Title of the work: Report

Student: Isaac Sánchez

Flores

Date of delivery: 27/11/2024

Subject: Introduction to data

science

Productivity **analysis**

Objective:

The objective of this project is to improve staff productivity and performance by a minimum of 15% in 6 months.

Problem

The company is going through a critical time in terms of productivity and performance, as reflected by recent evaluations showing that approximately 67% of the staff is scoring below 3 on a 5-point scale. This negative trend indicates the existence of problems that are affecting both performance and staff commitment. In addition, it is important to mention that the average employee satisfaction level is hovering around 3 on a 5-point scale, which could lead to mass resignations within the company, increasing the workload on other employees and negatively affecting satisfaction and performance within the company.

Technology and tools

We will be using the Python programming language with the pandas, matplotlib and numpy libraries to perform the initial analysis.

Data

To carry out this project, the database provided to us gives us information about the department, gender, age, job position, years worked in the company, education level, performance score from 1 to 5, performance score from 1 to 5 and performance score from 1 to 5.
worked in the company, education level, performance score from 1 to 5, monthly salary, hours worked per week, number of projects, overtime, sick days taken, frequency of remote work, team size, training hours, raises/promotions, employee satisfaction, and whether the employee resigned.

Initial Hypotheses

Hypothesis 1: Monthly salary is directly proportional to performance.

Hypothesis 2: Employees with many years working in the company and with few promotions have worse performance.

Hypothesis 3: Training hours directly influence performance.

Key Stakeholders

1. Executives: Project financing and evaluation.
2. Human Resources Department: Personnel management and policy implementation.
3. Department heads: Oversee that policies are followed.
4. Employee: Direct impact on project success.

Key Questions

1. What factors affect performance?
2. What is the average salary that shows the best performance?
3. What is the relationship between training hours and performance?
4. What size team performs best?
5. What is the relationship between raises and performance?
6. How does education level affect performance?
7. How can satisfaction scores be increased?
8. How does overtime affect performance?
9. What is the relationship between age and performance?
10. Which department has the worst performance?
11. How does seniority affect performance?
12. How can resignation or recession be avoided?

Data sources identified

- Employee satisfaction scores.
- Performance survey.
- Measurement of training hours.
- Educational attainment survey.
- Performance survey by department.

Project justification

Poor performance and low employee satisfaction within the company represent a critical problem that requires immediate attention. This situation significantly affects revenues and the quality of products and services offered by the company. Poor performance among employees often translates into a decrease in product quality, which negatively impacts competitiveness against other companies in the market. If adequate measures are not taken to address this problem, the company could face serious consequences, such as the loss of customers due to inferior products compared to those of competitors. In addition, the problem of the low level of job satisfaction can lead to higher employee turnover, which is another major challenge. Mass resignations, if left unchecked, will not only affect the continuity of operations, but will also increase the costs of hiring and training new personnel. In summary, both low productivity and employee dissatisfaction put the company's long-term stability and sustainability at risk, making it urgent to implement strategies that improve the work environment, motivate employees and improve their performance, thus ensuring the quality of our products and services.

Amount and type of data

The amount of data to be handled is around 100,000 per column, being 20 columns. Five columns are of type object, twelve are int64, two are float64 and one column is of type bool.

Data cleaning

Initial analysis

Summary statistics of the data before cleaning:

```
df.describe()

#Faltan varias columnas importantes en el resumen estadístico como el puntaje de rendimiento
```

[124] ✓ 0.0s

	Employee_ID	Years_At_Company	Projects_Handled	Overtime_Hours	Team_Size	Promotions	Employee_Satisfaction_Score
count	121825.000000	121825.000000	121825.000000	121825.000000	121825.000000	121825.000000	119393.000000
mean	50083.834878	4.473918	24.458929	14.508566	10.018247	1.000673	2.998747
std	28889.832419	2.872489	14.469846	8.656905	5.500860	0.815299	1.151397
min	1.000000	0.000000	0.000000	0.000000	1.000000	0.000000	1.000000
25%	25058.000000	2.000000	12.000000	7.000000	5.000000	0.000000	2.010000
50%	50129.000000	4.000000	24.000000	15.000000	10.000000	1.000000	3.000000
75%	75092.000000	7.000000	37.000000	22.000000	15.000000	2.000000	3.990000
max	100000.000000	10.000000	49.000000	29.000000	19.000000	2.000000	5.000000

\+ Code + Markdown

Table showing the percentage of missing data per column:

```
# Tabla que muestra el porcentaje de datos faltantes en cada columna
porcentaje_datos_faltantes = (df.isnull().mean() * 100).round(2).astype(str) + '%'
print(porcentaje_datos_faltantes)

#En total alrededor del 80% de datos es nulo
```

[127] ✓ 0.0s Python

```
Employee_ID                    4.0%
Department                     4.0%
Gender                         4.0%
Age                            4.0%
Job_Title                      4.0%
Hire_Date                      4.0%
Years_At_Company               4.0%
Education_Level                4.0%
Performance_Score              4.0%
Monthly_Salary                 4.0%
Work_Hours_Per_Week            4.0%
Projects_Handled               4.0%
Overtime_Hours                 4.0%
Sick_Days                      4.0%
Remote_Work_Frequency          4.0%
Team_Size                      4.0%
Training_Hours                 4.0%
Promotions                     4.0%
Employee_Satisfaction_Score   5.92%
Resigned                       4.0%
dtype: object
```

Total repeated data found: (Only in Employee_ID as it is irrelevant in the demos)

```
df.duplicated('Employee_ID').sum() #Comprobamos cuantos datos duplicados hay en Employee_ID

#Tenemos 30032 duplicados
[128] ✓ 0.0s

30032
```

Description of original data types and problems encountered:

```
df.info()

#La mayoría de datos está en formato object, tengo que cambiar a float o int según sea conveniente
[125] ✓ 0.0s                                                                                    Python

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 126901 entries, 0 to 126900
Data columns (total 20 columns):
 #   Column                     Non-Null Count   Dtype
---  ------                     --------------   -----
 0   Employee_ID                121825 non-null  float64
 1   Department                 121825 non-null  object
 2   Gender                     121825 non-null  object
 3   Age                        121825 non-null  object
 4   Job_Title                  121825 non-null  object
 5   Hire_Date                  121825 non-null  object
 6   Years_At_Company           121825 non-null  float64
 7   Education_Level            121825 non-null  object
 8   Performance_Score          121825 non-null  object
 9   Monthly_Salary             121825 non-null  object
 10  Work_Hours_Per_Week        121825 non-null  object
 11  Projects_Handled           121825 non-null  float64
 12  Overtime_Hours             121825 non-null  float64
 13  Sick_Days                  121825 non-null  object
 14  Remote_Work_Frequency      121825 non-null  object
 15  Team_Size                  121825 non-null  float64
 16  Training_Hours             121825 non-null  object
 17  Promotions                 121825 non-null  float64
 18  Employee_Satisfaction_Score 119393 non-null  float64
 19  Resigned                   121825 non-null  object
dtypes: float64(7), object(13)
memory usage: 19.4+ MB
```

The data originally was numeric type, mostly, but now all the data was in object format, this causes important data such as Perfomance_Score to not appear in the summary statistics.

Another problem I encountered when doing the initial analysis of the database was that many of the columns contained invalid values such as "bbb".

```python
lista_columnas = df.columns

for c in lista_columnas:
    print(f'En la columna {c} los bbb son: {df[df[c] == 'bbb'].shape[0]}')

#Revisamos cuantos bbb hay en cada columna
```

```
En la columna Employee_ID los bbb son: 0
En la columna Department los bbb son: 2430
En la columna Gender los bbb son: 0
En la columna Age los bbb son: 2436
En la columna Job_Title los bbb son: 0
En la columna Hire_Date los bbb son: 0
En la columna Years_At_Company los bbb son: 0
En la columna Education_Level los bbb son: 2445
En la columna Performance_Score los bbb son: 2434
En la columna Monthly_Salary los bbb son: 2439
En la columna Work_Hours_Per_Week los bbb son: 2425
En la columna Projects_Handled los bbb son: 0
En la columna Overtime_Hours los bbb son: 0
En la columna Sick_Days los bbb son: 2434
En la columna Remote_Work_Frequency los bbb son: 2441
En la columna Team_Size los bbb son: 0
En la columna Training_Hours los bbb son: 2445
En la columna Promotions los bbb son: 0
En la columna Employee_Satisfaction_Score los bbb son: 0
En la columna Resigned los bbb son: 0
```

Cleaning process

For the database cleanup I used the numpy library in addition to the pandas library for reasons that I will explain below. The pandas functions I used were to remove duplicates and NaN values, convert data to a specific format and fill NaN values either with a 'No Data' legend or with the average of each column. The only numpy function I used was the one that allows converting data to NaN, I did this to replace the string 'bbb' with values that were more convenient for me with the help of the pandas function mentioned above.

Before removing duplicates:

```python
df.duplicated('Employee_ID').sum() #Comprobamos cuantos datos duplicados hay en Employee_ID

#Tenemos 30032 duplicados
```

```
30032
```

After removing duplicates:

```python
#Primero eliminamos los duplicados en Employee_ID

df = df.drop_duplicates(subset=['Employee_ID'])

df = df.reset_index(drop=True)

df.duplicated(subset=['Employee_ID']).sum()
```

[131] ✓ 0.0s Python

```
0
```

Before transforming data to numeric:

```python
df.info()

#La mayoría de datos está en formato object, tengo que cambiar a float o int según sea conveniente
```

[150] ✓ 0.0s Python

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 126901 entries, 0 to 126900
Data columns (total 20 columns):
 #   Column                     Non-Null Count   Dtype
---  ------                     --------------   -----
 0   Employee_ID                121825 non-null  float64
 1   Department                 121825 non-null  object
 2   Gender                     121825 non-null  object
 3   Age                        121825 non-null  object
 4   Job_Title                  121825 non-null  object
 5   Hire_Date                  121825 non-null  object
 6   Years_At_Company           121825 non-null  float64
 7   Education_Level            121825 non-null  object
 8   Performance_Score          121825 non-null  object
 9   Monthly_Salary             121825 non-null  object
 10  Work_Hours_Per_Week        121825 non-null  object
 11  Projects_Handled           121825 non-null  float64
 12  Overtime_Hours             121825 non-null  float64
 13  Sick_Days                  121825 non-null  object
 14  Remote_Work_Frequency      121825 non-null  object
 15  Team_Size                  121825 non-null  float64
 16  Training_Hours             121825 non-null  object
 17  Promotions                 121825 non-null  float64
 18  Employee_Satisfaction_Score  119393 non-null  float64
 19  Resigned                   121825 non-null  object
dtypes: float64(7), object(13)
memory usage: 19.4+ MB
```

After transforming data to numeric:

```python
df.info()
```

[158] ✓ 0.0s Python

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 96869 entries, 0 to 96868
Data columns (total 20 columns):
 #   Column                     Non-Null Count  Dtype  
---  ------                     --------------  -----  
 0   Employee_ID                96868 non-null  float64
 1   Department                 92987 non-null  object 
 2   Gender                     92979 non-null  object 
 3   Age                        91058 non-null  float64
 4   Job_Title                  92971 non-null  object 
 5   Hire_Date                  92983 non-null  object 
 6   Years_At_Company           92990 non-null  float64
 7   Education_Level            92999 non-null  object 
 8   Performance_Score          91186 non-null  float64
 9   Monthly_Salary             91053 non-null  float64
 10  Work_Hours_Per_Week        91142 non-null  float64
 11  Projects_Handled           93005 non-null  float64
 12  Overtime_Hours             93028 non-null  float64
 13  Sick_Days                  91133 non-null  float64
 14  Remote_Work_Frequency      91200 non-null  float64
 15  Team_Size                  92950 non-null  float64
 16  Training_Hours             91104 non-null  float64
 17  Promotions                 93009 non-null  float64
 18  Employee_Satisfaction_Score 91204 non-null  float64
 19  Resigned                   92977 non-null  object 
dtypes: float64(14), object(6)
memory usage: 14.8+ MB
```

Before converting 'bbb' to NaN:

```python
df
```

[148] ✓ 0.0s

	Employee_ID	Department	Gender	Age	Job_Title	Hire_Date	Years_At_Company	Education_Level	Performan
0	1.0	IT	Male	55	Specialist	NaN	2.0	High School	
1	2.0	Finance	Male	29	Developer	2024-04-18 08:03:05.556036	0.0	High School	
2	3.0	Finance	Male	NaN	Specialist	2015-10-26 08:03:05.556036	8.0	High School	
3	4.0	bbb	Female	48	Analyst	2016-10-22 08:03:05.556036	7.0	Bachelor	
4	5.0	Engineering	Female	36	NaN	2021-07-23 08:03:05.556036	3.0	Bachelor	
...	...	...	...	...	...	...	...	...	
126896	9646.0	Finance	Male	38	Engineer	2021-06-07 08:03:05.556036	3.0	Bachelor	
						2023-09-05			

After converting 'bbb' to NaN:

```python
#Convertimos los bbb a NaN utilizando la librería de Numpy para luego llenar con el promedio o 'Sin dato', según sea conveniente

df.replace('bbb' , np.nan, inplace=True)

df
```

[159] ✓ 0.0s

	Employee_ID	Department	Gender	Age	Job_Title	Hire_Date	Years_At_Company	Education_Level	Performance_Score	Monthly_Sala
0	1.0	IT	Male	55.0	Specialist	NaN	2.0	High School	5.0	6750
1	2.0	Finance	Male	29.0	Developer	2024-04-18 08:03:05.556036	0.0	High School	5.0	7500
2	3.0	Finance	Male	NaN	Specialist	2015-10-26 08:03:05.556036	8.0	High School	3.0	Na
3	4.0	NaN	Female	48.0	Analyst	2016-10-22 08:03:05.556036	7.0	Bachelor	2.0	4800
4	5.0	Engineering	Female	36.0	NaN	2021-07-23 08:03:05.556036	3.0	Bachelor	2.0	4800
...	...	...	...	...	...	...	...	...	...	
96864	72705.0	Engineering	Female	27.0	Specialist	2019-07-23 08:03:05.556036	NaN	Bachelor	NaN	5400
96865	3248.0	Operations	Male	35.0	Analyst	2017-08-29 08:03:05.556036	NaN	Bachelor	2.0	4800

Before replacing NaN with 'No Data' or average:


```python
#Convertimos los bbb a NaN utilizando la librería de Numpy para luego llenar con el promedio o 'Sin dato', según sea conveniente

df.replace('bbb' , np.nan, inplace=True)

df
```

[159] ✓ 0.0s

	Employee_ID	Department	Gender	Age	Job_Title	Hire_Date	Years_At_Company	Education_Level	Performance_Score	Monthly_Sala
0	1.0	IT	Male	55.0	Specialist	NaN	2.0	High School	5.0	6750
1	2.0	Finance	Male	29.0	Developer	2024-04-18 08:03:05.556036	0.0	High School	5.0	7500
2	3.0	Finance	Male	NaN	Specialist	2015-10-26 08:03:05.556036	8.0	High School	3.0	Na
3	4.0	NaN	Female	48.0	Analyst	2016-10-22 08:03:05.556036	7.0	Bachelor	2.0	4800
4	5.0	Engineering	Female	36.0	NaN	2021-07-23 08:03:05.556036	3.0	Bachelor	2.0	4800
...	...	...	...	...	...	...	...	...	...	
96864	72705.0	Engineering	Female	27.0	Specialist	2019-07-23 08:03:05.556036	NaN	Bachelor	NaN	5400
96865	3248.0	Operations	Male	35.0	Analyst	2017-08-29 08:03:05.556036	NaN	Bachelor	2.0	4800

After replacing 'bbb' with 'No Data' or average:

	Employee_ID	Department	Gender	Age	Job_Title	Hire_Date	Years_At_Company	Education_Level	Performance_Score	Monthly_Salary	W
0	1.0	IT	Male	55.000000	Specialist	NaN	2.000000	High School	5.000000	6750.000000	
1	2.0	Finance	Male	29.000000	Developer	2024-04-18 08:03:05.556036	0.000000	High School	5.000000	7500.000000	
2	3.0	Finance	Male	41.024325	Specialist	2015-10-26 08:03:05.556036	8.000000	High School	3.000000	6404.100908	
3	4.0	Sin dato	Female	48.000000	Analyst	2016-10-22 08:03:05.556036	7.000000	Bachelor	2.000000	4800.000000	
4	5.0	Engineering	Female	36.000000	Sin dato	2021-07-23 08:03:05.556036	3.000000	Bachelor	2.000000	4800.000000	
...	...	...	...	...	...	...	...	...	...	...	
96864	72705.0	Engineering	Female	27.000000	Specialist	2019-07-23 08:03:05.556036	4.478632	Bachelor	2.996184	5400.000000	
96865	3248.0	Operations	Male	35.000000	Analyst	2017-08-29 08:03:05.556036	4.478632	Bachelor	2.000000	4800.000000	
96866	66901.0	Sales	Male	41.024325	Specialist	2015-02-01 08:03:05.556036	9.000000	High School	4.000000	6300.000000	
96867	21770.0	Marketing	Male	33.000000	Technician	2016-10-22 08:03:05.556036	7.000000	Bachelor	3.000000	4550.000000	
96868	33177.0	Operations	Other	39.000000	Specialist	2020-12-24 08:03:05.556036	3.000000	Bachelor	3.000000	5850.000000	

Before applying dropna on Employee_ID and Hire_Date:

```
df.info()
[212] ✓ 0.0s

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 96869 entries, 0 to 96868
Data columns (total 20 columns):
 #   Column                     Non-Null Count  Dtype
---  ------                     --------------  -----
 0   Employee_ID                96869 non-null  object
 1   Department                 96869 non-null  object
 2   Gender                     96869 non-null  object
 3   Age                        96869 non-null  float64
 4   Job_Title                  96869 non-null  object
 5   Hire_Date                  92983 non-null  object
 6   Years_At_Company           96869 non-null  float64
 7   Education_Level            96869 non-null  object
 8   Performance_Score          96869 non-null  float64
 9   Monthly_Salary             96869 non-null  float64
 10  Work_Hours_Per_Week        96869 non-null  float64
 11  Projects_Handled           96869 non-null  float64
 12  Overtime_Hours             96869 non-null  float64
 13  Sick_Days                  96869 non-null  float64
 14  Remote_Work_Frequency      96869 non-null  float64
 15  Team_Size                  96869 non-null  float64
 16  Training_Hours             96869 non-null  float64
 17  Promotions                 96869 non-null  float64
 18  Employee_Satisfaction_Score  96869 non-null  float64
 19  Resigned                   96869 non-null  object
dtypes: float64(13), object(7)
memory usage: 14.8+ MB
```

After applying dropna on Employee_ID and Hire_Date:

```
#Verificamos que no haya valores nulos

df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 92982 entries, 1 to 96868
Data columns (total 20 columns):
 #   Column                     Non-Null Count  Dtype  
---  ------                     --------------  -----  
 0   Employee_ID                92982 non-null  float64
 1   Department                 92982 non-null  object 
 2   Gender                     92982 non-null  object 
 3   Age                        92982 non-null  float64
 4   Job_Title                  92982 non-null  object 
 5   Hire_Date                  92982 non-null  object 
 6   Years_At_Company           92982 non-null  float64
 7   Education_Level            92982 non-null  object 
 8   Performance_Score          92982 non-null  float64
 9   Monthly_Salary             92982 non-null  float64
 10  Work_Hours_Per_Week        92982 non-null  float64
 11  Projects_Handled           92982 non-null  float64
 12  Overtime_Hours             92982 non-null  float64
 13  Sick_Days                  92982 non-null  float64
 14  Remote_Work_Frequency      92982 non-null  float64
 15  Team_Size                  92982 non-null  float64
 16  Training_Hours             92982 non-null  float64
 17  Promotions                 92982 non-null  float64
 18  Employee_Satisfaction_Score 92982 non-null  float64
 19  Resigned                   92982 non-null  object 
dtypes: float64(14), object(6)
memory usage: 14.9+ MB
```

Before converting all data to their respective format:

```
#Verificamos que no haya valores nulos

df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 92982 entries, 1 to 96868
Data columns (total 20 columns):
 #   Column                     Non-Null Count  Dtype  
---  ------                     --------------  -----  
 0   Employee_ID                92982 non-null  float64
 1   Department                 92982 non-null  object 
 2   Gender                     92982 non-null  object 
 3   Age                        92982 non-null  float64
 4   Job_Title                  92982 non-null  object 
 5   Hire_Date                  92982 non-null  object 
 6   Years_At_Company           92982 non-null  float64
 7   Education_Level            92982 non-null  object 
 8   Performance_Score          92982 non-null  float64
 9   Monthly_Salary             92982 non-null  float64
 10  Work_Hours_Per_Week        92982 non-null  float64
 11  Projects_Handled           92982 non-null  float64
 12  Overtime_Hours             92982 non-null  float64
 13  Sick_Days                  92982 non-null  float64
 14  Remote_Work_Frequency      92982 non-null  float64
 15  Team_Size                  92982 non-null  float64
 16  Training_Hours             92982 non-null  float64
 17  Promotions                 92982 non-null  float64
 18  Employee_Satisfaction_Score 92982 non-null  float64
 19  Resigned                   92982 non-null  object 
dtypes: float64(14), object(6)
memory usage: 14.9+ MB
```

After converting all data to their respective format:

```
df.info()
```

[272] ✓ 0.0s

```
<class 'pandas.core.frame.DataFrame'>
Index: 92982 entries, 1 to 96868
Data columns (total 20 columns):
 #   Column                     Non-Null Count  Dtype
---  ------                     --------------  -----
 0   Employee_ID                92982 non-null  float64
 1   Department                 92982 non-null  object
 2   Gender                     92982 non-null  object
 3   Age                        92982 non-null  int64
 4   Job_Title                  92982 non-null  object
 5   Hire_Date                  92982 non-null  datetime64[ns]
 6   Years_At_Company           92982 non-null  int64
 7   Education_Level            92982 non-null  object
 8   Performance_Score          92982 non-null  int64
 9   Monthly_Salary             92982 non-null  int64
 10  Work_Hours_Per_Week        92982 non-null  int64
 11  Projects_Handled           92982 non-null  int64
 12  Overtime_Hours             92982 non-null  int64
 13  Sick_Days                  92982 non-null  int64
 14  Remote_Work_Frequency      92982 non-null  int64
 15  Team_Size                  92982 non-null  int64
 16  Training_Hours             92982 non-null  int64
 17  Promotions                 92982 non-null  int64
 18  Employee_Satisfaction_Score  92982 non-null  float64
 19  Resigned                   92982 non-null  bool
dtypes: bool(1), datetime64[ns](1), float64(2), int64(12), object(4)
memory usage: 14.3+ MB
```

Results:

Final summary

```
df.describe()
```

[298] 0.0s Python

	Employee_ID	Age	Hire_Date	Years_At_Company	Performance_Score	Monthly_Salary	Work_Hours_Per_Week	Projects_Handled	Overtime_Hours	Sic
count	92982.000000	92982.000000	92982	92982.000000	92982.000000	92982.000000	92982.000000	92982.000000	92982.000000	92982.
mean	49989.427287	41.014745	2019-09-14 01:21:29.227714560	4.460401	2.997096	6404.305973	44.948227	24.401002	14.535297	7.
min	2.000000	22.000000	2014-09-07 08:03:05.556036	0.000000	1.000000	3850.000000	30.000000	0.000000	0.000000	0.
25%	25001.250000	32.000000	2017-03-18 08:03:05.556036096	2.000000	2.000000	5400.000000	38.000000	12.000000	7.000000	3.
50%	49998.500000	41.000000	2019-09-19 08:03:05.556036096	4.000000	3.000000	6404.000000	45.000000	24.000000	15.000000	7.
75%	74948.750000	50.000000	2022-03-13 08:03:05.556036096	7.000000	4.000000	7200.000000	52.000000	36.000000	22.000000	11.
max	100000.000000	60.000000	2024-09-03 08:03:05.556036	10.000000	5.000000	9000.000000	60.000000	49.000000	29.000000	14.
std	28857.808013	10.904000	NaN	2.813627	1.373266	1330.992928	8.672218	14.179277	8.491088	4.

Final missing data table

```
porcentaje_datos_faltantes_final = (df.isnull().mean() * 100).round(2).astype(str) + '%'
print(porcentaje_datos_faltantes_final)
```

[300] 0.0s

```
Employee_ID                  0.0%
Department                   0.0%
Gender                       0.0%
Age                          0.0%
Job_Title                    0.0%
Hire_Date                    0.0%
Years_At_Company             0.0%
Education_Level              0.0%
Performance_Score            0.0%
Monthly_Salary               0.0%
Work_Hours_Per_Week          0.0%
Projects_Handled             0.0%
Overtime_Hours               0.0%
Sick_Days                    0.0%
Remote_Work_Frequency        0.0%
Team_Size                    0.0%
Training_Hours               0.0%
Promotions                   0.0%
Employee_Satisfaction_Score  0.0%
Resigned                     0.0%
dtype: object
```

Checking for duplicates and invalid values

```
#Verificamos que ya no hay valores invalidos

for c in lista_columnas:
    print(f'En la columna {c} los bbb son: {df[df[c] == 'bbb'].shape[0]}')
```

[301] ✓ 0.0s Python

```
En la columna Employee_ID los bbb son: 0
En la columna Department los bbb son: 0
En la columna Gender los bbb son: 0
En la columna Age los bbb son: 0
En la columna Job_Title los bbb son: 0
En la columna Hire_Date los bbb son: 0
En la columna Years_At_Company los bbb son: 0
En la columna Education_Level los bbb son: 0
En la columna Performance_Score los bbb son: 0
En la columna Monthly_Salary los bbb son: 0
En la columna Work_Hours_Per_Week los bbb son: 0
En la columna Projects_Handled los bbb son: 0
En la columna Overtime_Hours los bbb son: 0
En la columna Sick_Days los bbb son: 0
En la columna Remote_Work_Frequency los bbb son: 0
En la columna Team_Size los bbb son: 0
En la columna Training_Hours los bbb son: 0
En la columna Promotions los bbb son: 0
En la columna Employee_Satisfaction_Score los bbb son: 0
En la columna Resigned los bbb son: 0
```

```
df.duplicated('Employee_ID').sum() #Comprobamos que no hay duplicados
```

[302] ✓ 0.0s Python

```
0
```

Exploratory data analysis

Overview

```python
df
```

Python

	Department	Gender	Age	Job_Title	Hire_Date	Years_At_Company	Education_Level	Performance_Score	Monthly_Salary	Work_Hours_Per_Week	Projects_Handle
0	Finance	Male	29	Developer	2024-04-18 08:03:05.556036	0	High School	5	7500	34	3
1	Finance	Male	41	Specialist	2015-10-26 08:03:05.556036	8	High School	3	6404	37	2
2	Marketing	Female	48	Analyst	2016-10-22 08:03:05.556036	7	Bachelor	2	4800	52	1
3	Engineering	Female	36	Specialist	2021-07-23 08:03:05.556036	3	Bachelor	2	4800	38	1
4	IT	Male	43	Manager	2016-08-14 08:03:05.556036	8	High School	3	7800	46	3
...	...	...	...	...	...	...	...	...	...	...	
92978	Engineering	Female	27	Specialist	2019-07-23 08:03:05.556036	4	Bachelor	3	5400	40	2
92979	Operations	Male	35	Analyst	2017-08-29 08:03:05.556036	4	Bachelor	2	4800	45	1
92980	Sales	Male	41	Specialist	2015-02-01 08:03:05.556036	9	High School	4	6300	59	4
92981	Marketing	Male	33	Technician	2016-10-22 08:03:05.556036	7	Bachelor	3	4550	34	3
92982	Operations	Male	39	Specialist	2020-12-24 08:03:05.556036	3	Bachelor	3	5850	34	3

92983 rows × 19 columns

The data frame I was working with had 92983 rows and 19 columns, the data collects different employee information such as department, gender, age, job title, and so on. The total number of records was 1859660 data.

Types of variables

```
df.dtypes
```

```
Department                          object
Gender                              object
Age                                  int64
Job_Title                           object
Hire_Date                   datetime64[ns]
Years_At_Company                     int64
Education_Level                     object
Performance_Score                    int64
Monthly_Salary                       int64
Work_Hours_Per_Week                  int64
Projects_Handled                     int64
Overtime_Hours                       int64
Sick_Days                            int64
Remote_Work_Frequency                int64
Team_Size                            int64
Training_Hours                       int64
Promotions                           int64
Employee_Satisfaction_Score        float64
Resigned                              bool
dtype: object
```

The data types in the columns are of type object which stores lines of text, int64 which stores numeric data without decimals, float64 which stores numeric data that has decimal numbers and boolean which is binary data such as true or false.

In this data frame department, gender, job title, education level describe employee characteristics in text format. While age and performance score also describe the characteristics, but in numerical format.

Resigned is of type boolean and stores if any employee has left the company.

Employee satisfaction score is of type float and stores the level of employee job satisfaction.

Summary statistics

	Age	Hire_Date	Years_At_Company	Performance_Score	Monthly_Salary	Work_Hours_Per_Week	Projects_Handled	Overtime_Hours	Sick_Days
count	92983.000000	92983	92983.000000	92983.000000	92983.000000	92983.000000	92983.000000	92983.000000	92983.000000
mean	41.014938	2019-09-14 01:08:27.098683904	4.460428	2.997118	6404.333889	44.948066	24.400977	14.535238	7.016272
min	22.000000	2014-09-07 08:03:05.556036	0.000000	1.000000	3850.000000	30.000000	0.000000	0.000000	0.000000
25%	32.000000	2017-03-18 08:03:05.556036096	2.000000	2.000000	5400.000000	38.000000	12.000000	7.000000	3.000000
50%	41.000000	2019-09-19 08:03:05.556036096	4.000000	3.000000	6404.000000	45.000000	24.000000	15.000000	7.000000
75%	50.000000	2022-03-13 08:03:05.556036096	7.000000	4.000000	7200.000000	52.000000	36.000000	22.000000	11.000000
max	60.000000	2024-09-03 08:03:05.556036	10.000000	5.000000	9000.000000	60.000000	49.000000	29.000000	14.000000
std	10.904101	NaN	2.813624	1.373275	1331.012991	8.672310	14.179203	8.491062	4.198860

Remote_Work_Frequency	Team_Size	Training_Hours	Promotions	Employee_Satisfaction_Score
92983.000000	92983.000000	92983.000000	92983.000000	92983.000000
50.079584	10.009841	49.555747	1.000678	2.998564
0.000000	1.000000	0.000000	0.000000	1.000000
25.000000	5.000000	26.000000	0.000000	2.070000
50.000000	10.000000	50.000000	1.000000	2.998583
75.000000	15.000000	73.000000	2.000000	3.930000
100.000000	19.000000	99.000000	2.000000	5.000000
34.282087	5.385067	27.994939	0.799358	1.116361

In the statistical summary we can see several data, being the most relevant, the age where the average is around 41 years old, with a minimum of 22 and a maximum of 60 years old, being that 75% of the data is below 50 years old.

It can be observed that the oldest date of hiring is 2014 and the most recent is 2024, which means that the company has been operating for approximately 10 years.

In terms of performance score it can be observed that the score scale goes from a minimum of 1 to a maximum of 5, being that the average performance is approximately 3 points, it can be observed that 25% of the employees have a lower performance, that is, almost a quarter of the employees present a bad performance, while only 25% of the employees have a performance higher than 4.

The average number of hours worked in a week is 45, assuming that it is a working week from Monday to Friday, which means that they work about 9 hours a day, however, it can be seen that the maximum working day recorded is 60 hours, which would imply that sometimes they may work Saturdays and 10 hours a day, not counting overtime.

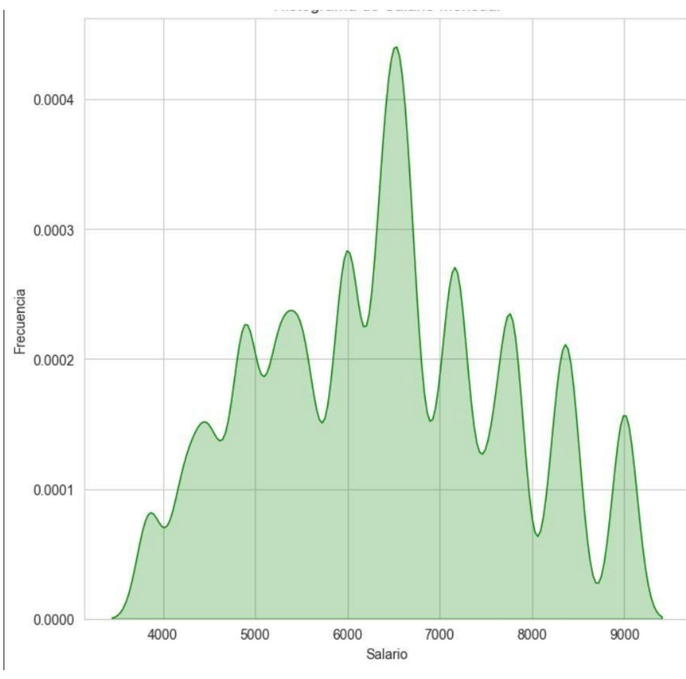
Speaking of overtime, it can be observed that on average they work 14 hours of overtime per week, that is, 2 hours of overtime daily, which would increase the average workday from 9 hours to 11 hours in the best case scenario, since there is a record of 29 hours of overtime worked in a week, which would considerably increase the average workday.

The monthly salary within the company is maintained at a minimum of 3,850 and a maximum of 9,000.

The projects managed in the company can vary from 0, being the minimum, and a maximum of 50. 75% of the data is kept below 36.

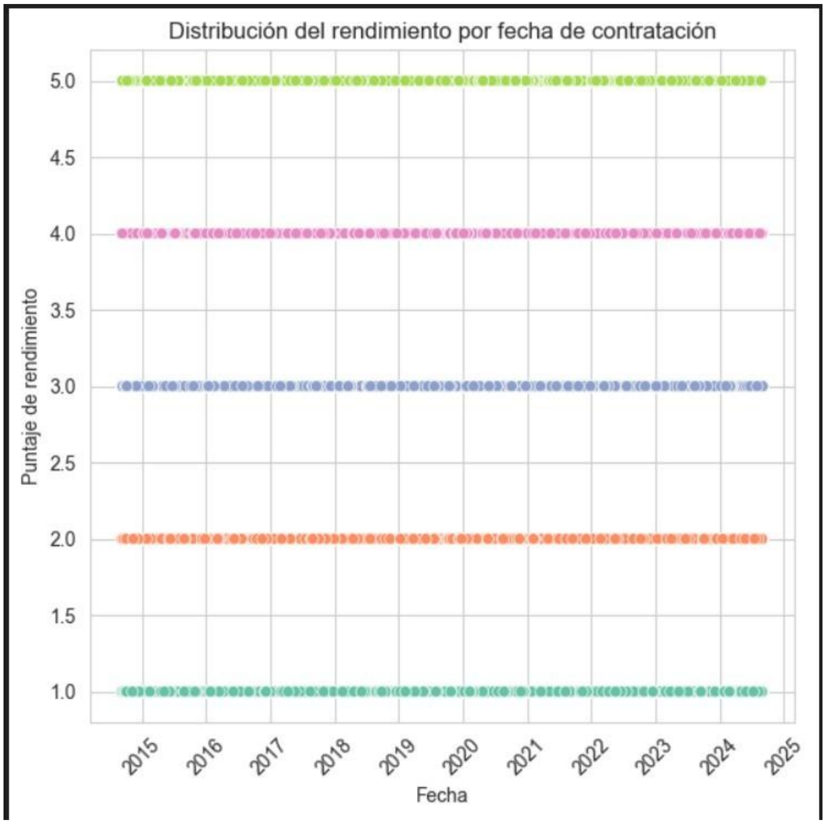
Visualization and Distribution of Individual Variables

Monthly Salary Density Graph

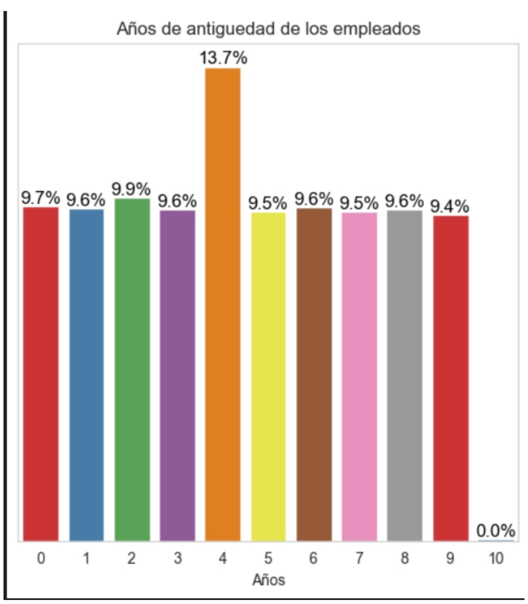


Most of the monthly salaries are between 6000 and 7000, however, it can be seen that there are a lot of peaks and valleys which could indicate that salaries are highly variable in the company.

Dispersion of performance by year of hire

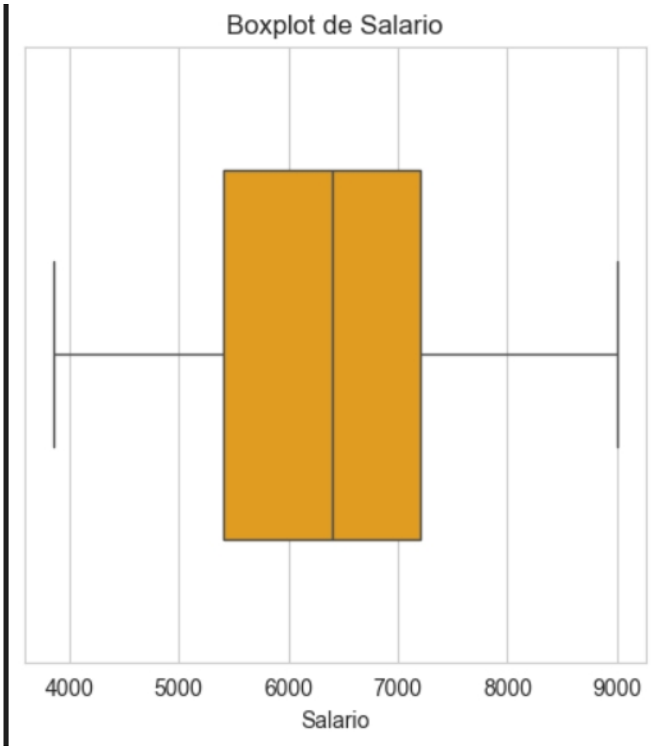


Performance is not affected by date of hire. Bar chart measuring employee seniority



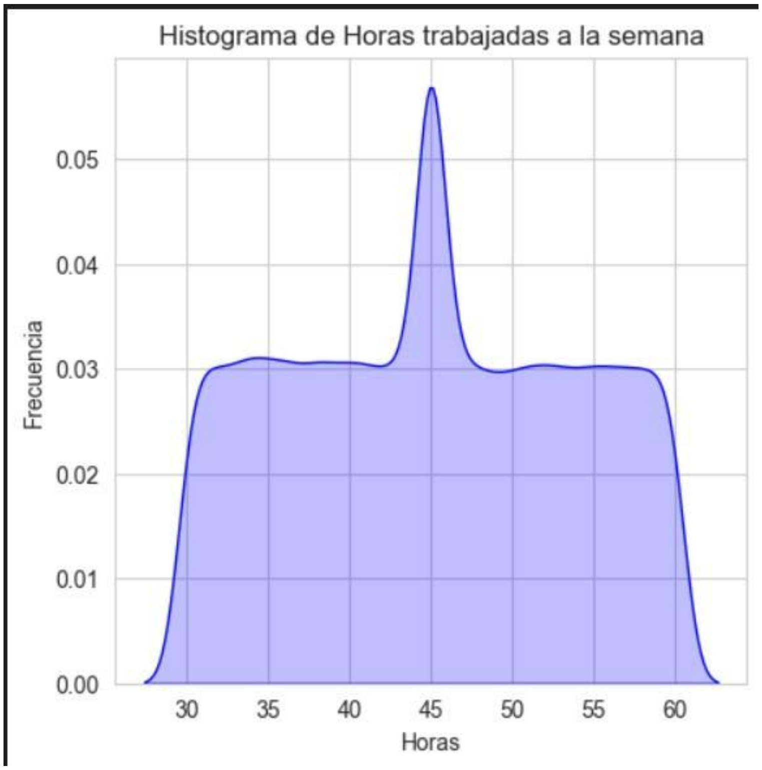
It can be observed that most of the personnel have 4 years of seniority in the company, all other years have the same frequency, however, there are no personnel with 10 years in the company. It could indicate that there is a bad working environment that makes employees resign.

Monthly salary boxplot



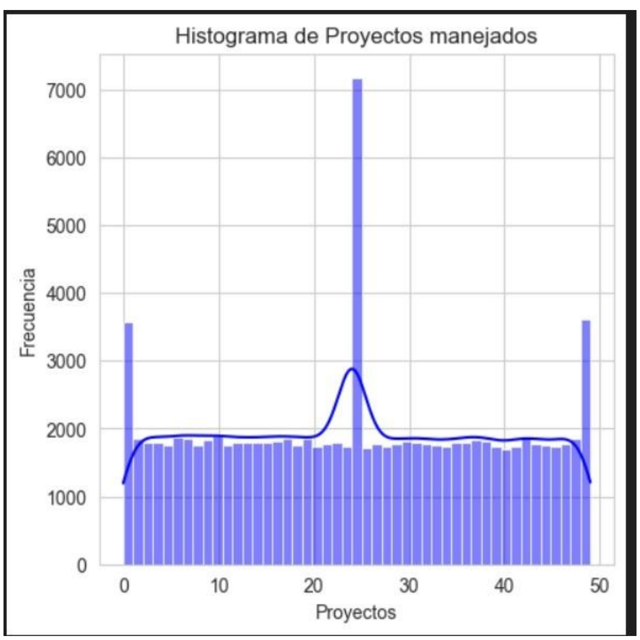
50% of wages are between 6000 and 7000, whiskers show wages ranging from approximately 3900 to 9000, there are no outliers.

Density of weekly hours worked



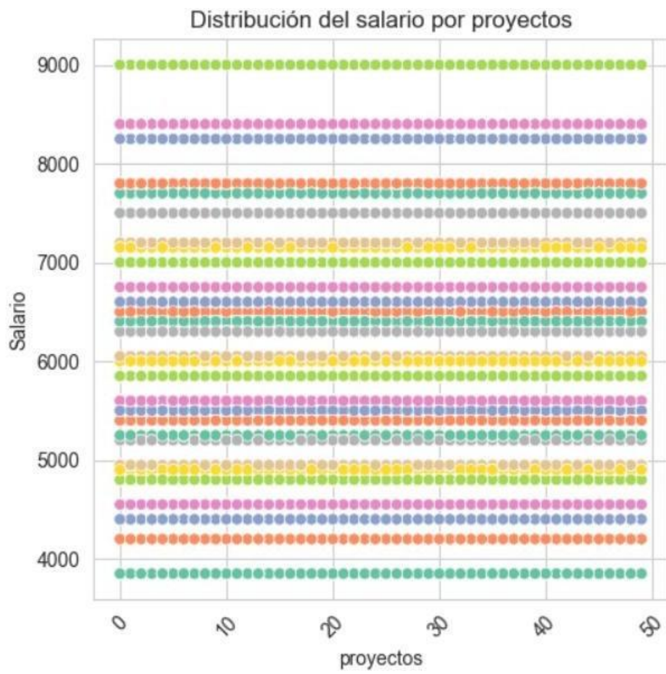
The weekly hours worked are evenly distributed, however, it can be observed that there is a peak at 45 hours, indicating that there is a large portion of the staff working more than the rest.

Histogram of projects handled



There are 3 peaks in the number of projects handled per employee, being that there are many staff handling between 20 and 30 projects, but it can also be seen that there is a significant segment that handles 50, the worrying thing is that the same number of employees handle 0 projects. This could be due to the job position or a lack of responsibility and commitment.

Salary distribution by project

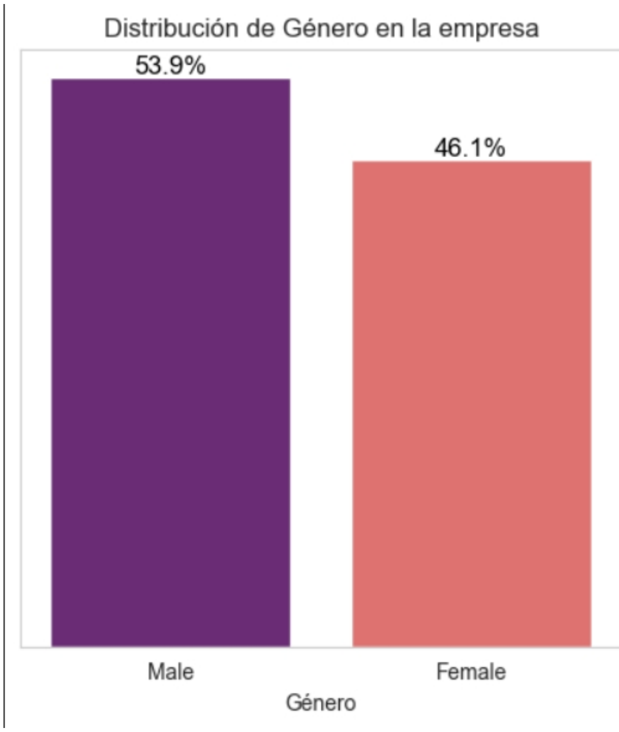


Salary is not affected by the number of projects handled; it can be observed that someone with 0 projects can earn from 3900 to 9000 as well as someone who has worked on 50 projects.

Density of employee satisfaction

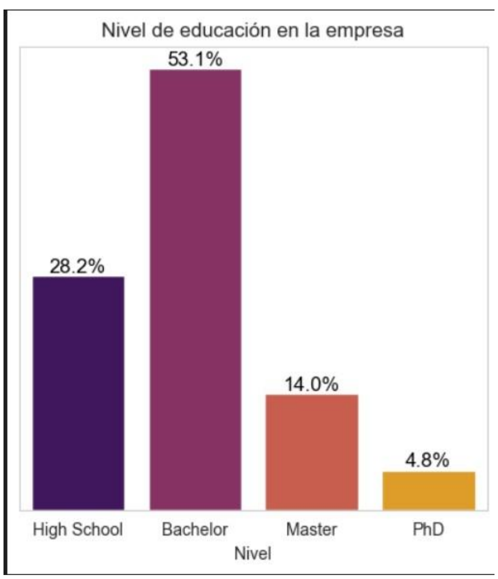
Employee satisfaction has a uniform density with a peak at 3. That is, satisfaction is neither good nor a bad company.

Gender in the company

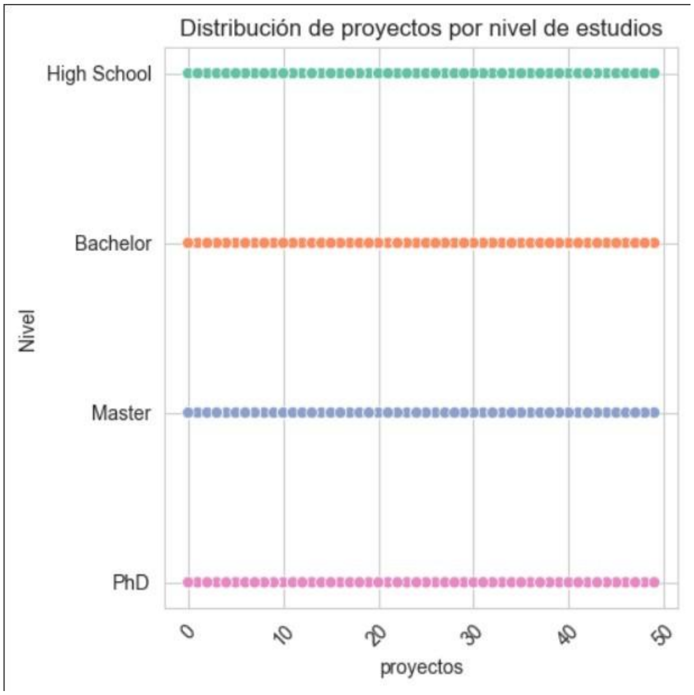


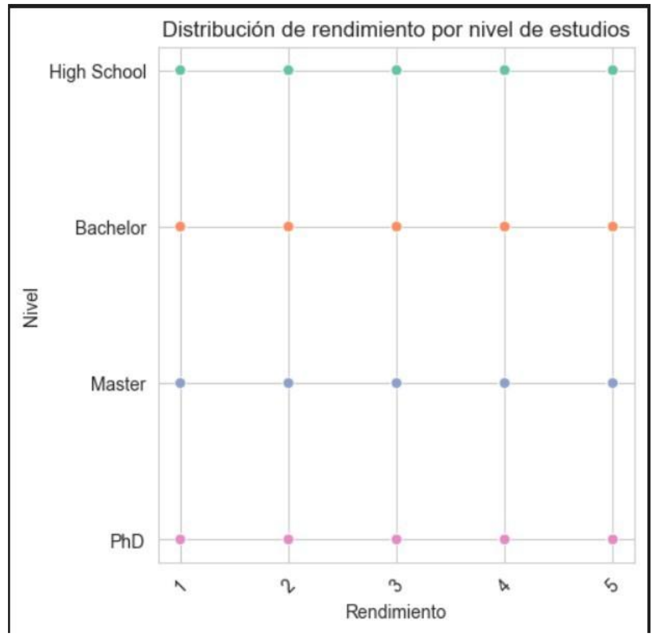
More than half of the staff is male.

Level of education in the company

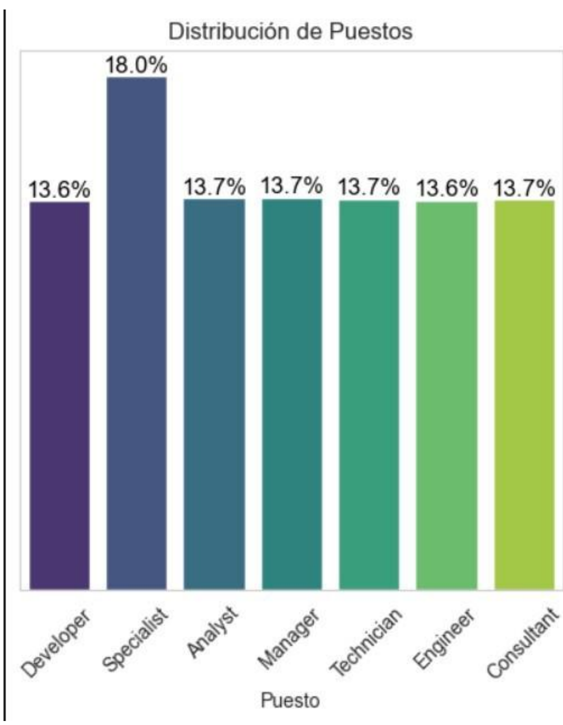


Staff in the company mostly have bachelor's and high school levels, with very few master’s and doctorates. However, this does not affect performance or salary as shown in the following graphs:



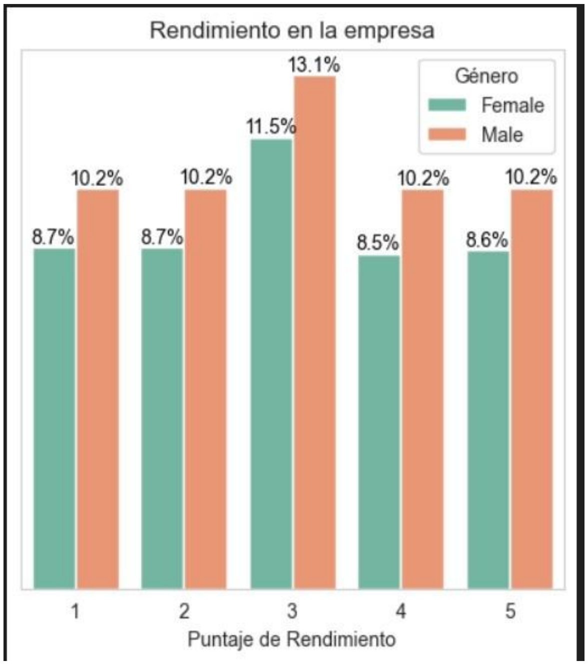


Distribution of positions in the company.



The company has more employees as specialists than in any other position.

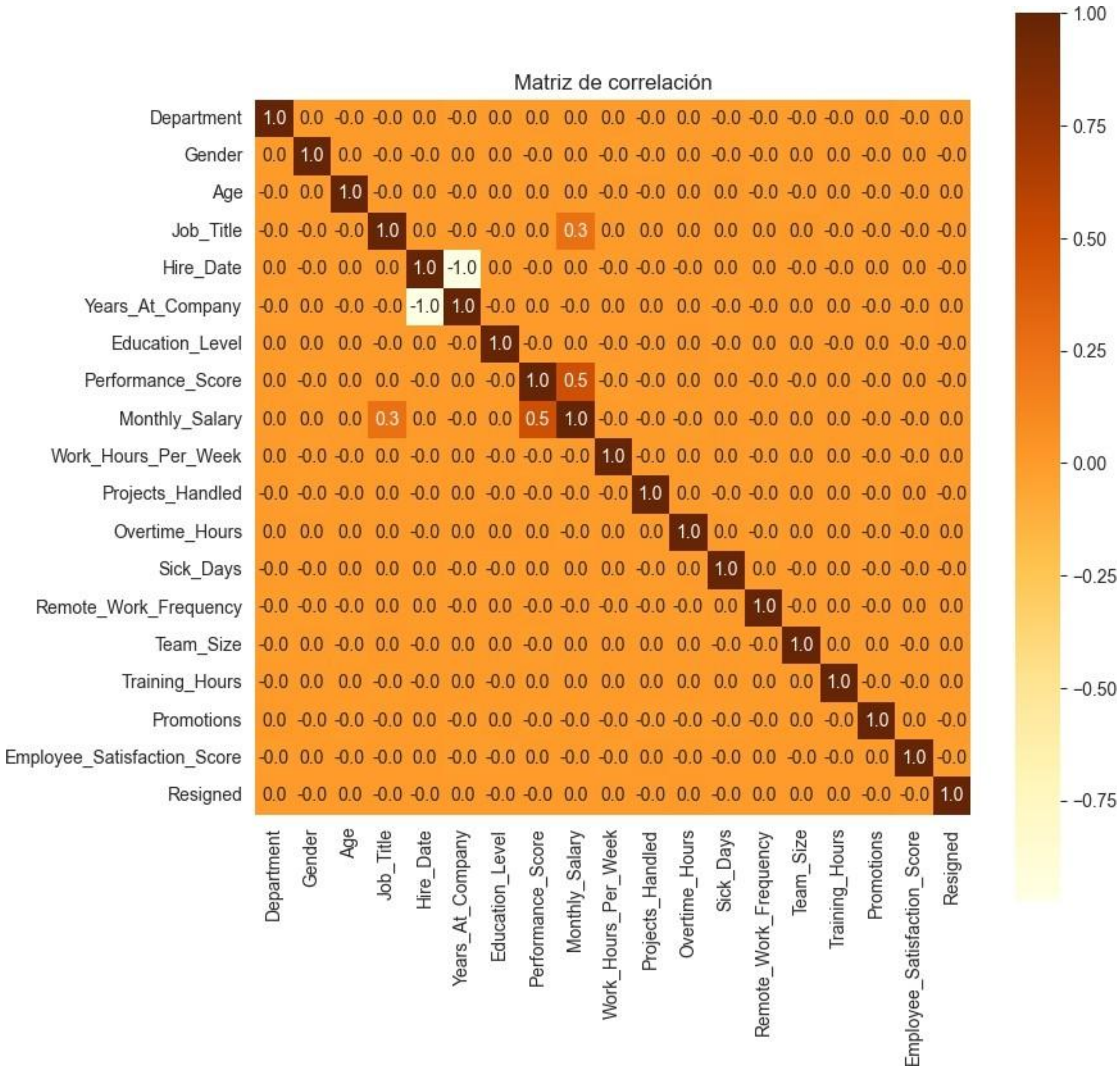
Company performance by gender



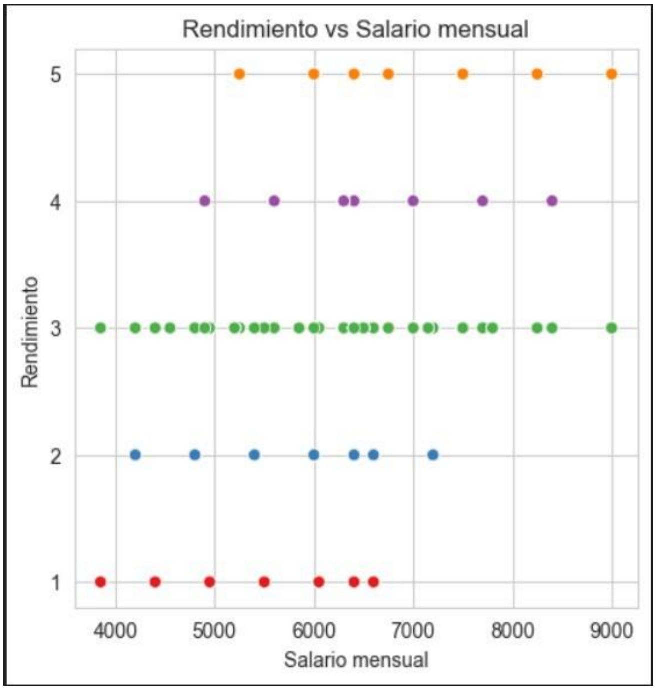
The most common performance in men and women is 3 on a scale ranging from 1 to 5, it can be observed that the difference between men and women is 2% per category, this is only a consequence of the fact that there are more men than women in the company.

Correlation between Variables

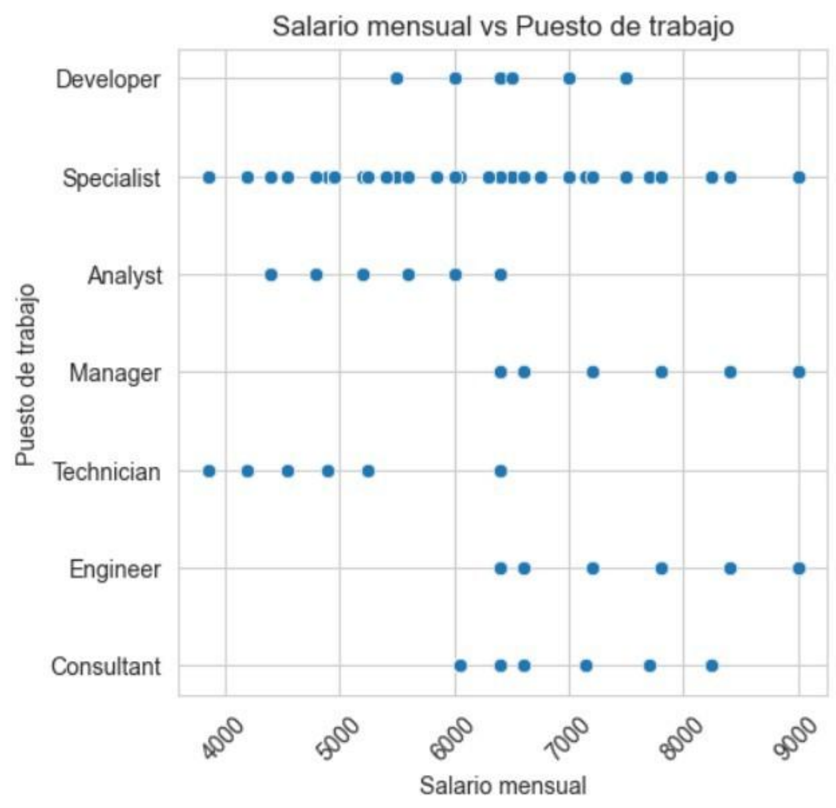
Correlation matrix



All the data in the dataframe have no correlation, except for the performance score and monthly salary. However, it is very low which means that it is not feasible to make a linear regression model, so I decided to make a decision tree.



Different salaries are clustered in a 3-point performance. Those with performances of 1 and 2 are mostly in the range of 3900 to 6500-6400, this suggests that those with lower-than-average salaries tend to have lower performance. At the same time, those with performance above 3 are slightly more clustered to the right where the highest salaries are found. However, this cannot be taken as an indicator that salary determines performance since it can be observed that different salaries are in a performance of 3 and even that salaries that fall within the low performance range have high performance.



It can be observed that the specialist position covers most of the salaries, this is since it is the position that has the most personnel, so it can be assumed that it is the position with the most varied performance.

In this graph the job position is not necessarily an indicator of good performance since there are positions that usually have a higher-than-average salary and still have a low performance.

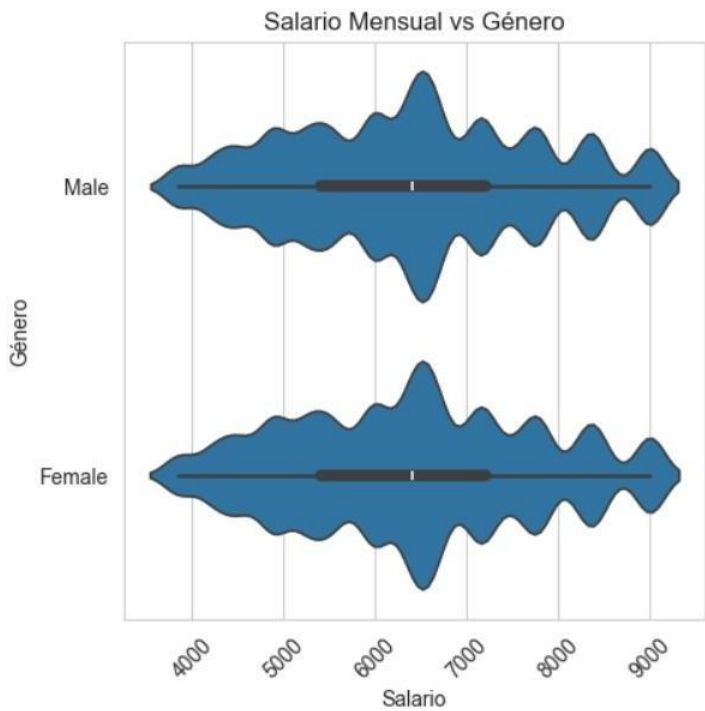
Missing Value Analysis

To clean up missing values, impute the values with the mode of each column.

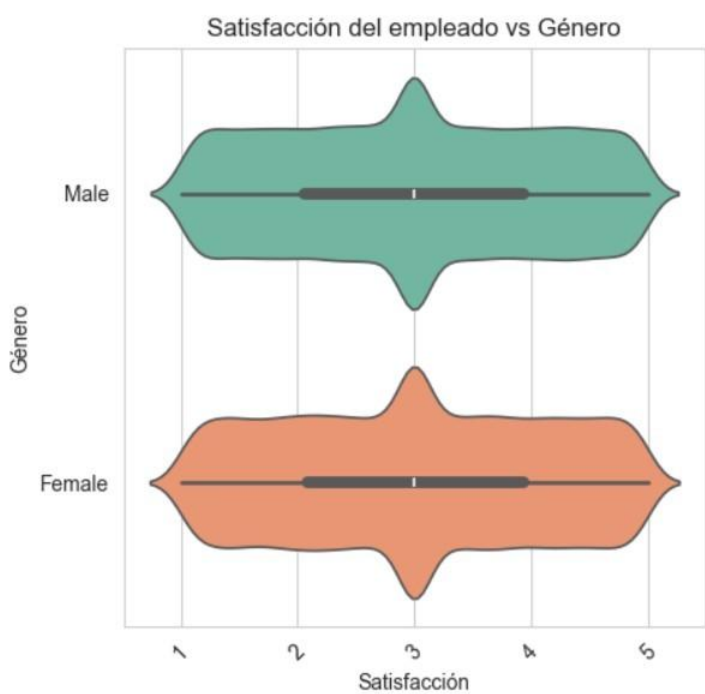
```
categorical_columns = ['Department', 'Job_Title', 'Education_Level' , 'Gender']
for col in categorical_columns:
    df[col] = df[col].replace(['Sin dato','Other'], pd.NA)

# Imputar los valores faltantes con la moda de cada columna
for col in categorical_columns:
    mode_value = df[col].mode().iloc[0]  # Obtener la moda
    df[col].fillna(mode_value, inplace=True)

# Confirmar que los valores han sido reemplazados
print(df[categorical_columns].isnull().sum())
```

✓ 0.0s

Relationship between Categorical and Numeric Variables



It can be observed that salaries do not differ between genders.



It can be observed that employee satisfaction has the same distribution in both men and women.

Important Observations and Findings

Performance is influenced, to some extent, by salary and job position. According to the scatter plots, those with a higher salary tend to perform better. As for job position, although the direct correlation is almost nonexistent, an indirect relationship could be assumed. This is because some positions carry greater responsibilities, but these are not always reflected in the salary, which could contribute to poor performance.

On the other hand, neither the weekly hours worked, nor the number of projects managed seem to impact on the salary of employees. As observed, a person with 0 projects can earn the same as one who manages 50 projects. This work environment can be extremely tiring and stressful, with working hours that can reach 9 to 11 hours a day without this being reflected in the monthly salary. However, as the correlation is very low, they will not be included in the model.

For the model this implies that the determining characteristics can be taken and thus create a decision tree to be able to predict the performance of future employees with the conditions they will have and to be able to make the necessary adjustments to achieve a good performance.

Machine Learning Model

The model that will be used for this project is a decision tree, which will be used to predict the performance of future employees based on the characteristics identified as influential. This model will allow us to optimize job performance.

Implementation and Training

```python
data = df_corr[['Performance_Score' , 'Job_Title'  , 'Monthly_Salary']]
x = data.drop('Performance_Score' , axis=1)
y = data['Performance_Score']
```
✓ 0.0s Python

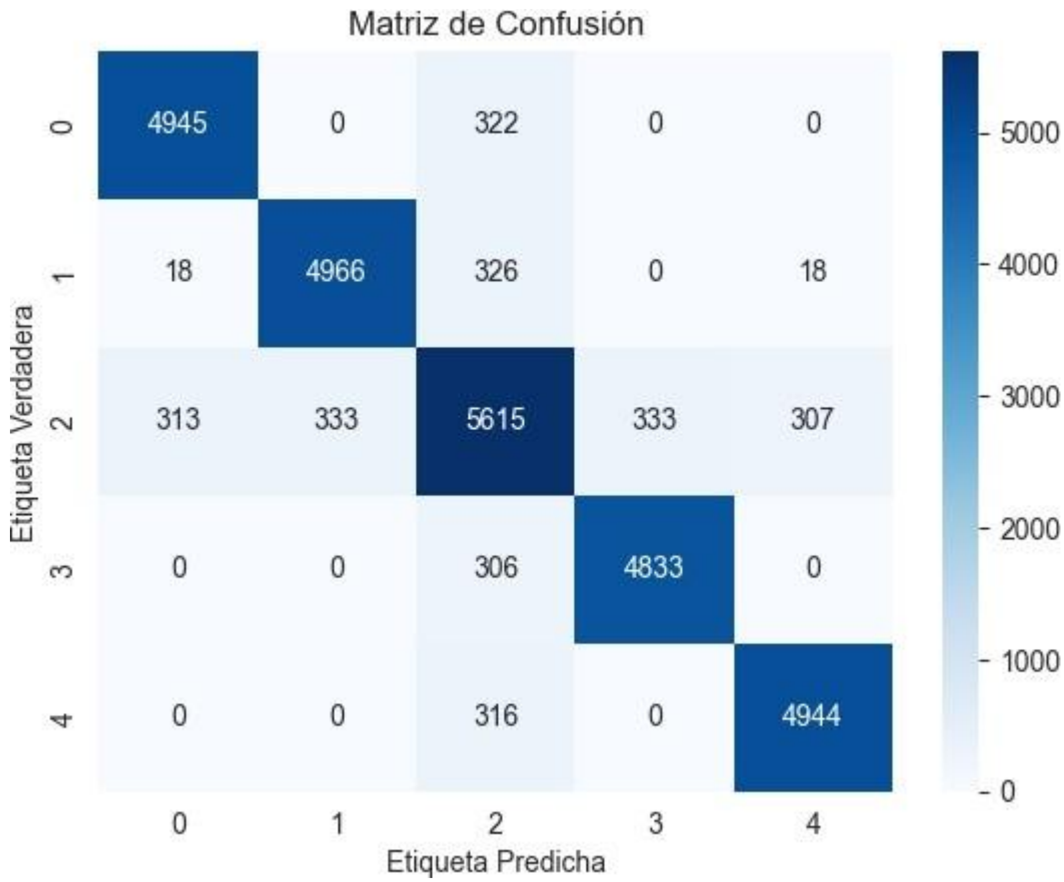
```python
#Dividir en entrenamiento y prueba

x_train , x_test , y_train , y_test = train_test_split(x , y , test_size=0.3 , random_state=42)
```
✓ 0.0s Python

```python
model = DecisionTreeClassifier(random_state=42)
model.fit(x_train , y_train)
```
✓ 0.0s Python

```
DecisionTreeClassifier
DecisionTreeClassifier(random_state=42)
```

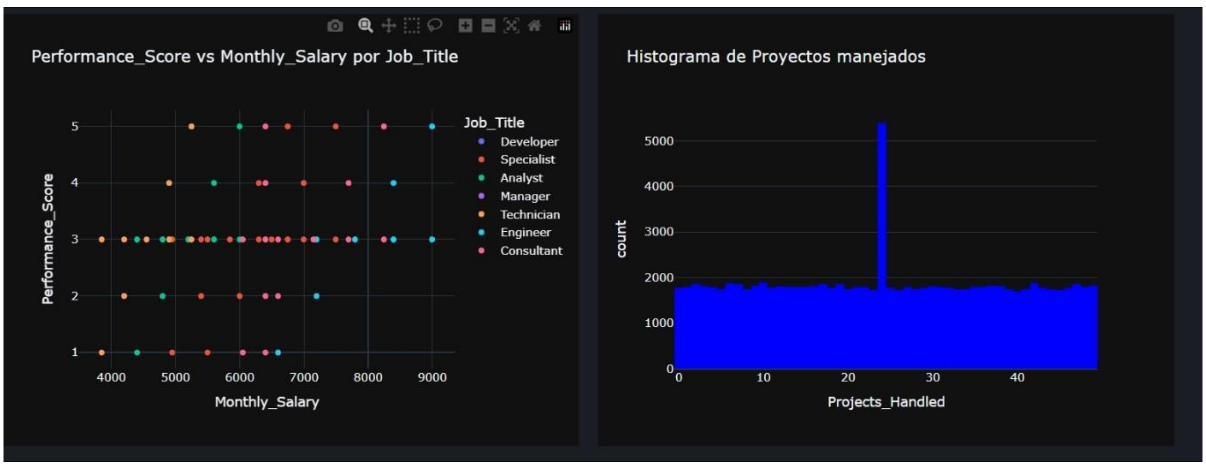
```python
y_pred = model.predict(x_test)
```
✓ 0.0s Python

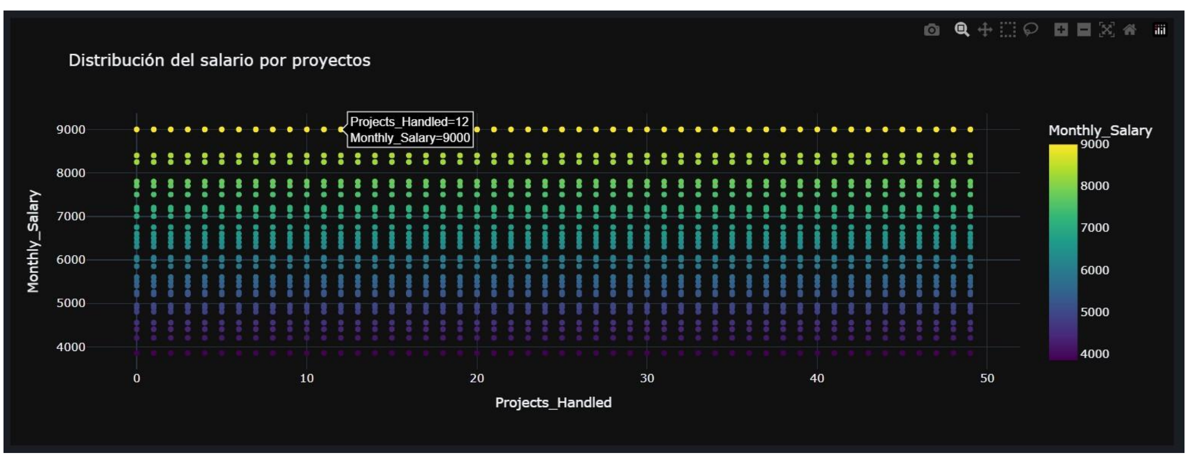
Results



The model after training is quite accurate with very few cases of erroneous predictions, being that most of the errors are found in label 2 which corresponds to performance level 3, this is caused by the large variety of data that usually has that level of performance.

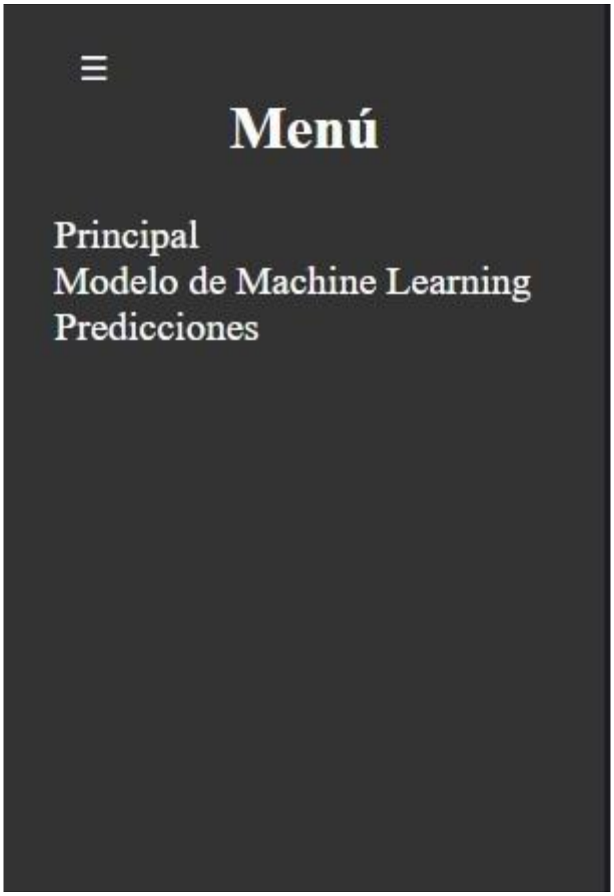
Dashboard



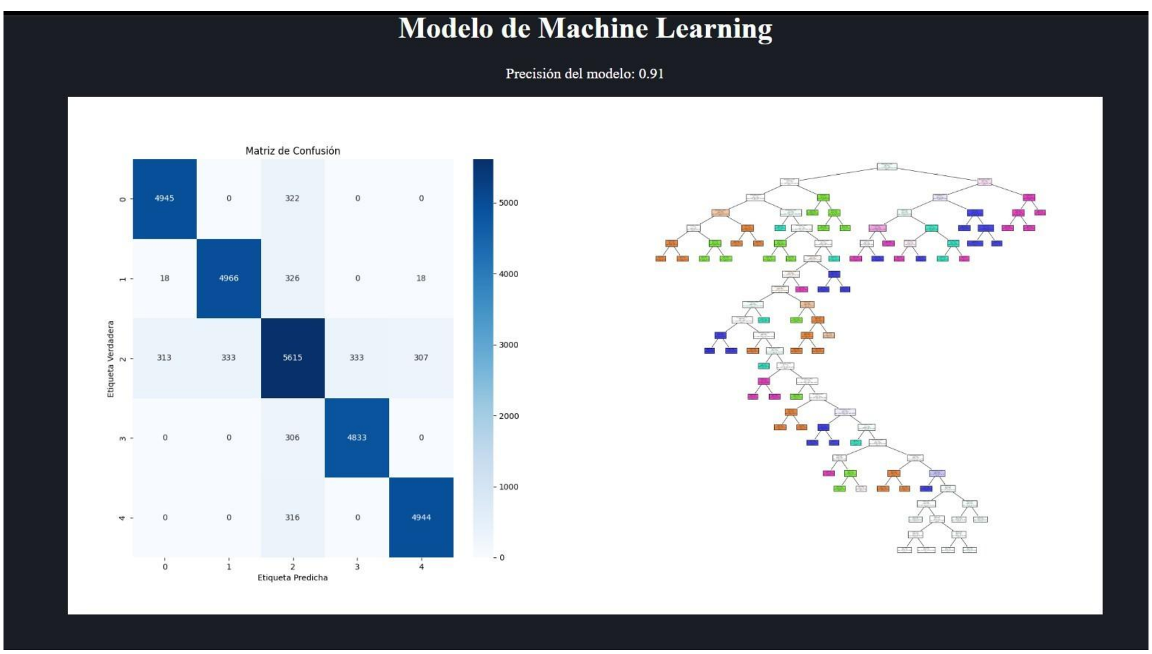


The main part of the dashboard includes graphs relevant to the analysis such as hours worked per week, monthly salary vs. performance, projects managed and salary distribution per project.

The dashboard also includes a drop-down menu with the different sections



The Machine Learning section includes information about the model, such as the accuracy and the graphed tree.



Finally, the predictions section has a simple data capture interface.

The dashboard can be useful for clients and users as it presents important information about employee performance, focused on different points, as well as a section to make predictions.

Conclusions and Future Lines of Work

The main objective of this project was to analyze the performance of the employees and what were the characteristics of the employees that tended to have a higher performance, we managed to find that salary was a factor that influenced performance, however, it is not the only factor.

To improve the analysis, I would like to do more in-depth data collection, such as how employees rate the offices, if they have experienced any harassment, distance they live from the job. In addition, I would highly recommend investigating employees' feelings about working hours, salary and salary distribution.