


```
import pandas as pd
import numpy as np
from sklearn import linear_model
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
```



```
from google.colab import files
uploaded = files.upload()
```




 **Choose Files** hiring.csv

- **hiring.csv**(text/csv) - 198 bytes, last modified: 3/17/2025 - 100% done  
Saving hiring.csv to hiring.csv

```
df_hiring = pd.read_csv('hiring.csv')
```


```
df_hiring
```

	experience	test_score(out of 10)	interview_score(out of 10)	salary(\$)	
0	NaN	8.0	9	50000	
1	NaN	8.0	6	45000	
2	five	6.0	7	60000	
3	two	10.0	10	65000	
4	seven	9.0	6	70000	
5	three	7.0	10	62000	
6	ten	NaN	7	72000	
7	eleven	7.0	8	80000	


Next steps: [Generate code with df\\_hiring](#) [View recommended plots](#) [New interactive sheet](#)

```
print("Missing values in the dataset:")
print(df_hiring.isnull().sum())
```

 Missing values in the dataset:


```
experience      2
test_score(out of 10)  1
interview_score(out of 10)  0
salary($)      0
dtype: int64
```

```
df_hiring['experience'].fillna(df_hiring['experience'].mode()[0], inplace=True)
df_hiring['test_score(out of 10)'].fillna(df_hiring['test_score(out of 10)'].mean(), inplace=True)
```

 <ipython-input-7-a370352093a7>:2: FutureWarning: A value is trying to be set on a copy of a DataFrame or Series through chained assignme  
The behavior will change in pandas 3.0. This inplace method will never work because the intermediate object on which we are setting valu  
For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({col: value}, inplace=True)' or df[col] = df[col].me

```
df_hiring['test_score(out of 10)'].fillna(df_hiring['test_score(out of 10)'].mean(), inplace=True)
```

```
print("Missing values in the dataset:")
print(df_hiring.isnull().sum())
```


 Missing values in the dataset:

```
experience      0
test_score(out of 10)  0
interview_score(out of 10)  0
salary($)      0
dtype: int64
```


```
experience_mapping = {
    'two': 2,
    'three': 3,
    'five': 5,
    'seven': 7,
```




```
'eight': 8,
'ten': 10,
'eleven': 11
}
```

```
# Replace the text values in 'experience' with corresponding numeric values
df_hiring['experience'] = df_hiring['experience'].replace(experience_mapping)
```

 <ipython-input-9-142fbbbeaee3f>:12: FutureWarning: Downcasting behavior in `replace` is deprecated and will be removed in a future version  
df\_hiring['experience'] = df\_hiring['experience'].replace(experience\_mapping)

df\_hiring





	experience	test_score(out of 10)	interview_score(out of 10)	salary(\$)	
0	11	8.000000	9	50000	
1	11	8.000000	6	45000	
2	5	6.000000	7	60000	
3	2	10.000000	10	65000	
4	7	9.000000	6	70000	
5	3	7.000000	10	62000	
6	10	7.857143	7	72000	
7	11	7.000000	8	80000	

Next steps: [Generate code with df\\_hiring](#) [View recommended plots](#) [New interactive sheet](#)


```
X_hiring = df_hiring[['experience', 'test_score(out of 10)', 'interview_score(out of 10)']] # Independent variables
y_hiring = df_hiring['salary($)'] # Dependent variable (Salary)
```

```
X_train_hiring, X_test_hiring, y_train_hiring, y_test_hiring = train_test_split(X_hiring, y_hiring, test_size=0.2, random_state=42)
```


```
hiring_model = LinearRegression()
hiring_model.fit(X_train_hiring, y_train_hiring)
```

  LinearRegression ⓘ ?  
LinearRegression()

```
predicted_salary_12_10_10 = hiring_model.predict([[12, 10, 10]])
print(f"\nPredicted salary for a candidate with 12 years of experience, 10 test score, and 10 interview score: ${predicted_salary_12_10_10[0]}
```

 Predicted salary for a candidate with 12 years of experience, 10 test score, and 10 interview score: \$64629.25  
/usr/local/lib/python3.11/dist-packages/sklearn/utils/validation.py:2739: UserWarning: X does not have valid feature names, but LinearRegression.warn()

```
predicted_salary_2_9_6 = hiring_model.predict([[2, 9, 6]])
print(f"Predicted salary for a candidate with 2 years of experience, 9 test score, and 6 interview score: ${predicted_salary_2_9_6[0]:.2f}")
```

 Predicted salary for a candidate with 2 years of experience, 9 test score, and 6 interview score: \$70576.11  
/usr/local/lib/python3.11/dist-packages/sklearn/utils/validation.py:2739: UserWarning: X does not have valid feature names, but LinearRegression.warn()

