Data Normalization with Pandas

February 14, 2025

1 Data Normalization with Pandas

https://www.geeksforgeeks.org/data-normalization-with-pandas/Steps Needed

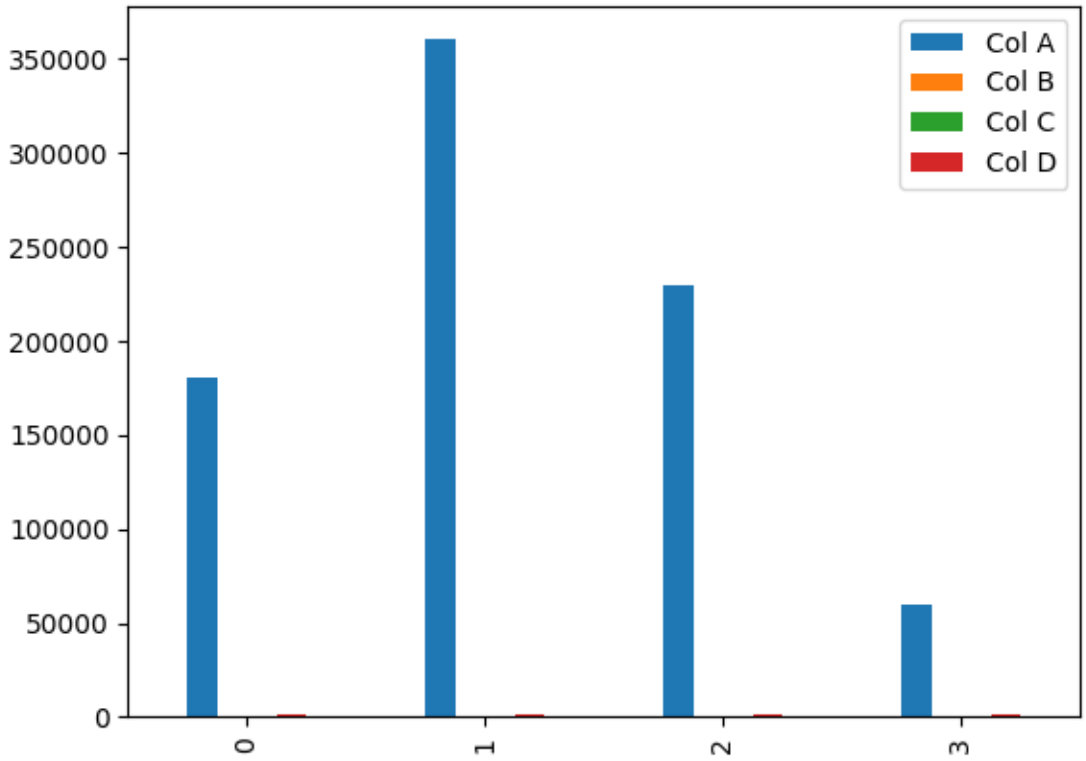
1.Import Library (Pandas) 2.Import / Load / Create data. 3.Use the technique to normalize the data.

[7]:
```
import pandas as pd

df = pd.DataFrame([[180000, 110, 18.9, 1400],
                   [360000, 905, 23.4, 1800],
                   [230000, 230, 14.0, 1300],
                   [60000, 450, 13.5, 1500]],

                    columns=['Col A', 'Col B', 'Col C', 'Col D']
                  )

display(df)
```

	Col A	Col B	Col C	Col D
0	180000	110	18.9	1400
1	360000	905	23.4	1800
2	230000	230	14.0	1300
3	60000	450	13.5	1500

[8]:
```
import matplotlib.pyplot as plt

df.plot(kind='bar')
```

[8]: <Axes: >



1.1 Applying Normalization Technique for the above

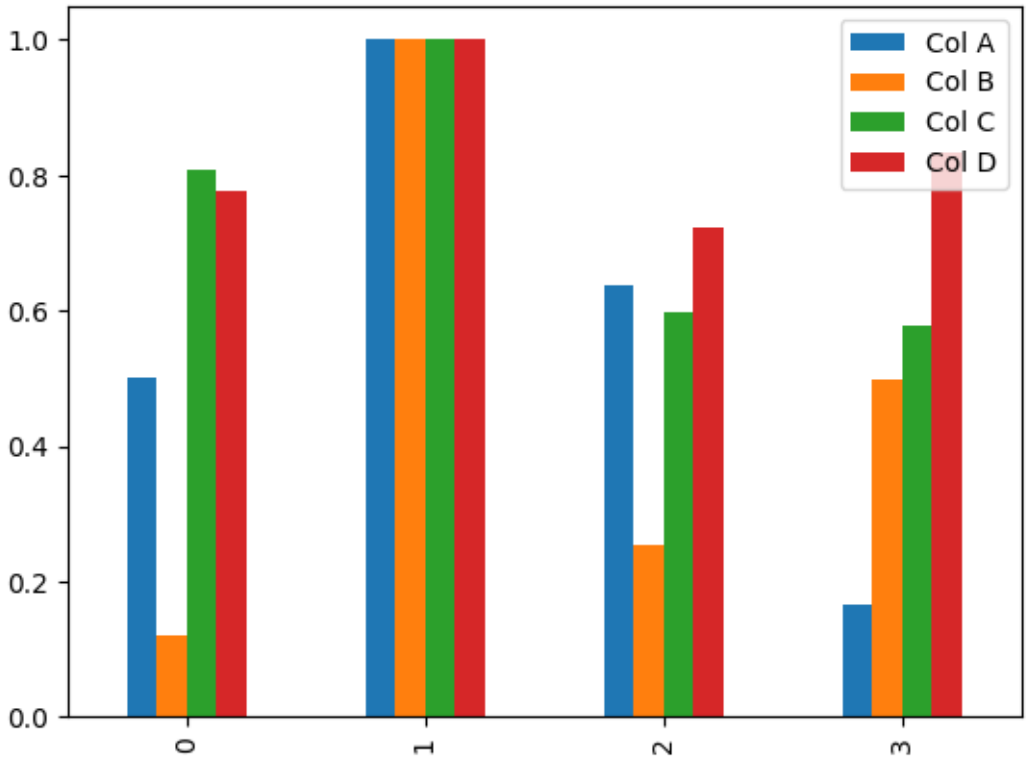
1.1.1 1. Maximum absolute scaling

[9]:
```
df_max_scaled = df.copy()

for column in df_max_scaled.columns:
    df_max_scaled[column] = df_max_scaled[column] / df_max_scaled[column].abs().
 ↪max()

display(df_max_scaled)
```

	Col A	Col B	Col C	Col D
0	0.500000	0.121547	0.807692	0.777778
1	1.000000	1.000000	1.000000	1.000000
2	0.638889	0.254144	0.598291	0.722222
3	0.166667	0.497238	0.576923	0.833333

[15]:
```
import matplotlib.pyplot as plt

df_max_scaled.plot(kind='bar')
```

[15]: <Axes: >



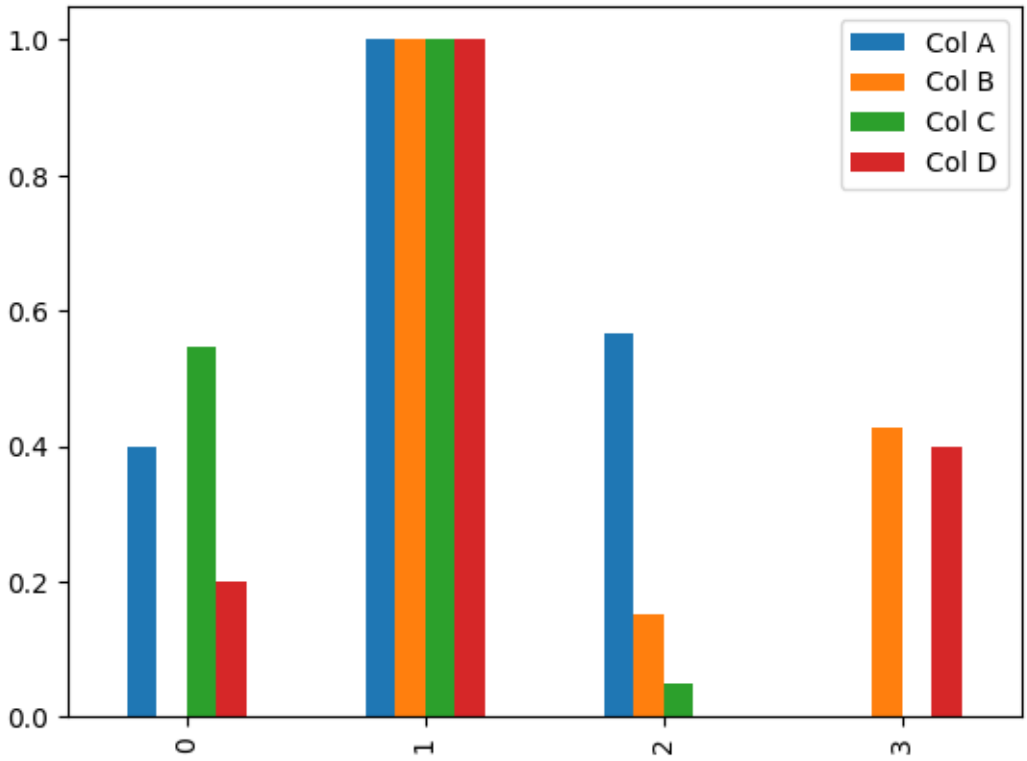
1.1.2 2. Using The min-max feature scaling / Normalization method

[20]:
```
df_min_max_scaled = df.copy()

for column in df_min_max_scaled.columns:
    df_min_max_scaled[column] = (df_min_max_scaled[column] -␣
 ↪df_min_max_scaled[column].min()) / (df_min_max_scaled[column].max() -␣
 ↪df_min_max_scaled[column].min())

print(df_min_max_scaled)
```

```
      Col A     Col B     Col C  Col D
0  0.400000  0.000000  0.545455    0.2
1  1.000000  1.000000  1.000000    1.0
2  0.566667  0.150943  0.050505    0.0
3  0.000000  0.427673  0.000000    0.4
```

[21]:
```
import matplotlib.pyplot as plt

df_min_max_scaled.plot(kind = 'bar')
```

[21]: <Axes: >



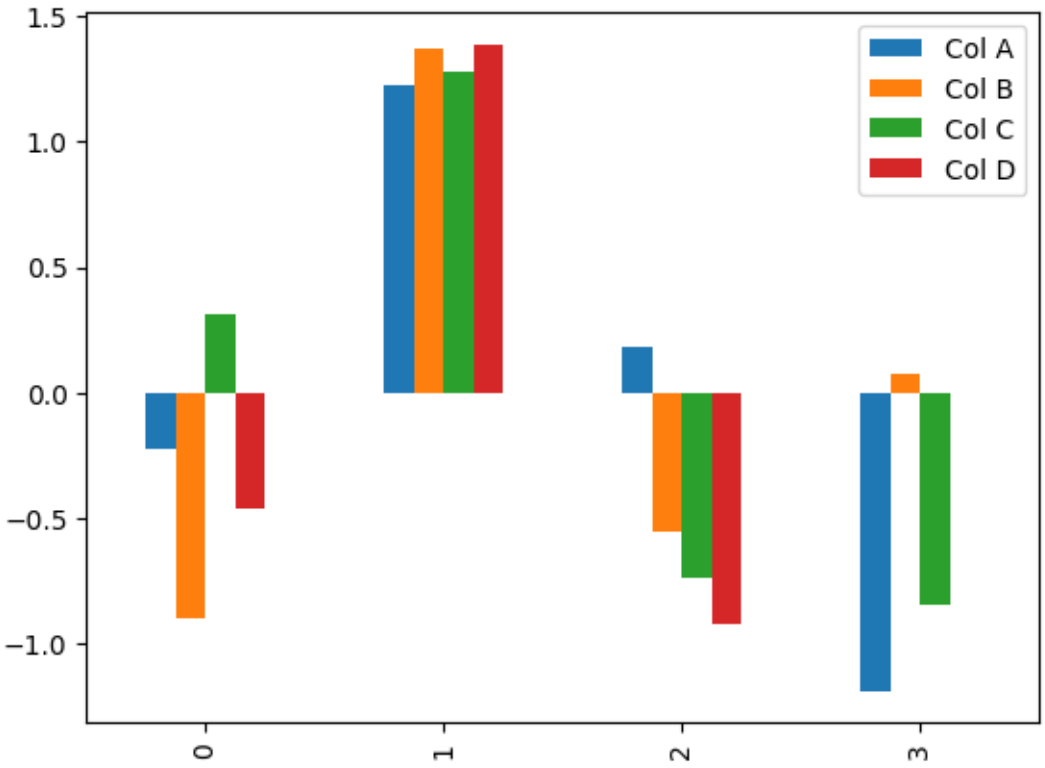
1.1.3 3. Using The z-score method / Standardization Method

[22]:
```
df_z_scaled = df.copy()

for column in df_z_scaled.columns:
    df_z_scaled[column] = (df_z_scaled[column] - df_z_scaled[column].mean()) /␣
 ↪df_z_scaled[column].std()

display(df_z_scaled)
```

```
      Col A     Col B     Col C    Col D
0 -0.221422 -0.895492  0.311486 -0.46291
1  1.227884  1.373564  1.278167  1.38873
2  0.181163 -0.552993 -0.741122 -0.92582
3 -1.187625  0.074922 -0.848531  0.00000
```

[23]:
```
import matplotlib.pyplot as plt

df_z_scaled.plot(kind = 'bar')
```

[23]: <Axes: >



1.2 Imported Data - employees.csv

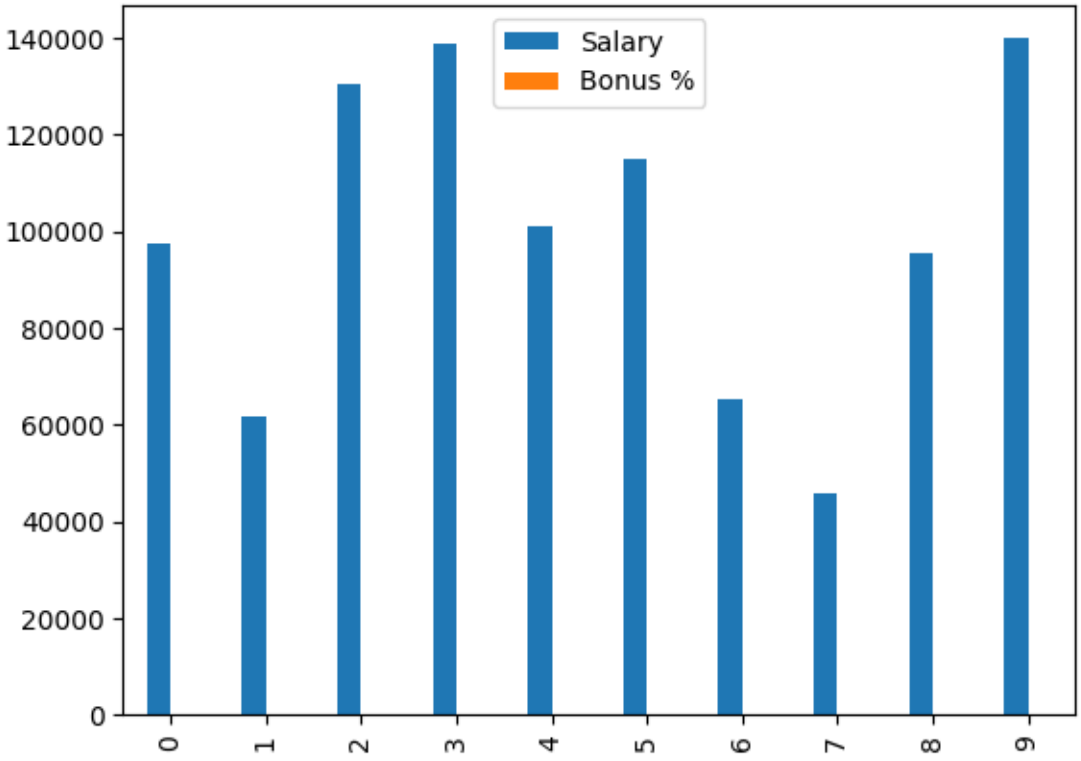
[64]:
```
import pandas as pd

imported_data = pd.read_csv("~/Desktop/myenv/Datasets/employees.csv")

display(imported_data.head(10))
```

```
  First Name  Gender  Start Date Last Login Time  Salary  Bonus %  \
0    Douglas    Male    8/6/1993        12:42 PM   97308    6.945
1     Thomas    Male   3/31/1996         6:53 AM   61933    4.170
2      Maria  Female   4/23/1993        11:17 AM  130590   11.858
3      Jerry    Male    3/4/2005         1:00 PM  138705    9.340
4      Larry    Male   1/24/1998         4:47 PM  101004    1.389
5     Dennis    Male   4/18/1987         1:35 AM  115163   10.125
6       Ruby  Female   8/17/1987         4:20 PM   65476   10.012
7        NaN  Female   7/20/2015        10:43 AM   45906   11.598
8     Angela  Female  11/22/2005         6:29 AM   95570   18.523
9    Frances  Female    8/8/2002         6:51 AM  139852    7.524

  Senior Management                 Team
0              True            Marketing
```

```
1              True                    NaN
2             False                Finance
3              True                Finance
4              True        Client Services
5             False                  Legal
6              True                Product
7               NaN                Finance
8              True            Engineering
9              True  Business Development
```

[65]:
```
import matplotlib.pyplot as plt

imported_data.head(10).plot(kind = 'bar')
```

[65]: <Axes: >



1.2.1 1. Maximum Absolute Scaling

[41]:
```
df_max_scaled = imported_data.copy()

for column in df_max_scaled.select_dtypes(include=[int,float]).columns:
```

```
    df_max_scaled[column] = df_max_scaled[column] / df_max_scaled[column].abs().
↪max()

display(df_max_scaled.head(10))
```

```
  First Name  Gender  Start Date Last Login Time    Salary   Bonus %  \
0    Douglas    Male    8/6/1993        12:42 PM  0.649118  0.348225
1     Thomas    Male   3/31/1996         6:53 AM  0.413140  0.209085
2      Maria  Female   4/23/1993        11:17 AM  0.871134  0.594565
3      Jerry    Male    3/4/2005         1:00 PM  0.925267  0.468311
4      Larry    Male   1/24/1998         4:47 PM  0.673773  0.069645
5     Dennis    Male   4/18/1987         1:35 AM  0.768225  0.507671
6       Ruby  Female   8/17/1987         4:20 PM  0.436775  0.502006
7        NaN  Female   7/20/2015        10:43 AM  0.306228  0.581528
8     Angela  Female  11/22/2005         6:29 AM  0.637524  0.928751
9    Frances  Female    8/8/2002         6:51 AM  0.932919  0.377256

  Senior Management                 Team
0              True            Marketing
1              True                  NaN
2             False              Finance
3              True              Finance
4              True       Client Services
5             False                Legal
6              True              Product
7               NaN              Finance
8              True          Engineering
9              True  Business Development
```

[60]:
```
import matplotlib.pyplot as plt

df_max_scaled.head(10).plot(kind = 'bar')
```

[60]: <Axes: >

1.2.2 2. Min Max feature scaling

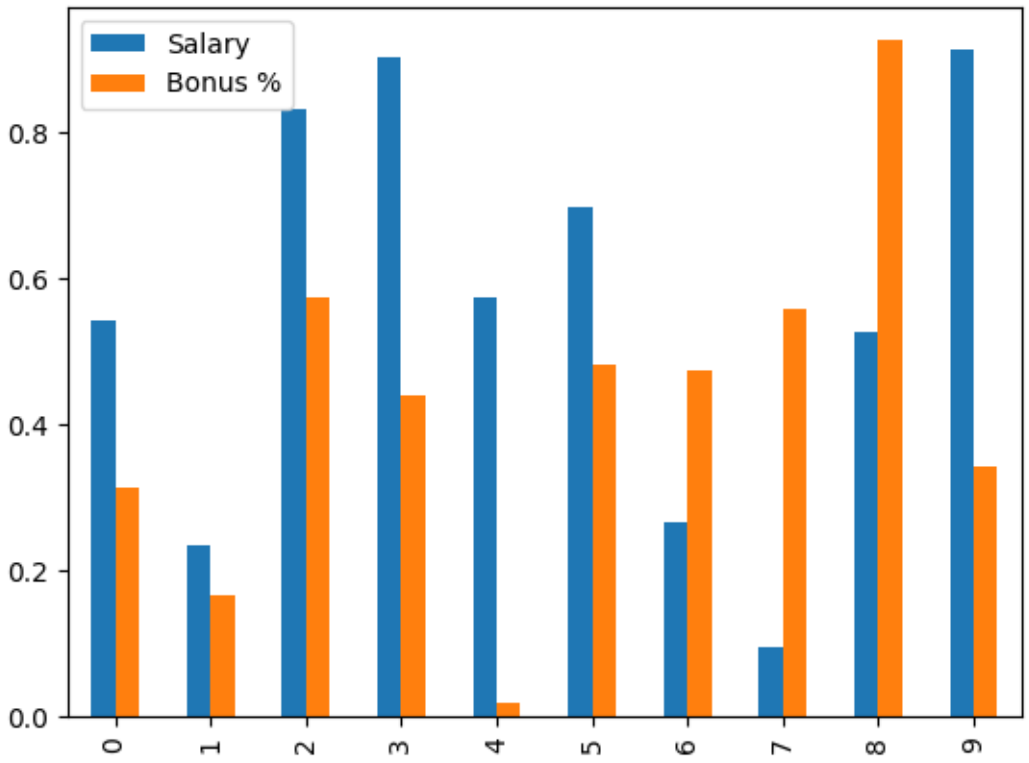
[66]:
```
df_min_max_scaled = imported_data.copy()

for column in df_min_max_scaled.select_dtypes(include=[int,float]).columns:
    df_min_max_scaled[column] = (df_min_max_scaled[column] -␣
 ↪df_min_max_scaled[column].min()) / (df_min_max_scaled[column].max() -␣
 ↪df_min_max_scaled[column].min())

display(df_min_max_scaled.head(10))
```

	First Name	Gender	Start Date	Last Login Time	Salary	Bonus %	\
0	Douglas	Male	8/6/1993	12:42 PM	0.542191	0.313276	
1	Thomas	Male	3/31/1996	6:53 AM	0.234301	0.166675	
2	Maria	Female	4/23/1993	11:17 AM	0.831864	0.572825	
3	Jerry	Male	3/4/2005	1:00 PM	0.902494	0.439801	
4	Larry	Male	1/24/1998	4:47 PM	0.574359	0.019758	
5	Dennis	Male	4/18/1987	1:35 AM	0.697593	0.481272	
6	Ruby	Female	8/17/1987	4:20 PM	0.265138	0.475302	
7	NaN	Female	7/20/2015	10:43 AM	0.094808	0.559089	
8	Angela	Female	11/22/2005	6:29 AM	0.527064	0.924930	
9	Frances	Female	8/8/2002	6:51 AM	0.912477	0.343864	

	Senior Management	Team
0	True	Marketing
1	True	NaN
2	False	Finance
3	True	Finance
4	True	Client Services
5	False	Legal
6	True	Product
7	NaN	Finance
8	True	Engineering
9	True	Business Development

[67]:
```
import matplotlib.pyplot as plt

df_min_max_scaled.head(10).plot(kind = 'bar')
```

[67]: <Axes: >



1.2.3 3. Z-Score Method

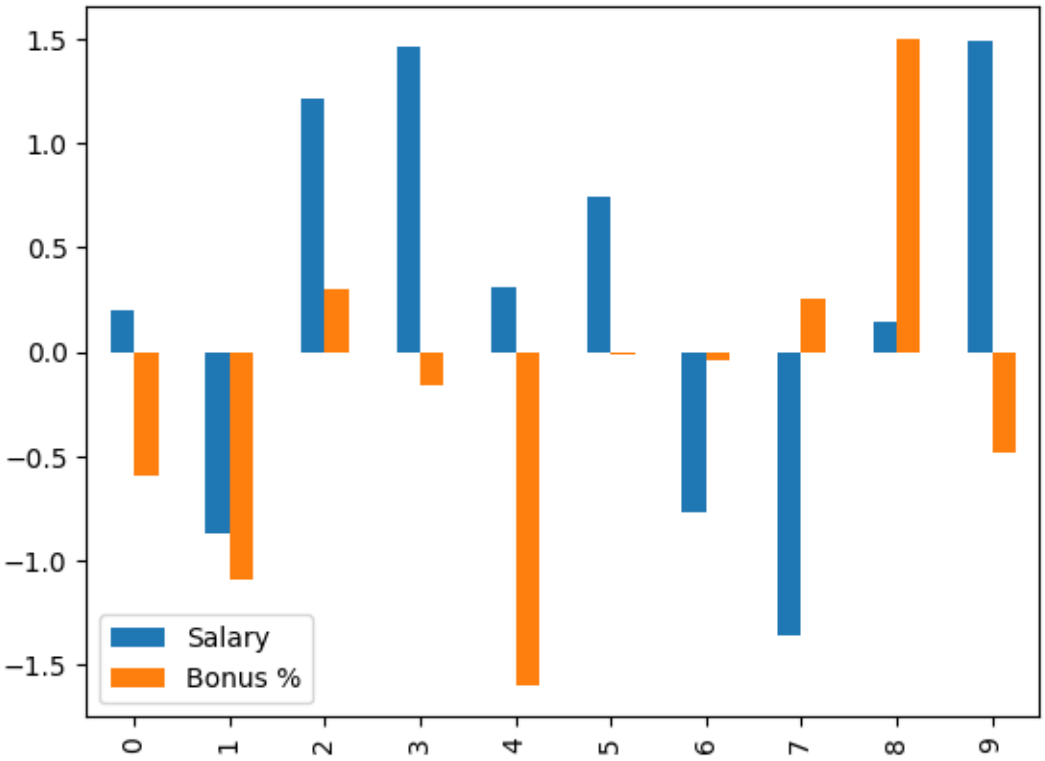
[51]:
```
df_z_scaled = imported_data.copy()

for column in df_z_scaled.select_dtypes(include=[int,float]).columns:
    df_z_scaled[column] = (df_z_scaled[column] - df_z_scaled[column].mean()) /␣
 ↪df_z_scaled[column].std()

display(df_z_scaled.head(10))
```

	First Name	Gender	Start Date	Last Login Time	Salary	Bonus %	\
0	Douglas	Male	8/6/1993	12:42 PM	0.201855	-0.590136	
1	Thomas	Male	3/31/1996	6:53 AM	-0.872599	-1.092082	
2	Maria	Female	4/23/1993	11:17 AM	1.212738	0.298535	
3	Jerry	Male	3/4/2005	1:00 PM	1.459217	-0.156925	
4	Larry	Male	1/24/1998	4:47 PM	0.314115	-1.595114	
5	Dennis	Male	4/18/1987	1:35 AM	0.744170	-0.014933	
6	Ruby	Female	8/17/1987	4:20 PM	-0.764987	-0.035372	
7	NaN	Female	7/20/2015	10:43 AM	-1.359391	0.251506	
8	Angela	Female	11/22/2005	6:29 AM	0.149066	1.504110	
9	Frances	Female	8/8/2002	6:51 AM	1.494055	-0.485406	

	Senior Management	Team
0	True	Marketing
1	True	NaN
2	False	Finance
3	True	Finance
4	True	Client Services
5	False	Legal
6	True	Product
7	NaN	Finance
8	True	Engineering
9	True	Business Development

[62]:
```
import matplotlib.pyplot as plt

df_z_scaled.head(10).plot(kind = 'bar')
```

[62]: <Axes: >



1.2.4 4. Decimal Scaling

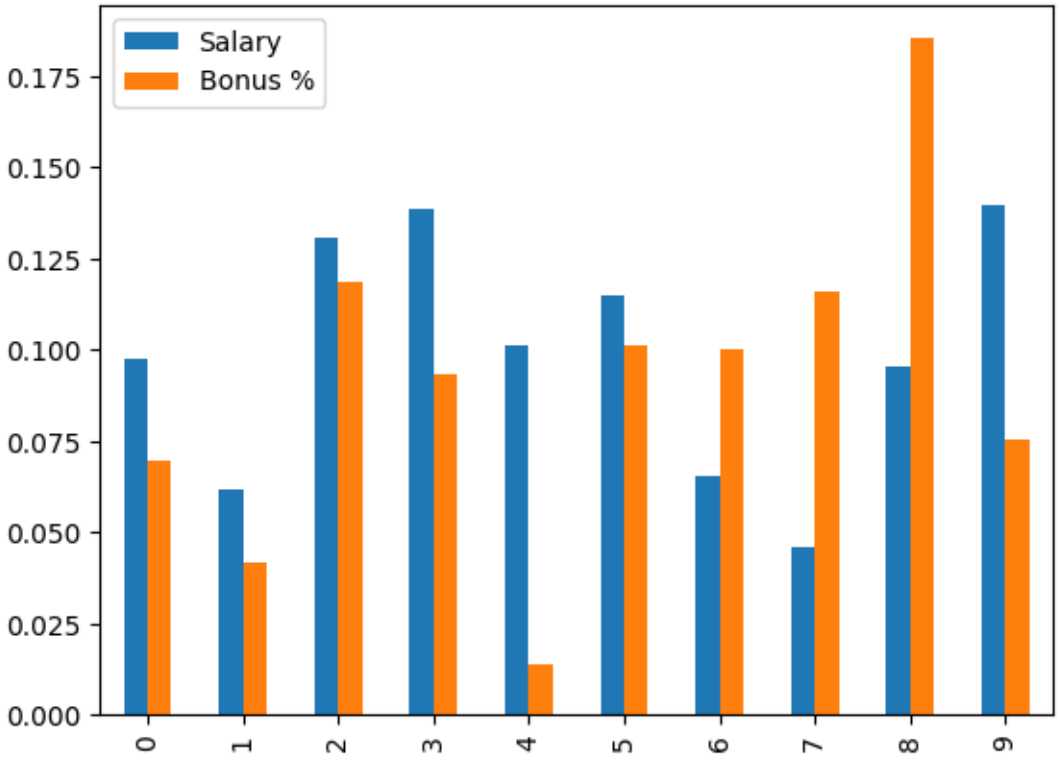
[72]:
```
import numpy as np

df_decimal_scaled = imported_data.copy()

for column in df_decimal_scaled.select_dtypes(include=[int,float]).columns:
    df_decimal_scaled[column] = df_decimal_scaled[column] / 10**np.ceil(np.
 ↪log10(df_decimal_scaled[column].abs().max()))

display(df_robust_scaled.head(10))
```

	First Name	Gender	Start Date	Last Login Time	Salary	Bonus %	\
0	Douglas	Male	8/6/1993	12:42 PM	97308	6.945	
1	Thomas	Male	3/31/1996	6:53 AM	61933	4.170	
2	Maria	Female	4/23/1993	11:17 AM	130590	11.858	
3	Jerry	Male	3/4/2005	1:00 PM	138705	9.340	
4	Larry	Male	1/24/1998	4:47 PM	101004	1.389	
5	Dennis	Male	4/18/1987	1:35 AM	115163	10.125	
6	Ruby	Female	8/17/1987	4:20 PM	65476	10.012	
7	NaN	Female	7/20/2015	10:43 AM	45906	11.598	
8	Angela	Female	11/22/2005	6:29 AM	95570	18.523	
9	Frances	Female	8/8/2002	6:51 AM	139852	7.524	

	Senior Management	Team
0	True	Marketing
1	True	NaN
2	False	Finance
3	True	Finance
4	True	Client Services
5	False	Legal
6	True	Product
7	NaN	Finance
8	True	Engineering
9	True	Business Development

[74]:
```
import matplotlib.pyplot as plt

df_decimal_scaled.head(10).plot(kind = 'bar')
```

[74]: <Axes: >



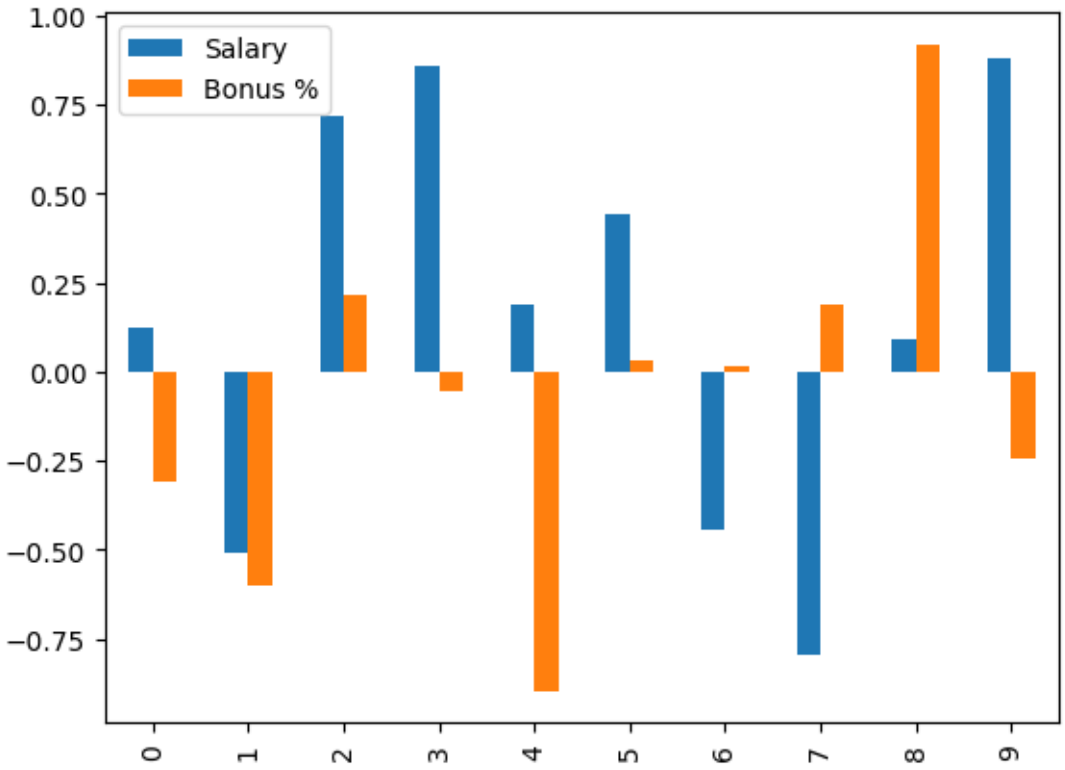
1.2.5 5. Robust Scaling

[78]:
```
from sklearn.preprocessing import RobustScaler

df_robust_scaled = imported_data.copy()
scaler = RobustScaler()

for column in df_robust_scaled.select_dtypes(include=[int,float]).columns:
    df_robust_scaled[column] = scaler.fit_transform(df_robust_scaled[column].
 ↪values.reshape(-1,1))

display(df_robust_scaled.head(10))
```

	First Name	Gender	Start Date	Last Login Time	Salary	Bonus %	Senior Management	Team
0	Douglas	Male	8/6/1993	12:42 PM	0.122579	-0.306637	True	Marketing
1	Thomas	Male	3/31/1996	6:53 AM	-0.507686	-0.600715	True	NaN
2	Maria	Female	4/23/1993	11:17 AM	0.715553	0.214015	False	Finance
3	Jerry	Male	3/4/2005	1:00 PM	0.860135	-0.052828	True	Finance
4	Larry	Male	1/24/1998	4:47 PM	0.188429	-0.895430	True	Client Services
5	Dennis	Male	4/18/1987	1:35 AM	0.440695	0.030362	False	Legal
6	Ruby	Female	8/17/1987	4:20 PM	-0.444561	0.018387	True	Product
7	NaN	Female	7/20/2015	10:43 AM	-0.793233	0.186462	NaN	Finance
8	Angela	Female	11/22/2005	6:29 AM	0.091613	0.920334	True	Engineering
9	Frances	Female	8/8/2002	6:51 AM	0.880570	-0.245278	True	Business Development

[79]:
```
import matplotlib.pyplot as plt

df_robust_scaled.head(10).plot(kind = 'bar')
```

[79]: <Axes: >



1.2.6 6. L2 Scaling

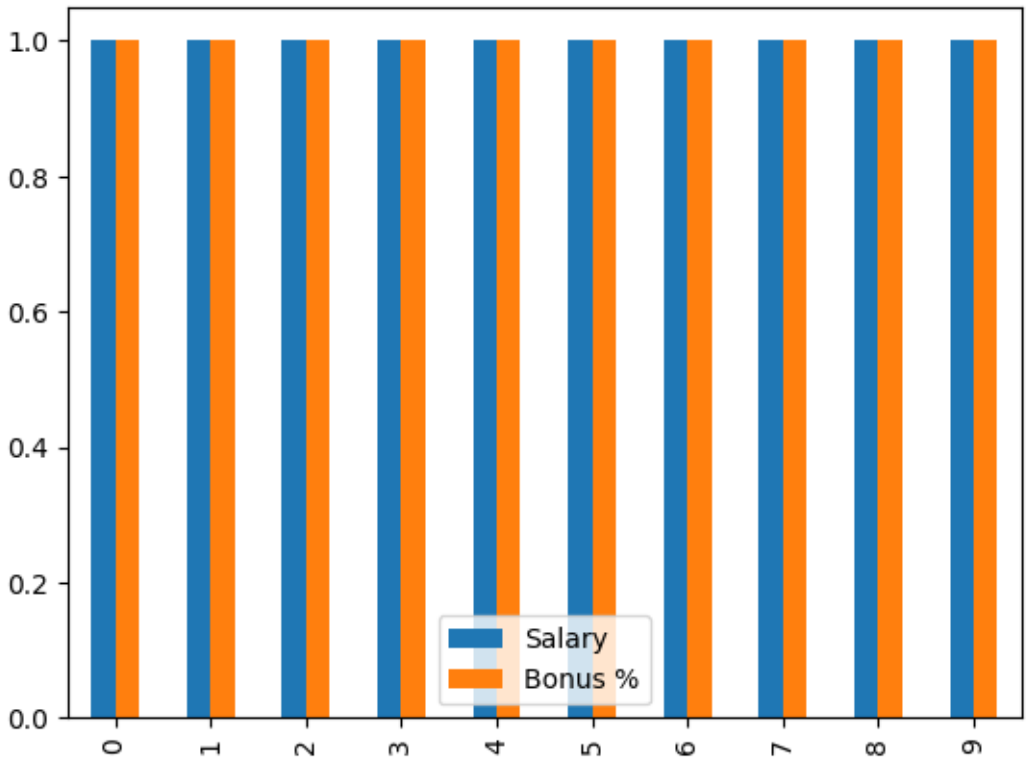
[80]:
```
from sklearn.preprocessing import Normalizer

df_l2_scaled = imported_data.copy()
scaler = Normalizer(norm='l2')

for column in df_l2_scaled.select_dtypes(include=[int,float]).columns:
    df_l2_scaled[column] = scaler.fit_transform(df_l2_scaled[column].values.
 ↪reshape(-1,1))

display(df_l2_scaled.head(10))
```

	First Name	Gender	Start Date	Last Login Time	Salary	Bonus %	\
0	Douglas	Male	8/6/1993	12:42 PM	1.0	1.0	
1	Thomas	Male	3/31/1996	6:53 AM	1.0	1.0	
2	Maria	Female	4/23/1993	11:17 AM	1.0	1.0	
3	Jerry	Male	3/4/2005	1:00 PM	1.0	1.0	
4	Larry	Male	1/24/1998	4:47 PM	1.0	1.0	
5	Dennis	Male	4/18/1987	1:35 AM	1.0	1.0	
6	Ruby	Female	8/17/1987	4:20 PM	1.0	1.0	
7	NaN	Female	7/20/2015	10:43 AM	1.0	1.0	

```
8     Angela  Female  11/22/2005         6:29 AM     1.0      1.0
9    Frances  Female    8/8/2002         6:51 AM     1.0      1.0

  Senior Management                  Team
0              True             Marketing
1              True                   NaN
2             False               Finance
3              True               Finance
4              True       Client Services
5             False                 Legal
6              True               Product
7               NaN               Finance
8              True           Engineering
9              True  Business Development
```

[81]:
```
import matplotlib.pyplot as plt

df_l2_scaled.head(10).plot(kind = 'bar')
```

[81]: <Axes: >



[ ]: