

```
In [2]: import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
import seaborn as sns
from mlxtend.frequent_patterns import apriori, association_rules
```

```
In [3]: salary = pd.read_csv('ds_salaries.csv')
```

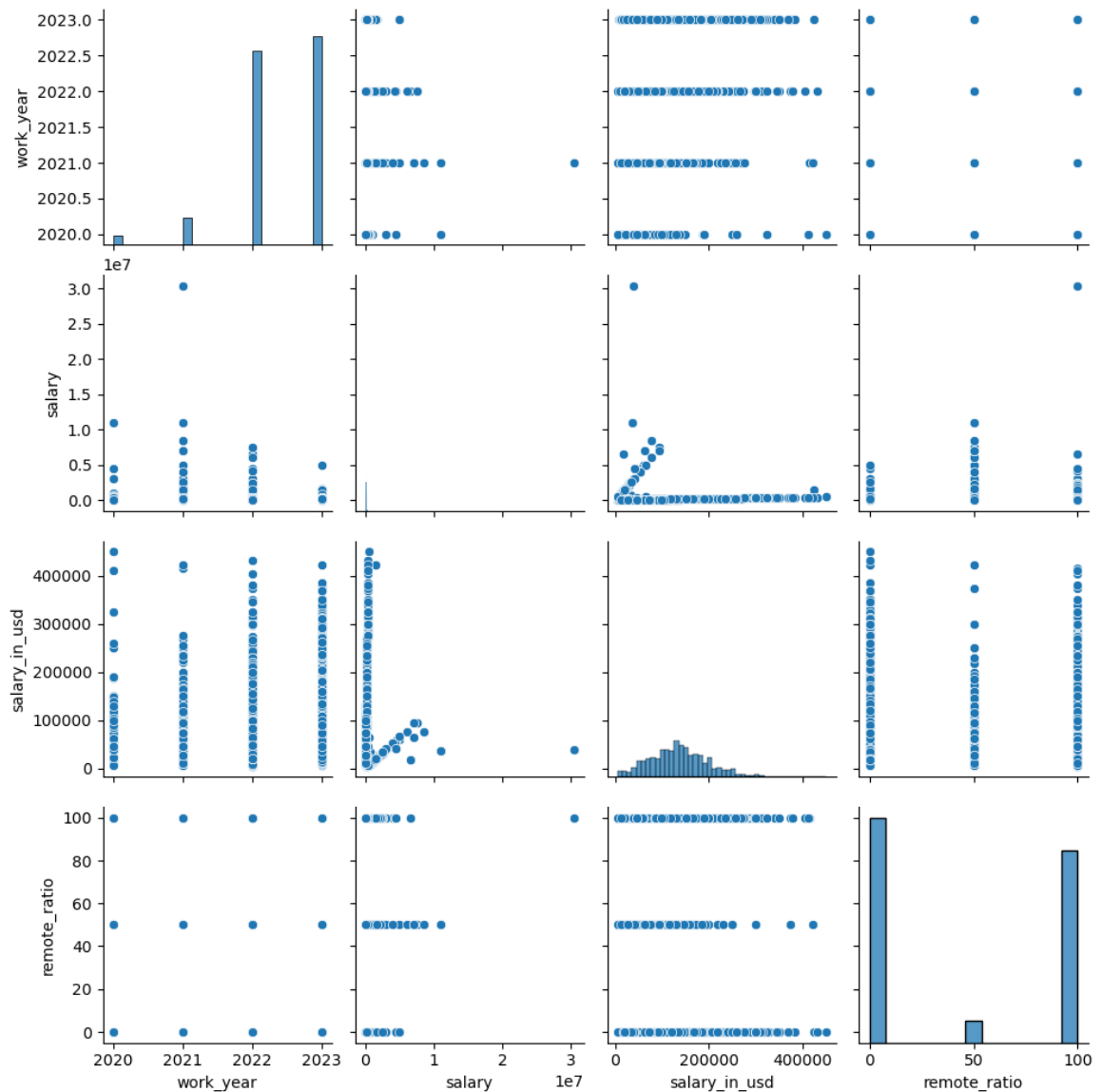
```
In [4]: salary.head()
```

```
Out[4]:
```

	work_year	experience_level	employment_type	job_title	salary	salary_
0	2023	SE	FT	Principal Data Scientist	80000	
1	2023	MI	CT	ML Engineer	30000	
2	2023	MI	CT	ML Engineer	25500	
3	2023	SE	FT	Data Scientist	175000	
4	2023	SE	FT	Data Scientist	120000	

```
In [5]: sns.pairplot(salary)
```

```
Out[5]: <seaborn.axisgrid.PairGrid at 0x12e0730b0>
```



```
In [24]: salary['experience_level'].unique()
```

```
Out[24]: array(['SE', 'MI', 'EN', 'EX'], dtype=object)
```

```
In [30]: from sklearn.preprocessing import OrdinalEncoder
order = ['SE', 'MI', 'EN', 'EX']
OE = OrdinalEncoder( categories = [order] )
salary['experience_level'] = OE.fit_transform(salary[['experience_l
```

```
In [34]: salary['experience_level'].unique()
```

```
Out[34]: array([0., 1., 2., 3.])
```

```
In [38]: salary['company_size'].unique()
```

```
Out[38]: array(['L', 'S', 'M'], dtype=object)
```

```
In [40]: from sklearn.preprocessing import OrdinalEncoder
order = ['L', 'S', 'M']
OE = OrdinalEncoder( categories = [order] )
salary['company_size'] = OE.fit_transform(salary[['company_size']])
```

In [42]: `salary.head()`

Out[42]:

	work_year	experience_level	employment_type	job_title	salary	salary_
0	2023	0.0	FT	Principal Data Scientist	80000	
1	2023	1.0	CT	ML Engineer	30000	
2	2023	1.0	CT	ML Engineer	25500	
3	2023	0.0	FT	Data Scientist	175000	
4	2023	0.0	FT	Data Scientist	120000	

In [44]:

```

from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LinearRegression

scaler = StandardScaler()
X = scaler.fit_transform(salary[['experience_level', 'remote_ratio',
y = salary['salary']

model = LinearRegression()
model.fit(X, y)

```

Out[44]:

LinearRegression ⓘ ?

LinearRegression()

In [46]:

```

y_pred = model.predict(X)

from sklearn.metrics import mean_absolute_error
score = mean_absolute_error(y, y_pred)
print(score)

```

128031.14717806886

In []: