DATA PREPROCESSING

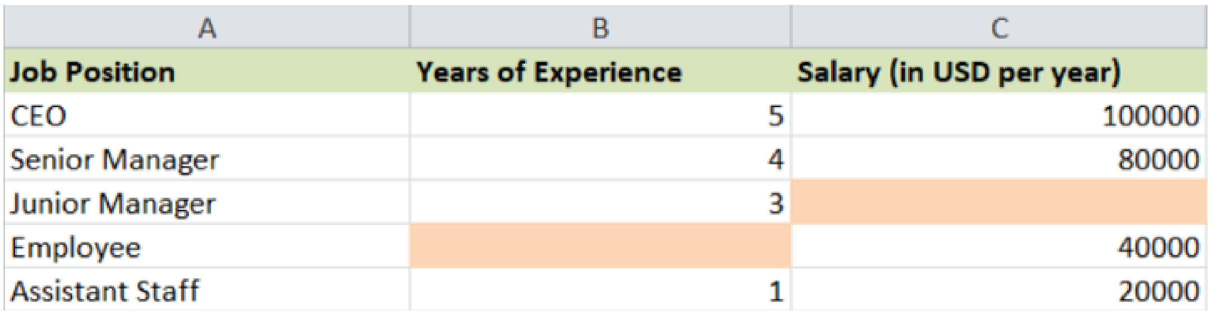
Handling Null Values

Date	21-10-2022
Team ID	PNT2022TMID434432
Project Name	Project - Web Phishing Detection

Techniques For Handling Missing Data

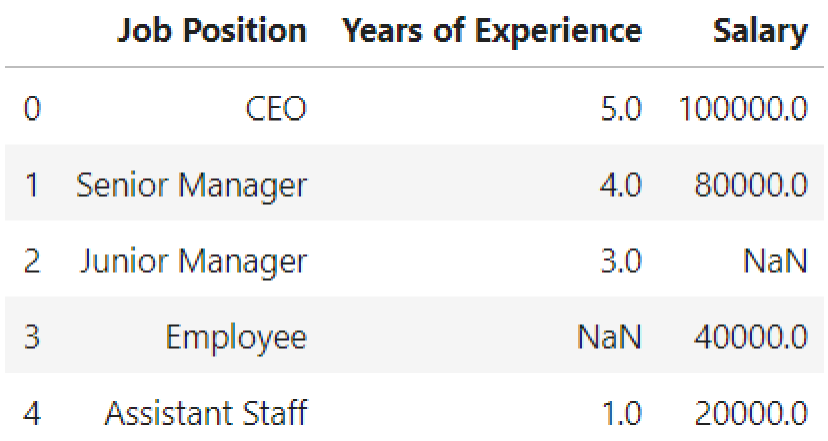
Data may not always be complete i.e. some of the values in the data may be missing or null. Thus, there are a specific set of ways to handle the missing data and make the data complete.

The following example shows that the 'Years of Experience' of 'Employee' is missing. Also, the 'Salary (in USD per year)' of 'Junior Manager' is missing.



A	B	C
Job Position	**Years of Experience**	**Salary (in USD per year)**
CEO	5	100000
Senior Manager	4	80000
Junior Manager	3	
Employee		40000
Assistant Staff	1	20000

```
import pandas as pd

# Creating the dataframe as shown above

df = pd.DataFrame({'Job Position': ['CEO', 'Senior Manager', 'Junior
Manager', 'Employee', 'Assistant Staff'], 'Years of Experience':[5, 4,
3, None, 1], 'Salary':[100000,80000,None,40000, 20000]})

# Viewing the contents of the dataframe
df.head()
```

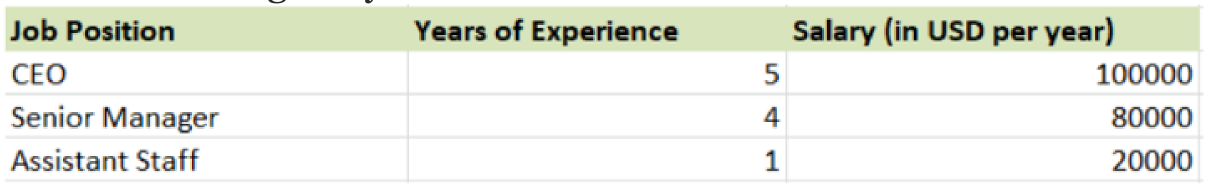


	Job Position	Years of Experience	Salary
0	CEO	5.0	100000.0
1	Senior Manager	4.0	80000.0
2	Junior Manager	3.0	NaN
3	Employee	NaN	40000.0
4	Assistant Staff	1.0	20000.0

Some of the ways to handle missing data are listed below:

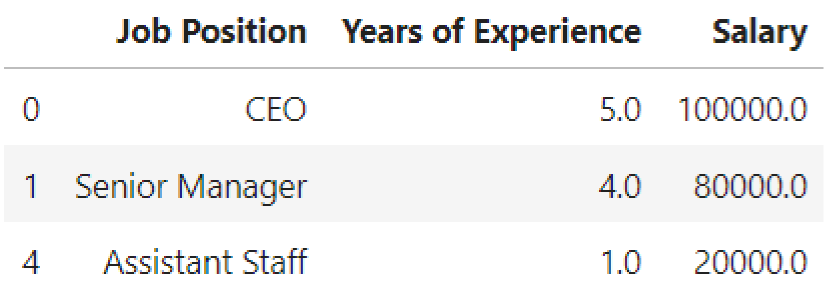
1. Data Removal

Remove the missing data rows (data points) from the dataset. However, when using this technique will decrease the available dataset and in turn result in less robustness of data point if the size of dataset is originally small.



Job Position	Years of Experience	Salary (in USD per year)
CEO	5	100000
Senior Manager	4	80000
Assistant Staff	1	20000

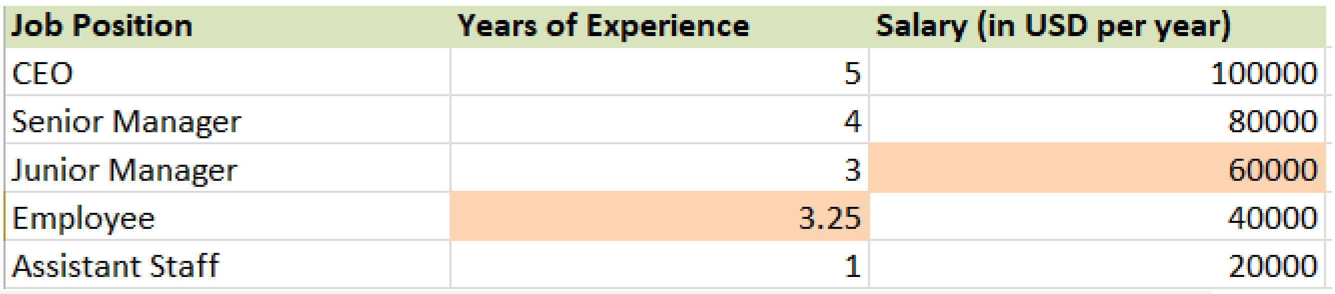
```
# Dropping the 2nd and 3rd index
dropped_df = df.drop([2,3],axis=0)

# Viewing the dataframe
dropped_df
```



	Job Position	Years of Experience	Salary
0	CEO	5.0	100000.0
1	Senior Manager	4.0	80000.0
4	Assistant Staff	1.0	20000.0

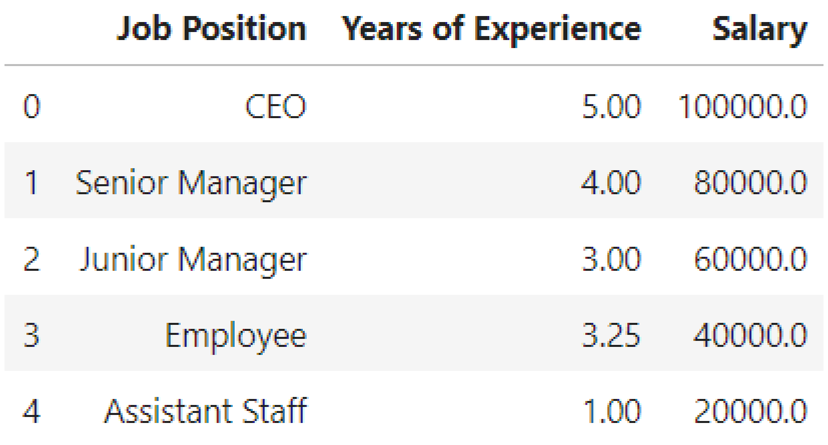
2. Fill missing value through statistical imputation

Fill the missing data by taking the mean or median of the available data points. Generally, the median of the data points is used to fill the missing values as it is not affected heavily by outliers like the mean. Here, we have used the median to fill the missing data.



Job Position	Years of Experience	Salary (in USD per year)
CEO	5	100000
Senior Manager	4	80000
Junior Manager	3	60000
Employee	3.25	40000
Assistant Staff	1	20000

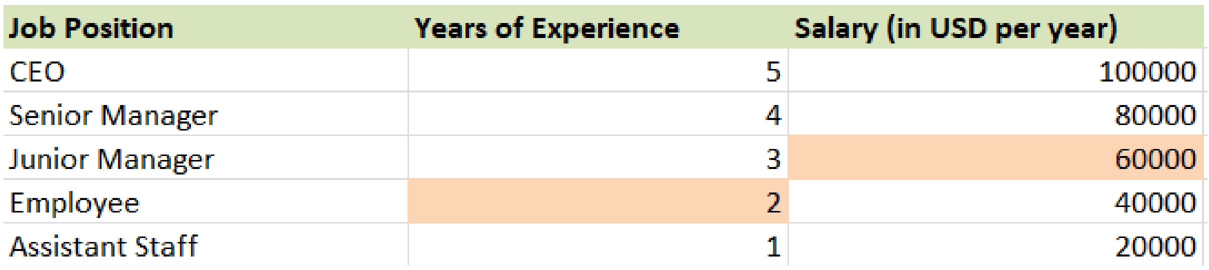
```
# Filling each column with their mean values

df['Years of Experience'] = df['Years of
Experience'].fillna(df['Years of Experience'].mean())

df['Salary'] = df['Salary'].fillna(df['Salary'].mean())

# Viewing the dataframe
df
```



	Job Position	Years of Experience	Salary
0	CEO	5.00	100000.0
1	Senior Manager	4.00	80000.0
2	Junior Manager	3.00	60000.0
3	Employee	3.25	40000.0
4	Assistant Staff	1.00	20000.0

3. Fill missing value using observation

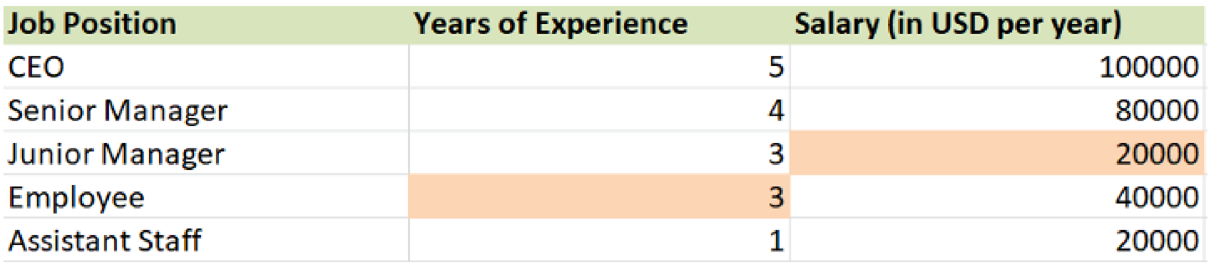
Manually fill in the missing data from observation. This may be possible sometimes for small datasets but for larger datasets it is very difficult to do so.



Job Position	Years of Experience	Salary (in USD per year)
CEO	5	100000
Senior Manager	4	80000
Junior Manager	3	60000
Employee	2	40000
Assistant Staff	1	20000

4. Fill in the most repeated value

Fill in the missing value using the most repeated value in the dataset. This is done when most of the data is repeated and there is good reasoning to do so. Since there are no repeated values in the example, we can fill it with any one of the numbers in the respective column.



Job Position	Years of Experience	Salary (in USD per year)
CEO	5	100000
Senior Manager	4	80000
Junior Manager	3	20000
Employee	3	40000
Assistant Staff	1	20000

5. Fill in with random value within the range of available data

Take the given range of data points and fill in the data by randomly selecting a value from the available range.

Job Position	Years of Experience	Salary (in USD per year)
CEO	5	100000
Senior Manager	4	80000
Junior Manager	3	20000
Employee	3	40000
Assistant Staff	1	20000

6. Fill in by regression

Use regression analysis to find the most probable data point for filling in the dataset.

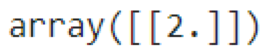
Job Position	Years of Experience	Salary (in USD per year)
CEO	5	100000
Senior Manager	4	80000
Junior Manager	3	60000
Employee	2	40000
Assistant Staff	1	20000

```
from sklearn.linear_model import LinearRegression

# Excluding the rows with the null data
train_df = df.drop([2,3],axis=0)

# Creating linear regression model
regr = LinearRegression()

# Here the target is the Salary and the feature is Years of Experience
regr.fit(train_df[['Years of Experience']], train_df[['Salary']])

# Predicting for 3 years of experience
regr.predict([[3]])
```

```
array([[60000.]])
```

Therefore, the salary for 3 years of experience by regression is 60000. Now, finding the years of experience based on salary.3.

```
from sklearn.linear_model import LinearRegression

# Excluding the rows with the null data
train_df = df.drop([2,3],axis=0)

# Creating linear regression model
regr = LinearRegression()

# Here the target is the Years of Experience and the feature is Salary
regr.fit(train_df[['Salary']], train_df[['Years of Experience']])

# Predicting for 40000 salary
regr.predict([[40000.0]])
```



```
array([[2.]])
```

Therefore, the years of experience for 40000 salary is 2