PREDICTING EMPLOYEE CHURN

Gracie Inman
DSC 630
Professor Hua
Final Paper
03/01/24

Final Paper

Introduction

Employee retention is a problem that companies across the United States face. Replacing employees can be costly for a company. According to Sam Steady, “A study by the Center for American Progress found that companies typically pay about one-fifth of an employee’s salary to replace that employee, and the cost can significantly increase if executives or highest-paid employees are to be replaced.” (2022). Not to mention there is a significant number of benefits to having successful employee retention. According to Marc Holliday, the 10 benefits of retaining employees are cost reduction, morale improvement, experienced employees, increased productivity, recruitment and training efficacy, better customer experiences, improved company culture, better employee experiences, increased revenue, and increased employee satisfaction (Holliday, 2021). The job market is always changing, but “According to Gartner, the pace of employee turnover is forecast to be 50–75% higher than companies have experienced previously, and the issue is compounded by it taking 18% longer to fill roles than pre-pandemic” (Tupper & Ellis, 2022). Companies and corporations being able to target and prevent employee turnover will benefit not only the companies but also their employees. The HR dataset was obtained from Kaggle (Steady, 2022). The dataset models employee surveys, workload, and other attributes such as salary and if they left the company which were used to create a model to predict if an employee was likely to leave.

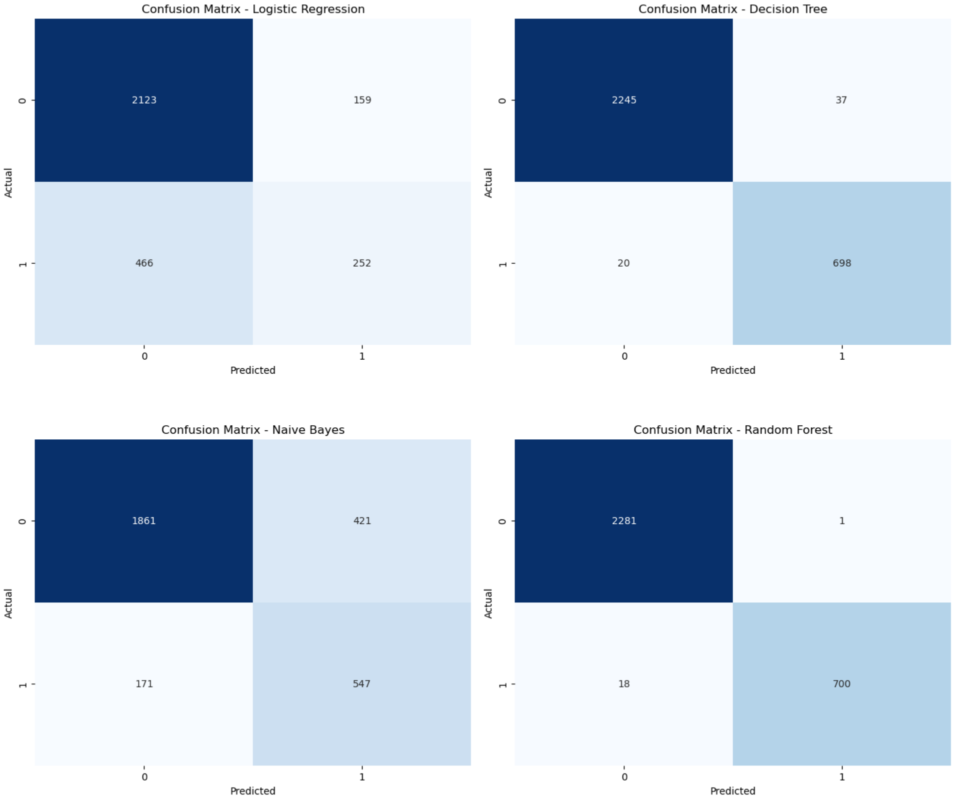
Methods/Results

The data was prepared by first importing the data into Jupyter Notebooks. The dataset was checked for missing values and only 2 entries out of 14,999 were found to contain missing values. Due to the small number of missing values, it was determined that dropping the values would be the best method as it would have very little or no effect on the overall analysis. The column salary needed to be encoded to allow for an accurate analysis. The employee-id column was then dropped as it was not necessary for analysis. When exploring the data, several charts were made to explore each column and its distribution. While some data appeared to be skewed, it was determined that the skew was an accurate representation of the data. It was noted and analysis continued.

When building models, logistic regression, decision trees, Naive Bayes, and random forest were chosen for their ability with classification tasks. The data was split, and each model was trained. Precision, accuracy, and a confusion matrix were used to evaluate each model shown below. These metrics were chosen because together they give a high-level overview of model performance.

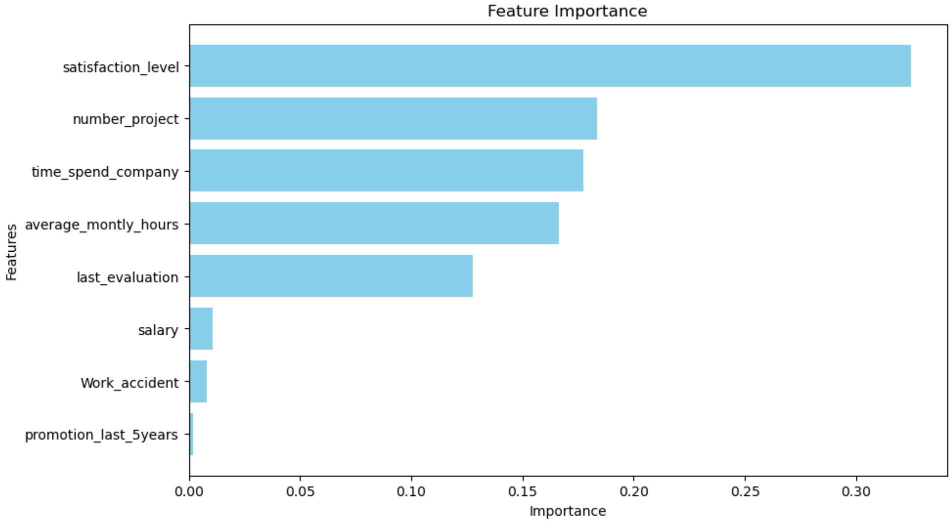
Table 1: Accuracy and Precision Scores by Model

Model	Accuracy	Precision
Logistic Regression	0.7917	0.6131
Decision Tree	0.9810	0.9497
Naïve Bayes	0.8027	0.5651
Random Forest	0.9937	0.9986



When considering all of the metrics and models, it appears that the random forest model appears to perform the best with this dataset. Random forest had the highest accuracy (0.9937) and precision (0.9986) of all the models. Looking at the confusion matrix, it also contained the highest number of true positives and true negatives as well as the least false negatives and false positives.

Feature importance was used to determine the most important attribute in predicting if an employee will leave or not. Shown below is a graph depicting the feature importance of each feature. The most important feature was determined to be the satisfaction level of the employee. This is followed by the number of projects and then the amount of time an employee has spent at a company.



Conclusion

Throughout this project, I have learned key steps and factors to consider when performing an analysis. I learned that employee churn is a real-world problem that companies are facing. For this dataset, random forest was the best model and employee satisfaction was the most important feature. I would recommend the company of this dataset look into ways to improve their employee satisfaction rating. Before the model is deployed it should be tested against real-world data to confirm the model is performing as expected. The data used did not state where it was obtained. This can cause ethical concerns regarding data accuracy and sourcing. The model should be confirmed using ethically sourced data that has a known source. When presenting the data, it will also be important to be transparent about the source of the data and my concerns. In addition, two entries were removed due to being incomplete. While this is not likely to affect results, it is still something to consider. While the model has the potential to benefit companies and employees, it could be used to negatively target employees who are already likely to leave. While little could be done to make sure the model is used ethically, offering training and support could help.

Milestone 4

In the preparation of the data, a number of factors were considered. Out of the 14,999 data entries, only 2 were incomplete. The two rows were dropped as it was an insignificant amount of data and was unlikely to affect the overall results. The salary column contained text entries such as “low”, “medium”, and “high” which were mapped to numerical values (low:0, medium:1, high:2) for analysis. Most of the variables within the dataset were already converted to Boolean so no additional conversions were needed. The employee ID column was dropped as it was unnecessary for analysis. It was determined in milestone 3 by using visualizations that there were no obvious outliers or data issues, and the data was split into training and testing sets.

Based on knowledge of models, logistic regression, naïve bayes, decision tree, and random forest were performed in hopes of finding a model that would be accurate in predicting whether an employee would turn. These models were chosen because they are all used for classification and have the ability to work with numerical and categorical values. Depicted below are the accuracy and precision scores for each model and the confusion matrices.

Table 1: Accuracy and Precision Scores by Model

Model	Accuracy	Precision
Logistic Regression	0.7917	0.6131
Decision Tree	0.9810	0.9497
Naïve Bayes	0.8027	0.5651
Random Forest	0.9937	0.9986

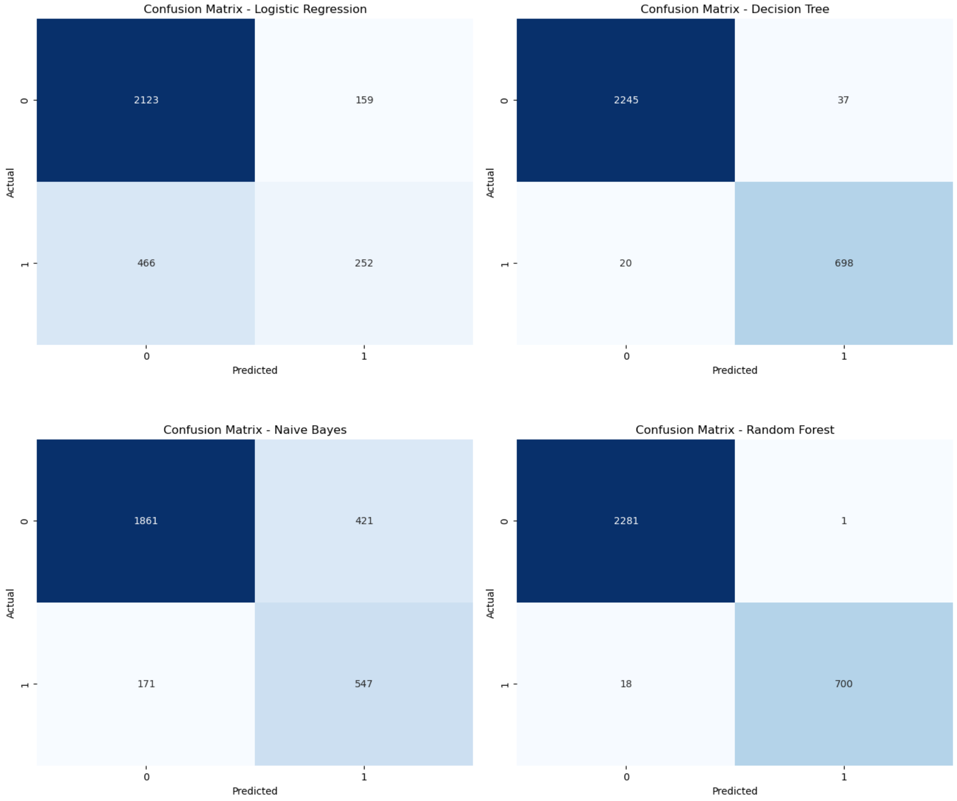


Figure 1 A-D: Illustrates the confusion matrix created for each model.

Based on the current analysis of the data, it was determined the best model for the dataset is a random forest model. This is confirmed by looking at the confusion matrix which has the highest number of true positives and the lowest number of false positives and negatives.

The goal of the project was to be able to successfully build a model that would help identify employees that were likely to churn. Employee retention is key to a successful business. According to Marc Holliday, the 10 benefits of retaining employees are cost reduction, morale improvement, experienced employees, increased productivity, recruitment and training efficacy, better customer experiences, improved company culture, better employee experiences, increased revenue, and increased employee satisfaction (Holliday, 2021). Employee retention benefits both the company and its employees.

Feature importance was performed to determine which feature was most important to determine if an employee would leave. As shown below, the most important feature was determined to be employee satisfaction. The next important feature was the number of projects. Salary, work accident, and promotion in the last five years were not significantly important in determining if an employee will quit.

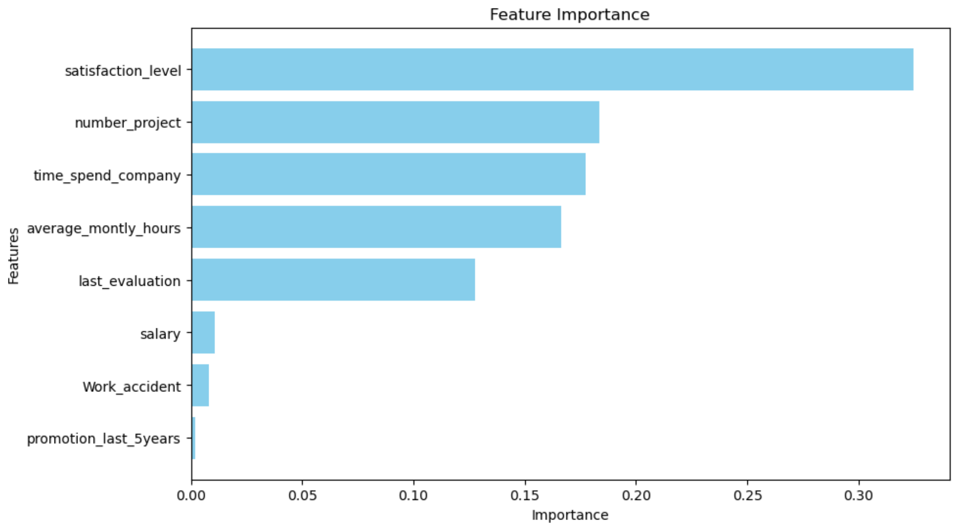
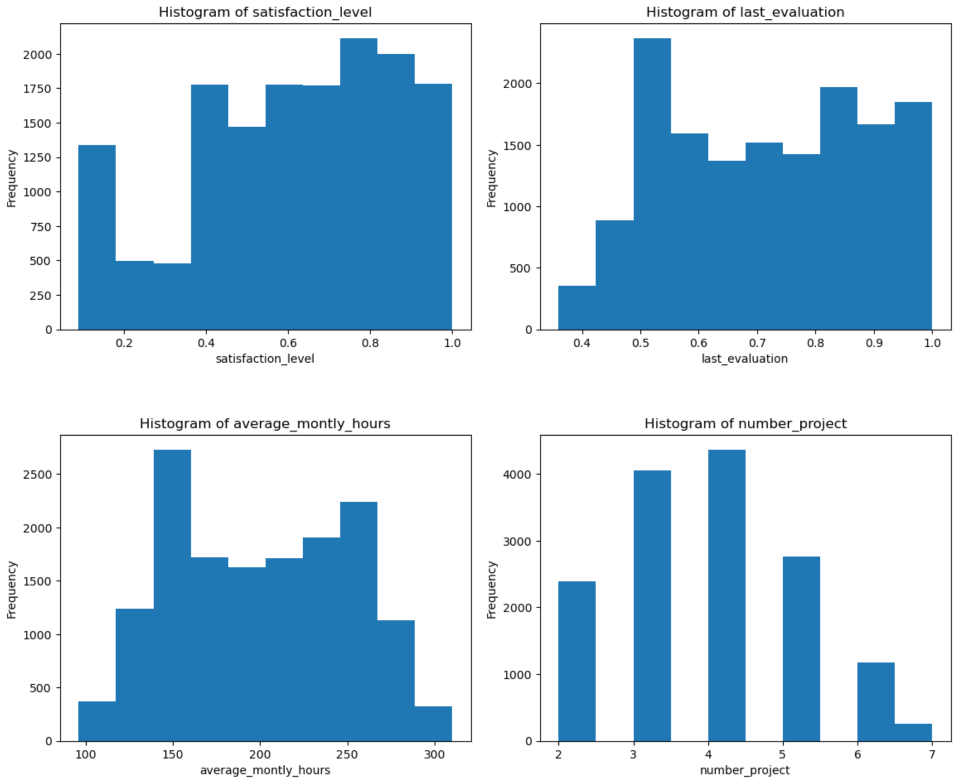


Figure 2: Illustrates feature importance for each variable.

Random forest was the best model for the dataset shown by the 99.4% accuracy, and 99.9% precision, and demonstrated by the confusion matrix. Employee retention is key for businesses and can have many benefits as stated by Marc Holliday (2021). The most important feature was the employee satisfaction level. This makes sense considering if you are not fulfilled in your current position, you are more likely to leave for a better opportunity. To improve employee retention, I would recommend looking into ways to improve employee satisfaction.

Milestone 3

After exploring my data, many visualizations will be useful in analyzing the data. Some of the visualizations I plan to use are bar graphs, pie charts, and scatter plots. Bar graphs will give me more insight into how many of each categorical variable is present in the dataset. This will help in ensuring the balance of the data. Pie charts and donut charts will also help with this and help visualize the amount of each variable present concerning the entire dataset. Scatter plots will help determine clusters or trends within the data. Shown below are the plots obtained from the data.

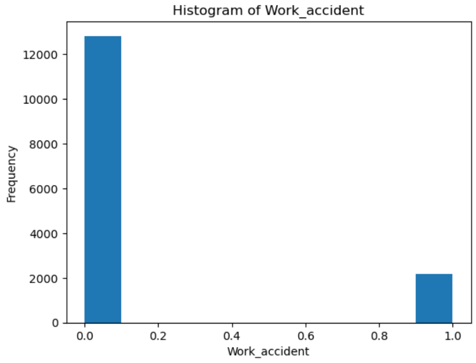


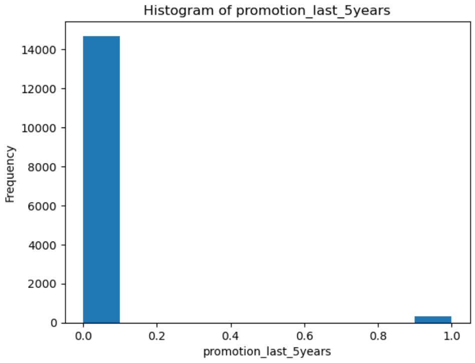
Histogram of satisfaction_level

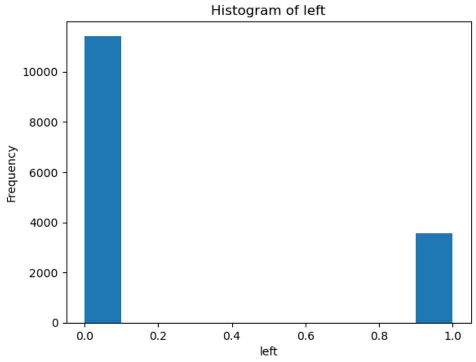
Histogram of last_evaluation

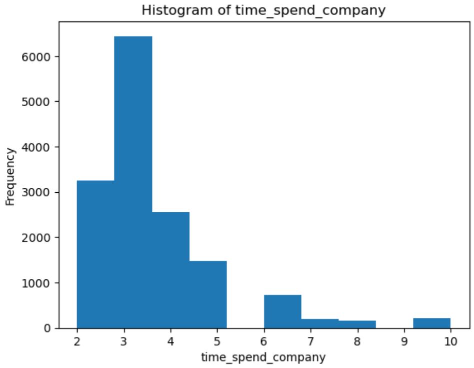
Histogram of average_montly_hours

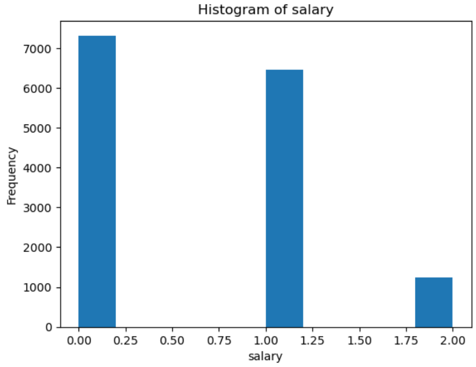
Histogram of number_project

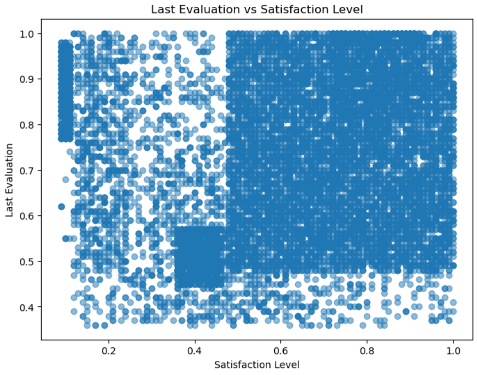

Histogram of Work_accident
Frequency
0
2000
4000
6000
8000
10000
12000
0.0
0.2
0.4
0.6
0.8
1.0
Work_accident

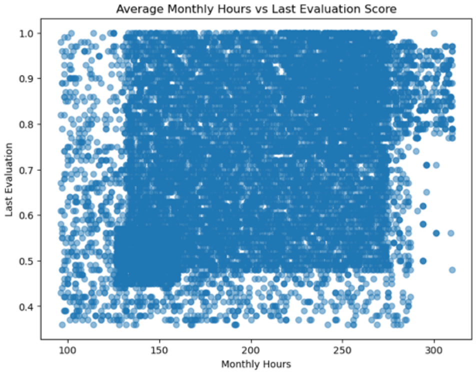

Histogram of promotion_last_5years
Frequency
0
2000
4000
6000
8000
10000
12000
14000
0.0
0.2
0.4
0.6
0.8
1.0
promotion_last_5years

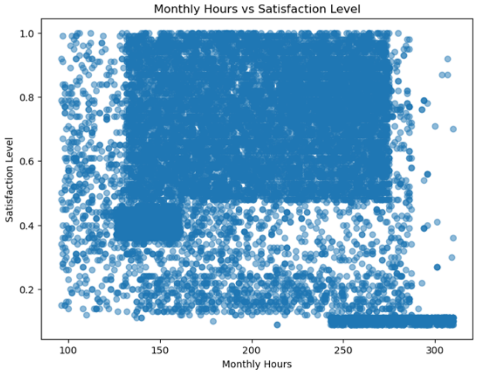

Histogram of left
Frequency
0
2000
4000
6000
8000
10000
0.0
0.2
0.4
0.6
0.8
1.0
left


Histogram of time_spend_company
Frequency
0
1000
2000
3000
4000
5000
6000
2
3
4
5
6
7
8
9
10
time_spend_company


Histogram of salary
Frequency
0
1000
2000
3000
4000
5000
6000
7000
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
2.00
salary


Last Evaluation vs Satisfaction Level
Last Evaluation
0.4
0.5
0.6
0.7
0.8
0.9
1.0
0.2
0.4
0.6
0.8
1.0
Satisfaction Level


Average Monthly Hours vs Last Evaluation Score
Last Evaluation
0.4
0.5
0.6
0.7
0.8
0.9
1.0
100
150
200
250
300
Monthly Hours


Monthly Hours vs Satisfaction Level
Satisfaction Level
0.2
0.4
0.6
0.8
1.0
100
150
200
250
300
Monthly Hours

According to the figures shown, the employee satisfaction level and last evaluation appear to be right skewed to a higher employee satisfaction level and evaluation. The number of hours monthly appears to be bimodal. While the number of projects appears to be skewed left. The graphs also show that significantly more people at the company have not had a workplace accident, been promoted in the last 5 years, or left the company. The number of years with the company currently is significantly skewed left meaning most people within the company have been there less than 5 years. It is also shown that a significant amount of the employees are in the low to medium range of salary. The last evaluation, satisfaction levels, and the average number of hours worked were visualized to see how each variable correlated with the other. This gave a good idea of where a majority of the employees were represented.

The goal of being able to predict if an employee will churn is obtainable with the current dataset. I feel as though I will be able to successfully answer my question with the data I have. Although, I do plan to supplement findings with outside sources. I do not feel it is necessary to change the driving question. The dataset will need to be adjusted and cleaned to build the model. There are null values within the dataset that will need to be handled by methods such as removing or filling. Categorical variables will also need to be transformed for analysis. I also plan to remove duplicate entries. This ensures the dataset is not oversaturated with the same data point. I still plan to use logistic regression, decision trees, naïve Bayes, and random forest for my models and do not feel as if they need to be adjusted currently. I plan to add more evaluation metrics in addition to the confusion matrix. I plan to look at accuracy, precision, false positives and negatives, and true positives and negatives. I believe my original expectations for this project are still obtainable.

Milestone 2

Employees are the backbone of most companies. Whether it's at the mall, at dinner, or when going to the movies, everyone has interacted with employees of some type of business. They are the people who serve meals, check customers out at the register, create marketing campaigns, etc. Without employees, most companies would not be successful.

Employees are essential to businesses for them to function however, it takes time, effort, and money to both hire and train an employee to be independently working. Employee turnover is a large issue for companies. Consistently, having to use resources to hire new employees can hinder a business by taking time, money, etc. to train them. According to Sam Steady, "A study by the Center for American Progress found that companies typically pay about one-fifth of an employee's salary to replace that employee, and the cost can significantly increase if executives or highest-paid employees are to be replaced." (2022). Another thing to consider is additional licensures, certificates, and tests (such as eye or drug tests) paid for by the company. Retaining current employees would be more beneficial and will allow for a more optimal workflow. If it was possible to predict which employees had the highest likelihood of turnover, they could be targeted to prevent turnover.

For a company with a high turnover rate, being able to predict and prevent employees from leaving will save time, money, and resources. Retaining long-term employees will also allow for the company to better know and understand employee strengths to optimize workflow. Reducing turnover also can improve morale. This is because often when a new employee is hired training and additional responsibilities fall on the other employees on the team. For some companies, saving these resources will increase morale and profits.

The HR employee retention data set was obtained from Kaggle (Steady, 2022). The HR dataset measures things like satisfaction level, last evaluation, number of projects, number of hours worked per month, years with the company, time spent commuting, if they have had a work accident, the number of promotions in the last 5 years, their salary category and if they left the company (Steady, 2022). Using this data, I am hoping I will be used to establish a model that will be able to predict if an employee will leave the company once the employee is identified that information can then be used to target the employee to prevent turnover.

I plan to use multiple different models to determine which is the best at identifying employees who will quit. The overall models I plan to try are logistic regression, decision trees, naïve Bayes, and random forest. These models all look at/ interpret the data in different ways and testing multiple models will allow for the opportunity to find which model works best with the data. They are all classification models that will help in the end goal of predicting the probability of an employee leaving. I plan to evaluate my models based on performance. The goal is to correctly identify the employees that have been identified as likely to quit. Therefore, using a confusion matrix to is a good way to visualize how well they can correctly identify those employees by looking at true positives, false positives, true negatives, and false negatives from each model.

Throughout this project, I hope to learn to be better at building and interpreting models. I also hope that throughout this project I learn new ways to optimize my code. I feel as though my code can contain unnecessary steps and I want to try to eliminate that. I also hope to learn how to improve upon my visualizations as they are very helpful and important in data analysis. I hope to also get better at finding and fixing issues within the dataset. This includes cleaning the dataset as this is a key part of data science and is important in getting an accurate model. I also hope to learn how to communicate more effectively with my peers.

Ethical implications and risks are important to consider with any datasets and analysis to maintain an accurate and transparent analysis. One thing to consider is that I don't know how this dataset was obtained. It is important that since I do not know if this data was fabricated or obtained from a reputable source, the data could be biased or false. It also could be considered unethical to offer incentives based on how likely an employee is to leave the company. It is possible that this system of employee retention has the ability to cause more issues later on with other employees and cause additional turnover. The dataset also appears to be currently imbalanced. Which can cause an issue, because I am unsure if the data is real or generated. The data will also be cleaned, and altering the data must be done with caution to prevent altering the data in a way that could significantly alter the analysis. I plan to minimize ethical concerns in any way I can.

Although there are ethical concerns, I believe it is okay to proceed with the current dataset and plan for the project.

The goal is to proceed with the project as planned, however, if I am unable to build a model to accurately predict employee turnover, I want to try and predict employee satisfaction or the number of projects that a person will be assigned using the given variables. If there is a problem overall with the dataset, I would try to find another dataset that highlights employee turnover. If I am unable to find another dataset for employee turnover, I will likely find a new dataset that highlights customer turnover as it is a similar issue.

References

Holliday, M. (2021, February 22). *10 Benefits of Employee Retention for Businesses*. Oracle NetSuite. https://www.netsuite.com/portal/resource/articles/human-resources/employee-retention-benefits.shtml

Steady, S. (2022, August 20). *HR employee Retention Dataset*. Kaggle. https://www.kaggle.com/datasets/shubham8983/hr-employee-retention-dataset

Tupper, H., & Ellis, S. (2022, July 4). *It's time to Reimagine employee retention*. Harvard Business Review. https://hbr.org/2022/07/its-time-to-reimagine-employee-retention

DSC 630

Inman, Gracie

Term Project Winter 23

02/10/24

In [37]:
```
# Load packages
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score
from sklearn.tree import DecisionTreeClassifier
from sklearn.naive_bayes import GaussianNB
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import confusion_matrix
import seaborn as sns
from sklearn.metrics import precision_score
import warnings
from sklearn.exceptions import ConvergenceWarning
```

In [2]:
```
# Read df
employee_df = pd.read_csv("hr_employee_churn_data.csv")
employee_df.head()
```

Out[2]:

	empid	satisfaction_level	last_evaluation	number_project	average_montly_hours	time_s
0	1	0.38	0.53	2	157	
1	2	0.80	0.86	5	262	
2	3	0.11	0.88	7	272	
3	4	0.72	0.87	5	223	
4	5	0.37	0.52	2	159	

The data frame was read using pandas and the first few rows were viewed to check that the data imported correctly.

In [3]:
```
# Check shape
employee_df.shape
```

Out[3]:
```
(14999, 10)
```

The shape was checked to get a sense of the size of the dataset.

In [4]:

```
# Check for missing values
miss_val = employee_df.isnull().sum()

print("Missing values per column:")
print(miss_val)
```

```
Missing values per column:
empid                    0
satisfaction_level       2
last_evaluation          0
number_project           0
average_montly_hours     0
time_spend_company       0
Work_accident            0
promotion_last_5years    0
salary                   0
left                     0
dtype: int64
```

It is important to handle missing values before analysis to prevent errors or biased results. Since there are only 2 missing values out of 14,999. I have decided to drop the missing values as it is likely to have little to no impact on analysis.

In [5]:

```
# Drop the incomplete data
employee_df = employee_df.dropna()
employee_df.shape
```

Out[5]:

```
(14997, 10)
```

The incomplete data entries were dropped and the shape was check to ensure the number of rows dropped was only two.

In [6]:

```
# Map values and encode column
salary_map = {'low': 0, 'medium': 1, 'high': 2}
employee_df['salary'] = employee_df['salary'].map(salary_map)
employee_df.head()
```

Out[6]:

	empid	satisfaction_level	last_evaluation	number_project	average_montly_hours	time_s
0	1	0.38	0.53	2	157	
1	2	0.80	0.86	5	262	
2	3	0.11	0.88	7	272	
3	4	0.72	0.87	5	223	
4	5	0.37	0.52	2	159	

The salary column must be encoded to ensure smooth analysis. This prevents errors and issues related to text in analysis.

In [7]:

```python
# Check column names
employee_df.columns
```

Out[7]:

```
Index(['empid', 'satisfaction_level', 'last_evaluation', 'number_project',
       'average_montly_hours', 'time_spend_company', 'Work_accident',
       'promotion_last_5years', 'salary', 'left'],
      dtype='object')
```

I check column names to determine if any columns could easily be removed

In [8]:

```python
# Drop employee ID
employee_df.drop(columns='empid', inplace=True)
employee_df.head()
```

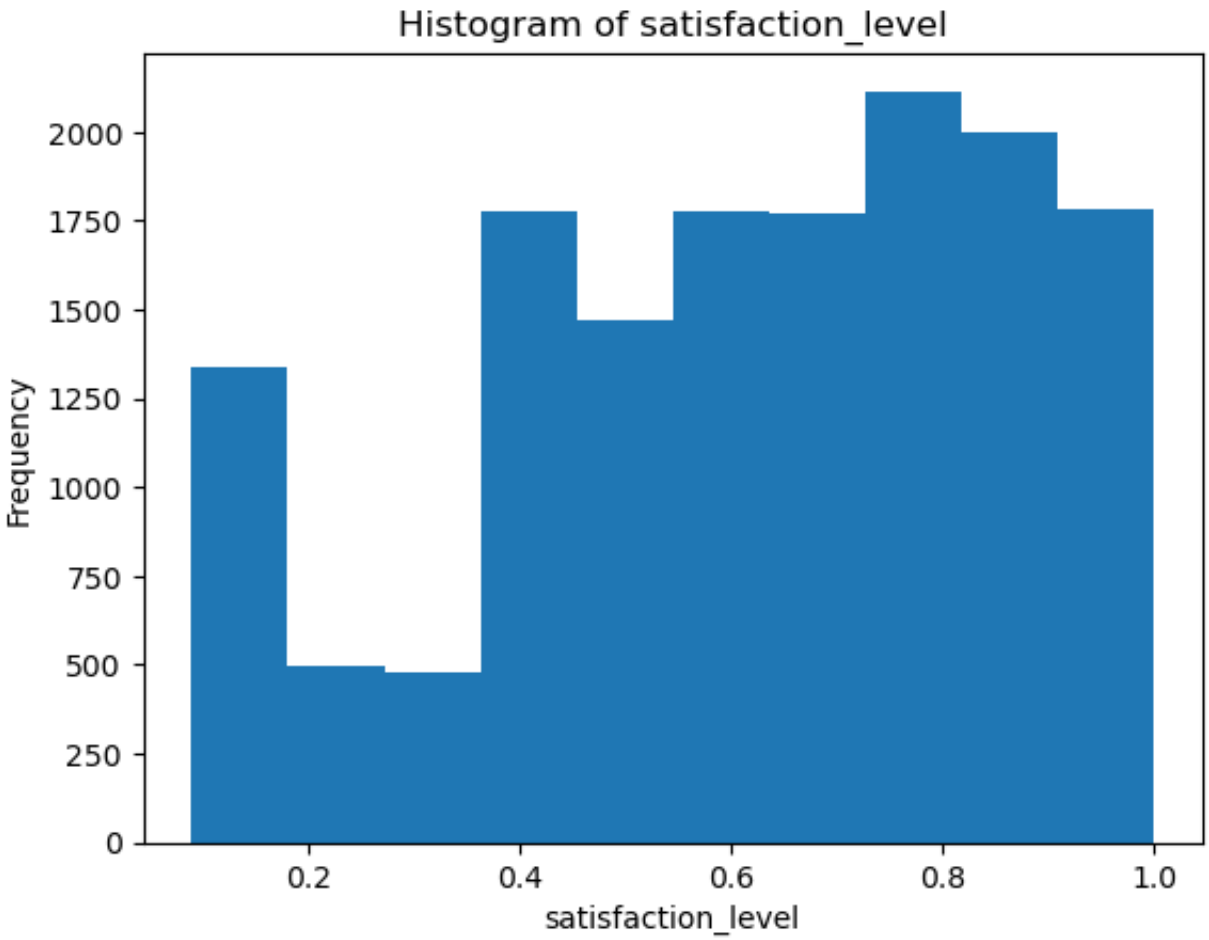
Out[8]:

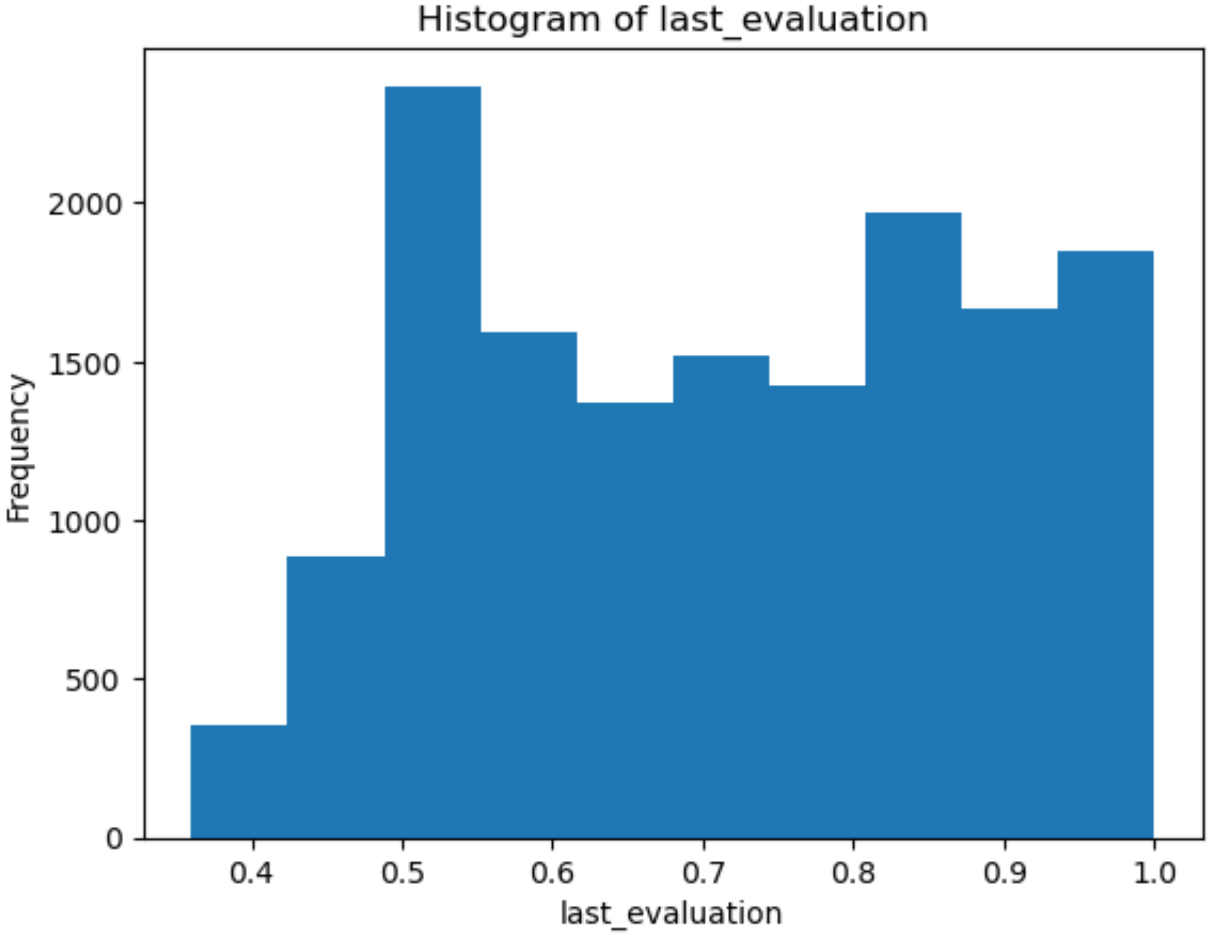
	satisfaction_level	last_evaluation	number_project	average_montly_hours	time_spend_cc
0	0.38	0.53	2	157	
1	0.80	0.86	5	262	
2	0.11	0.88	7	272	
3	0.72	0.87	5	223	
4	0.37	0.52	2	159	

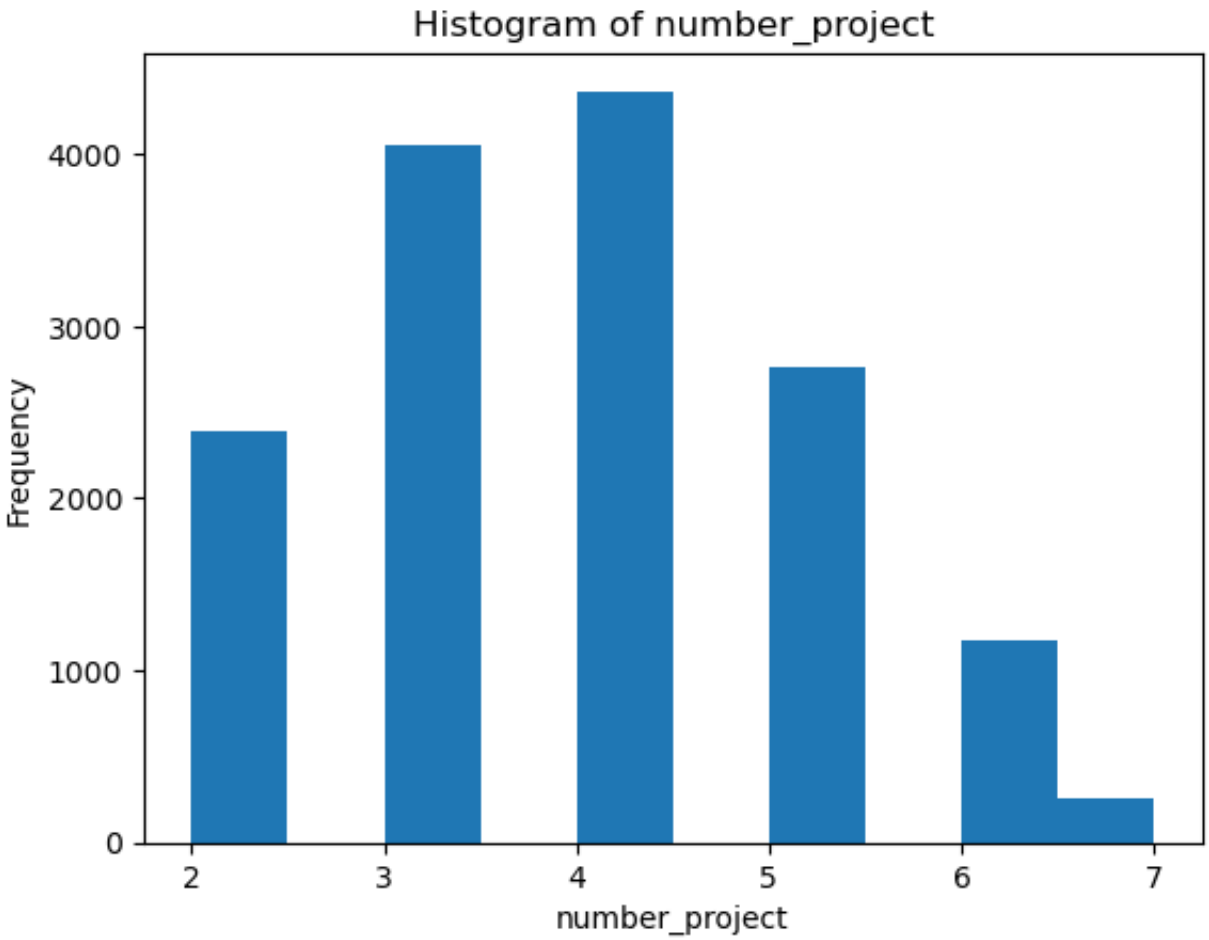
The employee ID column was removed as it has no benefit in the dataset.

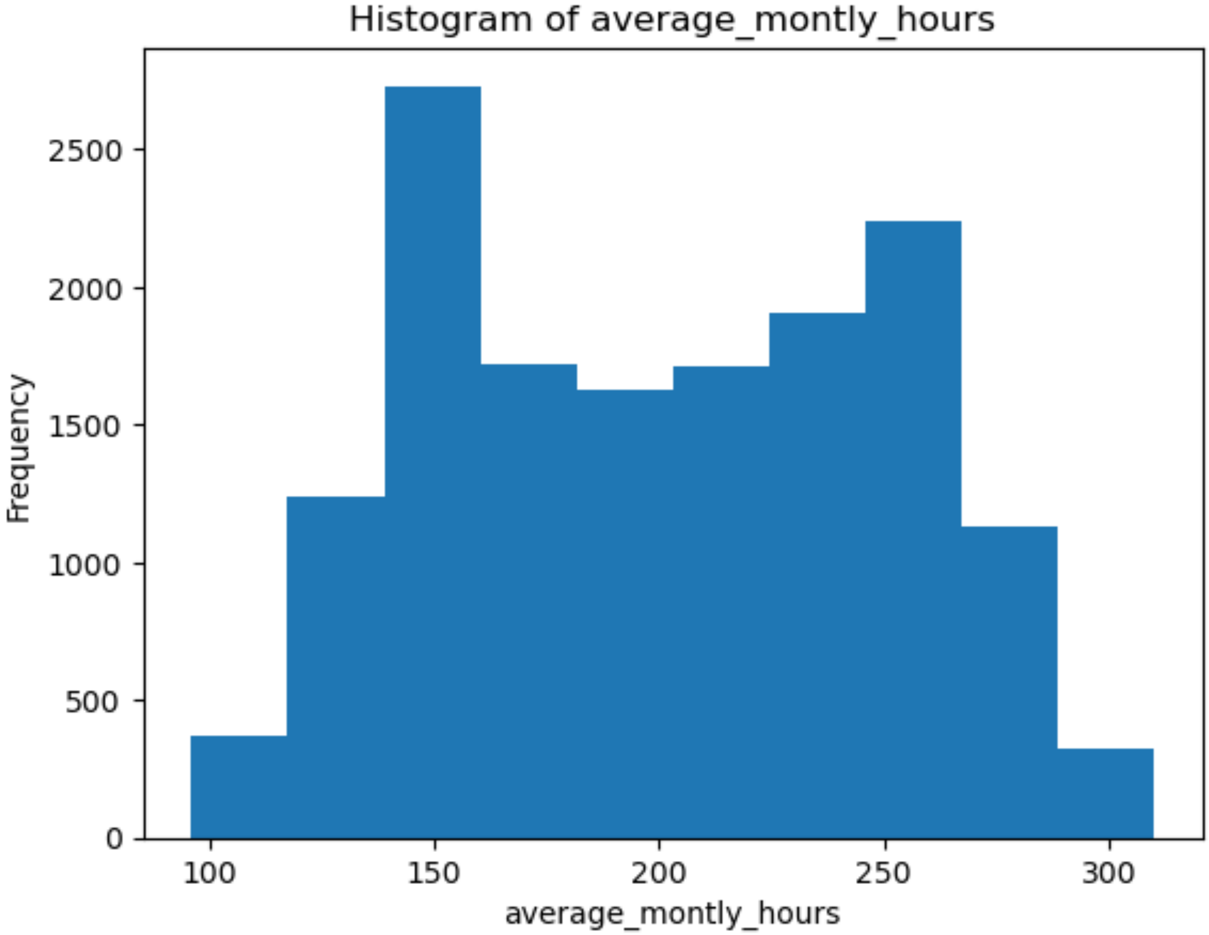
In [9]:

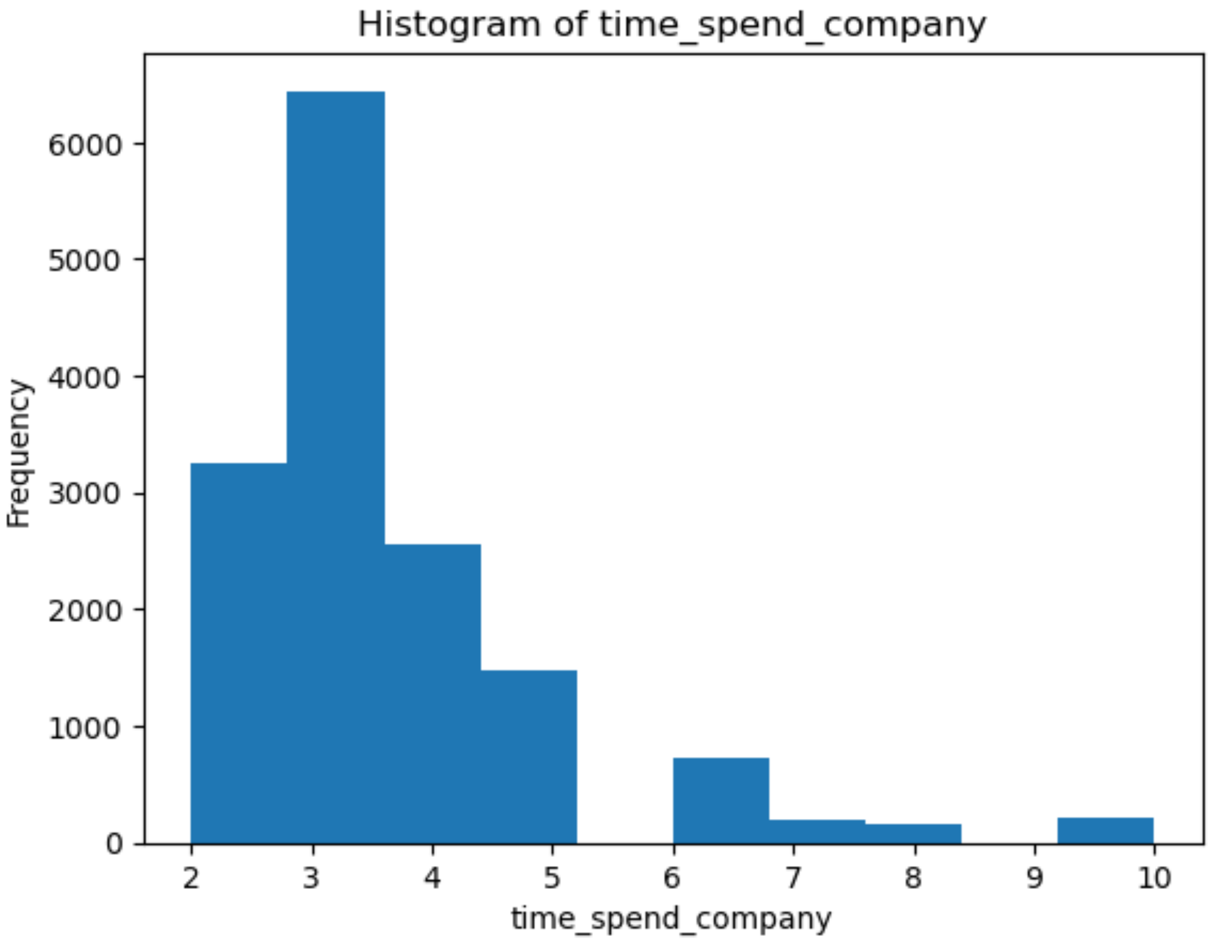
```python
# Plot each column
for column in employee_df.columns:
    plt.figure()
    plt.hist(employee_df[column])
    plt.xlabel(column)
    plt.ylabel("Frequency")
    plt.title(f"Histogram of {column}")

plt.show()
```

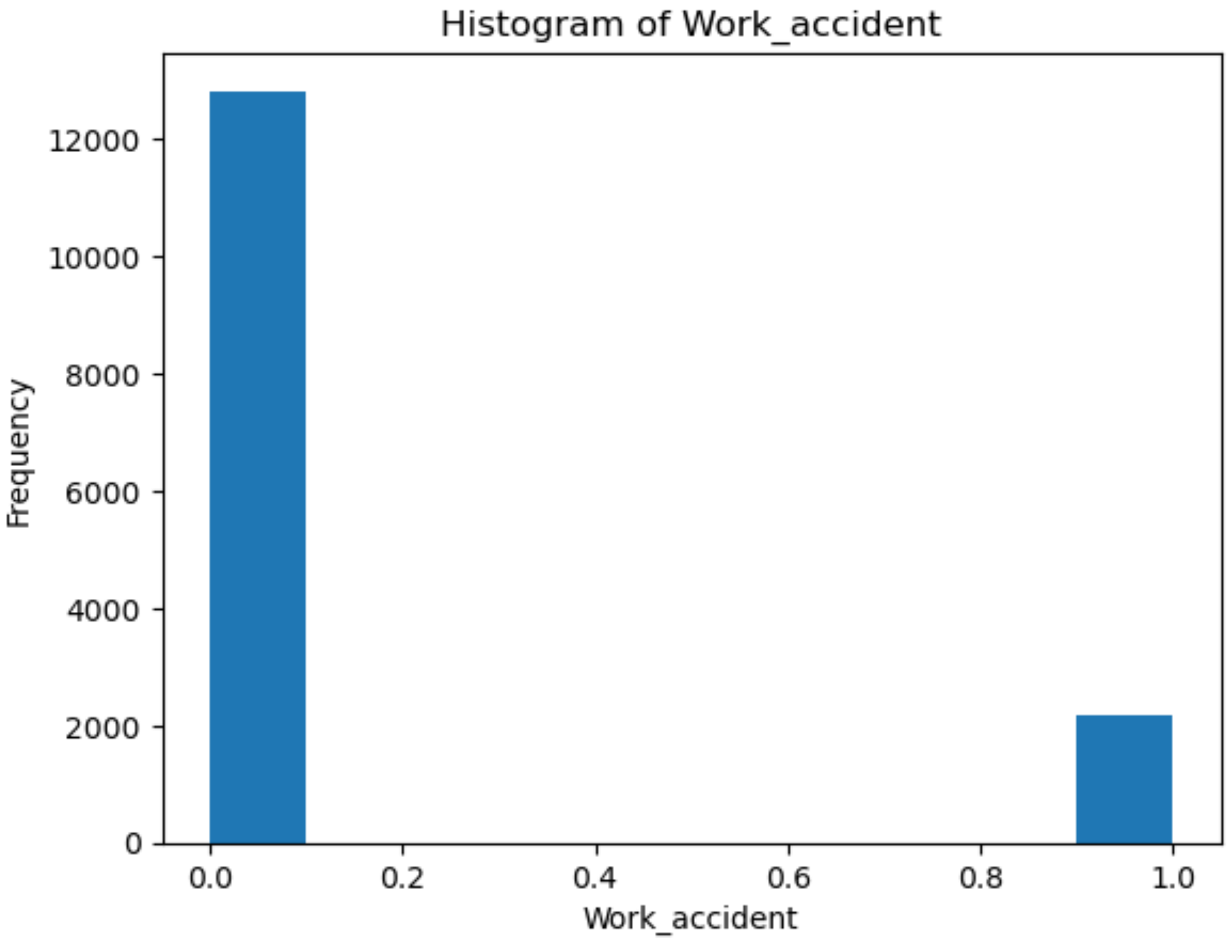

Histogram of satisfaction_level
Frequency
0
250
500
750
1000
1250
1500
1750
2000
0.2
0.4
0.6
0.8
1.0
satisfaction_level

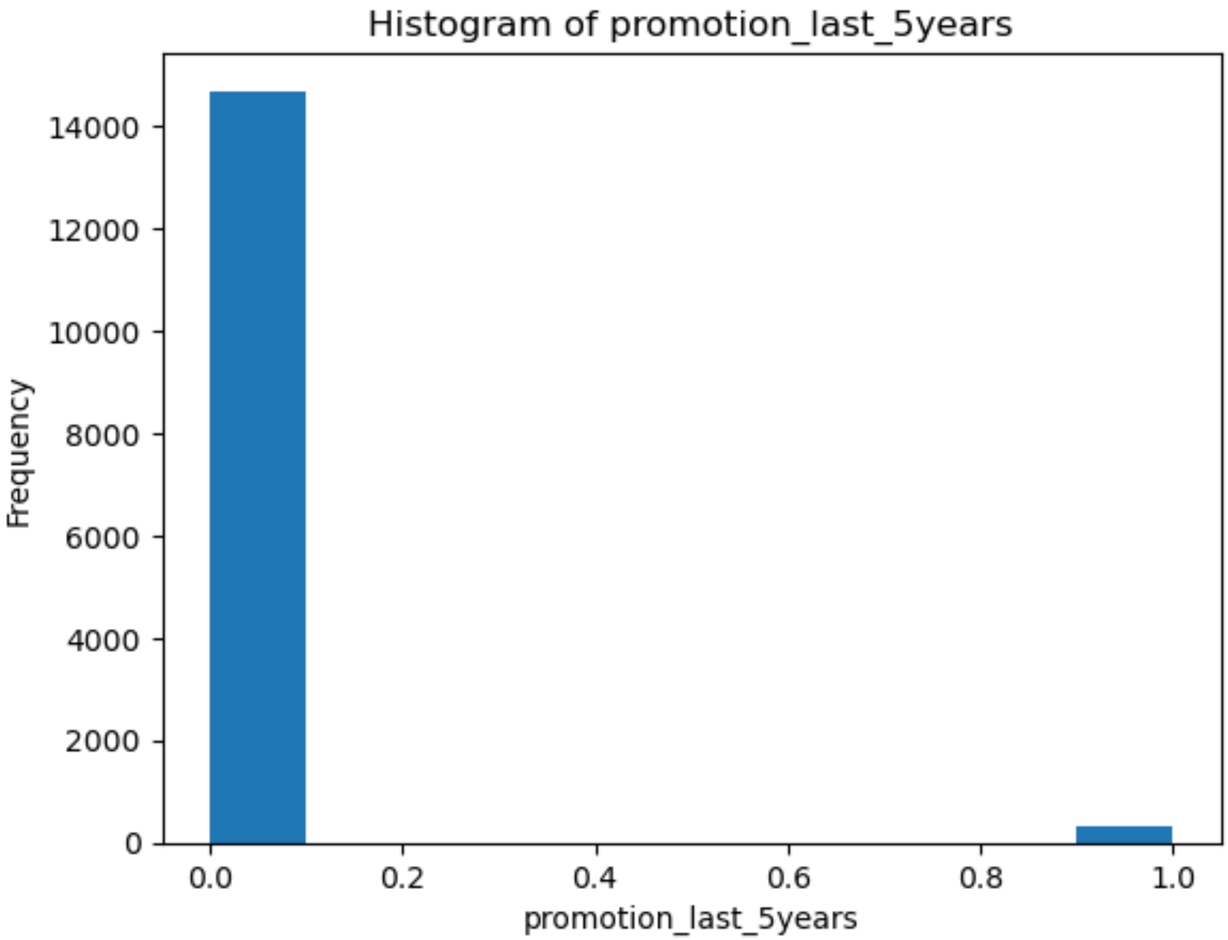

Histogram of last_evaluation
Frequency
0
500
1000
1500
2000
0.4
0.5
0.6
0.7
0.8
0.9
1.0
last_evaluation

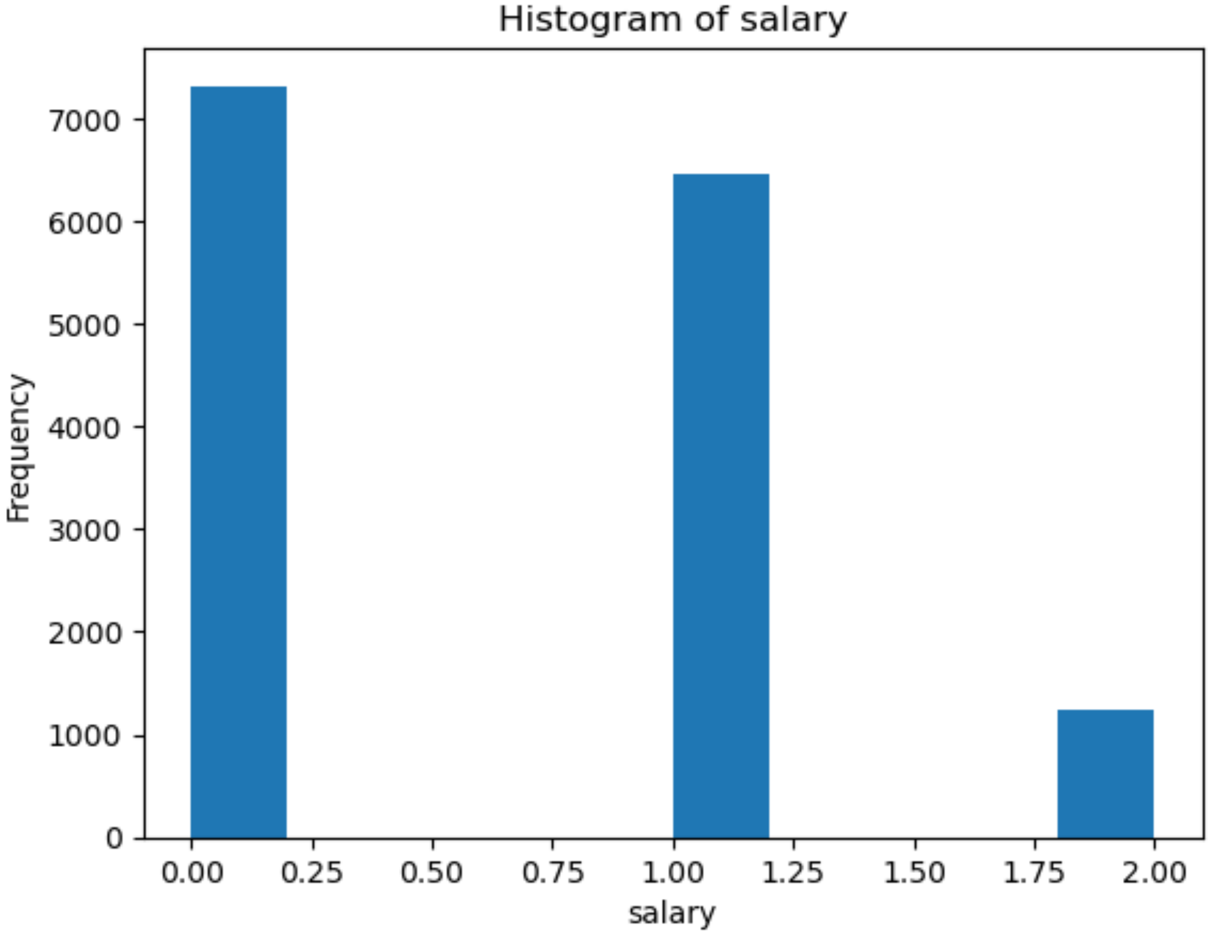

Histogram of number_project
Frequency
0
1000
2000
3000
4000
2
3
4
5
6
7
number_project

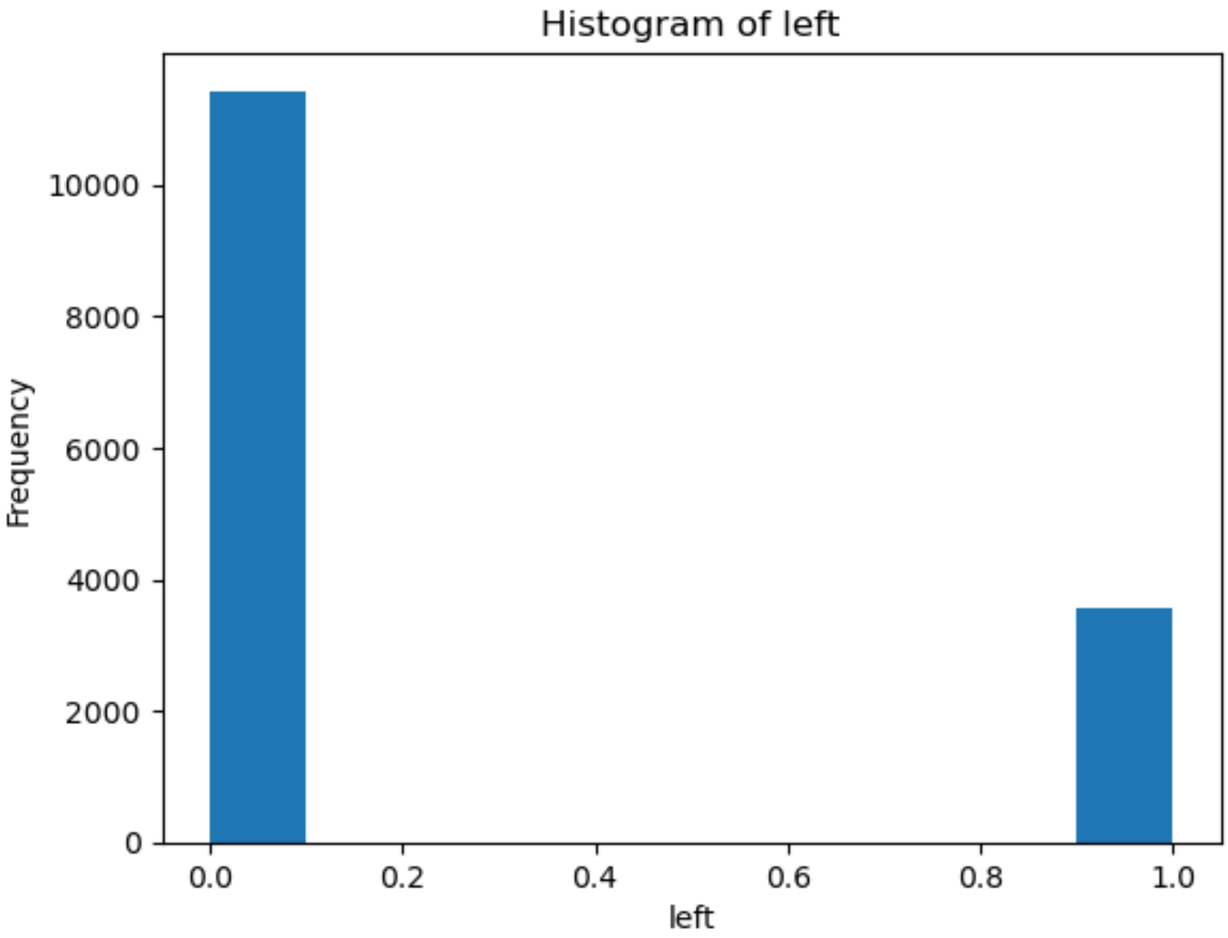

Histogram of average_montly_hours
Frequency
2500
2000
1500
1000
500
0
100
150
200
250
300
average_montly_hours


Histogram of time_spend_company
Frequency
0
1000
2000
3000
4000
5000
6000
2
3
4
5
6
7
8
9
10
time_spend_company


Histogram of Work_accident
Frequency
12000
10000
8000
6000
4000
2000
0
0.0
0.2
0.4
0.6
0.8
1.0
Work_accident

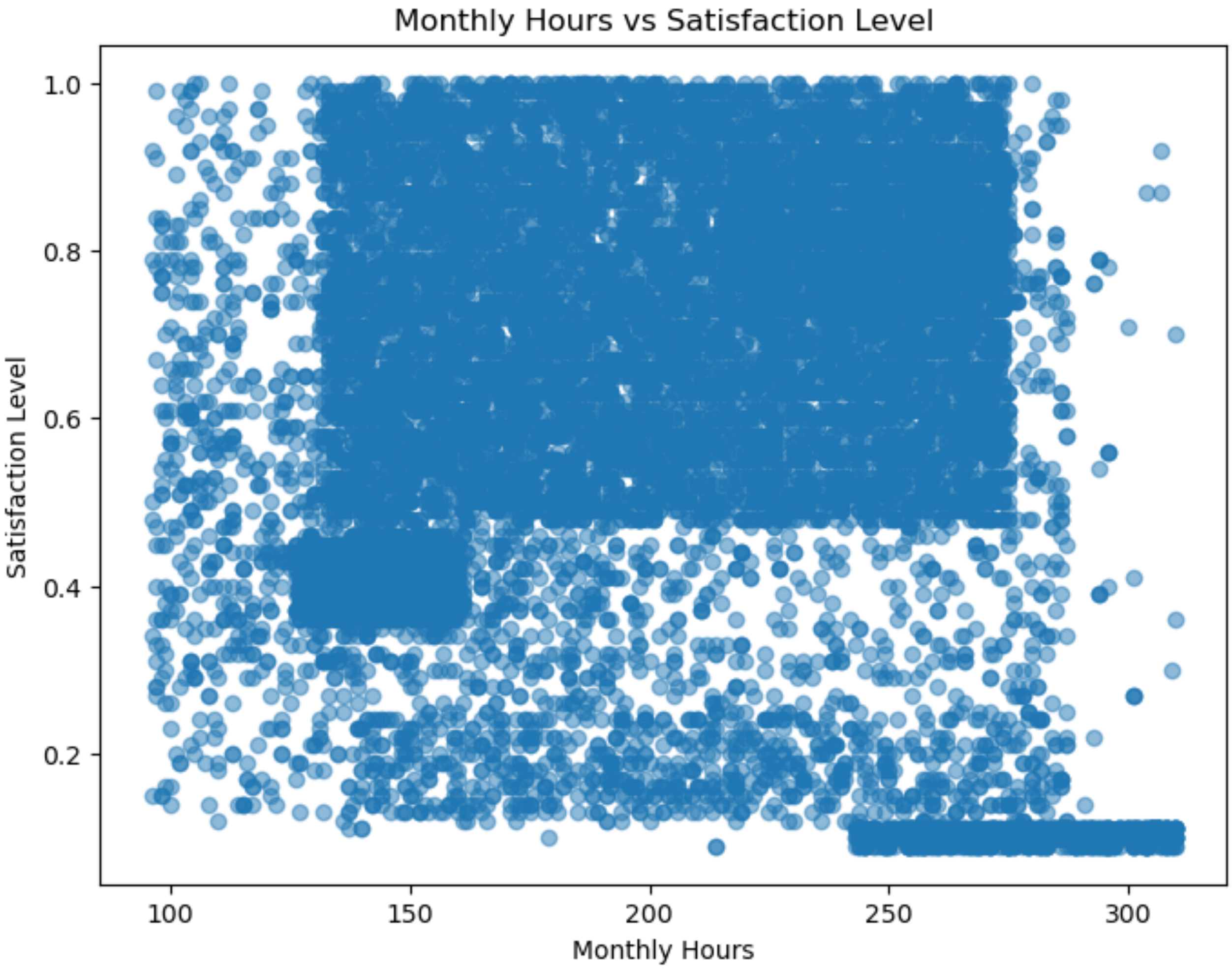

Histogram of promotion_last_5years
Frequency
0
2000
4000
6000
8000
10000
12000
14000
0.0
0.2
0.4
0.6
0.8
1.0
promotion_last_5years


Histogram of salary
Frequency
7000
6000
5000
4000
3000
2000
1000
0
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
2.00
salary

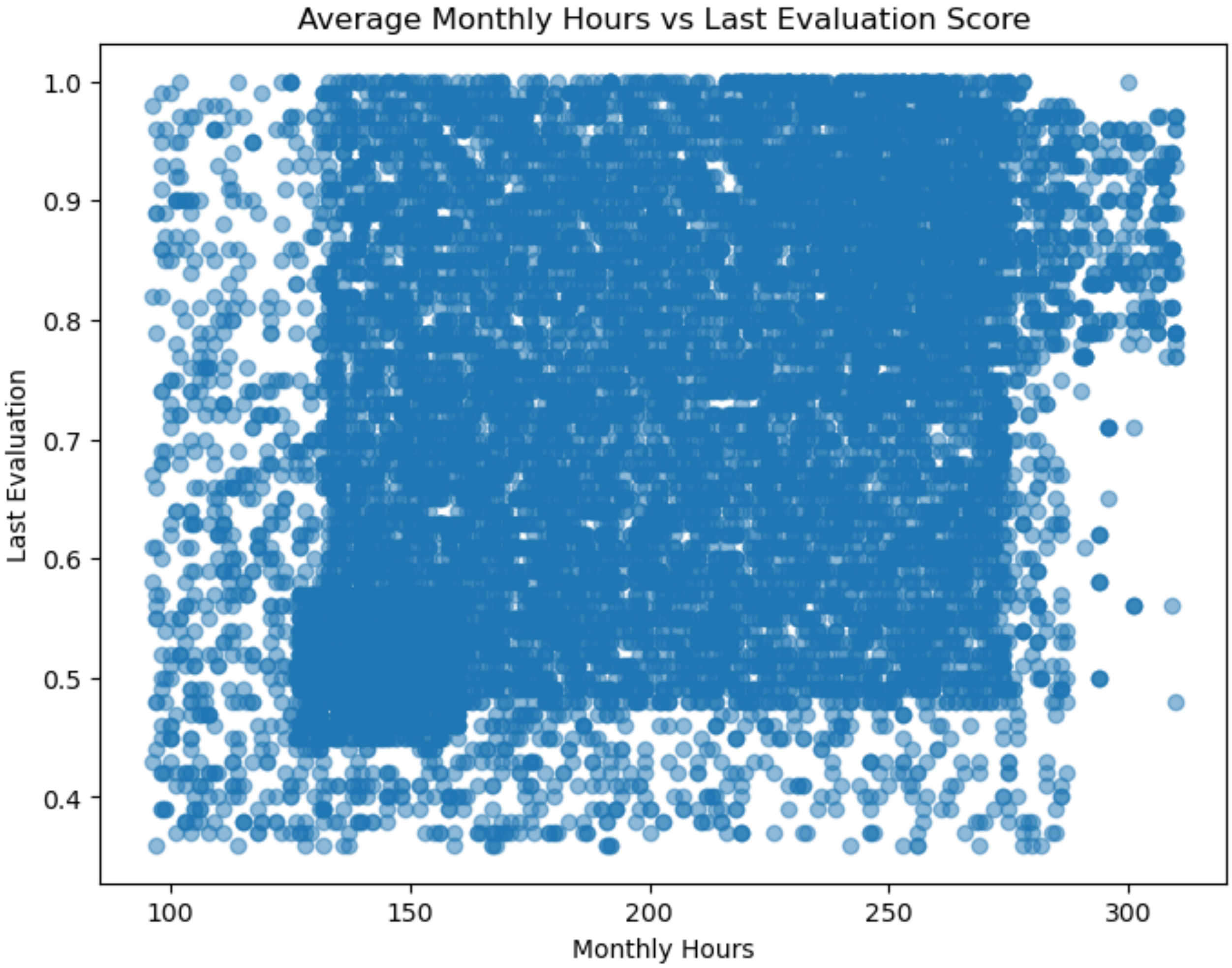


In [48]:

```
# Satisfaction level
plt.figure(figsize=(8, 6))
plt.scatter(employee_df['average_montly_hours'], employee_df['satisfaction_l
plt.title('Monthly Hours vs Satisfaction Level')
plt.xlabel('Monthly Hours')
plt.ylabel('Satisfaction Level')
plt.show()
```

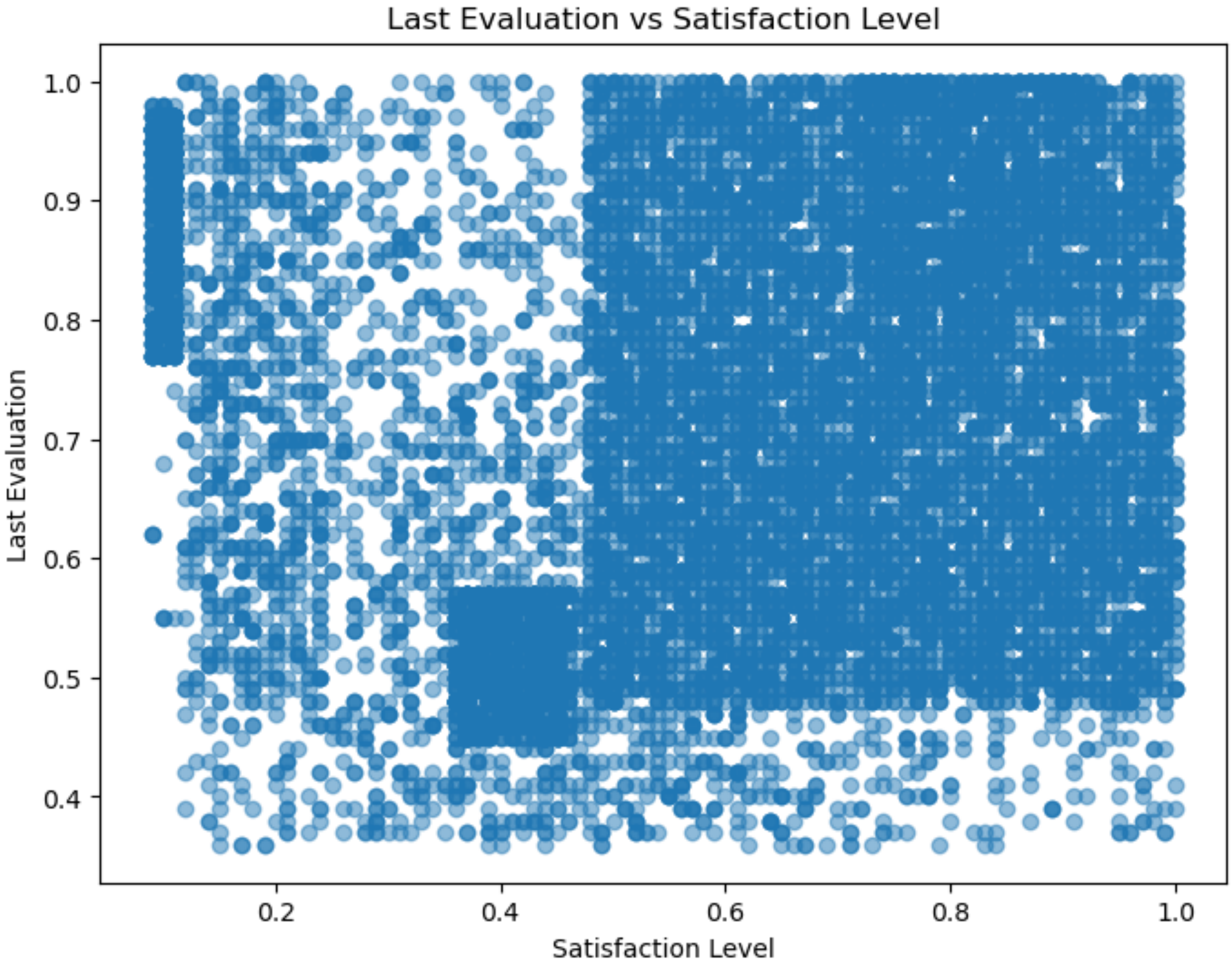


In [46]:
```
# Last evaluation
plt.figure(figsize=(8, 6))
plt.scatter(employee_df['average_montly_hours'], employee_df['last_evaluatic
plt.title('Average Monthly Hours vs Last Evaluation Score')
plt.xlabel('Monthly Hours')
plt.ylabel('Last Evaluation')
plt.show()
```



In [47]:

```
# Last evaluation
plt.figure(figsize=(8, 6))
plt.scatter(employee_df['satisfaction_level'], employee_df['last_evaluation'
plt.title('Last Evaluation vs Satisfaction Level')
plt.xlabel('Satisfaction Level')
plt.ylabel('Last Evaluation')
plt.show()
```



Each column is plotted to look at the distribution of the data and check for outliers. Histograms were chosen due to the number of categorical variables. The scatter plots were used to visualize the numerical columns and their distributions. No outliers appear to be present and distributions have been considered.

In [10]:
```
# Split data
X = employee_df.drop('left', axis=1)
y = employee_df['left']

# Training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, ran
```

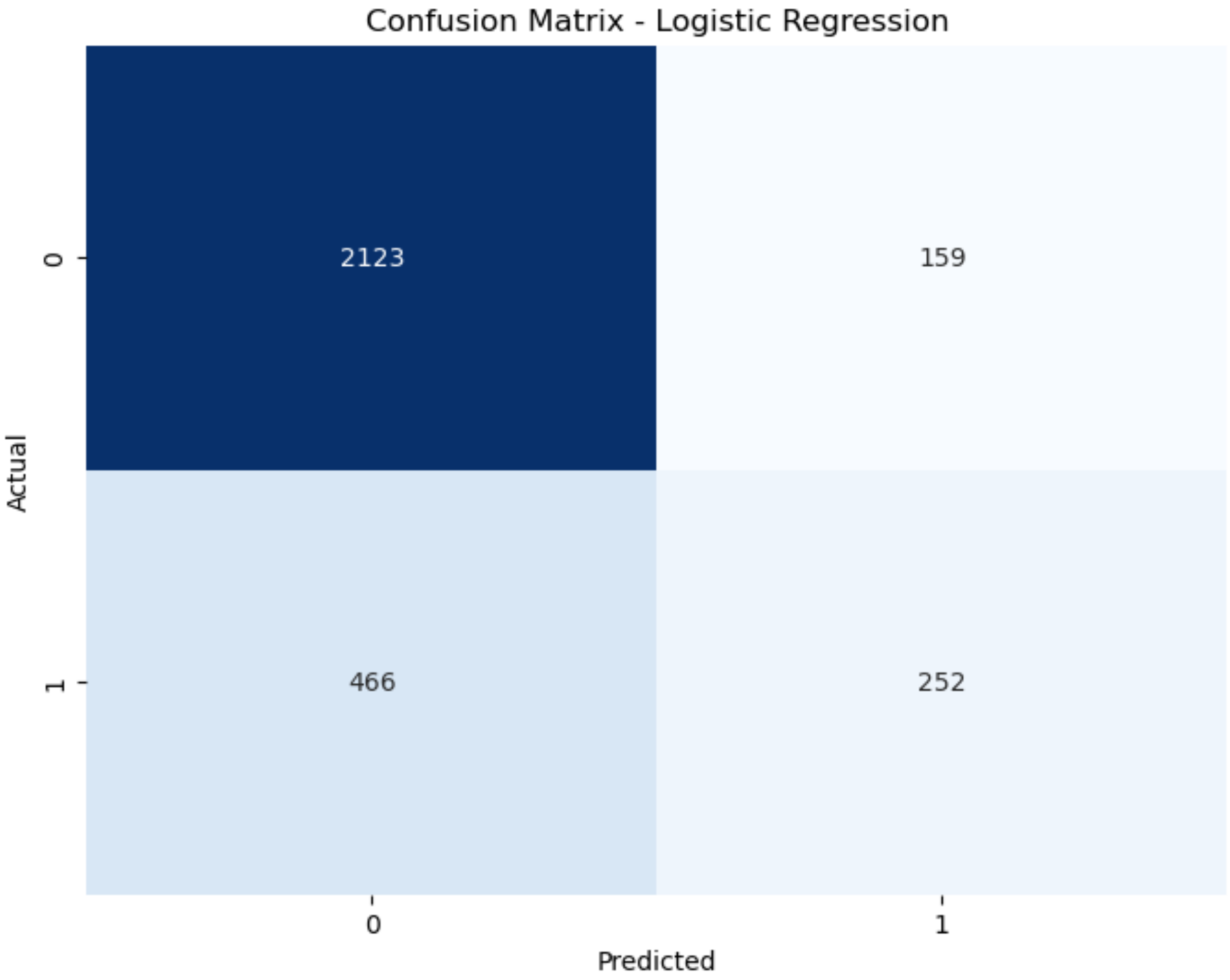
In [30]:

```
# Hide Warning
warnings.filterwarnings("ignore", category=ConvergenceWarning)

# Logistic regression model
logreg_model = LogisticRegression()
logreg_model.fit(X_train, y_train)
# Predict
y_pred = logreg_model.predict(X_test)

# Evaluate
print("Logistic Regression:")
accuracy = accuracy_score(y_test, y_pred)
print("Accuracy:", accuracy)

precision_logreg = precision_score(y_test, y_pred)
print("Precision", precision_logreg)
```

```
Logistic Regression:
Accuracy: 0.7916666666666666
Precision 0.6131386861313869
```

In [12]:

```
# Confusion Matrix for Logistic Regression
logreg_cm = confusion_matrix(y_test, y_pred)
plt.figure(figsize=(8, 6))
sns.heatmap(logreg_cm, annot=True, cmap='Blues', fmt='g', cbar=False)
plt.title('Confusion Matrix - Logistic Regression')
plt.xlabel('Predicted')
plt.ylabel('Actual')
plt.show()
```

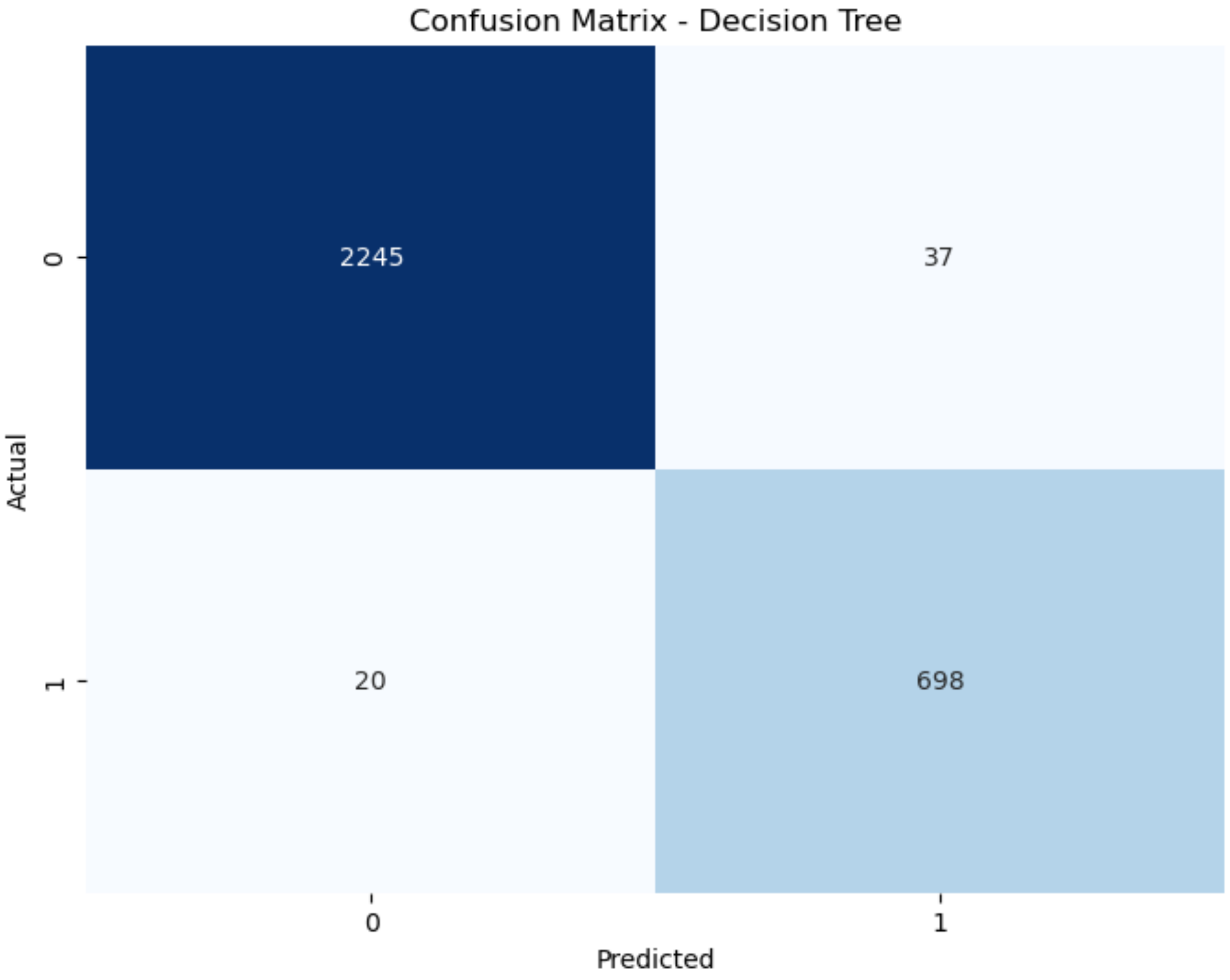


In [13]:

```
# Decision tree model
dt_model = DecisionTreeClassifier()
dt_model.fit(X_train, y_train)

# Predict
y_pred = dt_model.predict(X_test)

# Evaluate
print("Decision Tree:")
accuracy = accuracy_score(y_test, y_pred)
print("Accuracy:", accuracy)

precision_logreg = precision_score(y_test, y_pred)
print("Precision", precision_logreg)
```

```
Decision Tree:
Accuracy: 0.981
Precision 0.9496598639455782
```

In [14]:
```
# Confusion Matrix for Decision Tree
dt_cm = confusion_matrix(y_test, y_pred)
plt.figure(figsize=(8, 6))
sns.heatmap(dt_cm, annot=True, cmap='Blues', fmt='g', cbar=False)
plt.title('Confusion Matrix - Decision Tree')
plt.xlabel('Predicted')
plt.ylabel('Actual')
plt.show()
```



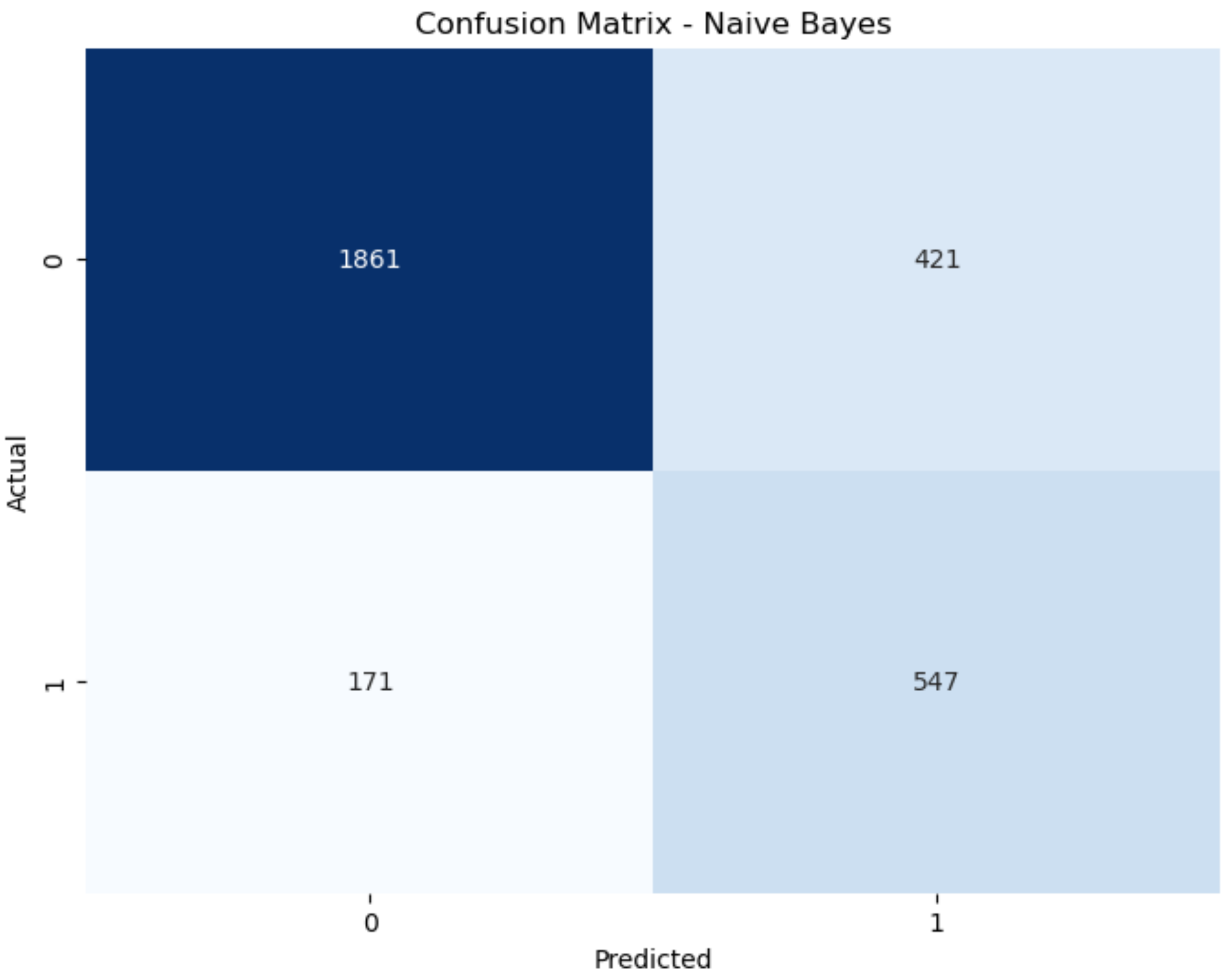
In [15]:

```
# Naive Bayes model
nb_model = GaussianNB()
nb_model.fit(X_train, y_train)

# Predict
y_pred = nb_model.predict(X_test)

# Evaluate
print("Naive Bayes:")
accuracy = accuracy_score(y_test, y_pred)
print("Accuracy:", accuracy)

precision_logreg = precision_score(y_test, y_pred)
print("Precision", precision_logreg)
```

```
Naive Bayes:
Accuracy: 0.8026666666666666
Precision 0.5650826446280992
```

In [16]:

```
# Confusion Matrix for Naive Bayes
nb_cm = confusion_matrix(y_test, y_pred)
plt.figure(figsize=(8, 6))
sns.heatmap(nb_cm, annot=True, cmap='Blues', fmt='g', cbar=False)
plt.title('Confusion Matrix - Naive Bayes')
plt.xlabel('Predicted')
plt.ylabel('Actual')
plt.show()
```

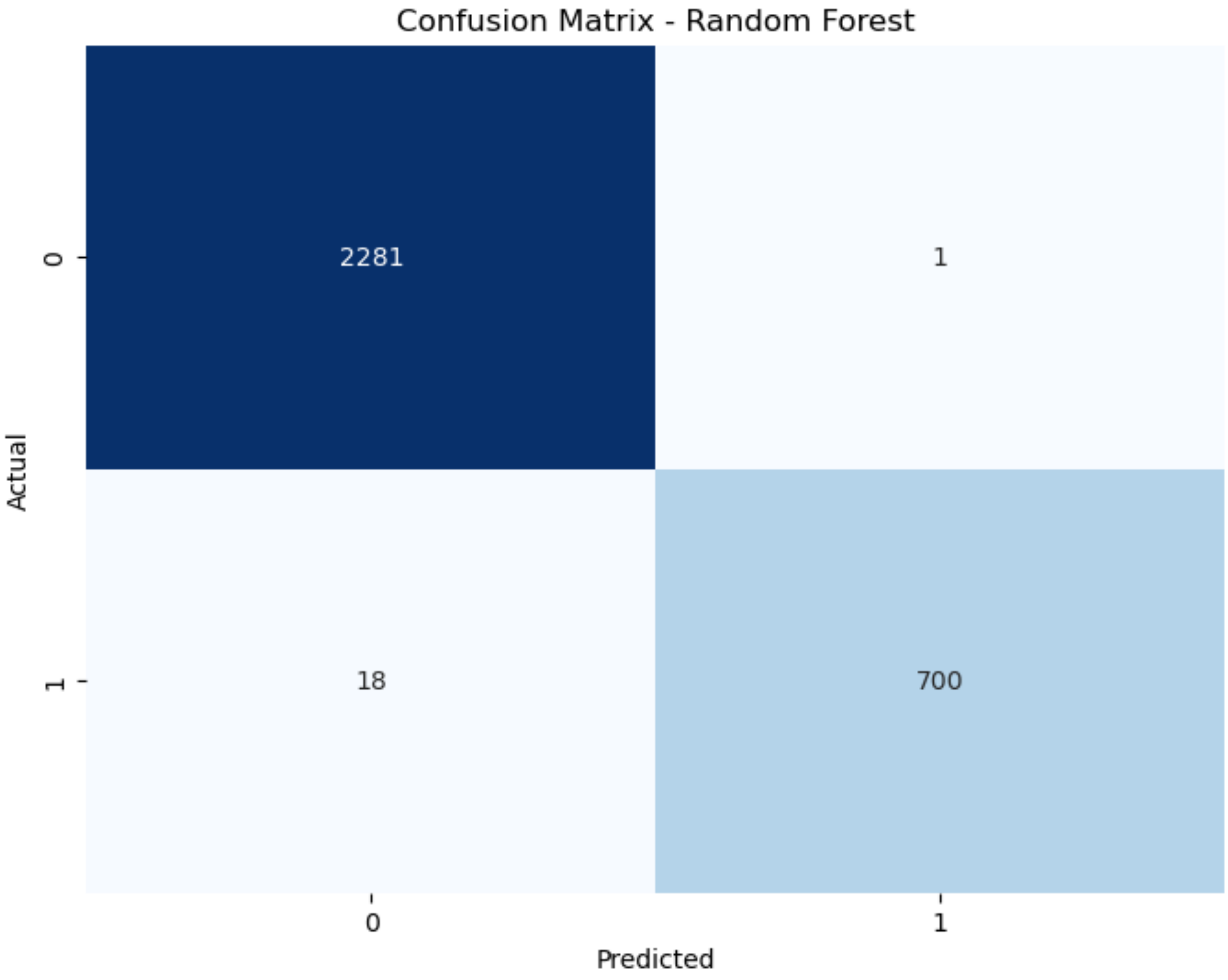


In [17]:

```
# Random forest model
rf_model = RandomForestClassifier()
rf_model.fit(X_train, y_train)

# Predict
y_pred = rf_model.predict(X_test)

# Evaluate
print("Random Forest:")
accuracy = accuracy_score(y_test, y_pred)
print("Accuracy:", accuracy)

precision_logreg = precision_score(y_test, y_pred)
print("Precision", precision_logreg)
```

```
Random Forest:
Accuracy: 0.9936666666666667
Precision 0.9985734664764622
```

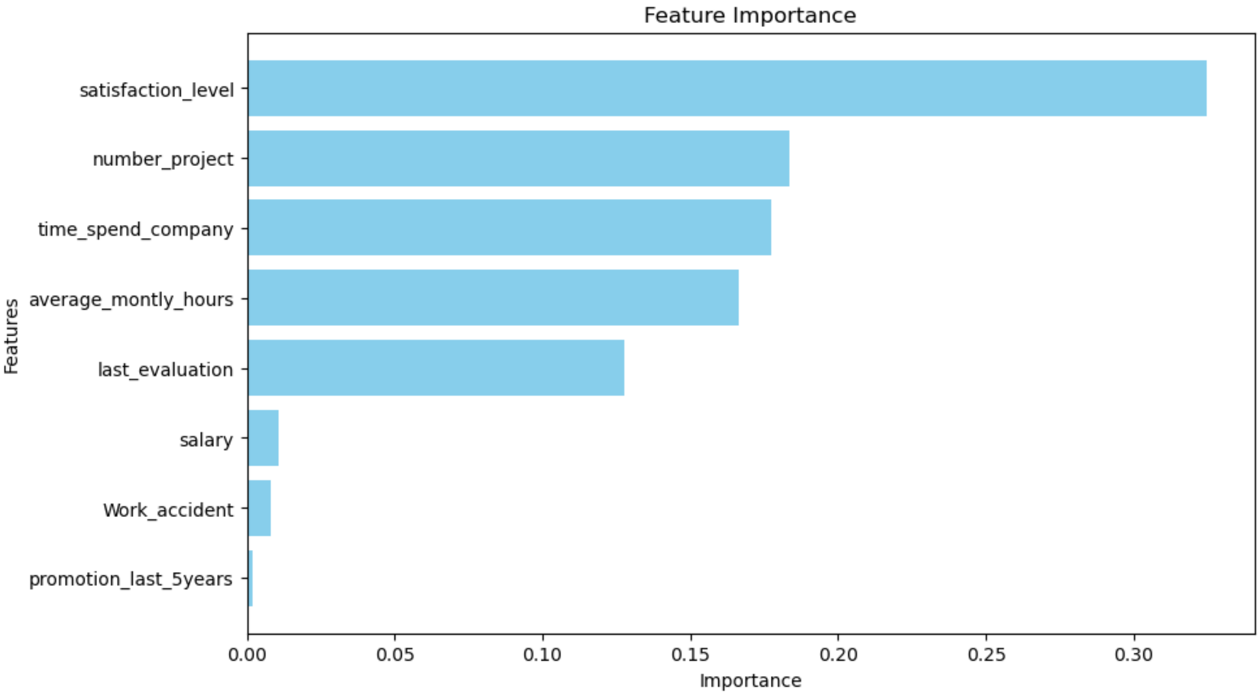
In [18]:
```
# Confusion Matrix for Random Forest
rf_cm = confusion_matrix(y_test, y_pred)
plt.figure(figsize=(8, 6))
sns.heatmap(rf_cm, annot=True, cmap='Blues', fmt='g', cbar=False)
plt.title('Confusion Matrix - Random Forest')
plt.xlabel('Predicted')
plt.ylabel('Actual')
plt.show()
```



When considering all the models, the accuracy, precision, and the confusion matrix the Random Forest model appears to preform the best. It has the highest number of true positives, accuracy score, and precision score.

In [36]:

```
# Checking feature importance
feature_importances = rf_model.feature_importances_
feature_importance_dict = dict(zip(employee_df.columns, feature_importances)

# Sort features by importance
sorted_features = sorted(feature_importance_dict.items(), key=lambda x: x[1]

# Extract feature names and importance scores
features = [x[0] for x in sorted_features]
importance = [x[1] for x in sorted_features]

# Plot
plt.figure(figsize=(10, 6))
plt.barh(features, importance, color='skyblue')
plt.xlabel('Importance')
plt.ylabel('Features')
plt.title('Feature Importance')
plt.gca().invert_yaxis()
plt.show()
```



The plot above shows that employee satisfaction level is the most important feature in determining if an employee will quit.

In [ ]: