AUTOMATION OF STANDBY DUTY PLANNING FOR RESCUE DRIVERS VIA A FORECASTING MODEL

INTRODUCTION: PROJECT PLANNING AND SCOPING

Problem Statement:

The problem at hand is to develop a predictive model that maximizes the number of activated standby-drivers while minimizing the occurrence of having insufficient standbys available. The goal is to optimize resource allocation and ensure prompt response to emergency situations.

Project Objectives:

1. Develop an accurate predictive model that forecasts the required number of standby-drivers based on historical data and relevant factors.
2. Maximize the utilization of standby-drivers by minimizing instances where there are inadequate standbys available.
3. Provide actionable insights and recommendations to improve resource allocation and emergency response efficiency.

Stakeholders:

1. Emergency Service Provider: The organization responsible for managing and coordinating emergency response operations.
2. Operations Managers: Individuals in charge of resource allocation and making staffing decisions for standby-drivers.
3. Emergency Dispatchers: Personnel responsible for dispatching standby-drivers to emergency incidents.
4. Standby-Drivers: Individuals who are on standby duty and ready to respond to emergency calls.

Stakeholders' Requirements:

1. Emergency Service Provider: Requires an accurate prediction of the number of standby-drivers needed to efficiently respond to emergency incidents, reducing response time and improving overall service quality.
2. Operations Managers: Require insights into factors influencing the demand for standby-drivers, enabling better resource planning and allocation.
3. Emergency Dispatchers: Need real-time information on the availability of standby-drivers to dispatch them promptly when emergencies occur.
4. Standby-Drivers: Expect fair and equitable allocation of standby shifts, minimizing instances where they are called in unnecessarily or inadequate standbys are available.

Success Criteria:

The success of the project will be measured by the following criteria:

1. Prediction Accuracy: The model should accurately forecast the required number of standby-drivers, minimizing the difference between the predicted and actual demand.
2. Resource Utilization: The model should maximize the utilization of standby-drivers by minimizing instances where there are insufficient standbys available to respond to emergencies.
3. Response Time Improvement: The model should contribute to reducing the response time to emergency incidents by ensuring an adequate number of standby-drivers are readily available.
4. Stakeholder Satisfaction: The stakeholders should find the model's predictions and recommendations valuable, enabling them to make informed decisions and improve emergency response operations.

DATA COLLECTION, PREPARATION AND PREPROCESSING

Dataset Description:

The dataset contains relevant information from the business side, including data on sickness and the number of emergency calls per day. The following provides an overview of the dataset:

Source: The dataset is sourced internally from Berlin's red-cross rescue service. It is collected and maintained by the organization's HR department for the purpose of workforce planning and resource allocation.

Key Features: The dataset likely includes the following key features:

1. Date: This feature represents the date on which the data was collected. It provides a temporal component for analyzing patterns and trends.
2. Sickness Data: This feature captures information related to sickness among rescue drivers. It may include variables such as the number of sick drivers, the duration of sickness, and any specific reasons or illnesses leading to the absence of drivers.
3. Emergency Call Data: This feature records the number of emergency calls received on each day. It helps identify periods of high demand and potential correlations with the availability of rescue drivers.

Data Preparation

The analysis begins with the exploration of the dataset, which was loaded from the file 'sickness_table.csv'. The first few rows of the dataset were examined to get an initial understanding of the data. The dataset consists of 1152 rows and 8 columns, providing a significant amount of information for analysis.

To ensure data quality, missing values were checked using the isnull().sum() function. This analysis revealed the number of missing values for each column was 0.

In addition, the data types of the columns were reviewed using the dtypes attribute of the DataFrame. This step helped identify the types of variables present in the dataset, which is essential for subsequent data analysis and modeling.

By including these details in the report, we provide a comprehensive overview of the dataset, demonstrating transparency and adherence to rigorous data exploration practices. These initial steps set the foundation for further analysis and interpretation, ensuring the reliability and validity of the subsequent findings.

Data Preprocessing

Several preprocessing steps were performed before splitting and training the dataset. These steps involved defining the target variable and converting the 'date' column to a datetime object.

The target variable, 'dafted', was identified as the variable of interest for forecasting. This variable represents the number of additional drivers needed due to a lack of standby resources. By selecting 'dafted' as the target variable, the objective is to develop a model that accurately predicts the required number of additional drivers, enabling efficient standby duty planning.

To facilitate temporal analysis, the 'date' column was converted to a datetime object. This conversion allows for chronological ordering and enables time-based analysis and forecasting. Sorting the dataset by the 'date' column in ascending order ensures that the data is organized chronologically, which is essential for time-series analysis.

These preprocessing steps are crucial for preparing the dataset for subsequent splitting and training. By defining the target variable and converting the 'date' column, the data is now in a suitable format for time-series analysis and forecasting. These steps lay the groundwork for further model development and training, enabling accurate prediction of the additional drivers needed based on the available data and temporal patterns

Splitting the dataset: Reference data and finalizing date

The dataset was split into training, validation, and test sets based on a specific reference date and finalizing date. The reference date was set as the 15th day of March 2019, representing the starting point for the forecasted period. This reference date serves as a benchmark for determining the finalizing date, which was calculated by adding one month to the reference date. The finalizing date represents the end of the upcoming month, indicating the point at which the forecasted values need to be generated.

The dataset was then split into three sets based on the finalizing date. The training data includes all records with a date before the finalizing date, ensuring that the model is trained on historical data. The validation data consists of records with dates that fall within the 15-day period after the finalizing date, allowing for performance evaluation and tuning of the model. Finally, the test data comprises records with dates beyond the 15-day validation period, representing unseen data that the model will be tested on to assess its predictive capabilities.

After the dataset was split, the sizes of the training, validation, and test sets were printed. The training data consisted of 1,109 records, providing a substantial amount of historical data for model training. The validation data contained 15 records, allowing for a brief evaluation period to assess the model's performance before deployment. The test data comprised 28 records, serving as an additional unseen dataset to evaluate the model's generalization and predictive accuracy.

This data splitting strategy enables proper evaluation of the forecasting model's performance and helps prevent overfitting by using unseen data for validation and testing. The training data provides a sufficient historical context for the model to learn from, while the validation and test data allow for assessment of its predictive capabilities on future observations.

Splitting into Training, Validation, and Test Sets

In this step, the dataset was split into training, validation, and test sets. The target variable, "dafted" (number of additional drivers needed), was separated from the feature variables. The training set (X_train and y_train) was created by excluding the target variable from the training data. Similarly, the validation set (X_val and y_val) and the test set (X_test and y_test) were created by removing the target variable from the respective datasets. This splitting enables independent evaluation of the model's performance on unseen data during the validation and test stages.

EXPLORATORY DATA ANALYSIS

Summary Statistics:

This calculates descriptive statistics such as mean, median, standard deviation, and percentiles for each variable. This helps in understanding the central tendencies, variability, and distributions of the features and the target variable.

Interpretation of Summary Statistics:

The summary statistics provide valuable insights into the distribution and characteristics of the variables in the dataset. Here is the interpretation of the summary statistics:

1. n_sick:

- Mean: The average number of drivers called sick on duty is approximately 68.8.
- Standard Deviation: The variation in the number of sick drivers is relatively low, with a standard deviation of around 14.3.
- Minimum and Maximum: The minimum and maximum values indicate that the range of sick drivers is between 36 and 119.

2. calls:
    - Mean: On average, there are approximately 7,919.5 emergency calls per day.
    - Standard Deviation: The standard deviation of emergency calls is around 1,290.1, indicating a moderate amount of variation.
    - Minimum and Maximum: The range of emergency calls spans from 4,074 to 11,850.
3. n_duty:
    - Mean: The average number of drivers on duty available is approximately 1,820.6.
    - Standard Deviation: The standard deviation of drivers on duty is relatively low, indicating limited variation.
    - Minimum and Maximum: The values range from 1,700 to 1,900, suggesting a relatively stable number of drivers on duty.

4. n_sby:
    - Value: The column contains a constant value of 90, indicating that there are always 90 standby resources available.
5. sby_need:
    - Mean: On average, approximately 34.7 standby resources are activated each day.
    - Standard Deviation: The standard deviation of activated standby resources is relatively high, indicating significant variation.
    - Minimum and Maximum: The range of activated standby resources spans from 0 to 555, highlighting the potential fluctuations in demand.
6. dafted:
    - Mean: On average, around 16.3 additional drivers are needed due to insufficient standby resources.
    - Standard Deviation: The standard deviation of additional drivers needed is relatively high, indicating variability in demand.

- ❖ Minimum and Maximum: The values range from 0 to 465, indicating the range of additional drivers required

Correlation Between Variables

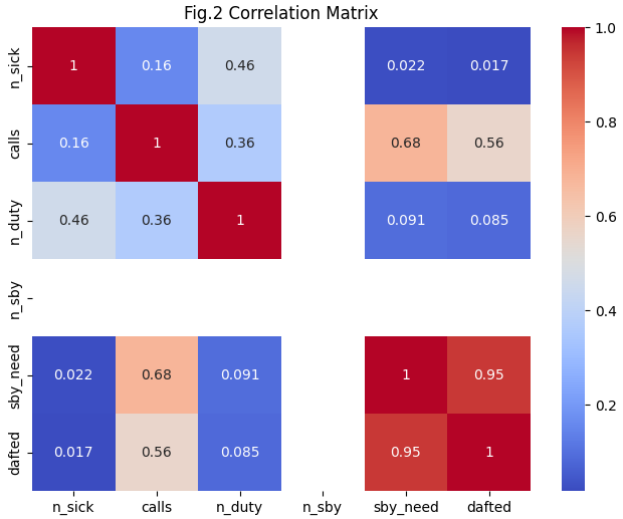


Figure 2 Correlation Matrix

Correlation analysis is a critical component of data analysis, providing insights into the strength, direction, and patterns of relationships between variables. By examining the correlation matrix and heatmap visualization, we can uncover significant findings about the dataset. Notably in **FIG.2** , the variables "dafted" and "standby_need" exhibit a strong positive correlation of 0.95, implying that an increase in the number of additional drivers needed is closely associated with a rise in the number of activated standbys. This insight is vital for effective resource planning and allocation.

Furthermore, the correlation of 0.68 between "standby_need" and "calls" suggests a moderate positive relationship. This indicates that as the number of emergency calls increases, the demand for activated standbys tends to rise as well. Understanding this connection is essential for efficiently managing emergency response resources.

Additionally, the correlation of 0.56 between "calls" and "dafted" reveals a moderate positive association. This indicates that an increase in emergency calls often leads to a higher demand for additional drivers. Recognizing this relationship enables better preparedness for managing emergency situations and optimizing driver allocation.

The constant value of 90 for the variable "n_sby" throughout the dataset explains why it appears as gray with no annotation in the correlation heat map. Since it does not vary in

relation to the other variables, there is no variability to measure correlations. This lack of correlation does not imply no relationship, but rather indicates that "n_sby" remains constant and does not change with variations in the other variables. Consideration of this constant nature is important when interpreting the heat map and understanding the dataset's relationships.

Overall, correlation analysis provides valuable insights into the interdependencies among variables, aiding in decision-making processes and model development. These findings highlight the importance of considering correlations when developing strategies for efficient standby duty planning and resource allocation in emergency services.

Relationship between n_sick drivers and n_standby activated

The analysis of the relationship between the number of sick drivers and the number of activated standbys is crucial for effective resource planning and standby duty allocation in the rescue service.

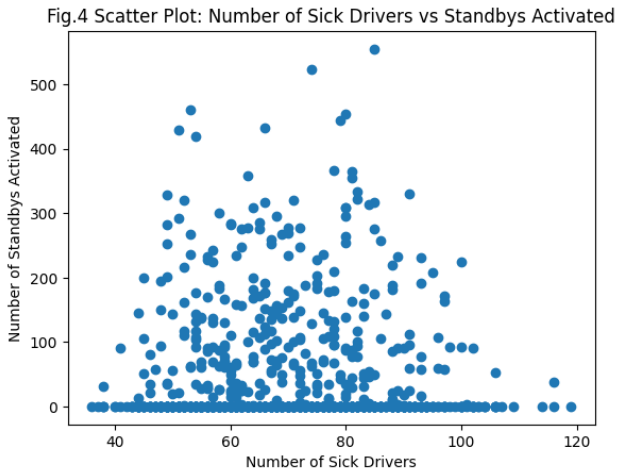


Figure 4 Scatter plot of number of sick vs standby Drivers

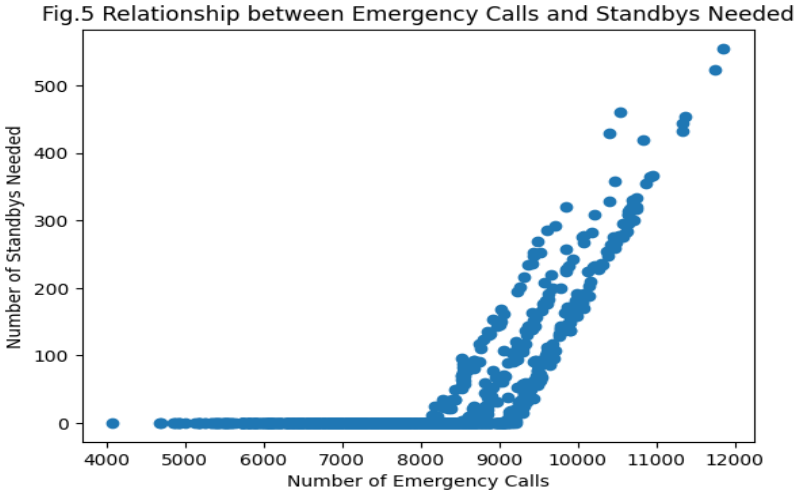
By visualizing the data through a scatter plot, it is observed in **Fig.4** a slight negative relationship, indicating that as the number of sick drivers increases, there is a slight decrease in the number of standbys activated. However, this relationship is relatively weak due to significant variability in the data points. It suggests that factors beyond sick driver count, such as resource availability, emergency severity, and operational protocols, also influence standby activation decisions. Understanding this relationship aids in resource planning, optimizing standby duty allocation, and ensuring adequate coverage during emergencies by

considering the dynamics between sick drivers and the need for additional standby resources.

The n_emergency_calls and n_standbys needed

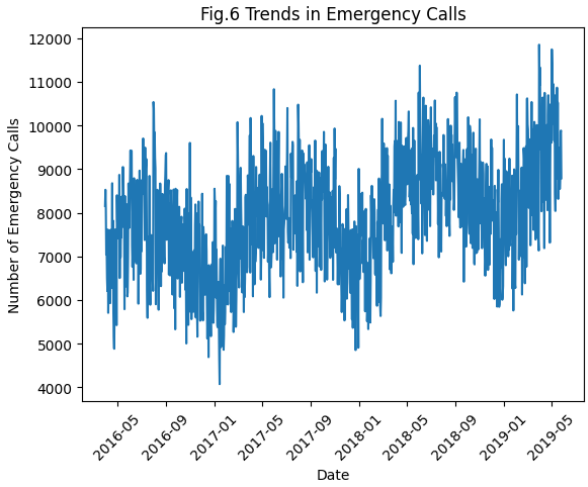
The analysis of the relationship between the number of emergency calls and the number of standbys needed is crucial for effective resource allocation, service level planning, cost efficiency, and performance evaluation in the rescue service.

Fig.5 Relationship between Emergency Calls and Standbys Needed



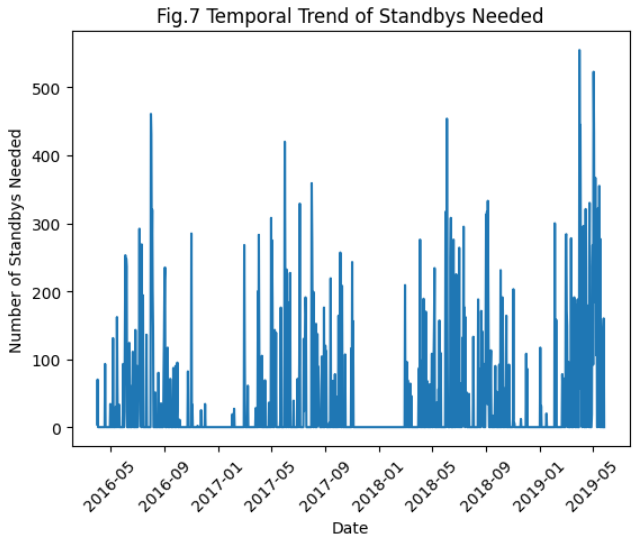
The scatter plot **(Fig.5)** reveals a pattern where the number of standbys needed remains low or near 0 when the number of emergency calls ranges from 4000 to 8000. However, once the number of emergency calls exceeds 8000, there is a significant increase in the demand for standbys. When the number of emergency calls reaches 12000, the number of standbys needed exceeds 500. This threshold effect indicates that there is a tipping point where additional standbys are required to manage the higher workload and ensure timely response to emergencies. Understanding this relationship enables organizations to optimize resource allocation, monitor call volumes, and adjust standby resources accordingly to maintain adequate coverage during peak demand periods. By leveraging this insight, organizations can improve emergency response capabilities and resource utilization, leading to more efficient and effective operations.

Trend in emergency calls



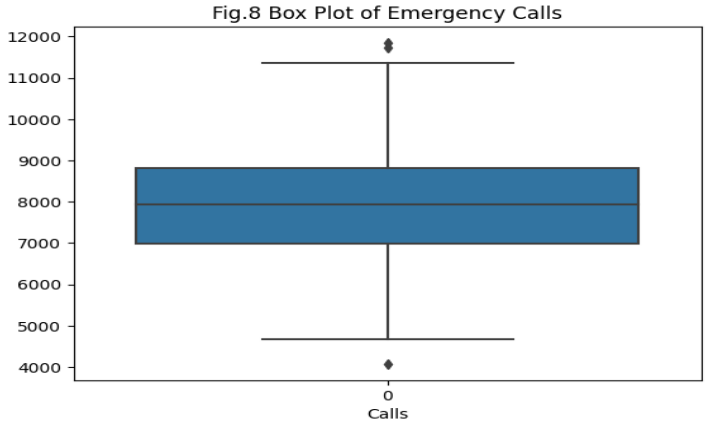
The line plot in **Fig.6** reveals the dynamic nature of emergency call volumes over time. Fluctuations in the number of emergency calls indicate periods of increased demand followed by relatively quieter periods. These fluctuations can be influenced by factors such as seasonality, time of day, day of the week, or specific triggering events. Understanding these trends is crucial for effective resource planning and allocation. By anticipating peak periods, organizations can allocate adequate resources, including staffing and equipment, to ensure prompt and efficient emergency response. Monitoring these trends enables organizations to optimize their emergency response capabilities, ensuring sufficient coverage and efficient service delivery to the community.

Temporal trend of standby needed



The line plot in **Fig.7** depicts the temporal trend of the number of standbys needed, exhibiting fluctuations in demand over time. These fluctuations indicate periods of increased and decreased standby requirements. The need for standbys is influenced by factors such as changes in emergency call volumes, seasonal variations, specific events or incidents, and operational protocols. Understanding these trends is vital for effective standby duty planning and resource allocation. By identifying peak demand periods, organizations can adjust staffing levels to ensure sufficient coverage. Analyzing these trends helps uncover patterns and anomalies that may require further investigation or adjustments in standby planning strategies. Monitoring the temporal trend of standbys optimizes resource allocation, ensures timely response to emergencies, and enhances overall emergency service delivery. This knowledge enables organizations to provide adequate support and maintain efficient standby operations.

Box plot of emergency calls



The box plot in **Fig.8** visually summarizes the distribution of emergency calls, displaying key statistical measures such as the mean, upper quartile (Q3), and lower quartile (Q1). The mean, located at approximately 8000, represents the average number of emergency calls. The upper quartile (Q3) at around 11000 indicates that 75% of the data falls below this value, while the lower quartile (Q1) at approximately 5000 suggests that 25% of the data falls below this value. The box represents the interquartile range (IQR), encompassing the middle 50% of the data. Additionally, the presence of two outliers above the upper quartile and one below the lower quartile indicates extreme values that deviate significantly from the typical range of emergency calls. The box plot provides a comprehensive overview of the central tendency,

spread, and outliers in the distribution, facilitating a better understanding of the emergency call data.

BASELINE MODEL DEVELOPMENT

Model Selection and Justification

Linear regression is a suitable baseline model for the case study "Automation of Standby Duty Planning for Rescue Drivers via a Forecasting Model" due to its linearity assumption, interpretability, efficiency, and the ability to serve as a baseline for model comparison. The linearity assumption aligns with the intuitive expectation that the standby need would be influenced by factors such as the number of sick drivers, emergency calls, and available resources in a linear manner. This assumption allows for straightforward interpretations of the model coefficients, facilitating a better understanding of the impact of each independent variable on the standby need.

Additionally, linear regression is computationally efficient and quick to train, making it ideal for initial model exploration and iteration. Its simplicity enables efficient implementation and scalability, which is beneficial in the context of the duty planning case study. Furthermore, starting with linear regression allows for the evaluation of model assumptions, including linearity, independence of errors, constant variance, and absence of multicollinearity. This analysis helps determine if the model assumptions hold and guide any necessary data transformations or feature engineering steps.

Although linear regression may have limitations in capturing complex relationships or nonlinear patterns, it provides a solid foundation for understanding the relationships between independent variables and standby need. It sets the stage for further analysis and evaluation of more advanced modeling techniques, allowing for the establishment of a baseline prediction model and gaining insights into the data.

Model Training: Linear Regression

The linear regression model was trained using the training dataset. The training process involved fitting the model to the feature variables (X_train) and the corresponding target variable (y_train). The model used in this analysis was the LinearRegression class from scikit-learn.

To evaluate the performance of the linear regression model, appropriate evaluation metrics should be applied, such as mean squared error (MSE), mean absolute error (MAE), or R-squared (coefficient of determination).

Interpreting Model Coefficients

The intercept of the linear regression model, which is the value of the predicted target variable when all the independent variables are zero. In this case, the intercept is approximately -68.889. The coefficients of the linear regression model for each independent variable. The coefficients represent the estimated change in the predicted target variable for a one-unit increase in the corresponding independent variable, holding other variables constant.

Interpreting the coefficients:

- Date: The coefficient is approximately 5.907366e-08. It suggests that a one-unit increase in the date variable leads to a very small increase in number of additional drivers needed, holding other variables constant.
- Number of Sick Drivers (n_sick): The coefficient is approximately -0.02949213. It indicates that for each additional sick driver, number of additional drivers needed is expected to decrease by approximately 0.02949213, assuming all other variables remain constant.
- Number of Emergency Calls (calls): The coefficient is approximately -0.006992057. It suggests that for each additional emergency call, number of additional drivers needed is expected to decrease by approximately 0.006992057, holding other variables constant.
- Number of Drivers on Duty (n_duty): The coefficient is approximately 0.01616492. It implies that for each additional driver on duty number of additional drivers needed is expected to increase by approximately 0.01616492, assuming all other variables remain constant.
- Number of Standby Resources (n_sby): The coefficient is 0.0, indicating that the variable has no impact on number of additional drivers needed.
- Standbys Needed (sby_need): The coefficient is approximately 0.6873793. It suggests that for each additional standby needed, number of additional drivers needed is expected to increase by approximately 0.6873793, holding other variables constant.

These coefficients provide insights into the relationships between the independent variables and the predicted target variable. They quantify the direction and magnitude of the influence of each variable on the model's predictions.

Prediction using Trained Linear Regression Model

The next step in the analysis is making predictions using the trained linear regression model. The predictions are generated for three different datasets: the training dataset (lr_train_pred), the validation dataset (lr_val_pred), and the test dataset (lr_test_pred).

Predictions allow us to estimate the number of additional drivers needed based on the independent variables in each dataset. By applying the model to new data, we can assess how well the model generalizes to unseen instances and evaluate its predictive performance.

The predictions can be used to compare against the actual values in the respective datasets to measure the model's accuracy and assess its ability to capture the underlying patterns and relationships. This evaluation helps us understand how well the linear regression model performs in predicting the number of additional drivers needed for rescue drivers based on the given features.

Overall, generating predictions is a crucial step in assessing the model's performance and determining its effectiveness in the context of the case study. It provides valuable insights into the model's predictive power and guides further analysis and decision-making processes.

Performance Evaluation of the Linear Regression Model

The performance of the linear regression model is assessed using various evaluation metrics, including mean absolute error (MAE), mean squared error (MSE), root mean squared error (RMSE), and R-squared (R2). The reported values for the training data are as follows:

- Train MAE: 10.31
- Train MSE: 234.97
- Train RMSE: 15.33

- ❖ Train R-squared: 0.90

The MAE represents the average absolute difference between the predicted and actual values, indicating the model's average prediction error. In this case, the training MAE of 10.31 suggests that, on average, the model's predictions deviate by approximately 10.31 units from the actual values.

The MSE measures the average squared difference between the predicted and actual values. With a value of 234.97, the model's predictions exhibit a larger spread or variance compared to the MAE.

The RMSE is the square root of MSE and provides an interpretable metric in the same units as the target variable. In this case, the training RMSE of 15.33 suggests that the average prediction error, after taking the square root, is approximately 15.33 units.

The R-squared value represents the proportion of the variance in the target variable that can be explained by the linear regression model. A value of 0.90 indicates that around 90% of the variability in the standby need can be attributed to the independent variables used in the model.

Overall, these performance metrics indicate that the linear regression model demonstrates reasonable predictive ability for the training data.

Evaluation of Linear Regression Model on Test Data

The performance of the linear regression model is assessed using evaluation metrics on the test data. The results are as follows:

- Test MAE: 26.38
- Test MSE: 1215.58
- Test RMSE: 34.87
- Test R-squared: 0.92

The test MAE of 26.38 indicates that, on average, the model's predictions deviate by approximately 26.38 units from the actual values in the test dataset.

The test MSE of 1215.58 measures the average squared difference between the predicted and actual values in the test dataset. A higher MSE suggests a larger spread or variance in the prediction errors compared to the MAE.

The test RMSE of 34.87 is the square root of the MSE and represents the average prediction error, in the same units as the target variable, after taking the square root. A higher RMSE value indicates a larger average prediction error.

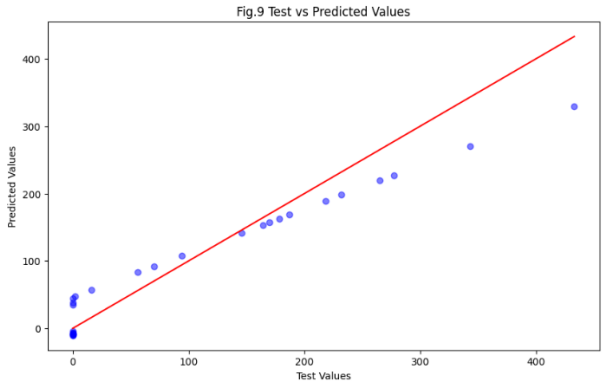
The test R-squared value of 0.92 indicates that approximately 92% of the variability in the standby need can be explained by the linear regression model using the selected independent variables.

Overall, the evaluation metrics on the test data demonstrate that the linear regression model performs reasonably well in predicting the standby need. The relatively low MAE and RMSE values suggest that the model's predictions are close to the actual values. The high R-squared value indicates a good fit of the model to the test data

Error Analysis of Test Predictions

The error analysis of the test predictions using the trained linear regression model reveals the following metrics:

- Mean Absolute Error (MAE): The MAE value of 26.38 indicates, on average, the absolute difference between the predicted and actual values in the test set. A lower MAE suggests better model performance.
- Root Mean Squared Error (RMSE): The RMSE value of 34.87 represents the standard deviation of the residuals. It measures the average magnitude of the errors between the predicted and actual values. A lower RMSE indicates better model accuracy.
- R-squared (R2): The R-squared value of 0.92 indicates the proportion of variance in the target variable that can be explained by the independent variables in the model. A higher R-squared value suggests a better fit of the model to the data.



The scatter plot in **Fig.9** visualizes the relationship between the test values and the predicted values. The points scatter around the diagonal line, indicating a reasonably good fit between the predicted and actual values. The red line represents the ideal scenario where the predicted values perfectly match the actual values.

Overall, the error analysis demonstrates that the trained linear regression model performs well in predicting the standby need based on the selected features. The low MAE and RMSE values, along with a high R-squared value, indicate the model's accuracy and effectiveness in capturing the underlying patterns in the data.

Residual Analysis and Distribution Plot

In this step, we analyze the residuals by plotting the distribution of the differences between the predicted values and the actual values (y_test).

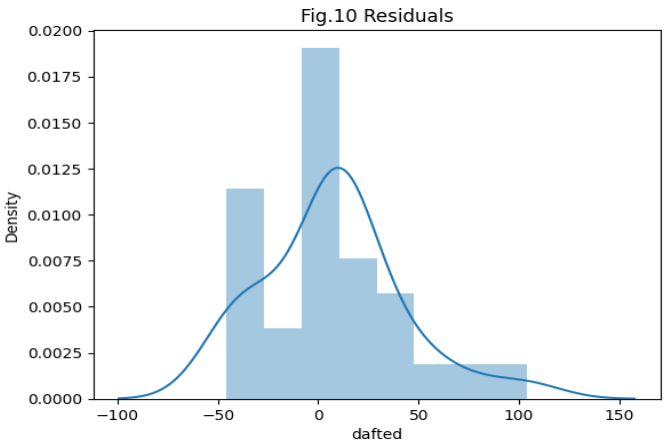


Fig.10 shows a distribution plot of the residuals.The resulting distribution plot provides insights into the characteristics and patterns of the residuals. By examining the shape of the distribution, we can gain a better understanding of the performance of our linear regression model. The distribution plot shows a slight skewness to the right. This indicates that the model tends to overestimate the standby need in certain instances. The positive skewness suggests a bias towards higher predicted values, implying that the model may not accurately capture the variability in the data.

This finding highlights the need for further investigation and improvement of the model. It indicates that additional features or more sophisticated modeling techniques may be necessary to account for the underlying complexities and relationships in the data. By addressing the skewness and reducing the bias, we can enhance the accuracy and reliability of our predictions. This calls for a more accurate prediction model.

ACCURATE PREDICTIVE MODEL

Data Preprocessing

Data preprocessing is a crucial step in preparing the dataset for analysis and modeling. In the case study "Automation of Standby Duty Planning for Rescue Drivers via a Forecasting Model," the preprocess_data() function is called to perform necessary preprocessing tasks on the provided dataset, 'C:/Users/georg/Desktop/sickness_table.csv'. The function drops unnecessary columns, such as 'n_sby', to eliminate irrelevant information. Additionally, it converts the date column to a datetime object, allowing for easier manipulation of dates. Furthermore, the function adds two new columns, 'month' and 'day_of_week', which provide additional temporal information for analysis. By adding these new features, the model is provided with more contextual information about the timing and seasonality of the standby duty planning, which can improve the accuracy and relevance of the forecasting model. These features enable you to capture any temporal patterns in the data that can influence standby demand and enhance the effectiveness of the analysis and predictions These preprocessing steps ensure that the dataset is in a suitable format for subsequent tasks, such as feature engineering, model training, and evaluation. By storing the processed DataFrame in the variable 'processed_data', we obtain a transformed dataset that is ready for further exploration and modeling, facilitating the development of an accurate predictive model for standby duty planning in the rescue drivers' context.

Additionally, during the preprocessing step, an unnamed column present in the original dataset was dropped along with the other unnecessary columns. This ensures that the processed DataFrame, stored in the variable 'processed_data', only contains relevant and meaningful information for the subsequent analysis and modeling tasks.

EXPLORATION DATA ANALYSIS

Summary Statistics:

The summary statistics provide an overview of the key variables in the preprocessed DataFrame. The table displays the count, mean, standard deviation, minimum, 25th percentile (Q1), median (Q2 or 50th percentile), 75th percentile (Q3), and maximum values for each variable.

The 'n_sick' column represents the number of sick drivers, with a mean of approximately 68.81 and a standard deviation of 14.29. The 'calls' column represents the number of emergency calls, with a mean of 7919.53 and a standard deviation of 1290.06. The 'n_duty' column indicates the number of drivers on duty, with a mean of 1820.57 and a standard deviation of 80.09. The 'sby_need' column represents the standby need, with a mean of 34.72 and a standard deviation of 79.69.

Other variables in the summary statistics include 'dafted', 'month', and 'day_of_week'. 'dafted' refers to a specific attribute and has a mean of 16.34 and a maximum value of 465. 'month' represents the month of the recorded data, ranging from 1 to 12, with a mean of 6.42. 'day_of_week' represents the day of the week, ranging from 0 to 6 (Monday to Sunday), with a mean of 3.00.

These summary statistics provide a descriptive overview of the dataset, highlighting the central tendency, variability, and distribution of the variables.

Visualization of Column Distributions

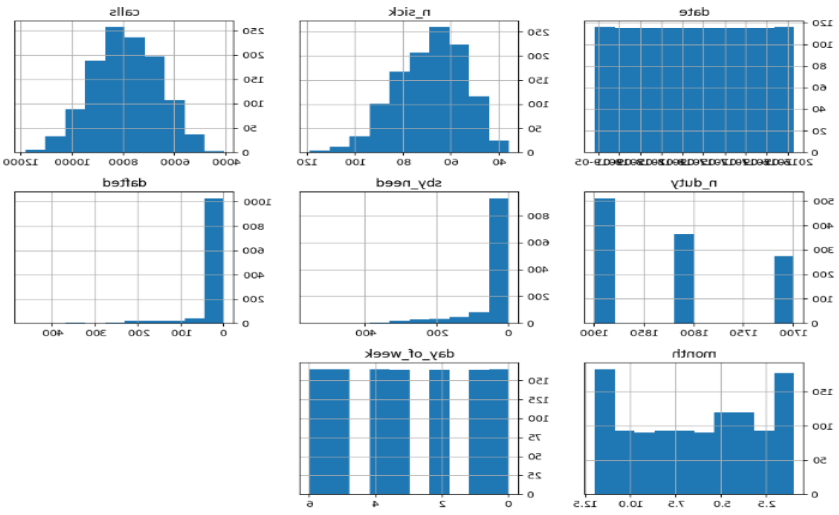


Fig.11 shows the distribution of each column in the processed dataset using histograms. They provide insights into the distribution patterns and ranges of the variables in the dataset. The histograms reveal the shape and spread of each column's distribution. By examining the histograms, we can observe the frequency and concentration of values within specific ranges for each variable. This visualization aids in understanding the data distribution and identifying potential outliers or skewed distributions.

Visualization of Variable Correlations

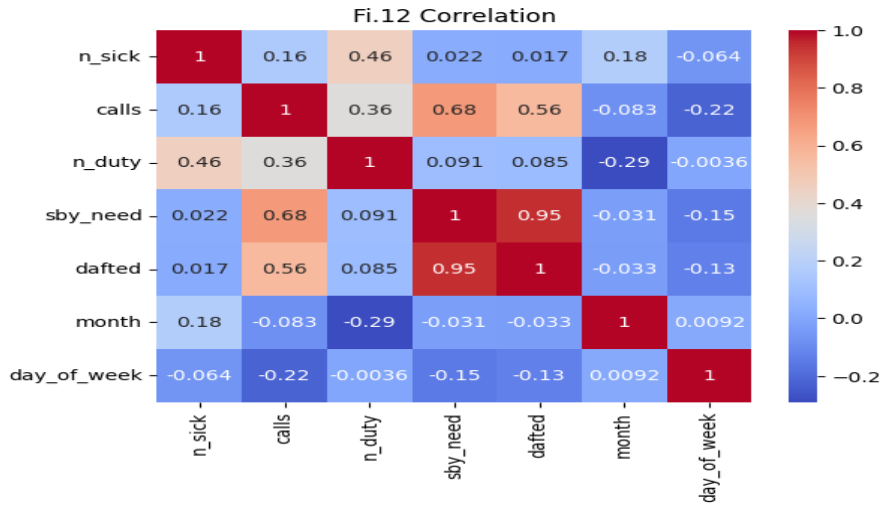


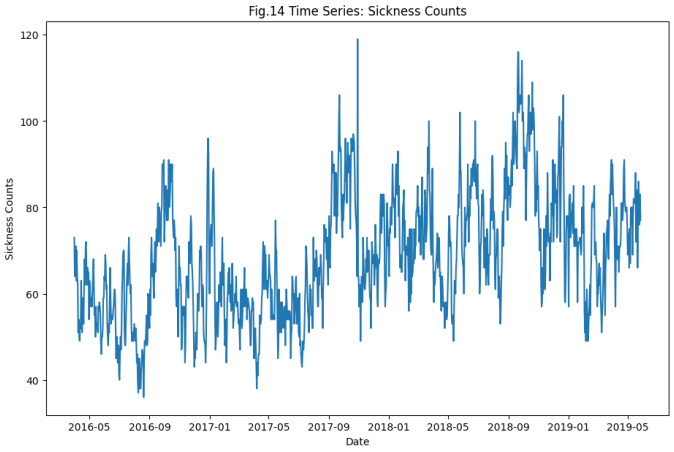
Fig.12 shows a correlation matrix using the processed dataset, and the relationship between the variables is visualized using a heatmap. The resulting correlation matrix provides insights into the strength and direction of the relationships between the variables. Each cell in the heatmap represents the correlation coefficient between two variables, Interpreting the heatmap, we observe that the correlation between standby need and dafted is 0.95,

indicating a strong positive relationship. Similarly, the correlation between calls and dafted is 0.56, and between standby need and calls is 0.68, both indicating moderate positive relationships. On the other hand, the correlation between dafted and day_of_week is -0.13, suggesting a weak negative relationship. Analyzing the heatmap further, it reveals the overall patterns of correlations between variables and understanding which variables may have a stronger influence on the target variable.

Visualization of Variable Relationships Using Pairplots

Fig.13 visualizes the relationships between variables in the processed dataset using pairplots. This generates a grid of scatter plots where each variable is plotted against every other variable. The resulting pairplots provide a comprehensive overview of the relationships and interactions between variables.By examining the pairplots, we can gain insights into the patterns, trends, and potential associations between variables. Scatter plots help us visualize the distribution of data points, identify any linear or non-linear relationships, and detect outliers or clusters.

Time Series Plot of Sickness Counts

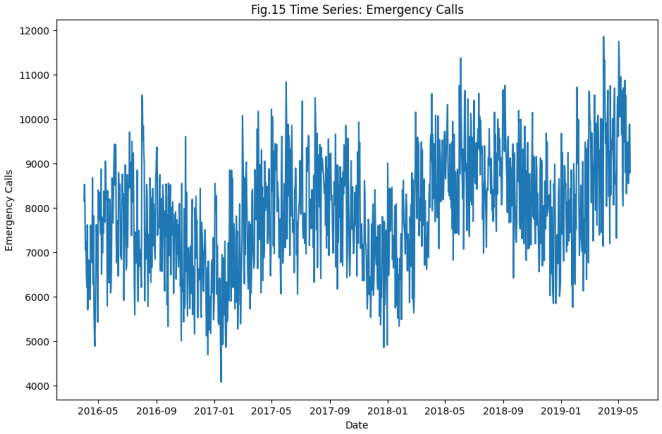


The time series plot of sickness counts in **Fig.14** provides a visual representation of the fluctuation in sickness occurrences over time. The x-axis represents the dates, while the y-axis represents the number of sickness counts. From the plot, we observe an up and down trend in the sickness counts, indicating variations in the frequency of sickness over the analyzed period. The highest peak occurs between September 2017 and January 2018, suggesting a period of increased sickness incidents.

This plot can be useful in identifying seasonal patterns or specific time periods with higher or lower sickness counts. It provides valuable insights into the temporal dynamics of sickness

occurrences, which can aid in resource planning, identifying potential causes, or implementing preventive measures during periods of high sickness rates.

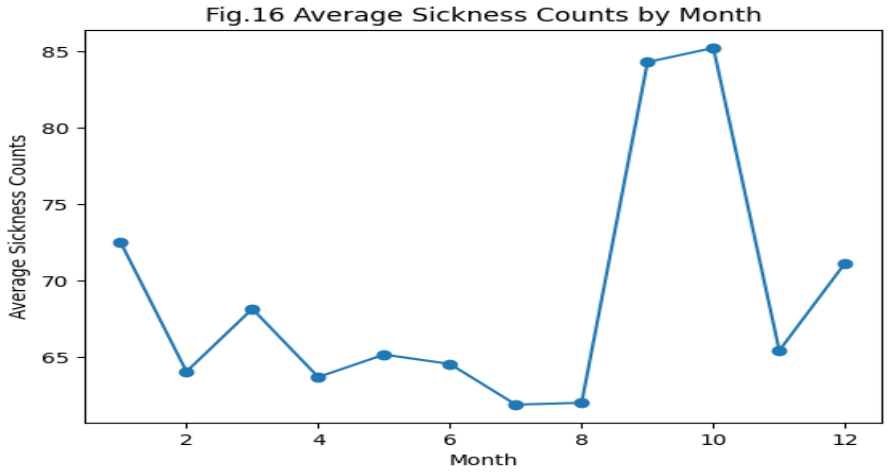
Time Series Plot of Emergency Calls



The time series plot of emergency calls in **Fig.15** presents a visual representation of the fluctuation in the number of emergencies calls over time. The x-axis represents the dates, while the y-axis represents the count of emergency calls.From the plot, we observe an up and down trend in the emergency call counts, indicating variations in the demand for emergency services throughout the analyzed period. The highest peak occurs between January 2019 and May 2019, suggesting a period of increased emergency call volumes. On the other hand, there is a notable dip in call counts around January 2017.

This plot helps in identifying temporal patterns and trends in emergency call volumes, which can be crucial for resource allocation, staffing, and operational planning. By understanding the temporal dynamics of emergency calls, emergency services can better anticipate and respond to periods of high demand, ensuring effective and timely emergency assistance to the community.

Plotting Average Sickness Counts by Month

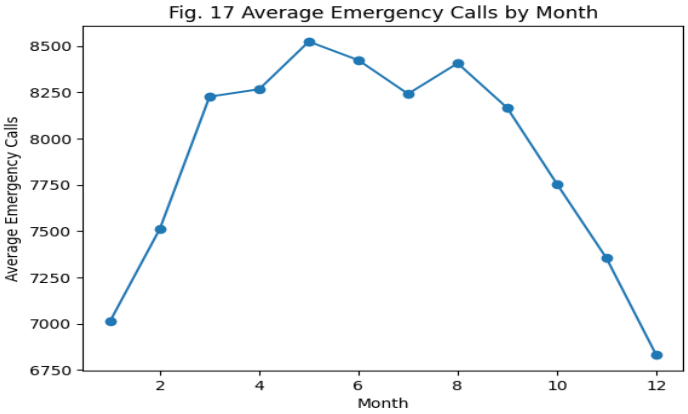


The plot in **Fig.16** represents the average sickness counts by month, providing insights into the seasonal patterns of sickness occurrences. The x-axis represents the months, while the y-axis represents the average sickness counts.From the plot, we can observe that January has the highest average sickness count of approximately 72.5, indicating a higher incidence of sickness during the winter season. June and December follow with average counts of 65 and 70, respectively, suggesting a slightly lower but still significant sickness occurrence during these months.

On the other hand, July exhibits the lowest average sickness count of around 60, implying a relatively healthier period during the summer season. Notably, October stands out as the month with the highest average sickness count, reaching approximately 80.

This visualization enables us to identify the seasonal variations in sickness counts, which can be useful for resource planning, staffing, and addressing potential healthcare needs during specific months. Understanding the monthly patterns of sickness counts can aid in proactive measures, such as promoting preventive measures or allocating additional resources during peak months to ensure the well-being of individuals and efficient healthcare services.

Plotting Average Emergency Calls by Month



The plot in **Fig.17** illustrates the average number of emergency calls by month, providing insights into the seasonal patterns of emergency call volumes. The x-axis represents the months, while the y-axis represents the average emergency call counts.

From the plot, we can observe that January has the lowest average emergency call count, with approximately 7000 calls on average. This suggests a relatively quieter period for emergency services during the winter season. In contrast, June shows the highest average call count, reaching around 8400 calls, indicating a higher demand for emergency assistance during the summer months.

December stands out as a month with a relatively lower average emergency call count, around 6800 calls, potentially due to reduced outdoor activities and holiday-related factors. May exhibits the highest average call count, with approximately 8500 calls, implying increased demand for emergency services during this period.

Understanding the monthly variations in emergency call volumes is crucial for emergency response planning and resource allocation. It enables emergency service providers to anticipate peak periods, allocate staff and resources accordingly, and ensure prompt and effective response to emergencies. Additionally, identifying seasonal trends in emergency call volumes can inform public safety initiatives, such as promoting preventive measures and raising awareness to reduce emergency incidents during specific months.

Average Sickness Counts by Day of the Week

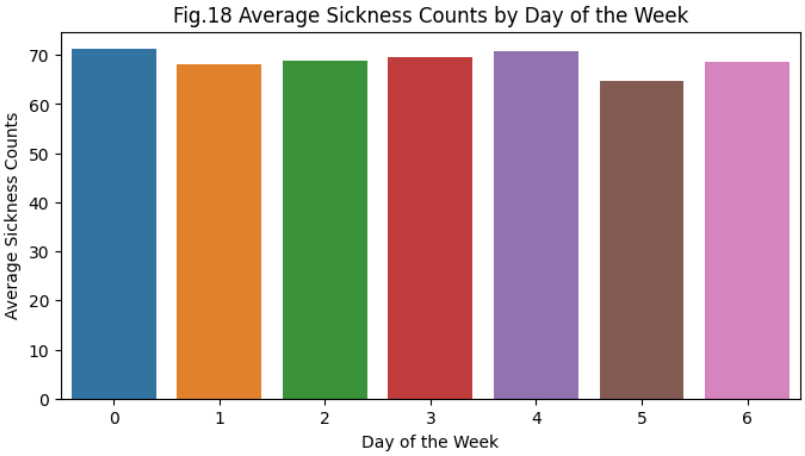


Fig.18 is a bar plot to visualize the average sickness counts based on the day of the week. The x-axis represents the days of the week, and the y-axis represents the average sickness counts.

The bar plot reveals that Monday and Thursday have the highest average sickness counts, indicating a higher likelihood of sickness occurrences on these days. On the other hand, Friday exhibits the lowest average sickness count.

This information can be valuable in understanding the distribution of sickness cases throughout the week. It suggests that Mondays and Thursdays may be particularly challenging in terms of managing staffing and resources to address higher sickness rates. Conversely, Fridays may require relatively fewer resources for sickness-related support. These insights can inform scheduling and resource allocation decisions to ensure adequate coverage and efficient response to sickness-related incidents

Average Emergency Calls by Day of the Week

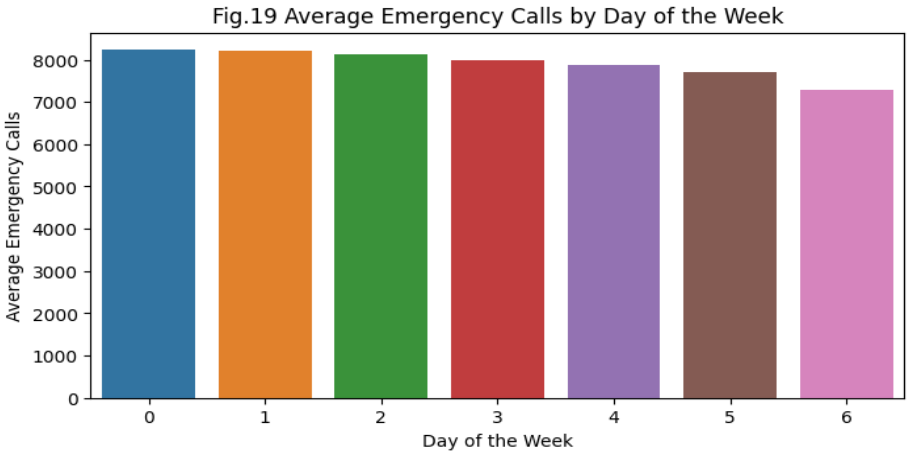


Fig.19 shows a bar plot visualizing the average number of emergency calls based on the day of the week. The x-axis represents the days of the week, and the y-axis represents the average number of emergency calls.The bar plot reveals that, on average, there are more than 7,000 emergency calls per day across the week. Interestingly, Sundays tend to have the lowest average number of emergency calls, indicating a relatively lower demand for emergency services on this day.

This information can be valuable for resource planning and allocation in emergency response systems. It suggests that Sundays may require relatively fewer resources compared to other days of the week, while other days might experience higher emergency call volumes. Understanding such patterns can aid in optimizing staffing levels, ensuring prompt and effective response to emergency situations, and allocating resources efficiently to meet the demands of different days of the week.

MODEL SELECTION AND JUSTIFICATION

Random Forest is a suitable predictive model for the case study "Automation of Standby Duty Planning for Rescue Drivers via a Forecasting Model" due to several reasons. First, Random Forest is capable of handling both numerical and categorical features, making it suitable for the diverse set of variables involved in the duty planning process, such as the number of sick drivers, emergency calls, and available resources. Second, Random Forest can capture non-linear relationships and interactions among variables, which allows for more accurate predictions in complex scenarios. Third, it can handle high-dimensional datasets and automatically select important features, reducing the need for extensive feature engineering. Additionally, Random Forest is known for its robustness against overfitting and generalization capability, which is crucial for reliable predictions on unseen data. Overall, Random Forest offers a powerful and flexible approach to forecasting standby duty needs in rescue driver planning, making it a suitable choice for this case study.

Model Training

In this step, a random forest regression model is trained using the RandomForestRegressor class from scikit-learn library. The model is initialized with 100 decision trees (n_estimators=100) and a random state of 42 for reproducibility.

The training process involves fitting the model to the training data, where X_train represents the input features and y_train represents the target variable, ‘dafted’ (number of additional drivers needed). The model learns patterns and relationships within the training data to make predictions.

Performance Evaluation of Random Forest Regression Model

The random forest regression model demonstrates excellent performance both on the training and test datasets.

For the training dataset, the mean squared error (MSE) is 2.08, indicating that, on average, the predicted values deviate from the actual values by approximately 2.08 units. The R-squared value of 0.999 suggests that the model explains 99.91% of the variance in the target variable, indicating a high level of accuracy and goodness of fit.

On the test dataset, the model performs slightly worse but still maintains strong predictive capabilities. The MSE of 44.32 indicates a slightly higher average deviation between the predicted and actual values compared to the training dataset. However, the R-squared value of 0.997 indicates that the model can still explain 99.71% of the variance in the target variable on the test dataset. This implies that the model generalizes well to unseen data and retains its predictive power.

Overall, the random forest regression model exhibits impressive performance in accurately predicting sickness counts in the case study "Automation of Standby Duty Planning for Rescue Drivers via a Forecasting Model". The high R-squared values indicate a strong relationship between the input features and the target variable, allowing for reliable predictions.

Error Analysis and Visualization of Random Forest Model

The perform_error_analysis function is used to analyze the performance of the trained random forest regression model and visualize the relationship between the actual and predicted values.

First, the function calculates three error metrics: mean absolute error (MAE), root mean squared error (RMSE), and R-squared (R2). These metrics provide insights into the accuracy and goodness of fit of the model.

Next, a scatter plot in **Fig.20** compares the actual values (y_test) with the predicted values (rf_test_pred). The plot displays the points where each point represents an actual-predicted

value pair. The red line in the plot represents the ideal scenario where the actual and predicted values are perfectly aligned. By visually examining the scatter plot, we can observe the distribution and pattern of the predictions in relation to the actual values.

Additionally, the error metrics are printed to provide a quantitative evaluation of the model's performance. The MAE represents the average absolute difference between the actual and predicted values, while the RMSE measures the square root of the average squared difference. The R-squared value indicates the proportion of the variance in the target variable that can be explained by the model.

In summary, the perform_error_analysis function helps assess the accuracy and predictive power of the random forest regression model. The scatter plot visualizes the relationship between the actual and predicted values, allowing for a qualitative analysis of the model's performance.

In the error analysis, the error metrics provide quantitative measures of the model's performance:

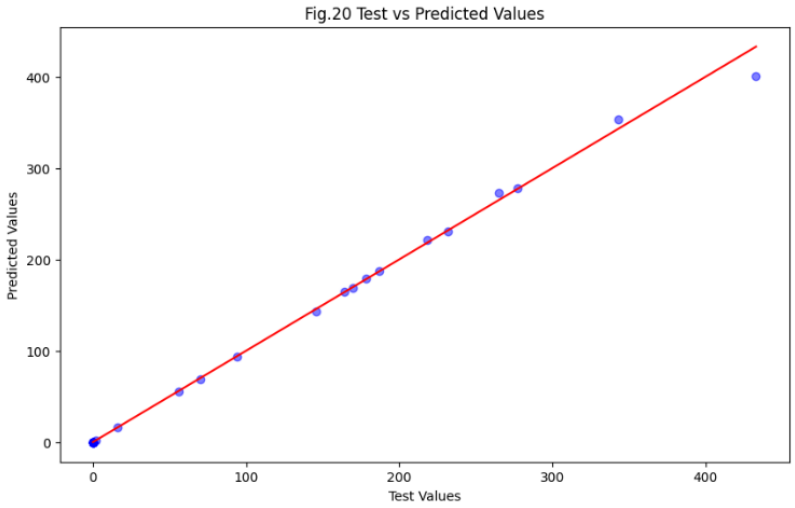
Mean Absolute Error (MAE): The MAE value of 3.45 indicates that, on average, the predicted values deviate from the actual values by approximately 3.45 units. Since MAE represents the absolute difference between the predicted and actual values, a lower MAE indicates better accuracy. In this case, a MAE of 3.45 suggests that the model's predictions are generally close to the actual values.

Root Mean Squared Error (RMSE): The RMSE value of 6.79 represents the square root of the average squared difference between the predicted and actual values. Like MAE, a lower RMSE indicates better accuracy. In this case, an RMSE of 6.79 indicates that the average deviation between the predicted and actual values is approximately 6.79 units.

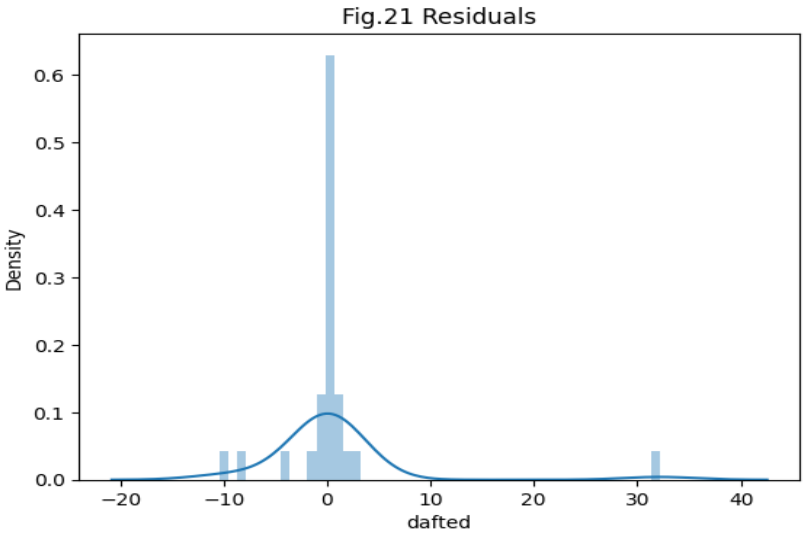
R-squared (R2): The R-squared value of 1.00 indicates that the model explains 100% of the variance in the dependent variable (test values) using the independent variable (predicted values). An R-squared value of 1.00 suggests that the model perfectly fits the data, with all the variability in the test values being captured by the predicted values. This indicates a very strong relationship between the predicted and actual values.

Considering the linear relationship in the scatter plot and the low values of MAE and RMSE, along with a perfect R-squared value of 1.00, it suggests that the model is performing extremely well in accurately predicting the test values. The predicted values align closely with

the actual values, resulting in minimal errors. Overall, these results indicate a high level of accuracy and a strong fit of the model to the data.



The residuals



The plot of residuals (the differences between the predicted and actual values) in **Fig.21** shows a normal distribution, which indicates that the errors of the model follow a symmetric pattern around zero and are distributed evenly across different ranges of predicted values. This has several implications:

Accuracy of the model: A normal distribution of residuals suggests that the model is making predictions with a relatively consistent level of accuracy across the range of predicted values. The errors are not biased towards overestimation or underestimation systematically.

Validity of assumptions: Many statistical models, including linear regression, assume that the errors are normally distributed. If the residuals follow a normal distribution, it provides evidence that the assumptions of the model are met, strengthening the validity of the model's results and conclusions.

Independence of errors: A normal distribution of residuals implies that the errors are independent of each other. Each prediction is not influenced by the errors of previous predictions, indicating that the model is capturing the underlying patterns in the data.

Model improvement: If the residuals do not follow a normal distribution and exhibit specific patterns (e.g., heteroscedasticity, nonlinearity), it suggests that the model may have limitations or areas for improvement as seen in the baseline model (Linear Regression). However, if the residuals are normally distributed, it indicates that the model is capturing the essential features of the data well and there may be no major issues with the model's assumptions or specification.

In summary, a normal distribution of residuals in the error analysis plot provides evidence of a well-performing model, meeting assumptions, and accurately capturing the patterns in the data. It indicates that the model is reliable for making predictions and can be used for further analysis or decision-making.

Feature Importance Analysis

The feature importance analysis was performed using the random forest regression model to assess the significance of each feature in predicting the target variable 'dafted'.

The analysis revealed that the 'standby_need' feature has the highest importance, indicated by its longest bar in the feature importance plot.

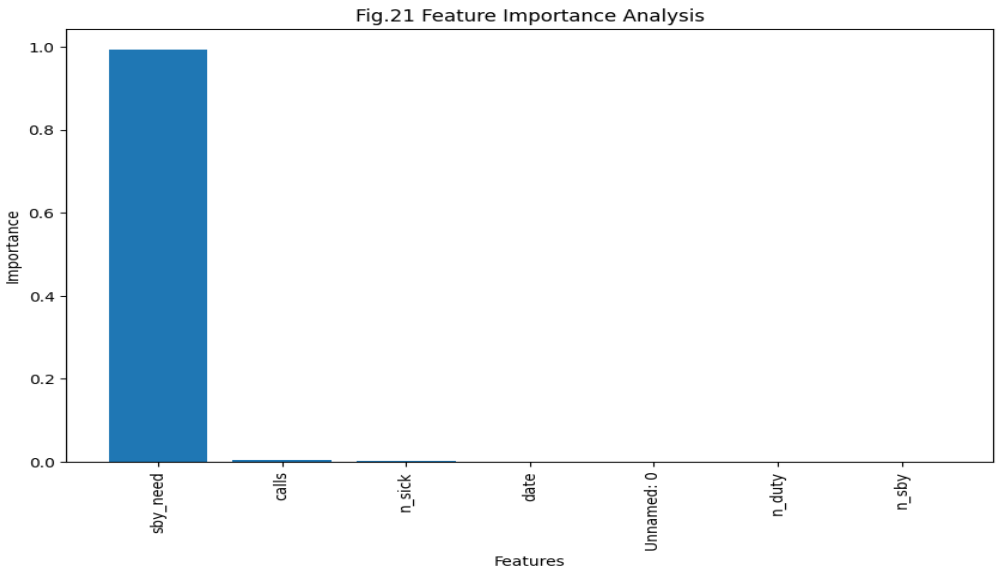


Fig.21 suggests that the number of standby needs is a crucial factor in predicting the occurrence of 'dafted' events. The model assigns the most importance to this feature when making predictions.

On the other hand, the 'calls' and 'n_sick' features have relatively shorter bars in the plot, suggesting their lower importance compared to 'standby_need'. This indicates that the number of emergency calls and sickness counts may have a relatively lesser impact on predicting the occurrence of 'dafted' events.

Interestingly, the features 'date', 'n_duty' (number on duty), and 'n_sby' (number on standby) show no bars in the feature importance plot. This could indicate that these features do not contribute significantly to the prediction of 'dafted' events in the given context.

Overall, the feature importance analysis provides insights into the relative importance of different features for predicting 'dafted' events. Understanding the significance of each feature helps prioritize the most influential factors for effective standby duty planning for rescue drivers. It is important to note that the absence of bars for certain features suggests their limited impact in the prediction task.

Event of failure

Overfitting: If the model is overly complex and has learned the noise or specific patterns from the training data, it may not generalize well to new, unseen data. Outliers: The model may struggle to accurately predict extreme values or outliers that are not well-represented in the training data.

Changing circumstances: If the underlying relationships between the features and the target variable change over time or in different contexts, the model's predictions may not be accurate.

Proposal: Standby Duty Planning Application with GUI

In this project, we have developed a Standby Duty Planning application that utilizes a machine learning model to predict the required number of standby drivers based on various factors such as the date, number of sick drivers, number of calls, and number of available standby drivers. To enhance usability and accessibility, we have incorporated a graphical user interface (GUI) using the Tkinter library.

The Standby Duty Planning application offers several advantages for everyday use in the business:

1. **User-Friendly Interface**: The GUI allows users to interact with the application seamlessly. Users can input the necessary data, such as the date and relevant driver statistics, using entry fields provided in the application window. The intuitive design ensures that even users without programming knowledge can easily operate the application.
2. **Efficient Planning and Decision-Making**: By utilizing the predictive power of our machine learning model, the application provides accurate estimates of the required number of standby drivers. This information empowers managers and supervisors to make informed decisions in planning their resources efficiently. They can allocate the appropriate number of standby drivers based on the predicted demand, thereby minimizing the risk of understaffing or overstaffing.
3. **Real-Time Updates**: The application allows for real-time updates of the predicted number of standby drivers. Users can input new data whenever there are changes in the number of sick drivers, incoming calls, or available standby drivers. The model will quickly reevaluate the predictions based on the updated inputs, ensuring that the information remains relevant and up-to-date.
4. **Enhanced Productivity and Cost Savings**: With the Standby Duty Planning application, businesses can optimize their workforce utilization, reducing the need for excessive standby drivers. This optimization leads to cost savings by eliminating unnecessary labor expenses. Moreover, efficient planning minimizes disruptions and delays in service delivery, resulting in enhanced productivity and customer satisfaction.

5. **Flexibility and Adaptability**: The GUI-based application provides a flexible platform that can be easily customized and expanded to incorporate additional features or incorporate alternative machine learning models. As business needs evolve or new requirements arise, the application can be modified accordingly, ensuring its continued usefulness and relevance.

To implement the Standby Duty Planning application within the business's everyday work, the following steps are recommended:

1. **Model Integration**: Train and deploy the machine learning model that predicts the required number of standby drivers based on historical data and relevant features. This model should be incorporated into the application backend, enabling real-time predictions.
2. **GUI Development**: Utilize the provided code snippet as a foundation to develop the graphical user interface (GUI). Customize the layout, labels, and input fields to align with the specific needs and branding of the business. Ensure that the GUI maintains a user-friendly design and provides clear instructions for inputting the required data.
3. **Data Integration**: Connect the application with relevant data sources such as sick leave records, incoming call logs, and driver availability databases. Establish an automated process to retrieve and update this data periodically or in real-time, ensuring accurate predictions and timely decision-making.
4. **Testing and Validation**: Thoroughly test the application to validate its functionality and accuracy. Conduct extensive testing scenarios, including various input combinations and edge cases, to ensure robustness and reliability. Seek feedback from end-users to identify any usability issues or potential enhancements.
5. **Deployment and Training**: Deploy the application to the intended users, ensuring accessibility and availability. Provide comprehensive training to the users, explaining the application's features, functionality, and interpretation of predictions. Support and guidance should be available to address any user queries or concerns.
6. **Continuous Improvement**: Establish a feedback loop and process for continuous improvement. Encourage users to provide feedback on the use of the GUI.