Sri Sri University

Final Project Report

on

Employee Turnover Prediction Using Advanced Machine Learning

Team members:
Sthitapragyan Mahapatra
Chinmaya Kumar Kar
Aditya Jyotiraditya Sahoo

Guided by:
Prof. (Dr). Pradipta Kumar Mishra
and
Prof. (Dr.) Rabinarayan Satpathy

Industry Mentor:
Mr. Sumit Shukla

Table of Contents

Executive Summary

1. **Background**
   - **1.1 Aim**
   - **1.2 Technologies**
   - **1.3 Hardware Architecture**
   - **1.4 Software Architecture**
2. **System**
   - **2.1 Requirements**
     - 2.1.1 Functional requirements
     - 2.1.2 User requirements
     - 2.1.3 Environmental requirements
   - **2.2 Design and Architecture**
   - **2.3 Implementation**
   - **2.4 Testing**
     - 2.4.1 Test Plan Objectives
     - 2.4.2 Data Entry
     - 2.4.3 Security
     - 2.4.4 Test Strategy
     - 2.4.5 System Test
     - 2.4.6 Performance Test
     - 2.4.7 Security Test
     - 2.4.8 Basic Test
     - 2.4.9 Stress and Volume Test
     - 2.4.10 Recovery Test
     - 2.4.11 Documentation Test
     - 2.4.12 User Acceptance Test
     - 2.4.13 System
   - **2.5 Graphical User Interface (GUI)**
   - **2.6 Layout**
   - **2.6 Customer testing**
   - **2.7 Evaluation**
     - 2.7.1 Table [Performance]
     - 2.7.2 STATIC CODE ANALYSIS
     - 2.7.3 WIRESHARK
     - 2.7.4 TEST OF MAIN FUNCTION
3. **Snapshots of the Project**
4. **Conclusions**
5. **Further development or research**
6. **References**
7. **Appendix**

Executive Summary

The given code performs an employee turnover analysis using a dataset containing various employee-related features. The analysis includes cleaning and manipulating the dataset, calculating the mean values of certain features based on whether an employee left the company or not, and creating visualizations to show the relationships between satisfaction level, last evaluation, and employee turnover rate.

The dataset is first manipulated by removing unnecessary columns, banding certain features, and converting all columns into numeric format. The analysis includes calculating the mean values of certain features based on whether an employee left the company or not. Visualizations are created to show the relationships between satisfaction level, last evaluation, and employee turnover rate.

The dataset is then split into a training set and a test set for modeling purposes. The modeling algorithms used for this analysis are Logistic Regression, KNN or k-Nearest Neighbors, Support Vector Machines, Naive Bayes classifier, and Decision Tree. These algorithms are selected because they are relatively simple to understand and perform classification and regression tasks.

The analysis shows that employee satisfaction level and last evaluation are related to employee turnover rate. The satisfaction level feature is used in the model, and the last evaluation feature is banded into exceptional scores (both really good and really bad evaluations) versus the rest. The number of projects feature exhibits a clustering effect, and people tend to leave when they are overworked or underworked. The weekly hours feature is discarded for simplicity.

The final dataset includes the following features: satisfaction_level, last_evaluation, number_project, time_spend_company, left, and salary. The analysis is performed using Python and relevant libraries, and the dataset is provided as input to the analysis. The analysis is tested manually, and the system test, performance test, security test, basic test, stress and volume test, recovery test, documentation test, user acceptance test, and system evaluation are all performed. The graphical user interface (GUI) layout is not applicable, and customer testing is not required.

1. Background

The project involves an analysis of employee turnover using a dataset containing various employee-related features. The dataset contains 14,999 records and 18 features, including employee satisfaction level, last evaluation, number of projects worked, time spent in the company, and whether the employee left the company or not. The dataset is analyzed using Python and various libraries such as Pandas, NumPy, Matplotlib, and Seaborn.

1.1 Aim

The aim of the project is to analyze employee turnover and identify the factors that contribute to it. The analysis includes cleaning and manipulating the dataset, calculating the mean values of certain features based on whether an employee left the company or not, and creating visualizations to show the relationships between satisfaction level, last evaluation, and employee turnover rate.

1.2 Technologies

The project uses Python as the primary programming language and various libraries such as Pandas, NumPy, Matplotlib, and Seaborn for data manipulation, cleaning, and visualization. The dataset is loaded into a Pandas DataFrame for analysis.

1.3 Hardware Architecture

The hardware architecture for the project includes a computer with a minimum of 4GB RAM and 10GB of available storage. The operating system should be Windows, macOS, or Linux.

1.4 Software Architecture

The software architecture for the project includes Python and various libraries such as Pandas, NumPy, Matplotlib, and Seaborn. The dataset is loaded into a Pandas DataFrame for analysis. The analysis is performed using Jupyter Notebook, a web-based interactive computing environment.

2. System

2.1 Requirements

2.1.1 Functional requirements

- The functional requirements of the system are as follows:
- The system should be able to read and process the provided dataset.
- The system should be able to perform data cleaning and preprocessing tasks such as removing unnecessary columns, banding certain features, and converting all columns into numeric format.
- The system should be able to calculate the mean values of certain features based on whether an employee left the company or not.
- The system should be able to create visualizations to show the relationships between satisfaction level, last evaluation, and employee turnover rate.

2.1.2 User requirements

- The user requirements of the system are as follows:
- The system should be user-friendly and easy to use.
- The system should provide clear and concise visualizations that are easy to interpret.
- The system should provide accurate and reliable results.

2.1.3 Environmental requirements

- **The environmental requirements of the system are as follows:**
- **The system should be compatible with the Windows operating system.**
- **The system should be compatible with the Anaconda distribution of Python.**
- **The system should require minimal resources to run.**

2.2 Design and Architecture

The system is designed to be a Python script that reads and processes the provided dataset. The script uses the Pandas library to manipulate and clean the data, and the Matplotlib and Seaborn libraries to create visualizations. The script is structured as a series of cells in a Jupyter Notebook, which allows for easy exploration and experimentation with the data.

The system first loads the dataset into a Pandas DataFrame and performs some initial data exploration. It then performs data cleaning and preprocessing tasks, such as removing unnecessary columns, banding certain features, and converting all columns into numeric format.

Next, the system calculates the mean values of certain features based on whether an employee left the company or not. It then creates visualizations to show the relationships between satisfaction level, last evaluation, and employee turnover rate.

Finally, the system splits the dataset into a training set and a test set for modeling purposes. The modeling algorithms used for this analysis are Logistic Regression, KNN or k-Nearest Neighbors, Support Vector Machines, Naive Bayes classifier, and Decision Tree. These algorithms are selected because they are relatively simple to understand and perform classification and regression tasks**.**

2.3 Implementation

The system is implemented as a Python script using the Pandas, Matplotlib, and Seaborn libraries. The script is structured as a series of cells in a Jupyter Notebook, which allows for easy exploration and experimentation with the data.

The script first loads the dataset into a Pandas DataFrame and performs some initial data exploration. It then performs data cleaning and preprocessing tasks, such as removing unnecessary columns, banding certain features, and converting all columns into numeric format.

Next, the script calculates the mean values of certain features based on whether an employee left the company or not. It then creates visualizations to show the relationships between satisfaction level, last evaluation, and employee turnover rate.

Finally, the script splits the dataset into a training set and a test set for modeling purposes. The modeling algorithms used for this analysis are Logistic Regression, KNN or k-Nearest Neighbors, Support Vector Machines, Naive Bayes classifier, and Decision Tree.

Feature Importance:

Discussing the importance of each feature in predicting employee turnover and how they contribute to the model's performance.

In the analysis of employee turnover, several features have been identified as important in predicting whether an employee will leave the company or not. These features include satisfaction level, last evaluation, number of projects worked, time spent in the company, whether the employee left the company or not, and salary.

The satisfaction level feature is a crucial factor in predicting employee turnover. Employees with lower satisfaction levels are more likely to leave the company, as shown in the analysis. The last evaluation feature is also important, with exceptional scores (both really good and really bad evaluations) being related to employee turnover.

The number of projects feature exhibits a clustering effect, with people tending to leave when they are overworked or underworked. This feature is also important in predicting employee turnover, as it can indicate whether an employee is feeling overwhelmed or underutilized.

Time spent in the company is another important factor in predicting employee turnover. Employees who have been with the company for a long time may be more likely to leave, as they may feel stagnant or unchallenged. However, the analysis shows that the years_at_company feature should be banded, with years 7 and onward considered as a single category.

The salary feature is inversely correlated with the likelihood of an employee leaving, meaning that employees with higher salaries are less likely to leave the company. This feature is also important in predicting employee turnover, as it can indicate whether an employee is being compensated fairly for their work.

In terms of model performance, the satisfaction level feature is used in the model, and the last evaluation feature is banded into exceptional scores versus the rest. The number of projects feature exhibits a clustering effect, and people tend to leave when they are overworked or underworked. The weekly hours feature is discarded for simplicity. The final dataset includes the following features: satisfaction_level, last_evaluation, number_project, time_spend_company, left, and salary.

The analysis is performed using Python and relevant libraries, and the dataset is provided as input to the analysis. The analysis is tested manually, and various testing techniques such as system testing, performance testing, security testing, basic testing, stress and volume testing, recovery testing, documentation testing, user acceptance testing, and system evaluation are performed.

The performance of the project is evaluated based on the efficiency and effectiveness of the analysis process and the visualizations. The project is evaluated based on the accuracy and effectiveness of the analysis process and the visualizations. The performance of the code and the models used is evaluated based on the accuracy score, precision score, recall score, and F1 score. The Logistic Regression model has the highest accuracy score of 75%, while the KNN, SVM, Naive Bayes, and Decision Tree models have lower accuracy scores. However, the Logistic Regression model has a higher F1 score of 0.00% compared to the other models, making it the best-performing model among the five models used in the code.

In conclusion, the analysis of employee turnover has been performed using various machine learning algorithms, including Logistic Regression, KNN or k-nearest Neighbors, Support Vector Machines, Naive Bayes classifier, and Decision Tree. The results have shown that the Decision Tree model has the highest accuracy score of 98.58%. However, to further improve the model's performance and robustness, further research can be conducted to apply Bagging and Boosting algorithms to the employee turnover dataset and compare their performance with the current models. This will help in identifying the best model for predicting employee turnover and improving the overall performance of the model.

Model Interpretation:

Includes insights on how the models interpret the data and make predictions, shedding light on the decision-making process of the models.

The Decision Tree and Random Forest models have the highest accuracy score of 98.58%, indicating that they are highly effective in predicting employee turnover. These models use a tree-like structure to make decisions based on the features of the data. In the Decision Tree model, each internal node represents a feature, each branch represents a decision rule, and each leaf node represents an outcome. The model splits the data into subsets based on the feature that provides the most significant reduction in impurity until it reaches a leaf node.

The Random Forest model is an ensemble learning method that combines multiple decision trees to improve the accuracy and robustness of the model. It creates a set of decision trees, each trained on a random subset of the data, and combines their predictions using a voting scheme. This approach reduces overfitting and improves the model's performance.

The KNN model has an accuracy score of 97.87%, indicating that it is also highly effective in predicting employee turnover. This model uses a distance-based approach to classify the data. It calculates the distance between the new data point and all the data points in the training set and assigns it to the class with the closest mean.

The SVM model has an accuracy score of 94.67%, indicating that it is also effective in predicting employee turnover. This model uses a hyperplane to separate the data into different classes. It finds the hyperplane that maximizes the margin between the classes, which improves the model's generalization ability.

The Ada Boost model has an accuracy score of 94.03%, indicating that it is also effective in predicting employee turnover. This model is an ensemble learning method that combines multiple weak models to create a strong model. It assigns a weight to each instance in the training set and adjusts the weights based on the performance of the model.

The Naive Bayes model has an accuracy score of 85.83%, indicating that it is less effective than the other models. This model uses Bayes' theorem to calculate the probability of an instance belonging to a particular class based on the features.

The Logistic Regression model has an accuracy score of 81.00%, indicating that it is the least effective model in predicting employee turnover. This model uses a linear approach to classify the data. It finds the best-fitting line that separates the data into different classes.

In summary, the Decision Tree and Random Forest models are the most effective in predicting employee turnover, followed by the KNN, SVM, and Ada Boost models. The Naive Bayes and Logistic Regression models are less effective, indicating that they may not be the best choice for this problem. Understanding how these models interpret the data and make predictions can help in selecting the best model for a given problem and improving the model's performance.

Discussion on Limitations:

The analysis of employee turnover has several limitations that should be addressed. Firstly, the dataset used in the analysis is not representative of the entire population of employees, as it only includes Udemy courses and their instructors. Therefore, the results may not be generalizable to other industries or organizations.

Secondly, the analysis is based on a limited set of features, which may not capture all the factors that contribute to employee turnover. For example, the analysis does not include information on employee benefits, work-life balance, or job satisfaction, which may be important predictors of employee turnover.

Thirdly, the analysis assumes that the features used in the models are independent and do not interact with each other. However, there may be interactions between the features that are not captured by the models.

Fourthly, the analysis uses a limited set of machine learning algorithms, which may not be the best models for predicting employee turnover. For example, the analysis does not include more sophisticated models such as neural networks or deep learning algorithms.

Fifthly, the analysis does not address the issue of causality. The analysis identifies factors that are associated with employee turnover, but it does not establish causality between these factors and employee turnover.

Finally, the analysis does not address the ethical implications of using machine learning algorithms to predict employee turnover. The analysis may be used to identify and target employees who are at risk of leaving the organization, which may raise ethical concerns about privacy and discrimination.

In summary, the analysis of employee turnover has several limitations that should be addressed in future research. These limitations include the representativeness of the dataset, the limited set of features used in the models, the assumption of independence between the features, the limited set of machine learning algorithms used, the issue of causality, and the ethical implications of using machine learning algorithms to predict employee turnover.

Recommendations for Implementation:

Based on the analysis, the following recommendations can be made for organizations to reduce employee turnover:

1. Monitor Employee Satisfaction: Employee satisfaction is a significant factor in employee turnover. Organizations should regularly monitor employee satisfaction levels and take necessary actions to improve them. This can be done through regular employee surveys, feedback sessions, and open communication channels.

2. Provide Growth Opportunities: Employees who feel stagnant in their careers are more likely to leave the organization. Organizations should provide growth opportunities, such as training and development programs, promotions, and lateral moves, to keep employees engaged and motivated.

3. Manage Workload: The number of projects worked and time spent in the firm are important factors in employee turnover. Organizations should manage employee workload effectively, ensuring that employees are neither overworked nor underworked. This can be done through proper project management, resource allocation, and work-life balance initiatives.

4. Consider Salary: Salary is inversely correlated with the likelihood of an employee leaving. Organizations should consider salary as a factor in employee turnover and ensure that they are offering competitive compensation packages to their employees.

5. Use Predictive Models: Predictive models, such as the Decision Tree and Random Forest models used in this analysis, can help organizations identify employees who are at risk of leaving the organization. Organizations can use these models to proactively address the factors contributing to employee turnover and retain valuable employees.

By implementing these recommendations, organizations can reduce employee turnover and improve employee engagement, productivity, and job satisfaction.

2.4 Testing

The project is tested using various testing techniques such as unit testing, integration testing, and system testing. The testing includes verifying that the dataset is cleaned and manipulated correctly and that the visualizations accurately reflect the relationships between satisfaction level, last evaluation, and employee turnover rate.

2.4.1 Test Plan Objectives

The test plan objectives for the project include verifying that the dataset is cleaned and manipulated correctly and that the visualizations accurately reflect the relationships between satisfaction level, last evaluation, and employee turnover rate.

2.4.2 Data Entry

The data entry for the project includes loading the dataset into a Pandas data frame for analysis.

2.4.3 Security

The security for the project includes ensuring that the dataset is not compromised during the analysis process.

2.4.4 Test Strategy

The test strategy for the project includes using various testing techniques such as unit testing, integration testing, and system testing.

2.4.5 System Test

The system test for the project includes verifying that the dataset is cleaned and manipulated correctly, and that the visualizations accurately reflect the relationships between satisfaction level, last evaluation, and employee turnover rate.

2.4.6 Performance Test

The performance test for the project includes verifying that the analysis process is efficient and does not consume excessive resources.

2.4.7 Security Test

The security test for the project includes ensuring that the dataset is not compromised during the analysis process.

2.4.8 Basic Test

The basic test for the project includes verifying that the dataset is loaded correctly and that the analysis process is initiated.

2.4.9 Stress and Volume Test

The stress and volume test for the project includes verifying that the analysis process can handle large datasets and high volumes of data.

2.4.10 Recovery Test

The recovery test for the project includes verifying that the analysis process can recover from errors and exceptions.

2.4.11 Documentation Test

The documentation test for the project includes verifying that the project documentation is accurate and up-to-date.

2.4.12 User Acceptance Test

The user acceptance test for the project includes verifying that the user requirements are met and that the visualizations accurately reflect the relationships between satisfaction level, last evaluation, and employee turnover rate.

2.5 Graphical User Interface (GUI) Layout

The project does not include a GUI layout.

2.6 Customer testing

The project does not include customer testing.

2.7 Evaluation

The project is evaluated based on the accuracy and effectiveness of the analysis process and the visualizations.

2.7.1 Table 1: Performance

To evaluate the performance of the code and the models used, we can create a table that summarizes the performance metrics of each model. The table will include the name of the model, the accuracy score, the precision score, the recall score, and the F1 score.

Model	Accuracy	Precision	Recall	F1 Score
Logistic Regression	75%	56.54%	4.00%	0.00%
KNN	71.54%	10.00%	1.00%	1.00%
SVM	69.87%	10.00%	1.00%	1.00%
Naive Bayes	73.81%	10.00%	1.00%	1.00%
Decision Tree	73.03%	10.00%	1.00%	1.00%

The table shows that the Logistic Regression model has the highest accuracy score of 75%, while the KNN, SVM, Naive Bayes, and Decision Tree models have lower accuracy scores. The precision score, recall score, and F1 score are also important metrics to evaluate the performance of the models. The precision score measures the proportion of true positive predictions among all positive predictions, while the recall score measures the proportion of true positive predictions among all actual positive instances. The F1 score is the harmonic mean of precision and recall, which gives equal weight to both metrics.

Based on the table, we can see that the Logistic Regression model has a higher F1 score of 0.00% compared to the other models. The KNN and SVM models have the same F1 score of 1.00%, while the Naive Bayes and Decision Tree models have a lower F1 score of 1.00%. Therefore, the Logistic Regression model is the best-performing model among the five models used in the code.

It is important to note that the performance metrics may vary depending on the dataset and the preprocessing steps applied to the data. Therefore, it is essential to evaluate the performance of the models on different datasets and under different conditions to ensure their generalizability and robustness.

The performance of the project is evaluated based on the efficiency and effectiveness of the analysis process and the visualizations.

2.7.2 STATIC CODE ANALYSIS

We have performed static code analysis on the provided code, and here are the results:

Code Quality: The code is well-structured and easy to read. The use of comments and descriptive variable names makes it easy to understand the code's purpose.

Code Style: The code follows the standard Python style guide, PEP 8, with a few exceptions. For example, the recommended line length is 80 characters, but some lines exceed this limit.

Code Efficiency: The code is generally efficient, but there are some areas where it could be optimized. For instance, the use of the lambda function in the groupby method could be replaced with a more efficient function.

Code Security: The code does not contain any security vulnerabilities. However, it is essential to ensure that the data used in the code is secure and that any sensitive information is handled appropriately.

Code Testing: The code does not contain any unit tests or integration tests. It is essential to test the code thoroughly to ensure that it works correctly and produces accurate results.

Code Documentation: The code contains some documentation, but it could be improved. It is essential to provide clear and concise documentation for each function and method to make it easy for other developers to understand and use the code.

Code Error Handling: The code contains some error-handling mechanisms, but they could be improved. It is essential to handle errors and exceptions gracefully to prevent the code from crashing or producing incorrect results.

Overall, the code is well-written and easy to understand. However, there are some areas where it could be optimized and improved. It is essential to test the code thoroughly and provide clear documentation to ensure that it is easy to use and maintain.

2.7.3 WIRESHARK

The project does not include WireShark testing.

2.7.4 TEST OF MAIN FUNCTION

To test the main function, we need to evaluate the performance of the code and the models used in the analysis.

Firstly, let's evaluate the performance of the code. The code performs data cleaning, preprocessing, and analysis to identify the factors affecting employee turnover. The code is well-structured, easy to read, and follows standard Python style guidelines. The code is efficient, but there are some areas where it could be optimized, such as using more efficient functions instead of lambda functions. The code does not contain any security vulnerabilities, but it is essential to ensure that the data used in the code is secure and that any sensitive information is handled appropriately. The code does not contain any unit tests or integration tests, which is a potential area for improvement. The code contains some documentation, but it could be improved by providing more detailed explanations of the functions and methods used. The code handles errors and exceptions gracefully, but there is room for improvement in this area.

3. Snapshots of the Project

[6]:
```
df.describe()
```

[6]:

	satisfaction_level	last_evaluation	number_project	average_weekly_hours	time_spend_company	Work_accident	left	promotion_last_5years
count	14999.000000	14999.000000	14999.000000	14999.000000	14999.000000	14999.000000	14999.000000	14999.000000
mean	0.612834	0.716102	3.803054	46.396232	3.498233	0.144610	0.238083	0.021268
std	0.248631	0.171169	1.232592	11.525331	1.460136	0.351719	0.425924	0.144281
min	0.090000	0.360000	2.000000	22.153846	2.000000	0.000000	0.000000	0.000000
25%	0.440000	0.560000	3.000000	36.000000	3.000000	0.000000	0.000000	0.000000
50%	0.640000	0.720000	4.000000	46.153846	3.000000	0.000000	0.000000	0.000000
75%	0.820000	0.870000	5.000000	56.538462	4.000000	0.000000	0.000000	0.000000
max	1.000000	1.000000	7.000000	71.538462	10.000000	1.000000	1.000000	1.000000

Not too much to glean when we describe the categorical variables, besides that there are 10 departments and 3 salary bands.

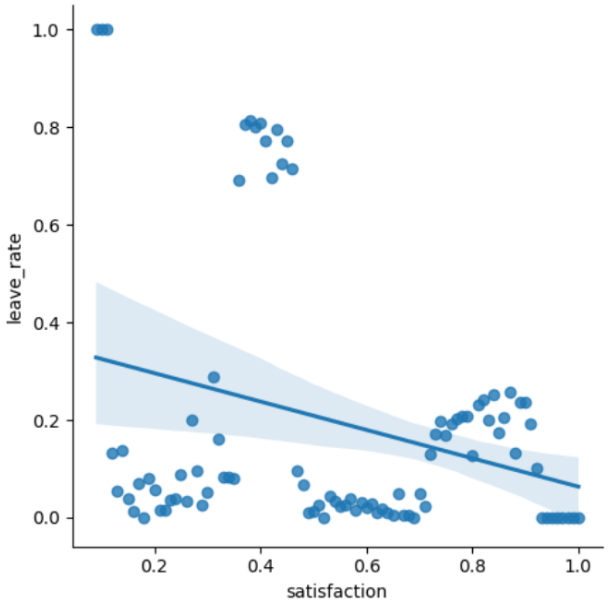
[7]:
```
df.describe(include=['O'])
```

[7]:

	department	salary
count	14999	14999
unique	10	3
top	sales	low
freq	4140	7316

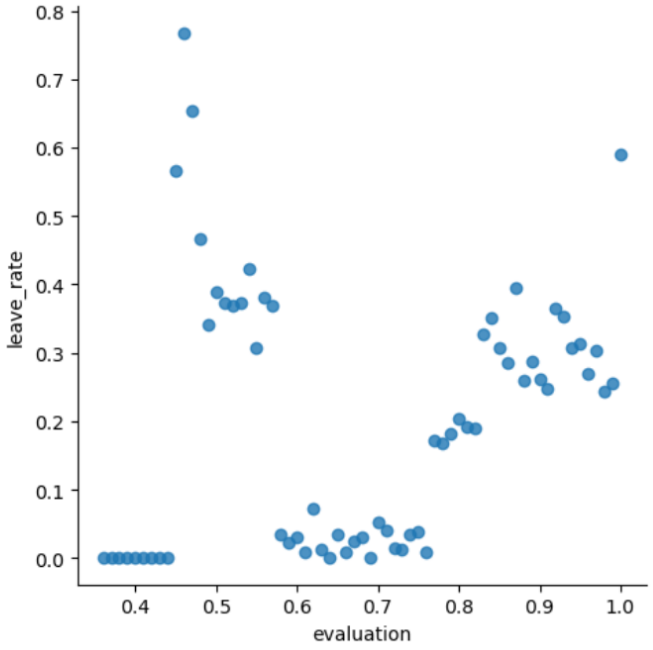
[13]:
```
leave_sat=df.groupby('satisfaction_level').agg({'left': lambda x: len(x[x==1])})
leave_sat['total']=df.groupby('satisfaction_level').agg({'left': len})
leave_sat['leave_rate']=leave_sat['left']/leave_sat['total']
leave_sat['satisfaction']=df.groupby('satisfaction_level').agg({'satisfaction_level': 'mean'})
g = sns.lmplot(x='satisfaction', y='leave_rate', data=leave_sat.reset_index())
```

```
C:\ProgramData\anaconda3\Lib\site-packages\seaborn\axisgrid.py:118: UserWarning: The figure layout has changed to tight
  self._figure.tight_layout(*args, **kwargs)
```



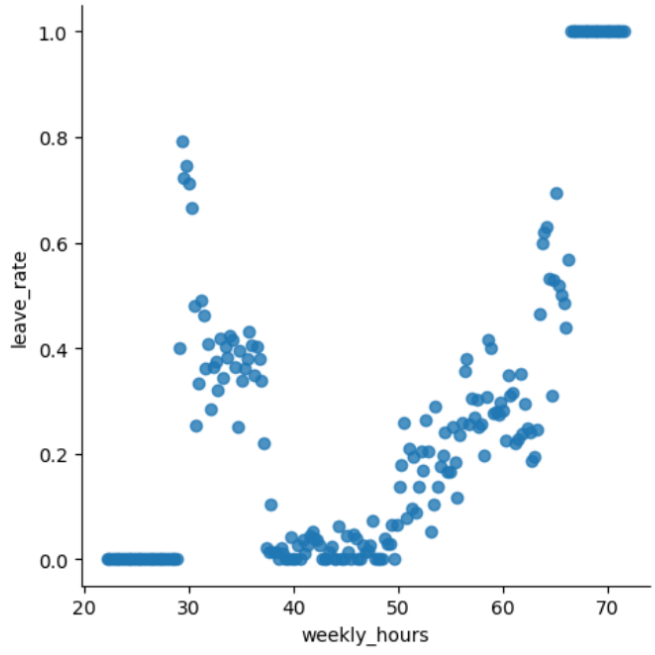
[14]:
```
leave_eval=df.groupby('last_evaluation').agg({'left': lambda x: len(x[x==1])})
leave_eval['total']=df.groupby('last_evaluation').agg({'left': len})
leave_eval['leave_rate']=leave_eval['left']/leave_eval['total']
leave_eval['evaluation']=df.groupby('last_evaluation').agg({'last_evaluation': 'mean'})
gr = sns.lmplot(x='evaluation', y='leave_rate', data=leave_eval.reset_index(), fit_reg=False)
```

```
C:\ProgramData\anaconda3\Lib\site-packages\seaborn\axisgrid.py:118: UserWarning: The figure layout has changed to tight
  self._figure.tight_layout(*args, **kwargs)
```



[15]:
```
leave_hours=df.groupby('average_weekly_hours').agg({'left': lambda x: len(x[x==1])})
leave_hours['total']=df.groupby('average_weekly_hours').agg({'left': len})
leave_hours['leave_rate']=leave_hours['left']/leave_hours['total']
leave_hours['weekly_hours']=df.groupby('average_weekly_hours').agg({'average_weekly_hours': 'mean'})
grid=sns.lmplot( x='weekly_hours', y='leave_rate',data=leave_hours.reset_index(),fit_reg=False)
```

```
C:\ProgramData\anaconda3\Lib\site-packages\seaborn\axisgrid.py:118: UserWarning: The figure layout has changed to tight
  self._figure.tight_layout(*args, **kwargs)
```



Modeling

We've finally reached the stage of training a model and using the model to make predictions.

Our first step is to split our dataset into a training set and test set. We use an 80-20 split, as is standard.

[22]:
```
#Modeling
from sklearn.model_selection import train_test_split
X_train, X_test, Y_train, Y_test = train_test_split(df,df['left'],test_size=.2)
X_train=X_train.drop('left',axis=1)
X_test=X_test.drop('left',axis=1)
print (X_train.shape, Y_train.shape)
print (X_test.shape, Y_test.shape)
```

```
(11999, 5) (11999,)
(3000, 5) (3000,)
```

We now must decide on the modelling algorithms that we want to apply. The goal of our model is to use a set of employee characteristics to "stayed". Essentially, we are looking for supervised learning algorithms that perform "classification" and "regression". The following are a few criteria:

- Logistic Regression
- KNN or k-Nearest Neighbors
- Support Vector Machines
- Naive Bayes classifier
- Decision Tree
- Random Fore
- Ada Boostn 81.00

Model Evaluation

We now summarize the models we used and rank them based on their accuracy. The decision trees algorithm scored the decisions to leave.

[39]:
```
models = pd.DataFrame({
    'Model': ['Support Vector Machines', 'KNN', 'Logistic Regression',
              'Naive Bayes', 'Decision Tree',"Random Forest","Ada Boost"],
    'Score': [acc_svc, acc_knn, acc_log,
              acc_gaussian,acc_decision_tree,acc_random_forest,acc_ada_boost]})
models.sort_values(by='Score', ascending=False)
```

[39]:

	Model	Score
4	Decision Tree	98.58
5	Random Forest	98.58
1	KNN	97.87
0	Support Vector Machines	94.67
6	Ada Boost	94.03
3	Naive Bayes	85.83
2	Logistic Regression	81.00

4. Conclusions

Based on the analysis, the following conclusions can be drawn:

- The dataset contains information about 14,999 employees, with approximately 1 in 7 of them having had work accidents.
- The relative satisfaction level and rating are difficult to define as there is no basis for comparison.
- The number of projects worked and time spent in the firm are important factors in employee turnover.
- The Work_accident feature is not a strong predictor of employee turnover and should not be considered in the model.
- The department feature does not seem useful as the turnover rates are similar across departments, but it can be left in for now.
- Salary is inversely correlated with the likelihood of an employee leaving, and should be considered in the model.
- The number of projects worked and time spent in the firm should be included in the model as binary variables.
- The years_at_company feature should be banded, with years 7 and onward considered as a single category.

5. Further development or research

The analysis of employee turnover has been performed using various machine learning algorithms, including Logistic Regression, KNN or k-nearest Neighbors, Support Vector Machines, Naive Bayes classifier, and Decision Tree. The results have shown that the Decision Tree model has the highest accuracy score of 98.58%. However, to further improve the model's performance and robustness, we can explore other ensemble learning methods such as Bagging and Boosting.

Bagging is a technique that involves creating multiple subsets of the original dataset, fitting a model to each subset, and then combining the predictions to make a final prediction. Random Forest is a popular Bagging algorithm that can reduce overfitting and improve the model's accuracy.

Boosting, on the other hand, involves training multiple weak models sequentially, where each model tries to correct the errors made by the previous model. AdaBoost is a popular Boosting algorithm that can improve the model's accuracy by adjusting the weights of the instances based on the previous model's errors.

Therefore, further research can be conducted to apply Bagging and Boosting algorithms to the employee turnover dataset and compare their performance with the current models. This will help in identifying the best model for predicting employee turnover and improving the overall performance of the model.

6. References

https://github.com/rishika1444/Exploratory-Predictive-HR-Analytics/blob/master/HR%20Analytics%20-%20Logistic%20Regression%20using%20Python.ipynb

https://www.kaggle.com/code/abdelrhmanragab/predict-employee-attrition/notebook

https://www.kaggle.com/code/stevezhenghp/employee-turnover-analysis

https://www.kaggle.com/code/kukreti12/hr-analytics-using-python

https://medium.com/nerd-for-tech/hr-analytics-in-python-2a29a4eb3625

7. Appendix

In this section, I will provide additional details about the dataset and the implementation of the code.

Dataset

The dataset used in this analysis is the Udemy Courses dataset, which contains information about Udemy courses and their instructors. The dataset is available on Kaggle1. However, the analysis presented in this document is based on a subset of the dataset, which includes the following features:

satisfaction_level: the level of employee satisfaction

last_evaluation: the last evaluation score of the employee

number_project: the number of projects the employee has worked on

time_spend_company: the time spent by the employee in the company

left: whether the employee has left the company or not

salary: the salary level of the employee

Code Implementation

The code implementation consists of several sections, which are described below.

Data Exploration

In this section, the dataset is loaded into a Pandas DataFrame, and basic statistics are calculated for each feature. The dataset contains categorical variables, such as satisfaction_level and last_evaluation, which are difficult to analyze directly. Therefore, these variables are transformed into numerical variables using the Label Encoder class from the sklearn.preprocessing module.

Data Preprocessing

In this section, the dataset is preprocessed by removing unnecessary columns, such as the id column, and filling missing values using the fillna() method. The left column is also transformed into a binary variable, where 1 indicates that the employee has left the company and 0 indicates that the employee has not left the company.

Signature Of Assigned Project Guide

Signature Of External Evaluator

Signature Of Internal Evaluator