Es. 11 (dal Dataset all'algoritmo)

April 3, 2024

1 DAL DATASET ALL'ALGORITMO: COME SI SVILLUPA UN MODELLO?

In questa esercitazione viene mostrato come da un semplice Dataset (in questo caso scaricato da Internet dalla community di Kaggle, https://www.kaggle.com/?utm_source=homescreen) si riesce a sviluppare un modello (quindi un algoritmo) per prevedere una o più variabili target. Per fare questo bisogna prima però eseguire dei passaggi preliminari che son fondamentali per la cura e la precisione del modello finale (come ad esempio quelli di gestire i NaN e gli Outliers)

1.1 FASE 1: SCEGLIERE (O CREARE), IMPORTARE E SALVARE IL DATASET

1) SCARICARE IL DATASET E INSERIRLO IN UN PATH (PER COMODITA LO METTO NELLA STESSA CARTELLA)
2) IMPORTARE LE LIBRERIE NECESSARIE: PANDAS (PER LEGGERE IL DATASET) E OS (PER GESTIRE I PATH)
3) IMPORTARE IL DATASET USANDO LE FUNZIONI DI PANDAS

[1]:
```
import pandas  as pd # Importare la libreria "Pandas" per poter gestire i␣
 ↪Dataset
import os # Importare la libreria "os" per gestire i path

# Per importare il Dataset possiamo usare due funzione di Pandas:
# 1) pd.read_csv(): per leggere il file CSV (comma separated values)
# 2) pd.read_excel(): per leggere i file Excel

path_dataset = r"C:\Users\matte\OneDrive - Scuola Paritaria S. Freud␣
 ↪SRL\Desktop\FREUD\2D\QUADERNI E ALTRO\ROBOTICA ED AI\ESERCIZI IN CLASSE␣
 ↪PYTHON\ds_salaries.csv" # Il prefisso "r" serve per evitare che ci siano␣
 ↪confusioni nell'interpretazione della stringa, come ad esempio: numeri,␣
 ↪caratteri speciali e backslash
dataset = pd.read_csv(path_dataset)
```

1.2 FASE 2: VISUALIZZAZIONE E ANALISI DEL DATASET (CON GRAFICI)

1) STAMPARE IL DATASET

2) PER OGNI FEATURE ANALIZZARE COME SIA COMPOSTA: CIOè CHE VALORI HA NEL DETTAGLIO (TIPO UNITà DI MISURA O VALUTE)
3) ANALIZZARE COSA SIA MEGLIO TENERE O COSA INVECE è MEGLIO BUTTARE

Esperienza lavorativa:

. SE = Senior . MI = Mid-level . EN = Entry-level

Tipo di impiego:

. FT = Full-time . CT = Contract

[2]:
```
dataset # Stampare il Dataset serve per poterlo analizzare nel dettaglio␣
 ↪meglio, come ad esempio visualizzare le Feature e le istanze per decidere␣
 ↪cose sia meglio tenere e cosa invece sia meglio eliminare
# Scrivendo solo il nome del dataset, quest'ultimo si stamperà (solo la parte␣
 ↪iniziale e finale)
```

[2]:
```
      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]
```

[3]:
```
# Stampare i valori unici (unique), nonchè tutti i possibili output per ogni␣
 ↪Feature, serve per analizzare meglio il Dataset nel dettaglio di ogni␣
 ↪Feature e capire così tutti i possibili ambiti
print("I valori di work_year sono:") # All'inizio viene stampata una stringa di␣
 ↪testo esplicativa
print(dataset["work_year"].unique()) # Poi si stampano i veri e propri valori␣
 ↪unici
print("I valori di experience_level sono:")
print(dataset["experience_level"].unique())
print("I valori di employment_type sono:")
print(dataset["employment_type"].unique())
print("I valori di job_title sono:")
print(dataset["job_title"].unique())
print("I valori di salary sono:")
print(dataset["salary"].unique())
print("I valori di salary_currency sono:")
print(dataset["salary_currency"].unique())
print("I valori di salary_in_usd sono:")
print(dataset["salary_in_usd"].unique())
print("I valori di employee_residence sono:")
print(dataset["employee_residence"].unique())
print("I valori di remote_ratio sono:")
print(dataset["remote_ratio"].unique())
print("I valori di company_location sono:")
print(dataset["company_location"].unique())
print("I valori di company_size sono:")
print(dataset["company_size"].unique())
```

```
I valori di work_year sono:
[2023 2022 2020 2021]
I valori di experience_level sono:
['SE' 'MI' 'EN' 'EX']
I valori di employment_type sono:
['FT' 'CT' 'FL' 'PT']
I valori di job_title sono:
['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']
I valori di salary sono:
[   80000    30000    25500   175000   120000   222200   136000   219000
   141000   147100    90700   130000   100000   213660   130760   170000
   150000   110000   275000   174000   230000   143200   225000   156400
   200000    90000    72000   253200   342810   184590   162500   105380
    64500  1650000   204620   110680   270703   221484   212750   185000
   262000   245000   275300   183500   218500   199098   203300   123600
   189110   139000   258750   231500   166000   172500   110500   238000
   176000   237000   201450   309400   159100   115000    81500   280000
   210000   280100   168100   193500   510000    65000   300000   185900
   129300   140000    45000    36000   105000    70000   163196   145885
   217000   202800   104300   145000   165000   132300   179170    94300
   152500   116450   247300   133800   203000   133000   220000    54000
   289800   214000   179820   143860   283200   188800   214200   252000
   129000   155000   161800   141600   342300   176100    85000   138784
```

83270	75000	204500	138900	318300	212200	95000	195000
160000	1700000	38000	35000	168400	105200	190000	241000
55000	15000	47500	250000	228000	186000	180000	50000
205000	215000	247500	172200	224000	1400000	128000	329500
269600	203500	152000	239000	122900	191765	134236	112000
84000	135000	105500	293000	148500	240500	123700	152900
117100	173000	113000	260000	184000	149500	127075	219535
146115	199000	162000	221000	153000	187000	179000	109000
142000	198800	125000	86000	106000	280700	150450	250500
159500	130001	71907	93918	51962	257000	147000	222000
133200	156000	304000	161200	84570	240000	183600	289076
202353	157750	104650	68000	60000	181000	154000	146000
64200	56100	208450	170550	171250	113750	153600	100500
182500	121500	203100	114500	92700	61800	258000	167500
106500	57000	286000	207000	223250	178600	353200	249300
297300	198200	151800	317070	170730	20000	108000	134000
124000	124500	148700	125600	120250	183000	1500000	216000
143865	115092	132000	208049	128500	149600	102000	106800
151000	7000	40000	143000	42000	111000	265000	235000
60400	164000	56000	83500	52500	201036	134024	62000
58000	172000	163800	126000	139500	109400	205600	105700
239748	159832	186300	102500	149040	113900	172600	107900
180180	106020	376080	213120	206500	121600	194500	115500
115934	81666	206000	138000	92000	48000	87000	299500
245100	115100	73900	141288	94192	210914	116704	185700
169000	110600	193000	136850	276000	178500	161000	83300
112700	128750	106250	188500	117000	104500	127000	94000
210550	153300	161500	119500	148750	146300	153400	122700
123900	340000	121700	310000	149076	82365	85500	97750
201000	122000	116990	82920	142200	205920	171600	78000
116000	36050	34320	93800	67000	1300000	1000000	104000
152380	121904	128280	106900	192000	170500	60027	44737
131899	104891	124740	65488	72200	64980	179975	86466
168000	167580	87980	202000	148000	269000	158000	197000
290000	172800	300240	200160	370000	137500	323300	184700
153088	183310	144000	66000	126277	126500	272000	259000
101400	288000	215050	198000	114000	209300	182200	227000
52000	226700	133300	124999	800000	63000	253750	169200
213580	163625	12000	375000	1350000	231250	138750	284310
153090	225900	385000	93919	241871	133832	192500	216100
140800	284000	236000	248100	145900	155850	102544	151410
115360	1050000	25000	107000	23000	182750	314100	195800
350000	262500	209450	158677	103200	61200	59000	174500
107250	119000	285800	154600	5000000	124234	74540	79000
141290	74178	107500	1060000	6000	1440000	840000	1250000
182000	234100	223800	172100	232200	167200	291500	196200
150900	167000	96100	196000	126100	187500	24000	165750
89700	55250	175308	100706	229000	4000000	272550	64000

```
   143100   180560   115440  1125000   261500   134500  1100000    94500
   127500    51000   248400  4460000   149000   246000    10000  2500000
  2800000   249500   149850   122500   102640    66100   122600   159000
   255000   166700   194000   129400    89200   178750   197430   134760
    99000   105120    75360   171000    13000   213000   227200    61000
   243000   178000    96000   137000   189750   140250   191200   179500
    26000   118000   177000   131000   193750   116250   208000    45555
  6600000   140700    33000   154560   123648   177500   192564   144854
   179305   142127   315000   243900   156600    77300    45600   184100
   198440    47000   187200   116100   159699   138938    76000   125404
   123000    92250    97000   157000   345600   230400   175950   130050
   236600    27000   400000     8000   123400    88100   139600    85700
    98200    98000   144200  3000000   188700   160395   191475   141525
   156868   178800   132100   229998   154545    99750    68400   236900
   159200   243225   179775   218000   145300   195400   131300   195700
   130500   141300   102100    83000  1800000   633000   179400   193900
   222640   182160   297500    93000    73000    40300   136994   101570
    97500   212800   142800   500000   130240    83376    65004    84958
    66822    81000    46000   204100   136100     7500    77000    28500
   119300   146200   124270   185800   137400   148800  7500000    82000
    32400   216200   144100   175100   189650   164996    99450   188100
   139860   248700   167100   450000   189500   140100   177600   202900
   900000  4200000   260500    73400    49500  2400000   206699    99100
   221300    74000   249260   185400   128875    93700   136260   109280
   150075   110925    22800   112900    90320    62500   105400    43200
   215300   158200   209100   165400   132320   208775   147800  6000000
   100800   140400    82900    63900   112300   108800   242000   165220
   120160   124190   181940   220110   160080   106260   120600    84900
   136620    99360   161342   137141   211500   138600   192400    61300
    95550   136600   167875   205300   200100    70500   116150    99050
   192600   266400   150260    69000   324000   185100   104890    53000
    88000    66500   121000    29000    69999    52800   405000   380000
  8500000  7000000    38400    82500   700000     8760    51999    41000
    13400   103000   270000    45760    44000  2250000    37456 11000000
    14000  2200000   188000  2100000    51400    61500   720000    31000
    91000  1600000   256000    72500    65720   111775    93150    21600
  4900000  1200000    21000  1799997     9272   120500    21844    22000
    76760  1672000   420000 30400000    32000   416000    40900  4450000
   423000   325000    34000    69600   435000    37000    19000    18000
    39600  1335000  1450000   190200   138350   130800   412000]
I valori di salary_currency sono:
['EUR' 'USD' 'INR' 'HKD' 'CHF' 'GBP' 'AUD' 'SGD' 'CAD' 'ILS' 'BRL' 'THB'
 'PLN' 'HUF' 'CZK' 'DKK' 'JPY' 'MXN' 'TRY' 'CLP']
I valori di salary_in_usd sono:
[ 85847  30000  25500 …  28369 412000  94665]
I valori di employee_residence sono:
['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']
I valori di remote_ratio sono:
[100   0  50]
I valori di company_location sono:
['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']
I valori di company_size sono:
['L' 'S' 'M']
```

1.3 FASE 3: MODIFICA DEL DATASET

1) VOGLIAMO MODIFICARE IL DATASET CONSIDERANDO SOLO TRE FEATURES E CON TUTTI I SALARI IN DOLLARI
2) ELIMINARE LE FEATURE INUTILI AL NOSTRO ALGORITMO FINALE
3) SALVARE SOVRASCRIVENDO IL DATASET
4) STAMPARE IL NUOVO DATASET PER VERIFICARE SE LE OPERAZIONE FATTE PRECEDENENTE HANNO AVUTO UN SEGUITO POSITIVO

TUTTE LE MODIFICHE VENGONO FATTE SU UN DATASET CLONE, IN MODO POI DA POTERLO COMPARARE CON L'ORIGINALE

[4]:
```
job_titles = ['Data Scientist', 'Machine Learning Engineer', 'Data Analyst',␣
 ↪'Data Engineer', 'Data Architect', 'Business Intelligence Engineer', 'Data␣
 ↪Strategist', 'Data Quality Analyst', 'Data Science Manager', 'Data␣
 ↪Operations Engineer']
print(len(job_titles))
dataset_ridotto = dataset[dataset['job_title'].isin(job_titles)]
dataset_ridotto["job_title"].unique() # Controllare che l'unico valore
```

```
10
```

[4]:
```
array(['Data Scientist', 'Data Analyst', 'Business Intelligence Engineer',
       'Machine Learning Engineer', 'Data Strategist', 'Data Engineer',
       'Data Quality Analyst', 'Data Architect', 'Data Science Manager',
       'Data Operations Engineer'], dtype=object)
```

[5]:
```
print("I valori di job_title sono:")
print(dataset_ridotto["job_title"].unique())
```

```
I valori di job_title sono:
['Data Scientist' 'Data Analyst' 'Business Intelligence Engineer'
 'Machine Learning Engineer' 'Data Strategist' 'Data Engineer'
```

```
 'Data Quality Analyst' 'Data Architect' 'Data Science Manager'
 'Data Operations Engineer']
```

[6]:
```
dataset_ridotto=dataset[dataset["salary_currency"] == "USD"] # Filtrare le␣
 ↪righe (istanze) del dataset in cui i valori di salary currency è "USD"
dataset_ridotto["salary_currency"].unique() # Controllare che l'unico valore in␣
 ↪salary currency sia "USD"
```

[6]:
```
array(['USD'], dtype=object)
```

[7]:
```
print("I valori di salary_currency sono:")
print(dataset_ridotto["salary_currency"].unique())
```

```
I valori di salary_currency sono:
['USD']
```

[8]:
```
dataset_ridotto=dataset[dataset["company_location"] == "US"] # Filtrare le␣
 ↪righe (istanze) del dataset in cui i valori di company location è "US"
dataset_ridotto["company_location"].unique() # Controllare che l'unico valore␣
 ↪in company location sia "US"
```

[8]:
```
array(['US'], dtype=object)
```

[9]:
```
print("I valori di company_location sono:")
print(dataset_ridotto["company_location"].unique())
```

```
I valori di company_location sono:
['US']
```

[10]:
```
dataset_ridotto=dataset_ridotto[dataset_ridotto["work_year"] == 2023] #␣
 ↪Filtrare le righe (istanze) del dataset
dataset_ridotto["work_year"].unique() # Controllare che l'unico valore
```

[10]:
```
array([2023], dtype=int64)
```

[11]:
```
print("I valori di work_year sono:")
print(dataset_ridotto["work_year"].unique())
```

```
I valori di work_year sono:
[2023]
```

[12]:
```
dataset=dataset[dataset["work_year"] == 2023] # Filtrare le righe (istanze) del␣
 ↪dataset in cui i valori di work_year è "2023"
dataset["work_year"].unique() # Controllare che l'unico valore in work_year è␣
 ↪"2023"
```

[12]:
```
array([2023], dtype=int64)
```

[13]:
```
print("I valori di work_year sono:")
print(dataset["work_year"].unique())
```

```
I valori di work_year sono:
[2023]
```

[14]:
```
dataset_ridotto =␣
↪dataset_ridotto[["experience_level","job_title","salary","company_location"]]␣
↪# Filtrare solo le features scelte e il target (salary). Le altre features␣
↪non scritte veranno eliminate
dataset_ridotto
```

[14]:

	experience_level	job_title	salary	company_location
1	MI	ML Engineer	30000	US
2	MI	ML Engineer	25500	US
5	SE	Applied Scientist	222200	US
6	SE	Applied Scientist	136000	US
9	SE	Data Scientist	147100	US
…	…	…	…	…
1815	SE	Machine Learning Engineer	134500	US
1817	MI	Data Scientist	130000	US
1818	MI	Data Scientist	90000	US
1819	EN	Data Engineer	160000	US
1820	EN	Data Engineer	135000	US

[1570 rows x 4 columns]

[15]:
```
dataset = dataset[["experience_level","job_title","salary","company_location"]]␣
↪# Filtrare solo le features scelte e il target (salary). Le altre features␣
↪non scritte veranno eliminate
dataset
```

[15]:

	experience_level	job_title	salary	company_location
0	SE	Principal Data Scientist	80000	ES
1	MI	ML Engineer	30000	US
2	MI	ML Engineer	25500	US
3	SE	Data Scientist	175000	CA
4	SE	Data Scientist	120000	CA
…	…	…	…	…
1815	SE	Machine Learning Engineer	134500	US
1817	MI	Data Scientist	130000	US
1818	MI	Data Scientist	90000	US
1819	EN	Data Engineer	160000	US
1820	EN	Data Engineer	135000	US

[1785 rows x 4 columns]

1.4 FASE 4: LE DISTRIBUZIONI E I GRAFICI SULLE MODIFICHE DEL DATASET RISPETTO AL DATASET ORIGINALE

1) CONFRONTIAMO LE DISTRIBUZIONE DEI TITOLI DI LAVORI "MONDIALE" VS CON QUELLA AMERICANA

[16]:
```
from matplotlib import pyplot as plt

persone_totali = len(dataset)

# Calcolare le percentuali dei titoli di lavoro mondiali rispetto ad una␣
 ↪singola categoria di lavoro

# Calcolare percentuali di "Data Scientist" mondiali

DataScientist_mondiali = dataset[dataset["job_title"]=="Data Scientist"]
numero_DataScientist_mondiali = len(DataScientist_mondiali)
percentuale_DataScientist_mondiali = numero_DataScientist_mondiali/
 ↪persone_totali*100

# Calcolare percentuali di "Machine Learning Engineer" mondiali

Machine_Learning_Engineer_mondiali = dataset[dataset["job_title"]=="Machine␣
 ↪Learning Engineer"]
numero_Machine_Learning_Engineer_mondiali =␣
 ↪len(Machine_Learning_Engineer_mondiali)
percentuale_Machine_Learning_Engineer_mondiali =␣
 ↪numero_Machine_Learning_Engineer_mondiali/persone_totali*100

# Calcolare percentuali di "Data Analyst" mondiali

Data_Analyst_mondiali = dataset[dataset["job_title"]=="Data Analyst"]
numero_Data_Analyst_mondiali = len(Data_Analyst_mondiali)
percentuale_Data_Analyst_mondiali = numero_Data_Analyst_mondiali/
 ↪persone_totali*100

# Calcolare percentuali di "Data Engineer" mondiali

Data_Engineer_mondiali = dataset[dataset["job_title"]=="Data Engineer"]
numero_Data_Engineer_mondiali = len(Data_Engineer_mondiali)
percentuale_Data_Engineer_mondiali = numero_Data_Engineer_mondiali/
 ↪persone_totali*100

# Calcolare percentuali di "Data Architect" mondiali

Data_Architect_mondiali = dataset[dataset["job_title"]=="Data Architect"]
numero_Data_Architect_mondiali = len(Data_Architect_mondiali)
```

```
percentuale_Data_Architect_mondiali = numero_Data_Architect_mondiali/
 ↪persone_totali*100

# Calcolare percentuali di "Business Intelligence Engineer" mondiali

Business_Intelligence_Engineer_mondiali =␣
 ↪dataset[dataset["job_title"]=="Business Intelligence Engineer"]
numero_Business_Intelligence_Engineer_mondiali =␣
 ↪len(Business_Intelligence_Engineer_mondiali)
percentuale_Business_Intelligence_Engineer_mondiali =␣
 ↪numero_Business_Intelligence_Engineer_mondiali/persone_totali*100

# Calcolare percentuali di "Data Strategist" mondiali

Data_Strategist_mondiali = dataset[dataset["job_title"]=="Data Strategist"]
numero_Data_Strategist_mondiali = len(Data_Strategist_mondiali)
percentuale_Data_Strategist_mondiali = numero_Data_Strategist_mondiali/
 ↪persone_totali*100

# Calcolare percentuali di "Data Quality Analyst" mondiali

Data_Quality_Analyst_mondiali = dataset[dataset["job_title"]=="Data Quality␣
 ↪Analyst"]
numero_Data_Quality_Analyst_mondiali = len(Data_Quality_Analyst_mondiali)
percentuale_Data_Quality_Analyst_mondiali =␣
 ↪numero_Data_Quality_Analyst_mondiali/persone_totali*100

# Calcolare percentuali di "Data Science Manager" mondiali

Data_Science_Manager_mondiali = dataset[dataset["job_title"]=="Data Science␣
 ↪Manager"]
numero_Data_Science_Manager_mondiali = len(Data_Science_Manager_mondiali)
percentuale_Data_Science_Manager_mondiali =␣
 ↪numero_Data_Science_Manager_mondiali/persone_totali*100

# Calcolare percentuali di "Data Operations Engineer" mondiali

Data_Operations_Engineer_mondiali = dataset[dataset["job_title"]=="Data␣
 ↪Operations Engineer"]
numero_Data_Operations_Engineer_mondiali =␣
 ↪len(Data_Operations_Engineer_mondiali)
percentuale_Data_Operations_Engineer_mondiali =␣
 ↪numero_Data_Operations_Engineer_mondiali/persone_totali*100
```

[17]:
```
from matplotlib import pyplot as plt
```

```
persone_totali = len(dataset_ridotto)

# Calcolare le percentuali dei titoli di lavoro americani rispetto ad una␣
 ↪singola categoria di lavoro

# Calcolare percentuali di "Data Scientist" americani

DataScientist_americani = dataset_ridotto[dataset_ridotto["job_title"]=="Data␣
 ↪Scientist"]
numero_DataScientist_americani = len(DataScientist_americani)
percentuale_DataScientist_americani = numero_DataScientist_americani/
 ↪persone_totali*100

# Calcolare percentuali di "Machine Learning Engineer" americani

Machine_Learning_Engineer_americani =␣
 ↪dataset_ridotto[dataset_ridotto["job_title"]=="Machine Learning Engineer"]
numero_Machine_Learning_Engineer_americani =␣
 ↪len(Machine_Learning_Engineer_americani)
percentuale_Machine_Learning_Engineer_americani =␣
 ↪numero_Machine_Learning_Engineer_americani/persone_totali*100

# Calcolare percentuali di "Data Analyst" americani

Data_Analyst_americani = dataset_ridotto[dataset_ridotto["job_title"]=="Data␣
 ↪Analyst"]
numero_Data_Analyst_americani = len(Data_Analyst_americani)
percentuale_Data_Analyst_americani = numero_Data_Analyst_americani/
 ↪persone_totali*100

# Calcolare percentuali di "Data Engineer" americani

Data_Engineer_americani = dataset_ridotto[dataset_ridotto["job_title"]=="Data␣
 ↪Engineer"]
numero_Data_Engineer_americani = len(Data_Engineer_americani)
percentuale_Data_Engineer_americani = numero_Data_Engineer_americani/
 ↪persone_totali*100

# Calcolare percentuali di "Data Architect" americani

Data_Architect_americani = dataset_ridotto[dataset_ridotto["job_title"]=="Data␣
 ↪Architect"]
numero_Data_Architect_americani = len(Data_Architect_americani)
percentuale_Data_Architect_americani = numero_Data_Architect_americani/
 ↪persone_totali*100
```

```
# Calcolare percentuali di "Business Intelligence Engineer" americani

Business_Intelligence_Engineer_americani =␣
 ↪dataset_ridotto[dataset_ridotto["job_title"]=="Business Intelligence␣
 ↪Engineer"]
numero_Business_Intelligence_Engineer_americani =␣
 ↪len(Business_Intelligence_Engineer_americani)
percentuale_Business_Intelligence_Engineer_americani =␣
 ↪numero_Business_Intelligence_Engineer_americani/persone_totali*100

# Calcolare percentuali di "Data Strategist" americani

Data_Strategist_americani = dataset_ridotto[dataset_ridotto["job_title"]=="Data␣
 ↪Strategist"]
numero_Data_Strategist_americani = len(Data_Strategist_americani)
percentuale_Data_Strategist_americani = numero_Data_Strategist_americani/
 ↪persone_totali*100

# Calcolare percentuali di "Data Quality Analyst" americani

Data_Quality_Analyst_americani =␣
 ↪dataset_ridotto[dataset_ridotto["job_title"]=="Data Quality Analyst"]
numero_Data_Quality_Analyst_americani = len(Data_Quality_Analyst_americani)
percentuale_Data_Quality_Analyst_americani =␣
 ↪numero_Data_Quality_Analyst_americani/persone_totali*100

# Calcolare percentuali di "Data Science Manager" americani

Data_Science_Manager_americani =␣
 ↪dataset_ridotto[dataset_ridotto["job_title"]=="Data Science Manager"]
numero_Data_Science_Manager_americani = len(Data_Science_Manager_americani)
percentuale_Data_Science_Manager_americani =␣
 ↪numero_Data_Science_Manager_americani/persone_totali*100

# Calcolare percentuali di "Data Operations Engineer" americani

Data_Operations_Engineer_americani =␣
 ↪dataset_ridotto[dataset_ridotto["job_title"]=="Data Operations Engineer"]
numero_Data_Operations_Engineer_americani =␣
 ↪len(Data_Operations_Engineer_americani)
percentuale_Data_Operations_Engineer_americani =␣
 ↪numero_Data_Operations_Engineer_americani/persone_totali*100
```

[18]:
```
print("Le percentuali mondiali di \"Data Scientist\" sono:")
print(percentuale_DataScientist_mondiali)
print("Le percentuali mondiali di \"Machine Learning Engineer\" sono:")
```

```
print(percentuale_Machine_Learning_Engineer_mondiali)
print("Le percentuali mondiali di \"Data Analyst\" sono:")
print(percentuale_Data_Analyst_mondiali)
print("Le percentuali mondiali di \"Data Engineer\" sono:")
print(percentuale_Data_Engineer_mondiali)
print("Le percentuali mondiali di \"Data Architect\" sono:")
print(percentuale_Data_Architect_mondiali)
print("Le percentuali mondiali di \"Business Intelligence Engineer\" sono:")
print(percentuale_Business_Intelligence_Engineer_mondiali)
print("Le percentuali mondiali di \"Data Strategist\" sono:")
print(percentuale_Data_Strategist_mondiali)
print("Le percentuali mondiali di \"Data Quality Analyst\" sono:")
print(percentuale_Data_Quality_Analyst_mondiali)
print("Le percentuali mondiali di \"Data Science Manager\" sono:")
print(percentuale_Data_Science_Manager_mondiali)
print("Le percentuali mondiali di \"Data Operations Engineer\" sono:")
print(percentuale_Data_Operations_Engineer_mondiali)
```

```
Le percentuali mondiali di "Data Scientist" sono:
20.72829131652661
Le percentuali mondiali di "Machine Learning Engineer" sono:
8.8515406162465
Le percentuali mondiali di "Data Analyst" sono:
17.198879551820728
Le percentuali mondiali di "Data Engineer" sono:
27.955182072829132
Le percentuali mondiali di "Data Architect" sono:
2.9131652661064424
Le percentuali mondiali di "Business Intelligence Engineer" sono:
0.22408963585434172
Le percentuali mondiali di "Data Strategist" sono:
0.11204481792717086
Le percentuali mondiali di "Data Quality Analyst" sono:
0.39215686274509803
Le percentuali mondiali di "Data Science Manager" sono:
1.2324929971988796
Le percentuali mondiali di "Data Operations Engineer" sono:
0.11204481792717086
```

[19]:
```
print("Le percentuali americane di \"Data Scientist\" sono:")
print(percentuale_DataScientist_americani)
print("Le percentuali americane di \"Machine Learning Engineer\" sono:")
print(percentuale_Machine_Learning_Engineer_americani)
print("Le percentuali americane di \"Data Analyst\" sono:")
print(percentuale_Data_Analyst_americani)
print("Le percentuali americane di \"Data Engineer\" sono:")
print(percentuale_Data_Engineer_americani)
```

```
print("Le percentuali americane di \"Data Architect\" sono:")
print(percentuale_Data_Architect_americani)
print("Le percentuali americane di \"Business Intelligence Engineer\" sono:")
print(percentuale_Business_Intelligence_Engineer_americani)
print("Le percentuali americane di \"Data Strategist\" sono:")
print(percentuale_Data_Strategist_americani)
print("Le percentuali americane di \"Data Quality Analyst\" sono:")
print(percentuale_Data_Quality_Analyst_americani)
print("Le percentuali americane di \"Data Science Manager\" sono:")
print(percentuale_Data_Science_Manager_americani)
print("Le percentuali americane di \"Data Operations Engineer\" sono:")
print(percentuale_Data_Operations_Engineer_americani)
```

```
Le percentuali americane di "Data Scientist" sono:
20.127388535031848
Le percentuali americane di "Machine Learning Engineer" sono:
8.535031847133757
Le percentuali americane di "Data Analyst" sono:
17.51592356687898
Le percentuali americane di "Data Engineer" sono:
30.063694267515924
Le percentuali americane di "Data Architect" sono:
3.1847133757961785
Le percentuali americane di "Business Intelligence Engineer" sono:
0.25477707006369427
Le percentuali americane di "Data Strategist" sono:
0.0
Le percentuali americane di "Data Quality Analyst" sono:
0.3821656050955414
Le percentuali americane di "Data Science Manager" sono:
1.2738853503184715
Le percentuali americane di "Data Operations Engineer" sono:
0.12738853503184713
```

[20]:
```
# con
percentuale_totale_mondiale = percentuale_Data_Analyst_mondiali +␣
  ↪percentuale_Data_Engineer_mondiali + percentuale_DataScientist_mondiali +␣
  ↪percentuale_Data_Architect_mondiali +␣
  ↪percentuale_Data_Quality_Analyst_mondiali +␣
  ↪percentuale_Data_Science_Manager_mondiali +␣
  ↪percentuale_Data_Operations_Engineer_mondiali +␣
  ↪percentuale_Machine_Learning_Engineer_mondiali +␣
  ↪percentuale_Business_Intelligence_Engineer_mondiali +␣
  ↪percentuale_Data_Strategist_mondiali
print(f"La percentuale totale mondiale è pari a:␣
  ↪{int(percentuale_totale_mondiale)}%")
```

```
La percentuale totale mondiale è pari a: 79%
```

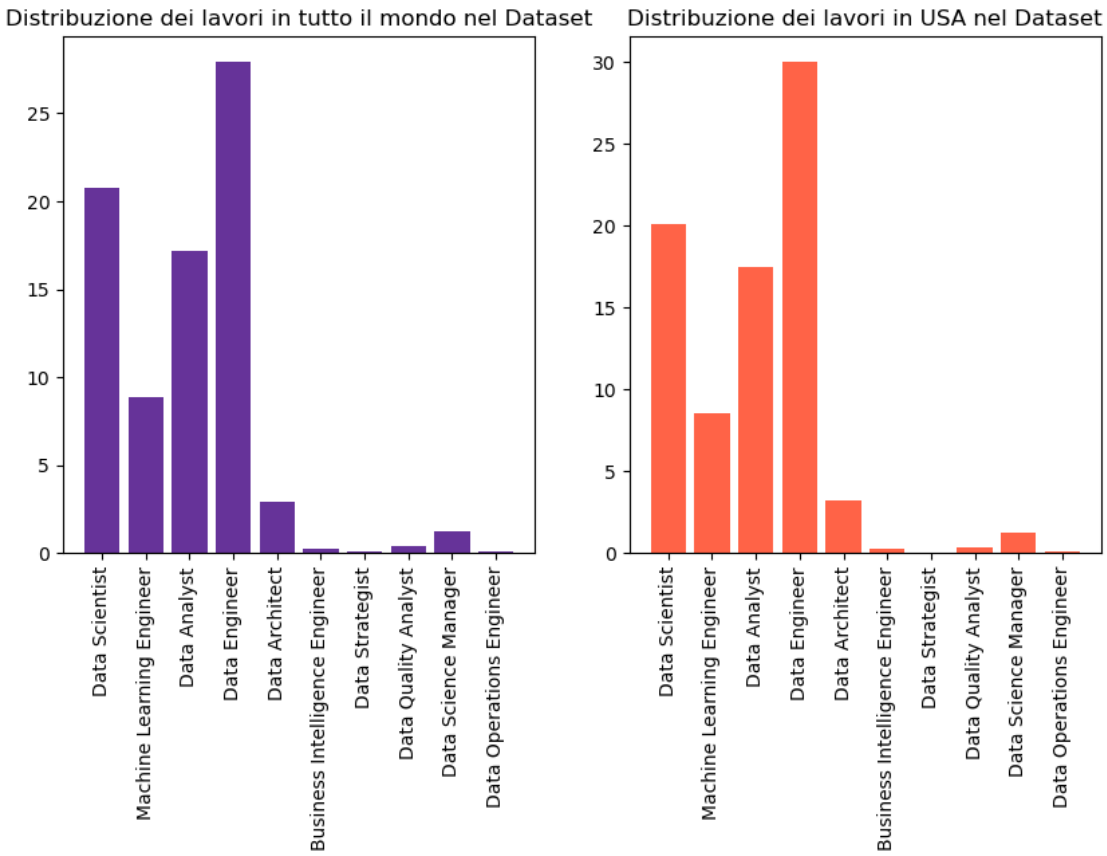
[21]:
```
# con
percentuale_totale_americana = percentuale_Data_Analyst_americani +␣
 ↪percentuale_Data_Engineer_americani + percentuale_DataScientist_americani +␣
 ↪percentuale_Data_Architect_americani +␣
 ↪percentuale_Data_Quality_Analyst_americani +␣
 ↪percentuale_Data_Science_Manager_americani +␣
 ↪percentuale_Data_Operations_Engineer_americani +␣
 ↪percentuale_Machine_Learning_Engineer_americani +␣
 ↪percentuale_Business_Intelligence_Engineer_americani +␣
 ↪percentuale_Data_Strategist_americani
print(f"La percentuale totale americana è pari a:␣
 ↪{int(percentuale_totale_americana)}%")
```

```
La percentuale totale americana è pari a: 81%
```

[22]:
```
labels = job_titles
percentuali_mondiali = [percentuale_DataScientist_mondiali,␣
 ↪percentuale_Machine_Learning_Engineer_mondiali,␣
 ↪percentuale_Data_Analyst_mondiali, percentuale_Data_Engineer_mondiali,␣
 ↪percentuale_Data_Architect_mondiali,␣
 ↪percentuale_Business_Intelligence_Engineer_mondiali,␣
 ↪percentuale_Data_Strategist_mondiali,␣
 ↪percentuale_Data_Quality_Analyst_mondiali,␣
 ↪percentuale_Data_Science_Manager_mondiali,␣
 ↪percentuale_Data_Operations_Engineer_mondiali]
print(len(percentuali_mondiali))
plt.figure()
fig,axs = plt.subplots(1,2,figsize=(10,5))
axs[0].set_title("Distribuzione dei lavori in tutto il mondo nel Dataset")
axs[0].bar(labels,percentuali_mondiali, color="rebeccapurple")
axs[0].tick_params(axis='x',rotation=90)

percentuali_americani = [percentuale_DataScientist_americani,␣
 ↪percentuale_Machine_Learning_Engineer_americani,␣
 ↪percentuale_Data_Analyst_americani, percentuale_Data_Engineer_americani,␣
 ↪percentuale_Data_Architect_americani,␣
 ↪percentuale_Business_Intelligence_Engineer_americani,␣
 ↪percentuale_Data_Strategist_americani,␣
 ↪percentuale_Data_Quality_Analyst_americani,␣
 ↪percentuale_Data_Science_Manager_americani,␣
 ↪percentuale_Data_Operations_Engineer_americani]

axs[1].set_title("Distribuzione dei lavori in USA nel Dataset")
axs[1].bar(labels,percentuali_americani, color="tomato")
axs[1].tick_params(axis='x',rotation=90)

plt.show()
```

```
10
```

```
<Figure size 640x480 with 0 Axes>
```



1.5 FASE 5: LE CORRELAZIONI TRA TUTTE LE FEATURES E IL SALARIO (CON LA MATRICE DI CORRELAZIONE)

[23]:
```
dataset.corr()
```

```
C:\Users\matte\AppData\Local\Temp\ipykernel_34736\2191645083.py:1:
FutureWarning: The default value of numeric_only in DataFrame.corr is
deprecated. In a future version, it will default to False. Select only valid
columns or specify the value of numeric_only to silence this warning.
  dataset.corr()
```

[23]:

	salary
salary	1.0

NON HO DELLE VARIABILI NUMERICHE PER FARE LA CORRELAZIONE, HO SOLO DEI VALORI NUMERICI PER LA FEATURE SALARY ED è PER QUESTO CHE NON MI VIENE LA MATRICE DI CORRELAZIONE

1.6 FASE 6: L'ANALISI DELLA PRESENZA DI NAN NEL DATASET, LA GESTIONE DI QUEST'ULTIMI ED EVENTUALI GRAFICI

[24]:
```
# Calcolo del totale delle righe con dati mancanti
totale_dati_mancanti_dataset = dataset.isnull().any(axis=1).sum() # Calcola il␣
 ↪totale delle righe con almeno un dato mancante

# Determinazione delle colonne con dati mancanti
colonne_dati_mancanti_dataset = dataset.isnull().any(axis=0) # True se almeno␣
 ↪un valore nella colonna è mancante (None o NaN)
```

[25]:
```
# Stampa delle colonne con dati mancanti e del totale dei dati mancanti
print("Colonne con dati mancanti nel Dataset originale:")
print(colonne_dati_mancanti_dataset)
print(f"Totale delle righe con dati mancanti nel Dataset originale:␣
 ↪{totale_dati_mancanti_dataset}")
```

```
Colonne con dati mancanti nel Dataset originale:
experience_level    False
job_title           False
salary              False
company_location    False
dtype: bool
Totale delle righe con dati mancanti nel Dataset originale: 0
```

[26]:
```
# Calcolo del totale delle righe con dati mancanti
totale_dati_mancanti_dataset_ridotto = dataset_ridotto.isnull().any(axis=1).
 ↪sum() # Calcola il totale delle righe con almeno un dato mancante

# Determinazione delle colonne con dati mancanti
colonne_dati_mancanti_dataset_ridotto = dataset_ridotto.isnull().any(axis=0) #␣
 ↪True se almeno un valore nella colonna è mancante (None o NaN)
```

[27]:
```
# Stampa delle colonne con dati mancanti e del totale dei dati mancanti
print("Colonne con dati mancanti nel Dataset ridotto:")
print(colonne_dati_mancanti_dataset_ridotto)
print(f"Totale delle righe con dati mancanti nel Dataset ridotto:␣
 ↪{totale_dati_mancanti_dataset_ridotto}")
```

```
Colonne con dati mancanti nel Dataset ridotto:
experience_level    False
job_title           False
salary              False
company_location    False
dtype: bool
Totale delle righe con dati mancanti nel Dataset ridotto: 0
```

1.7 FASE 7: L'ANALISI DELLA PRESENZA DI OUTLIERS NEL DATASET, LA GESTIONE DI QUEST'ULTIMI ED EVENTUALI GRAFICI

[28]:
```
# la formula è:   = √(Σ(xi - x̄)² / n)
    # √ = radice quadrata
    # Σ = sommatoria di tutti gli elementi dentro la parentesi quadra
    # xi = sono i singoli valori dei dati
    # x̄ = è la media dei dati
    # n = è il numero totale di dati
```

[29]:
```
# Calcolare la media del Dataset
mean_value_dataset = dataset["salary"].mean()
print("La media dei valori del Dataset originario è: ")
print(mean_value_dataset)
```

```
La media dei valori del Dataset originario è:
160381.4806722689
```

[30]:
```
# Calcolare la media del Dataset ridotto
mean_value_dataset_ridotto = dataset_ridotto["salary"].mean()
print("La media dei valori del Dataset ridotto è: ")
print(mean_value_dataset_ridotto)
```

```
La media dei valori del Dataset ridotto è:
156784.98089171975
```

[31]:
```
# Calcolare la deviazione standard del Dataset
std_dev_dataset = dataset["salary"].std()
print("La deviazione standard del Dataset originario è: ")
print(std_dev_dataset)
```

```
La deviazione standard del Dataset originario è:
162009.12823787233
```

[32]:
```
# Calcolare la deviazione standard del Dataset ridotto
std_dev_dataset_ridotto = dataset_ridotto["salary"].std()
print("La deviazione standard del Dataset ridotto è: ")
print(std_dev_dataset_ridotto)
```

```
La deviazione standard del Dataset ridotto è:
56862.76763170857
```

[33]:
```
#Identifica gli outliers consiederando +3 sigma dalla media
outliers_dataset=dataset[(dataset["salary"]>mean_value_dataset+3*std_dev_dataset)
 ↪| (dataset["salary"]<mean_value_dataset-3*std_dev_dataset)]
outliers_dataset
```

[33]:

	experience_level	job_title	salary	company_location
156	MI	Applied Data Scientist	1700000	IN
217	EN	Data Engineer	1400000	IN

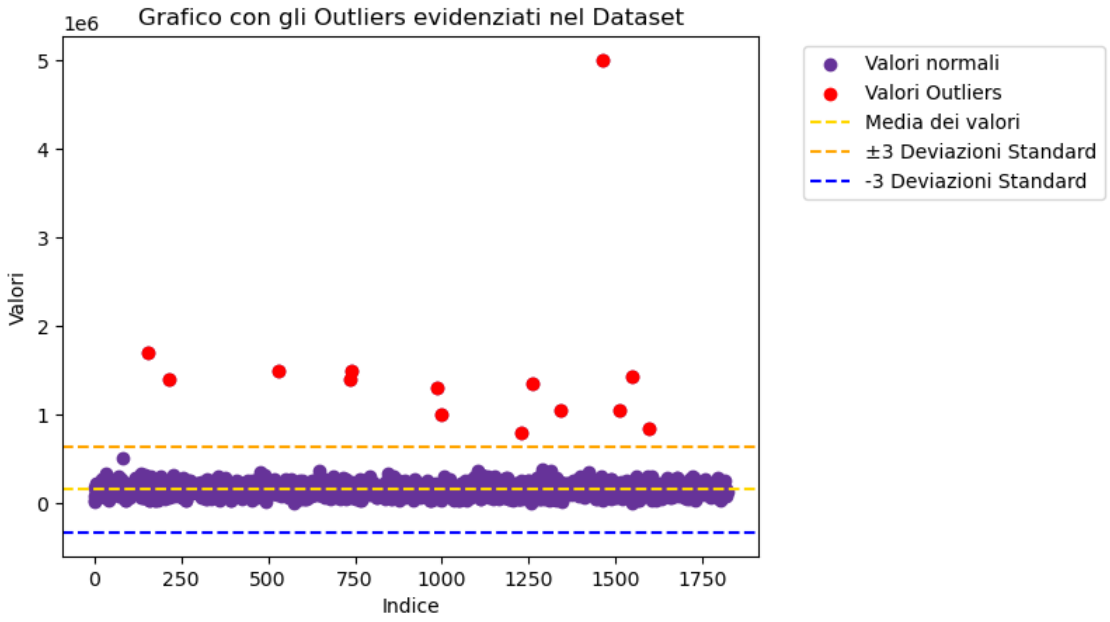
```
528                SE              AI Scientist  1500000              IL
735                MI           Data Scientist  1400000              IN
738                MI         Lead Data Analyst  1500000              IN
988                SE              Data Analyst  1300000              IN
998                SE  Data Science Consultant  1000000              TH
1230               EN           Data Scientist   800000              IN
1260               MI     Product Data Analyst  1350000              IN
1341               EN           Data Scientist  1050000              IN
1462               MI     Head of Data Science  5000000              IN
1512               EN           Data Scientist  1060000              IN
1549               MI      Data Analytics Lead  1440000              SG
1595               MI           Data Scientist   840000              TH
```

[34]:
```
#Identifica gli outliers consiederando +3 sigma dalla media
outliers_dataset_ridotto=dataset_ridotto[(dataset_ridotto["salary"]>mean_value_dataset_ridotto
 ↪|␣
 ↪(dataset_ridotto["salary"]<mean_value_dataset_ridotto-3*std_dev_dataset_ridotto)]
outliers_dataset_ridotto
```

[34]:
```
     experience_level                job_title  salary company_location
33                 SE   Computer Vision Engineer  342810               US
133                SE  Machine Learning Engineer  342300               US
228                EX               Head of Data  329500               US
478                EX   Director of Data Science  353200               US
649                SE             Data Architect  376080               US
845                MI         Research Scientist  340000               US
1105               SE             Data Scientist  370000               US
1288               SE               Data Analyst  385000               US
1311               SE         Research Scientist  370000               US
1421               SE          Applied Scientist  350000               US
```

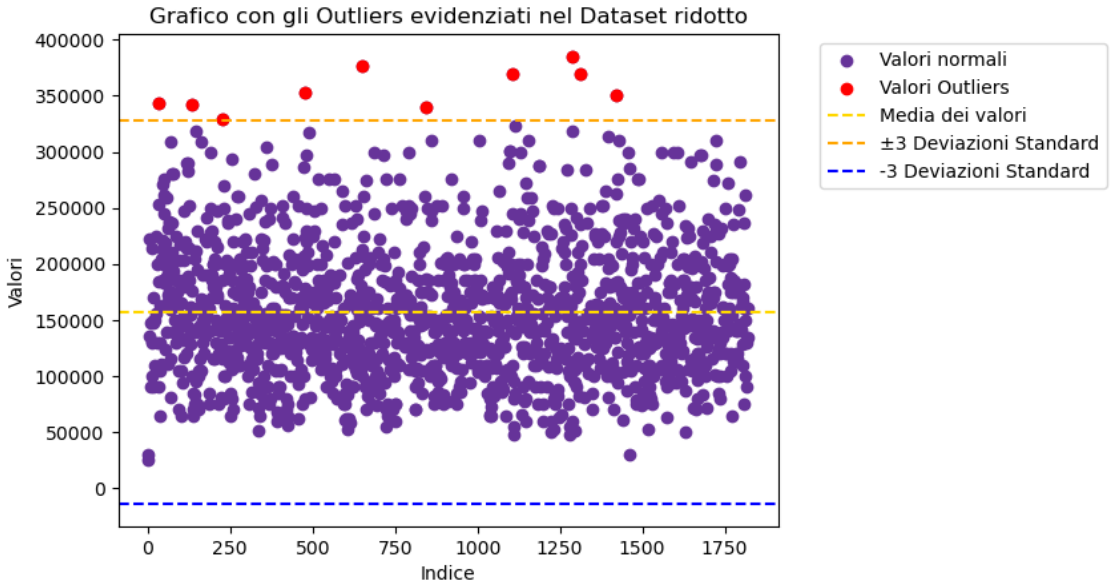
[35]:
```
# Crea un grafico a dispersione
plt.scatter(dataset.index, dataset['salary'], label='Valori normali',␣
 ↪color="rebeccapurple")

# Evidenzia gli outliers nel grafico con un colore diverso
plt.scatter(outliers_dataset.index, outliers_dataset['salary'], color='red',␣
 ↪label='Valori Outliers')

# Aggiungi la media e la deviazione standard al grafico
plt.axhline(y=mean_value_dataset, color='gold', linestyle='--', label='Media␣
 ↪dei valori')
plt.axhline(y=mean_value_dataset + 3 * std_dev_dataset, color='orange',␣
 ↪linestyle='--', label='±3 Deviazioni Standard')
plt.axhline(y=mean_value_dataset - 3 * std_dev_dataset, color='blue',␣
 ↪linestyle='--', label='-3 Deviazioni Standard')
```

```
# Aggiungi etichette e legenda al grafico
plt.xlabel('Indice')
plt.ylabel('Valori')
plt.title('Grafico con gli Outliers evidenziati nel Dataset')
plt.legend(bbox_to_anchor=(1.05, 1), loc='upper left')

# Mostra il grafico
plt.show()
```



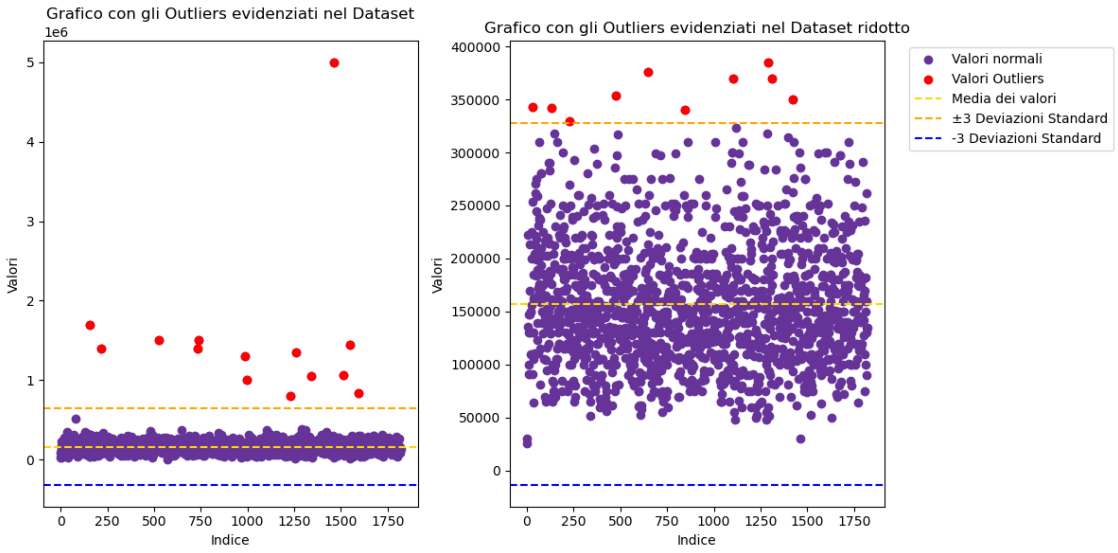
[36]:
```
# Crea un grafico a dispersione
plt.scatter(dataset_ridotto.index, dataset_ridotto['salary'], label='Valori␣
 ↪normali', color="rebeccapurple")

# Evidenzia gli outliers nel grafico con un colore diverso
plt.scatter(outliers_dataset_ridotto.index, outliers_dataset_ridotto['salary'],␣
 ↪color='red', label='Valori Outliers')

# Aggiungi la media e la deviazione standard al grafico
plt.axhline(y=mean_value_dataset_ridotto, color='gold', linestyle='--',␣
 ↪label='Media dei valori')
plt.axhline(y=mean_value_dataset_ridotto + 3 * std_dev_dataset_ridotto,␣
 ↪color='orange', linestyle='--', label='±3 Deviazioni Standard')
plt.axhline(y=mean_value_dataset_ridotto - 3 * std_dev_dataset_ridotto,␣
 ↪color='blue', linestyle='--', label='-3 Deviazioni Standard')

# Aggiungi etichette e legenda al grafico
```

```
plt.xlabel('Indice')
plt.ylabel('Valori')
plt.title('Grafico con gli Outliers evidenziati nel Dataset ridotto')
plt.legend(bbox_to_anchor=(1.05, 1), loc='upper left')

# Mostra il grafico
plt.show()
```



[37]:
```
# Importa la libreria matplotlib
import matplotlib.pyplot as plt

# Crea una figura e due assi (subplot)
fig, axs = plt.subplots(1, 2, figsize=(12, 6))

# Grafico con outliers nel dataset originale
axs[0].scatter(dataset.index, dataset['salary'], label='Valori normali',␣
 ↪color="rebeccapurple")
axs[0].scatter(outliers_dataset.index, outliers_dataset['salary'], color='red',␣
 ↪label='Valori Outliers')
axs[0].axhline(y=mean_value_dataset, color='gold', linestyle='--', label='Media␣
 ↪dei valori')
axs[0].axhline(y=mean_value_dataset + 3 * std_dev_dataset, color='orange',␣
 ↪linestyle='--', label='±3 Deviazioni Standard')
axs[0].axhline(y=mean_value_dataset - 3 * std_dev_dataset, color='blue',␣
 ↪linestyle='--', label='-3 Deviazioni Standard')
axs[0].set_xlabel('Indice')
axs[0].set_ylabel('Valori')
```

```
axs[0].set_title('Grafico con gli Outliers evidenziati nel Dataset')

# Grafico con outliers nel dataset ridotto
axs[1].scatter(dataset_ridotto.index, dataset_ridotto['salary'], label='Valori␣
 ↪normali', color="rebeccapurple")
axs[1].scatter(outliers_dataset_ridotto.index,␣
 ↪outliers_dataset_ridotto['salary'], color='red', label='Valori Outliers')
axs[1].axhline(y=mean_value_dataset_ridotto, color='gold', linestyle='--',␣
 ↪label='Media dei valori')
axs[1].axhline(y=mean_value_dataset_ridotto + 3 * std_dev_dataset_ridotto,␣
 ↪color='orange', linestyle='--', label='±3 Deviazioni Standard')
axs[1].axhline(y=mean_value_dataset_ridotto - 3 * std_dev_dataset_ridotto,␣
 ↪color='blue', linestyle='--', label='-3 Deviazioni Standard')
axs[1].set_xlabel('Indice')
axs[1].set_ylabel('Valori')
axs[1].set_title('Grafico con gli Outliers evidenziati nel Dataset ridotto')
plt.legend(bbox_to_anchor=(1.05, 1), loc='upper left')

# Regola la disposizione e lo spazio tra i subplot
plt.tight_layout()

# Mostra i grafici
plt.show()
```



[38]:
```
# Definisci il numero minimo di features che devono superare la soglia per␣
 ↪considerare un dato un outlier
min_features_threshold = 1
k = 3  # intervallo di confidenza
```

```
# Lista per salvare gli indici degli outliers
outlier_indices_dataset = []

# Calcola la media e la deviazione standard della feature "salary"
mean_salary_dataset = dataset['salary'].mean()
std_dev_salary_dataset = dataset['salary'].std()

# Identifica gli outliers per la feature "salary"
dataset['Outlier_salary'] = (dataset['salary'] > mean_salary_dataset + k *␣
 ↪std_dev_salary_dataset) | (dataset['salary'] < mean_salary_dataset - k *␣
 ↪std_dev_salary_dataset)

dataset
```

[38]:

	experience_level	job_title	salary	company_location	Outlier_salary
0	SE	Principal Data Scientist	80000	ES	False
1	MI	ML Engineer	30000	US	False
2	MI	ML Engineer	25500	US	False
3	SE	Data Scientist	175000	CA	False
4	SE	Data Scientist	120000	CA	False
…	…	…	…	…	…
1815	SE	Machine Learning Engineer	134500	US	False
1817	MI	Data Scientist	130000	US	False
1818	MI	Data Scientist	90000	US	False
1819	EN	Data Engineer	160000	US	False
1820	EN	Data Engineer	135000	US	False

[1785 rows x 5 columns]

[39]:
```
# Definisci il numero minimo di features che devono superare la soglia per␣
 ↪considerare un dato un outlier
min_features_threshold = 1
k = 3  # intervallo di confidenza
```

```
# Lista per salvare gli indici degli outliers
outlier_indices_dataset_ridotto = []

# Calcola la media e la deviazione standard della feature "salary"
mean_salary_dataset_ridotto = dataset_ridotto['salary'].mean()
std_dev_salary_dataset_ridotto = dataset_ridotto['salary'].std()

# Identifica gli outliers per la feature "salary"
dataset_ridotto['Outlier_salary'] = (dataset_ridotto['salary'] >␣
 ↪mean_salary_dataset_ridotto + k * std_dev_salary_dataset_ridotto) |␣
 ↪(dataset_ridotto['salary'] < mean_salary_dataset_ridotto - k *␣
 ↪std_dev_salary_dataset_ridotto)

dataset_ridotto
```

[39]:

	experience_level	job_title	salary	company_location	\
1	MI	ML Engineer	30000	US	
2	MI	ML Engineer	25500	US	
5	SE	Applied Scientist	222200	US	
6	SE	Applied Scientist	136000	US	
9	SE	Data Scientist	147100	US	
…	…	…	…	…	
1815	SE	Machine Learning Engineer	134500	US	
1817	MI	Data Scientist	130000	US	
1818	MI	Data Scientist	90000	US	
1819	EN	Data Engineer	160000	US	
1820	EN	Data Engineer	135000	US	

	Outlier_salary
1	False
2	False
5	False
6	False
9	False
…	…
1815	False
1817	False
1818	False
1819	False
1820	False

[1570 rows x 5 columns]

[40]:
```
#Elimina le righe corrispondenti agli outliers quelli che hanno una features␣
 ↪fuoriscala
```

```
outliers_dataset = dataset['Num_Outliers_nella_riga'] = dataset.
 ↪filter(like='Outlier_').sum(axis=1)
dataset
```

[40]:
```
     experience_level                  job_title  salary company_location  \
0                  SE   Principal Data Scientist   80000               ES
1                  MI                ML Engineer   30000               US
2                  MI                ML Engineer   25500               US
3                  SE             Data Scientist  175000               CA
4                  SE             Data Scientist  120000               CA
…                 …                          …     …              …
1815               SE  Machine Learning Engineer  134500               US
1817               MI             Data Scientist  130000               US
1818               MI             Data Scientist   90000               US
1819               EN              Data Engineer  160000               US
1820               EN              Data Engineer  135000               US

      Outlier_salary  Num_Outliers_nella_riga
0              False                        0
1              False                        0
2              False                        0
3              False                        0
4              False                        0
…              …                          …
1815           False                        0
1817           False                        0
1818           False                        0
1819           False                        0
1820           False                        0

[1785 rows x 6 columns]
```

[41]:
```
#Elimina le righe corrispondenti agli outliers quelli che hanno una features␣
 ↪fuoriscala
outliers_dataset_ridotto = dataset_ridotto['Num_Outliers_nella_riga'] =␣
 ↪dataset_ridotto.filter(like='Outlier_').sum(axis=1)
dataset_ridotto
```

[41]:
```
     experience_level                  job_title  salary company_location  \
1                  MI                ML Engineer   30000               US
2                  MI                ML Engineer   25500               US
5                  SE          Applied Scientist  222200               US
6                  SE          Applied Scientist  136000               US
9                  SE             Data Scientist  147100               US
…                 …                          …     …              …
1815               SE  Machine Learning Engineer  134500               US
1817               MI             Data Scientist  130000               US
```

```
1818               MI             Data Scientist   90000               US
1819               EN              Data Engineer  160000               US
1820               EN              Data Engineer  135000               US

      Outlier_salary  Num_Outliers_nella_riga
1              False                        0
2              False                        0
5              False                        0
6              False                        0
9              False                        0
…                  …                      …
1815           False                        0
1817           False                        0
1818           False                        0
1819           False                        0
1820           False                        0

[1570 rows x 6 columns]
```

[42]:
```
# Filtra i dati per mantenere solo le righe con almeno il numero minimo di␣
 ↪features superanti la soglia
outliers_dataset = dataset[dataset['Num_Outliers_nella_riga'] >=␣
 ↪min_features_threshold]
outliers_dataset
```

[42]:
```
     experience_level                job_title   salary company_location  \
156                MI   Applied Data Scientist  1700000               IN
217                EN            Data Engineer  1400000               IN
528                SE              AI Scientist  1500000               IL
735                MI           Data Scientist  1400000               IN
738                MI        Lead Data Analyst  1500000               IN
988                SE             Data Analyst  1300000               IN
998                SE  Data Science Consultant  1000000               TH
1230               EN           Data Scientist   800000               IN
1260               MI     Product Data Analyst  1350000               IN
1341               EN           Data Scientist  1050000               IN
1462               MI     Head of Data Science  5000000               IN
1512               EN           Data Scientist  1060000               IN
1549               MI      Data Analytics Lead  1440000               SG
1595               MI           Data Scientist   840000               TH

      Outlier_salary  Num_Outliers_nella_riga
156             True                        1
217             True                        1
528             True                        1
735             True                        1
738             True                        1
```

```
988             True                        1
998             True                        1
1230            True                        1
1260            True                        1
1341            True                        1
1462            True                        1
1512            True                        1
1549            True                        1
1595            True                        1
```

[43]:
```
# Filtra i dati per mantenere solo le righe con almeno il numero minimo di␣
 ↪features superanti la soglia
outliers_dataset_ridotto =␣
 ↪dataset_ridotto[dataset_ridotto['Num_Outliers_nella_riga'] >=␣
 ↪min_features_threshold]
outliers_dataset_ridotto
```

[43]:
```
     experience_level                job_title  salary company_location  \
33                 SE   Computer Vision Engineer  342810               US
133                SE  Machine Learning Engineer  342300               US
228                EX               Head of Data  329500               US
478                EX   Director of Data Science  353200               US
649                SE             Data Architect  376080               US
845                MI         Research Scientist  340000               US
1105               SE             Data Scientist  370000               US
1288               SE               Data Analyst  385000               US
1311               SE         Research Scientist  370000               US
1421               SE          Applied Scientist  350000               US

      Outlier_salary  Num_Outliers_nella_riga
33              True                        1
133             True                        1
228             True                        1
478             True                        1
649             True                        1
845             True                        1
1105            True                        1
1288            True                        1
1311            True                        1
1421            True                        1
```

[44]:
```
# Aggiungi una colonna che indica se il record è un outlier o meno
dataset['Is_Outlier'] = dataset.index.isin(outliers_dataset.index)
dataset
```

[44]:
```
     experience_level                job_title  salary company_location  \
0                  SE   Principal Data Scientist   80000               ES
```

```
1                  MI                ML Engineer   30000               US
2                  MI                ML Engineer   25500               US
3                  SE             Data Scientist  175000               CA
4                  SE             Data Scientist  120000               CA
…                 …                          …     …                 …
1815               SE  Machine Learning Engineer  134500               US
1817               MI             Data Scientist  130000               US
1818               MI             Data Scientist   90000               US
1819               EN              Data Engineer  160000               US
1820               EN              Data Engineer  135000               US

      Outlier_salary  Num_Outliers_nella_riga  Is_Outlier
0              False                        0       False
1              False                        0       False
2              False                        0       False
3              False                        0       False
4              False                        0       False
…              …                          …         …
1815           False                        0       False
1817           False                        0       False
1818           False                        0       False
1819           False                        0       False
1820           False                        0       False

[1785 rows x 7 columns]
```

[45]:
```
# Aggiungi una colonna che indica se il record è un outlier o meno
dataset_ridotto['Is_Outlier'] = dataset_ridotto.index.
 ↪isin(outliers_dataset_ridotto.index)
dataset_ridotto
```

[45]:
```
     experience_level                job_title  salary company_location  \
1                  MI                ML Engineer   30000               US
2                  MI                ML Engineer   25500               US
5                  SE          Applied Scientist  222200               US
6                  SE          Applied Scientist  136000               US
9                  SE             Data Scientist  147100               US
…                 …                          …     …                 …
1815               SE  Machine Learning Engineer  134500               US
1817               MI             Data Scientist  130000               US
1818               MI             Data Scientist   90000               US
1819               EN              Data Engineer  160000               US
1820               EN              Data Engineer  135000               US

      Outlier_salary  Num_Outliers_nella_riga  Is_Outlier
1              False                        0       False
2              False                        0       False
```

```
5              False                         0       False
6              False                         0       False
9              False                         0       False
…                …                         …         …
1815           False                         0       False
1817           False                         0       False
1818           False                         0       False
1819           False                         0       False
1820           False                         0       False

[1570 rows x 7 columns]
```

[46]:
```
# Rimuovi colonne ausiliarie
dataset.drop(dataset.filter(like='Outlier_').columns, axis=1, inplace=True)
dataset.drop('Num_Outliers_nella_riga', axis=1, inplace=True)
dataset
```

[46]:
```
     experience_level                  job_title  salary company_location  \
0                  SE   Principal Data Scientist   80000               ES
1                  MI                ML Engineer   30000               US
2                  MI                ML Engineer   25500               US
3                  SE             Data Scientist  175000               CA
4                  SE             Data Scientist  120000               CA
…                 …                          …     …                 …
1815               SE  Machine Learning Engineer  134500               US
1817               MI             Data Scientist  130000               US
1818               MI             Data Scientist   90000               US
1819               EN              Data Engineer  160000               US
1820               EN              Data Engineer  135000               US

     Is_Outlier
0         False
1         False
2         False
3         False
4         False
…           …
1815      False
1817      False
1818      False
1819      False
1820      False

[1785 rows x 5 columns]
```

[47]:
```
# Rimuovi colonne ausiliarie
```

```
dataset_ridotto.drop(dataset_ridotto.filter(like='Outlier_').columns, axis=1,␣
↪inplace=True)
dataset_ridotto.drop('Num_Outliers_nella_riga', axis=1, inplace=True)
dataset_ridotto
```

[47]:
```
     experience_level                  job_title  salary company_location  \
1                  MI                ML Engineer   30000               US
2                  MI                ML Engineer   25500               US
5                  SE          Applied Scientist  222200               US
6                  SE          Applied Scientist  136000               US
9                  SE             Data Scientist  147100               US
…                 …                          …     …                …
1815               SE  Machine Learning Engineer  134500               US
1817               MI             Data Scientist  130000               US
1818               MI             Data Scientist   90000               US
1819               EN              Data Engineer  160000               US
1820               EN              Data Engineer  135000               US

     Is_Outlier
1         False
2         False
5         False
6         False
9         False
…         …
1815      False
1817      False
1818      False
1819      False
1820      False

[1570 rows x 5 columns]
```

[48]:
```
dataset_filtered = dataset[dataset['Is_Outlier'] == False ]
dataset_filtered
```

[48]:
```
     experience_level                  job_title  salary company_location  \
0                  SE   Principal Data Scientist   80000               ES
1                  MI                ML Engineer   30000               US
2                  MI                ML Engineer   25500               US
3                  SE             Data Scientist  175000               CA
4                  SE             Data Scientist  120000               CA
…                 …                          …     …                …
1815               SE  Machine Learning Engineer  134500               US
1817               MI             Data Scientist  130000               US
1818               MI             Data Scientist   90000               US
1819               EN              Data Engineer  160000               US
```

```
1820              EN              Data Engineer  135000              US

      Is_Outlier
0          False
1          False
2          False
3          False
4          False
…          …
1815       False
1817       False
1818       False
1819       False
1820       False

[1771 rows x 5 columns]
```

[49]:
```
dataset_ridotto_filtered = dataset_ridotto[dataset_ridotto['Is_Outlier'] ==␣
↪False ]
dataset_ridotto_filtered
```

[49]:
```
     experience_level                job_title  salary company_location  \
1                  MI               ML Engineer   30000               US
2                  MI               ML Engineer   25500               US
5                  SE         Applied Scientist  222200               US
6                  SE         Applied Scientist  136000               US
9                  SE            Data Scientist  147100               US
…                 …                          …     …                …
1815               SE  Machine Learning Engineer  134500               US
1817               MI             Data Scientist  130000               US
1818               MI             Data Scientist   90000               US
1819               EN              Data Engineer  160000               US
1820               EN              Data Engineer  135000               US

      Is_Outlier
1          False
2          False
5          False
6          False
9          False
…          …
1815       False
1817       False
1818       False
1819       False
1820       False
```

[1560 rows x 5 columns]

1.8 FASE 8: LO SCALING ED ENCODING DEI DATI NELLE FEATURE (CON I GRAFICI)

[54]:
```
# Escludi le colonne non numeriche dal DataFrame
numeric_columns = dataset_ridotto_filtered.select_dtypes(include=['number']).
 ↪columns
dataset_numeric = dataset_ridotto_filtered[numeric_columns]

# Min-Max scaling solo delle colonne numeriche
min_max_scaler = MinMaxScaler()
min_max_scaled_data = min_max_scaler.fit_transform(dataset_numeric)
min_max_scaled_dataset_numeric = pd.DataFrame(min_max_scaled_data,␣
 ↪columns=dataset_numeric.columns)

# Concatena le colonne non numeriche con quelle scalate
non_numeric_columns = dataset_ridotto_filtered.
 ↪select_dtypes(exclude=['number']).columns
min_max_scaled_dataset_ridotto_filtered = pd.
 ↪concat([min_max_scaled_dataset_numeric,␣
 ↪dataset_ridotto_filtered[non_numeric_columns]], axis=1)

# Visualizza i DataFrame dopo lo scaling
print("\nDataFrame ridotto filtrato dopo Min-Max scaling:")
min_max_scaled_dataset_ridotto_filtered
```

DataFrame ridotto filtrato dopo Min-Max scaling:

[54]:

	salary	experience_level	job_title	company_location	\
0	0.015111	NaN	NaN	NaN	
1	0.000000	MI	ML Engineer	US	
2	0.660510	MI	ML Engineer	US	
3	0.371054	NaN	NaN	NaN	
4	0.408328	NaN	NaN	NaN	
…	…	…	…	…	
1815	NaN	SE	Machine Learning Engineer	US	
1817	NaN	MI	Data Scientist	US	
1818	NaN	MI	Data Scientist	US	
1819	NaN	EN	Data Engineer	US	
1820	NaN	EN	Data Engineer	US	

	Is_Outlier
0	NaN
1	False
2	False

```
3           NaN
4           NaN
…           …
1815      False
1817      False
1818      False
1819      False
1820      False

[1786 rows x 5 columns]
```

[53]:
```
# Escludi le colonne non numeriche dal DataFrame
numeric_columns = dataset_filtered.select_dtypes(include=['number']).columns
dataset_numeric = dataset_filtered[numeric_columns]

# Min-Max scaling solo delle colonne numeriche
min_max_scaler = MinMaxScaler()
min_max_scaled_data = min_max_scaler.fit_transform(dataset_numeric)
min_max_scaled_dataset_numeric = pd.DataFrame(min_max_scaled_data,␣
 ↪columns=dataset_numeric.columns)

# Concatena le colonne non numeriche con quelle scalate
non_numeric_columns = dataset_filtered.select_dtypes(exclude=['number']).columns
min_max_scaled_dataset_filtered = pd.concat([min_max_scaled_dataset_numeric,␣
 ↪dataset_filtered[non_numeric_columns]], axis=1)

# Visualizza i DataFrame dopo lo scaling
print("\nDataFrame filtrato dopo Min-Max scaling:")
min_max_scaled_dataset_filtered
```

```
DataFrame ridotto  dopo Min-Max scaling:
```

[53]:
```
        salary experience_level                  job_title company_location  \
0     0.145129               SE   Principal Data Scientist               ES
1     0.045726               MI                ML Engineer               US
2     0.036779               MI                ML Engineer               US
3     0.333996               SE             Data Scientist               CA
4     0.224652               SE             Data Scientist               CA
…          …                …                          …              …
1815       NaN               SE  Machine Learning Engineer               US
1817       NaN               MI             Data Scientist               US
1818       NaN               MI             Data Scientist               US
1819       NaN               EN              Data Engineer               US
1820       NaN               EN              Data Engineer               US

     Is_Outlier
```

```
0          False
1          False
2          False
3          False
4          False
…          …
1815       False
1817       False
1818       False
1819       False
1820       False

[1818 rows x 5 columns]
```

1.9 FASE 9: LO SPLITTING DATASET

[ ]:
```
import numpy as np
from sklearn.model_selection import train_test_split # in questo caso viene␣
 ↪solo importata una parte di libreria poichè è strettamente necessaria quella␣
 ↪determinata funzione
# Suddividere il dataset in training set (70%) e test set (30%) formando due␣
 ↪DataSet
X_train, X_test, y_train, y_test = train_test_split(salary, job_title,␣
 ↪test_size=0.3, random_state=42) # riprendendo la formula di prima: le X sono␣
 ↪i valori delle altezze perchè sono le Feature del DataSet, cioè l'input.␣
 ↪Invece le Y sono gli output o target del DataSet, cioè i valori dei pesi.␣
 ↪"test_size=0.3" vuol dire che il DataSet di Test è il 30% di quello totale␣
 ↪mentre random_state sceglie in modo randomico i valori del DataSet per il␣
 ↪Training e il Test
# Stampare le dimensioni dei training set e test set
print("Dimensioni del Training Set (salary e job title):", X_train.shape,␣
 ↪y_train.shape) # shape = dimensione dei DataSet di Training
print("Dimensioni del Test Set (salary e job title):", X_test.shape, y_test.
 ↪shape) # shape = dimensione dei DataSet di Test
```