      0.39        95
           M       0.89      0.97      0.93       628
           S       0.00      0.00      0.00        28

    accuracy                           0.85       751
   macro avg       0.46      0.43      0.44       751
```

```
weighted avg       0.80      0.85      0.82       751
```

We can see here that f1-score for the Medium companies is 0.93 which is high in comparaison with Large and Small companies which are 0.39 and 0 respectively

1.4.4 Decision Trees

Now, we're trying using Decision trees to decide the class of a company

[41]:
```
from sklearn.tree import DecisionTreeClassifier

dt = DecisionTreeClassifier(criterion='entropy',max_depth=6)
```

Training the model

[42]:
```
dt.fit(train_x,train_y)
```

[42]:
```
DecisionTreeClassifier(criterion='entropy', max_depth=6)
```

Prediction

[43]:
```
yhat = dt.predict(test_x)

yhat[:5]
```

[43]:
```
array(['M', 'M', 'M', 'M', 'S'], dtype=object)
```

Let us now plot the decision tree to see the logic behind It

[44]:
```
import sklearn.tree as tree

tree.plot_tree(dt)
```

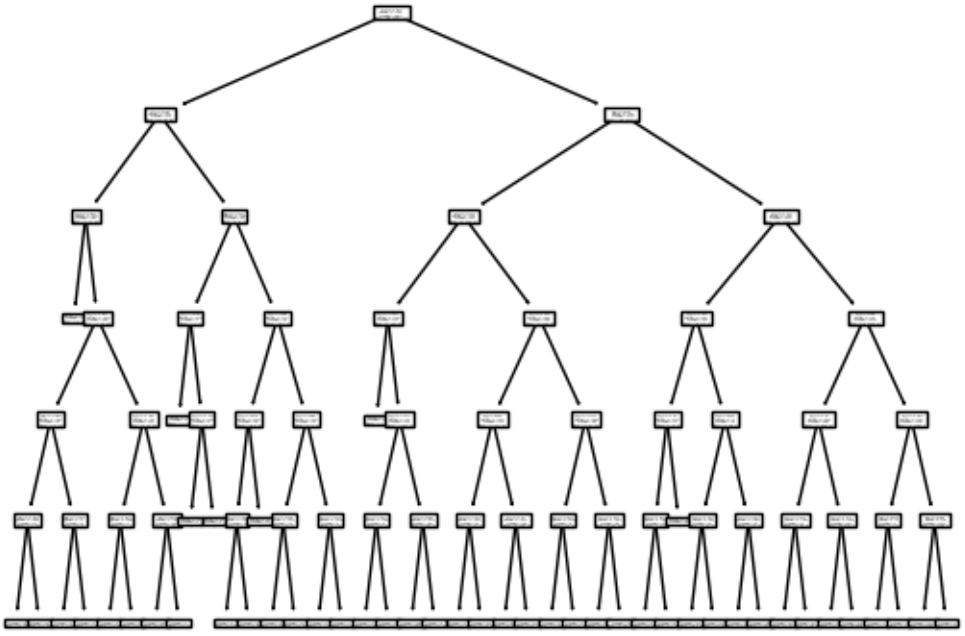
[44]:
```
[Text(0.4054878048780488, 0.9285714285714286, 'x[0] <= -1.264\nentropy =
0.762\nsamples = 3004\nvalue = [359, 2525, 120]'),
 Text(0.16310975609756098, 0.7857142857142857, 'x[4] <= 0.025\nentropy =
1.452\nsamples = 242\nvalue = [131, 52, 59]'),
 Text(0.08536585365853659, 0.6428571428571429, 'x[3] <= -2.059\nentropy =
1.519\nsamples = 190\nvalue = [91, 47, 52]'),
 Text(0.07317073170731707, 0.5, 'entropy = 0.0\nsamples = 2\nvalue = [0, 2,
0]'),
 Text(0.0975609756097561, 0.5, 'x[4] <= -1.837\nentropy = 1.513\nsamples =
188\nvalue = [91, 45, 52]'),
 Text(0.04878048780487805, 0.35714285714285715, 'x[5] <= -0.658\nentropy =
1.487\nsamples = 27\nvalue = [9, 5, 13]'),
 Text(0.024390243902439025, 0.21428571428571427, 'x[5] <= -1.568\nentropy =
1.268\nsamples = 21\nvalue = [6, 2, 13]'),
 Text(0.012195121951219513, 0.07142857142857142, 'entropy = 0.722\nsamples =
5\nvalue = [0, 1, 4]'),
```

```
 Text(0.036585365853658534, 0.07142857142857142, 'entropy = 1.248\nsamples =
16\nvalue = [6, 1, 9]'),
 Text(0.07317073170731707, 0.21428571428571427, 'x[5] <= -0.034\nentropy =
1.0\nsamples = 6\nvalue = [3, 3, 0]'),
 Text(0.06097560975609756, 0.07142857142857142, 'entropy = 0.811\nsamples =
4\nvalue = [1, 3, 0]'),
 Text(0.08536585365853659, 0.07142857142857142, 'entropy = 0.0\nsamples =
2\nvalue = [2, 0, 0]'),
 Text(0.14634146341463414, 0.35714285714285715, 'x[4] <= -1.783\nentropy =
1.49\nsamples = 161\nvalue = [82, 40, 39]'),
 Text(0.12195121951219512, 0.21428571428571427, 'x[5] <= -1.906\nentropy =
0.985\nsamples = 7\nvalue = [3, 4, 0]'),
 Text(0.10975609756097561, 0.07142857142857142, 'entropy = 0.0\nsamples =
3\nvalue = [0, 3, 0]'),
 Text(0.13414634146341464, 0.07142857142857142, 'entropy = 0.811\nsamples =
4\nvalue = [3, 1, 0]'),
 Text(0.17073170731707318, 0.21428571428571427, 'x[5] <= 0.512\nentropy =
1.486\nsamples = 154\nvalue = [79, 36, 39]'),
 Text(0.15853658536585366, 0.07142857142857142, 'entropy = 1.478\nsamples =
152\nvalue = [79, 34, 39]'),
 Text(0.18292682926829268, 0.07142857142857142, 'entropy = 0.0\nsamples =
2\nvalue = [0, 2, 0]'),
 Text(0.24085365853658536, 0.6428571428571429, 'x[5] <= -0.112\nentropy =
1.005\nsamples = 52\nvalue = [40, 5, 7]'),
 Text(0.1951219512195122, 0.5, 'x[1] <= 0.034\nentropy = 1.379\nsamples =
7\nvalue = [2, 1, 4]'),
 Text(0.18292682926829268, 0.35714285714285715, 'entropy = 0.0\nsamples =
2\nvalue = [2, 0, 0]'),
 Text(0.2073170731707317, 0.35714285714285715, 'x[7] <= -1.779\nentropy =
0.722\nsamples = 5\nvalue = [0, 1, 4]'),
 Text(0.1951219512195122, 0.21428571428571427, 'entropy = 0.0\nsamples =
1\nvalue = [0, 1, 0]'),
 Text(0.21951219512195122, 0.21428571428571427, 'entropy = 0.0\nsamples =
4\nvalue = [0, 0, 4]'),
 Text(0.2865853658536585, 0.5, 'x[3] <= 1.286\nentropy = 0.777\nsamples =
45\nvalue = [38, 4, 3]'),
 Text(0.25609756097560976, 0.35714285714285715, 'x[4] <= 0.213\nentropy =
0.353\nsamples = 30\nvalue = [28, 2, 0]'),
 Text(0.24390243902439024, 0.21428571428571427, 'x[3] <= -0.063\nentropy =
0.764\nsamples = 9\nvalue = [7, 2, 0]'),
 Text(0.23170731707317074, 0.07142857142857142, 'entropy = 0.0\nsamples =
5\nvalue = [5, 0, 0]'),
 Text(0.25609756097560976, 0.07142857142857142, 'entropy = 1.0\nsamples =
4\nvalue = [2, 2, 0]'),
 Text(0.2682926829268293, 0.21428571428571427, 'entropy = 0.0\nsamples =
21\nvalue = [21, 0, 0]'),
 Text(0.3170731707317073, 0.35714285714285715, 'x[4] <= 1.831\nentropy =
```

```
1.242\nsamples = 15\nvalue = [10, 2, 3]'),
 Text(0.2926829268292683, 0.21428571428571427, 'x[3] <= 1.448\nentropy =
1.041\nsamples = 12\nvalue = [9, 2, 1]'),
 Text(0.2804878048780488, 0.07142857142857142, 'entropy = 0.0\nsamples =
1\nvalue = [0, 0, 1]'),
 Text(0.3048780487804878, 0.07142857142857142, 'entropy = 0.684\nsamples =
11\nvalue = [9, 2, 0]'),
 Text(0.34146341463414637, 0.21428571428571427, 'x[2] <= -7.463\nentropy =
0.918\nsamples = 3\nvalue = [1, 0, 2]'),
 Text(0.32926829268292684, 0.07142857142857142, 'entropy = 1.0\nsamples =
2\nvalue = [1, 0, 1]'),
 Text(0.35365853658536583, 0.07142857142857142, 'entropy = 0.0\nsamples =
1\nvalue = [0, 0, 1]'),
 Text(0.6478658536585366, 0.7857142857142857, 'x[5] <= 0.434\nentropy =
0.561\nsamples = 2762\nvalue = [228, 2473, 61]'),
 Text(0.4817073170731707, 0.6428571428571429, 'x[6] <= -0.438\nentropy =
1.128\nsamples = 431\nvalue = [77, 309, 45]'),
 Text(0.4024390243902439, 0.5, 'x[7] <= -3.036\nentropy = 0.462\nsamples =
175\nvalue = [11, 161, 3]'),
 Text(0.3902439024390244, 0.35714285714285715, 'entropy = 0.0\nsamples =
2\nvalue = [2, 0, 0]'),
 Text(0.4146341463414634, 0.35714285714285715, 'x[1] <= -2.173\nentropy =
0.42\nsamples = 173\nvalue = [9, 161, 3]'),
 Text(0.3902439024390244, 0.21428571428571427, 'x[4] <= -2.055\nentropy =
1.406\nsamples = 8\nvalue = [3, 4, 1]'),
 Text(0.3780487804878049, 0.07142857142857142, 'entropy = 0.0\nsamples =
1\nvalue = [0, 0, 1]'),
 Text(0.4024390243902439, 0.07142857142857142, 'entropy = 0.985\nsamples =
7\nvalue = [3, 4, 0]'),
 Text(0.43902439024390244, 0.21428571428571427, 'x[3] <= 2.635\nentropy =
0.319\nsamples = 165\nvalue = [6, 157, 2]'),
 Text(0.4268292682926829, 0.07142857142857142, 'entropy = 0.235\nsamples =
156\nvalue = [6, 150, 0]'),
 Text(0.45121951219512196, 0.07142857142857142, 'entropy = 0.764\nsamples =
9\nvalue = [0, 7, 2]'),
 Text(0.5609756097560976, 0.5, 'x[7] <= -1.579\nentropy = 1.389\nsamples =
256\nvalue = [66, 148, 42]'),
 Text(0.5121951219512195, 0.35714285714285715, 'x[6] <= 0.591\nentropy =
1.203\nsamples = 170\nvalue = [30, 117, 23]'),
 Text(0.4878048780487805, 0.21428571428571427, 'x[3] <= -2.032\nentropy =
1.531\nsamples = 32\nvalue = [14, 11, 7]'),
 Text(0.47560975609756095, 0.07142857142857142, 'entropy = 0.0\nsamples =
2\nvalue = [0, 0, 2]'),
 Text(0.5, 0.07142857142857142, 'entropy = 1.475\nsamples = 30\nvalue = [14, 11,
5]'),
 Text(0.5365853658536586, 0.21428571428571427, 'x[1] <= -1.07\nentropy =
1.013\nsamples = 138\nvalue = [16, 106, 16]'),
```

```
 Text(0.524390243902439, 0.07142857142857142, 'entropy = 1.538\nsamples =
23\nvalue = [9, 9, 5]'),
 Text(0.5487804878048781, 0.07142857142857142, 'entropy = 0.777\nsamples =
115\nvalue = [7, 97, 11]'),
 Text(0.6097560975609756, 0.35714285714285715, 'x[2] <= -3.718\nentropy =
1.538\nsamples = 86\nvalue = [36, 31, 19]'),
 Text(0.5853658536585366, 0.21428571428571427, 'x[4] <= -0.913\nentropy =
0.918\nsamples = 6\nvalue = [0, 2, 4]'),
 Text(0.573170731707317, 0.07142857142857142, 'entropy = 0.0\nsamples = 4\nvalue
= [0, 0, 4]'),
 Text(0.5975609756097561, 0.07142857142857142, 'entropy = 0.0\nsamples =
2\nvalue = [0, 2, 0]'),
 Text(0.6341463414634146, 0.21428571428571427, 'x[4] <= -1.931\nentropy =
1.502\nsamples = 80\nvalue = [36, 29, 15]'),
 Text(0.6219512195121951, 0.07142857142857142, 'entropy = 1.361\nsamples =
10\nvalue = [4, 1, 5]'),
 Text(0.6463414634146342, 0.07142857142857142, 'entropy = 1.446\nsamples =
70\nvalue = [32, 28, 10]'),
 Text(0.8140243902439024, 0.6428571428571429, 'x[3] <= -1.196\nentropy =
0.405\nsamples = 2331\nvalue = [151, 2164, 16]'),
 Text(0.725609756097561, 0.5, 'x[3] <= -1.897\nentropy = 0.907\nsamples =
152\nvalue = [38, 112, 2]'),
 Text(0.6951219512195121, 0.35714285714285715, 'x[1] <= -1.07\nentropy =
0.273\nsamples = 77\nvalue = [1, 74, 2]'),
 Text(0.6829268292682927, 0.21428571428571427, 'x[3] <= -1.951\nentropy =
1.096\nsamples = 11\nvalue = [1, 8, 2]'),
 Text(0.6707317073170732, 0.07142857142857142, 'entropy = 0.918\nsamples =
3\nvalue = [0, 1, 2]'),
 Text(0.6951219512195121, 0.07142857142857142, 'entropy = 0.544\nsamples =
8\nvalue = [1, 7, 0]'),
 Text(0.7073170731707317, 0.21428571428571427, 'entropy = 0.0\nsamples =
66\nvalue = [0, 66, 0]'),
 Text(0.7560975609756098, 0.35714285714285715, 'x[6] <= 0.591\nentropy =
1.0\nsamples = 75\nvalue = [37, 38, 0]'),
 Text(0.7317073170731707, 0.21428571428571427, 'x[3] <= -1.627\nentropy =
0.83\nsamples = 42\nvalue = [31, 11, 0]'),
 Text(0.7195121951219512, 0.07142857142857142, 'entropy = 0.459\nsamples =
31\nvalue = [28, 3, 0]'),
 Text(0.7439024390243902, 0.07142857142857142, 'entropy = 0.845\nsamples =
11\nvalue = [3, 8, 0]'),
 Text(0.7804878048780488, 0.21428571428571427, 'x[4] <= -0.041\nentropy =
0.684\nsamples = 33\nvalue = [6, 27, 0]'),
 Text(0.7682926829268293, 0.07142857142857142, 'entropy = 0.985\nsamples =
7\nvalue = [4, 3, 0]'),
 Text(0.7926829268292683, 0.07142857142857142, 'entropy = 0.391\nsamples =
26\nvalue = [2, 24, 0]'),
 Text(0.9024390243902439, 0.5, 'x[0] <= 0.183\nentropy = 0.35\nsamples =
```

```
2179\nvalue = [113, 2052, 14]'),
 Text(0.8536585365853658, 0.35714285714285715, 'x[1] <= -1.07\nentropy =
0.523\nsamples = 998\nvalue = [92, 896, 10]'),
 Text(0.8292682926829268, 0.21428571428571427, 'x[4] <= -0.128\nentropy =
1.106\nsamples = 71\nvalue = [14, 51, 6]'),
 Text(0.8170731707317073, 0.07142857142857142, 'entropy = 1.394\nsamples =
34\nvalue = [13, 17, 4]'),
 Text(0.8414634146341463, 0.07142857142857142, 'entropy = 0.48\nsamples =
37\nvalue = [1, 34, 2]'),
 Text(0.8780487804878049, 0.21428571428571427, 'x[4] <= -1.246\nentropy =
0.456\nsamples = 927\nvalue = [78, 845, 4]'),
 Text(0.8658536585365854, 0.07142857142857142, 'entropy = 0.811\nsamples =
12\nvalue = [0, 9, 3]'),
 Text(0.8902439024390244, 0.07142857142857142, 'entropy = 0.433\nsamples =
915\nvalue = [78, 836, 1]'),
 Text(0.9512195121951219, 0.35714285714285715, 'x[4] <= -1.563\nentropy =
0.161\nsamples = 1181\nvalue = [21, 1156, 4]'),
 Text(0.926829268292683, 0.21428571428571427, 'x[5] <= 0.512\nentropy =
1.0\nsamples = 2\nvalue = [1, 0, 1]'),
 Text(0.9146341463414634, 0.07142857142857142, 'entropy = 0.0\nsamples =
1\nvalue = [0, 0, 1]'),
 Text(0.9390243902439024, 0.07142857142857142, 'entropy = 0.0\nsamples =
1\nvalue = [1, 0, 0]'),
 Text(0.975609756097561, 0.21428571428571427, 'x[5] <= 0.486\nentropy =
0.15\nsamples = 1179\nvalue = [20, 1156, 3]'),
 Text(0.9634146341463414, 0.07142857142857142, 'entropy = 0.145\nsamples =
1178\nvalue = [19, 1156, 3]'),
 Text(0.9878048780487805, 0.07142857142857142, 'entropy = 0.0\nsamples =
1\nvalue = [1, 0, 0]')]
```



Model evaluation

[45]:
```
print(f"The accuracy score for Decision Trees is {accuracy_score(test_y,yhat)}")
print(f"The Jaccard score for Decision Trees is␣
  ↪{jaccard_score(test_y,yhat,average='micro')}")
print(classification_report(test_y,yhat))
accuracies.append(accuracy_score(test_y,yhat))
```

```
The accuracy score for Decision Trees is 0.8615179760319573
The Jaccard score for Decision Trees is 0.7567251461988304
              precision    recall  f1-score   support

           L       0.53      0.55      0.54        95
           M       0.93      0.94      0.94       628
           S       0.14      0.07      0.10        28

    accuracy                           0.86       751
   macro avg       0.53      0.52      0.52       751
weighted avg       0.85      0.86      0.85       751
```

We see now that we have improved the accuracy score from 0.84 to 0.86, at the end we will do a comparaison of evaluations to choose the optimal model

Logistic Regression Training

[46]:
```
from sklearn.linear_model import LogisticRegression

lr = LogisticRegression(C=0.01,solver='liblinear')

lr.fit(train_x,train_y)
```

[46]:
```
LogisticRegression(C=0.01, solver='liblinear')
```

Prediction

[47]:
```
yhat = lr.predict(test_x)
yhat[:5]
```

[47]:
```
array(['M', 'M', 'M', 'M', 'L'], dtype=object)
```

We can also see the probability of these predictions

[48]:
```
yhat_proba = lr.predict_proba(test_x)
yhat_proba[:5]
```

[48]:
```
array([[0.15253828, 0.76517686, 0.08228487],
       [0.17032773, 0.74279451, 0.08687775],
       [0.08069931, 0.86720341, 0.05209728],
       [0.28849756, 0.46628218, 0.24522026],
       [0.48883585, 0.38658212, 0.12458203]])
```

Model evaluation

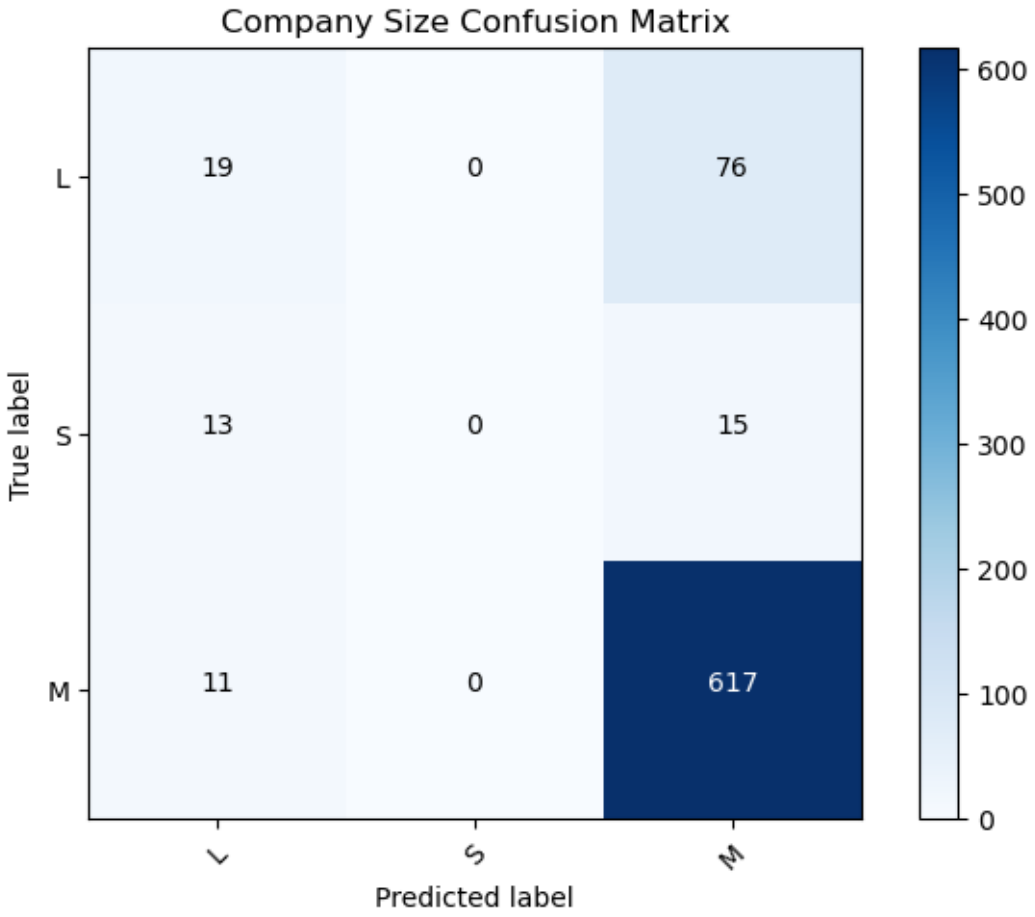
[49]:
```
print(f"The accuracy score for Decision Trees is {accuracy_score(test_y,yhat)}")
print(f"The Jaccard score for Decision Trees is␣
  ↪{jaccard_score(test_y,yhat,average='micro')}")
print(classification_report(test_y,yhat))
accuracies.append(accuracy_score(test_y,yhat))
```

```
The accuracy score for Decision Trees is 0.8468708388814914
The Jaccard score for Decision Trees is 0.7344110854503464
              precision    recall  f1-score   support

           L       0.44      0.20      0.28        95
           M       0.87      0.98      0.92       628
           S       0.00      0.00      0.00        28

    accuracy                           0.85       751
   macro avg       0.44      0.39      0.40       751
weighted avg       0.78      0.85      0.81       751
```

[50]:
```
cnf_matrix = confusion_matrix(test_y,yhat,labels = df['company_size'].unique())
```

```
plot_confusion_matrix(cnf_matrix,df['company_size'].unique(),title="Company␣
 ↪Size Confusion Matrix")
```

```
Confusion matrix, without normalization
[[ 19   0  76]
 [ 13   0  15]
 [ 11   0 617]]
```



We see that this model got all of the S business wrong, but he got most of the Medium companies correct

1.4.5 SVM

Now, we're going to try and work with an SVM model with 4 parameters (linear, rbf, polynomial and sigmoid)

[51]:
```
from sklearn import svm

sv_model = svm.SVC(kernel='rbf')
```

Training model

[52]:
```
sv_model.fit(train_x,train_y)
```

[52]:
```
SVC()
```

Predictions

[53]:
```
yhat = sv_model.predict(test_x)

yhat[:5]
```

[53]:
```
array(['M', 'M', 'M', 'M', 'L'], dtype=object)
```

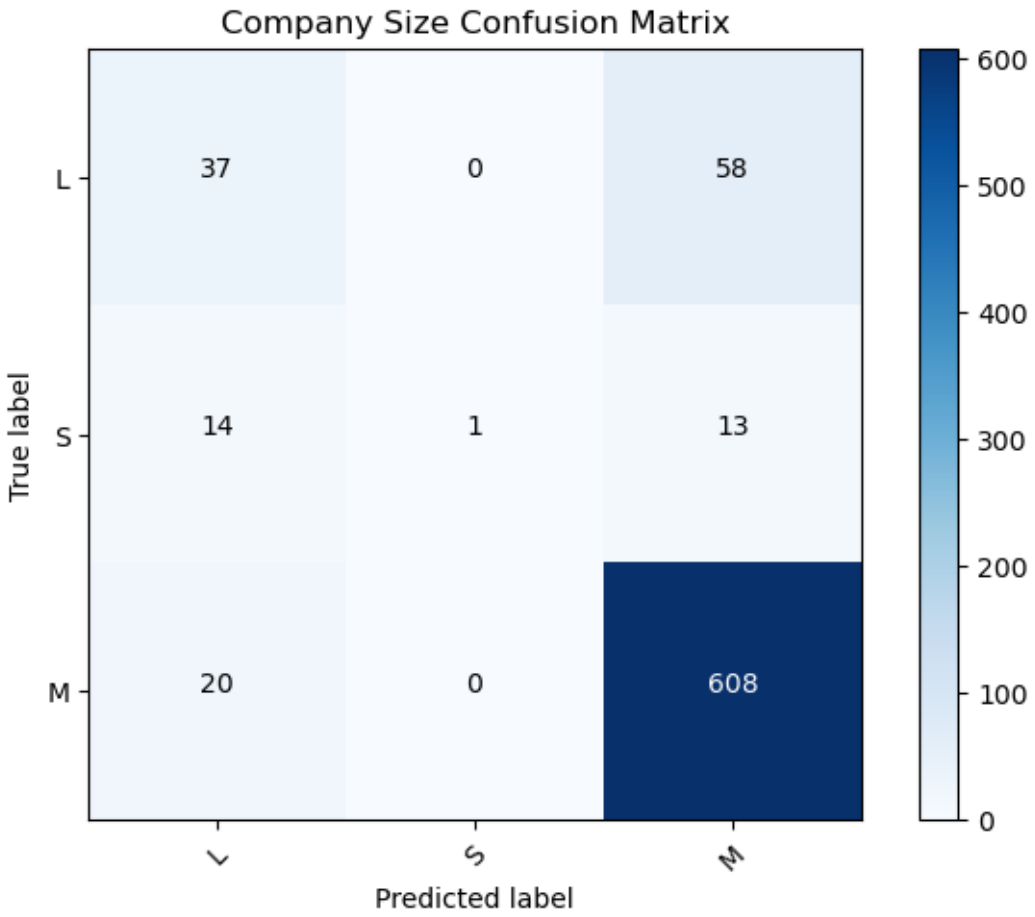
Model evaluation

[54]:
```
print(f"The accuracy score for SVM is {accuracy_score(test_y,yhat)}")
print(f"The Jaccard score for SVM is␣
  ↪{jaccard_score(test_y,yhat,average='micro')}")
print(classification_report(test_y,yhat))
accuracies.append(accuracy_score(test_y,yhat))
```

```
The accuracy score for SVM is 0.8601864181091877
The Jaccard score for SVM is 0.7546728971962616
              precision    recall  f1-score   support

           L       0.52      0.39      0.45        95
           M       0.90      0.97      0.93       628
           S       1.00      0.04      0.07        28

    accuracy                           0.86       751
   macro avg       0.81      0.46      0.48       751
weighted avg       0.85      0.86      0.84       751
```

[55]:
```
cnf_matrix = confusion_matrix(test_y,yhat,labels = df['company_size'].unique())

plot_confusion_matrix(cnf_matrix,df['company_size'].unique(),title="Company␣
  ↪Size Confusion Matrix")
```

```
Confusion matrix, without normalization
[[ 37   0  58]
 [ 14   1  13]
 [ 20   0 608]]
```



After using the SVM, we can see that we got an accuracy of 0.86 using the rbf which was the most optimal in our case, after comparing It to other kernels

1.5 Model Comparaison

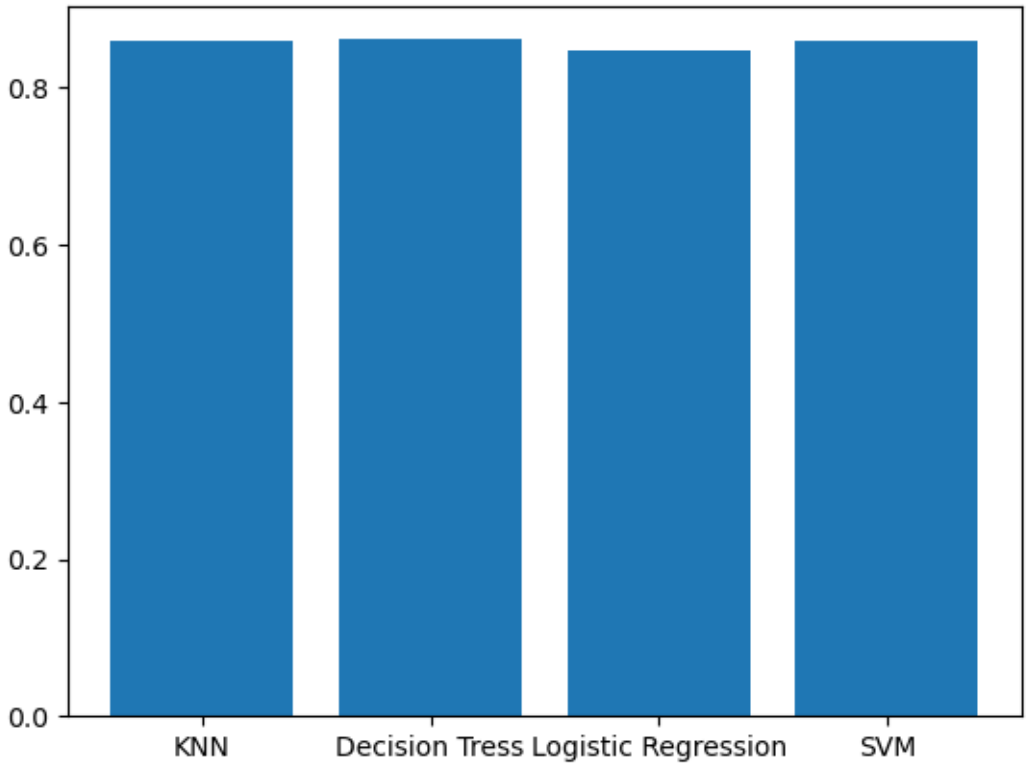
In this section, we're going to compare the accuracy that we got from all of the models

[56]:
```
accuracies
```

[56]:
```
[0.8601864181091877,
 0.8615179760319573,
 0.8468708388814914,
 0.8601864181091877]
```

[57]:
```
models = ['KNN','Decision Tress','Logistic Regression','SVM']

plt.bar(models,accuracies)
```

[57]:
```
<BarContainer object of 4 artists>
```



We can see that all of the models are close to each other in the matter of the accuracy, but we can see that the model with the most accuracy are SVM and KNN with an accuracy of **0.8601864181091877**

1.6 Using cross-validation

[58]:
```
from sklearn.model_selection import cross_val_score
from sklearn.model_selection import KFold
```

[84]:
```
kfold = KFold(n_splits=3)

scores = cross_val_score(lr,X,Y,cv=kfold,scoring='accuracy')
# Print the mean and standard deviation of the scores
print("\n===== Logistic Regression =====")
print(f"Mean accuracy: {scores.mean()}")
print(f"Standard deviation: {scores.std()}")

scores = cross_val_score(sv_model,X,Y,cv=kfold,scoring='accuracy')
# Print the mean and standard deviation of the scores
print("\n===== SVM =====")
print(f"Mean accuracy: {scores.mean()}")
print(f"Standard deviation: {scores.std()}")
```

```
scores = cross_val_score(dt,X,Y,cv=kfold,scoring='accuracy')
# Print the mean and standard deviation of the scores
print("\n===== Decision Trees =====")
print(f"Mean accuracy: {scores.mean()}")
print(f"Standard deviation: {scores.std()}")

scores = cross_val_score(knc,X,Y,cv=kfold,scoring='accuracy')
# Print the mean and standard deviation of the scores
print("\n===== KNN =====")
print(f"Mean accuracy: {scores.mean()}")
print(f"Standard deviation: {scores.std()}")
```

```
===== Logistic Regression =====
Mean accuracy: 0.8417655651836359
Standard deviation: 0.12047562539652273

===== SVM =====
Mean accuracy: 0.842830315087653
Standard deviation: 0.12178400560948695

===== Decision Trees =====
Mean accuracy: 0.8361774903825608
Standard deviation: 0.1088393026964943

===== KNN =====
Mean accuracy: 0.8359061419660864
Standard deviation: 0.12229744692622098
```