# Explainability in Regression: Prediction of the number of fire incidents in Dublin

Donghyeok Lee
*Human Centered Artificial Intelligence*
*Master School of Computing*
*Technological University Dublin*

**Abstract.** Increased urbanization and industrial activity are leading to the frequency of fire incidents, which causes significant damage to life and property and increases social and economic costs. This study presents a comparative analysis of seven machine learning models for predicting fire incidents in Dublin. Utilizing various regression techniques, including Prophet, ARIMA, Random Forest Regression, and others, the research evaluates the performance of each model based on metrics like Mean Squared Error (MSE), Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE) and R-Squared scores. The Prophet model emerged as the most effective with a high R-Squared score of 0.92, enabling accurate predictions of fire incidents in Dublin for the next 2 years. Furthermore, the study underscores the importance of Explainable AI (XAI) in enhancing the transparency and understanding of these technologies. By focusing on data transparency, model explainability, and effective visualizations, the study aims to make AI models more comprehensible and trustworthy, especially in domains with significant social impact. This approach not only advances the field of AI but also provides valuable insights for urban planning and emergency response strategies in Dublin.

Keywords: Fire incidents, Prophet, ARIMA, Simple Linear Regression, Polynomial Regression, Support Vector Regression (SVR), Decision Tree Regression, Random Forest Regression, Machine Learning Regression, Explainable AI (XAI), Black box, Gray Box, White box

## 1. Research Questions

- Which regression model has the best performance for predicting time series data?
- What time period exhibits the most distinct seasonality for prediction?
- How can the AI project be made explainable for users?

## 2. Introduction

Due to the rise in urbanization and industrial activities, the frequency of fire incidents in Dublin has increased. It can also bring destructive disasters that lead to fatalities and injuries as well as great losses of property and damage of environmental elements as well as elevated social and economic costs[1]. This study focused on predicting the number of fire incidents in Dublin and the explainability of regression model. The aim of the study was to identify the most accurate machine learning model for prediction and improving the performance of the model, which is crucial to public safety, the future of the Dublin Fire Brigade, and the city's planning. This study explored the predictability of the number of fire incidents by analyzing fire incident data in Dublin from 2016 to 2022.

The regression models which were used in this study are Prophet, ARIMA, simple linear regression, polynomial regression, support vector regression, decision tree regression, and random forest regression. The performances of each model were compared through Mean Squared Error (MSE), Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE) and R-Squared scores. This comparative analysis showed that before tuning the model, the Prophet model has the best predictive performance among these models. Furthermore, by tuning the hyperparameters, the Prophet's performance was improved as well.

The results of the study showed potential for use in urban planning and management. regression models can provide critical information for developing city risk management and emergency response strategies.

Additionally, in this study, the concept of Explainable AI (XAI) played a crucial role in enhancing the transparency and understandability of regression models. In general, XAI can be perceived as the topic or research field concerned with developing approaches to explain and make artificial systems understandable to human stakeholders[2]. In the field of artificial intelligence (AI), explainable AI (XAI) is a method and technique that makes the outcomes of AI decisions more understandable to humans. It involves making AI models that can explain or justify their behavior and decisions in a transparent and comprehensible manner. The goal of explainable AI (XAI) is to connect the gap between human decision making and AI processes, ensuring that AI decisions are transparent, understandable, and trustable, particularly in critical applications where understanding AI reasoning is essential.

In this study, by integrating explainable AI (XAI), this study aimed to not only identify the best regression model but also provide clear, understandable reasons behind these predictions. The use of various models such as Prophet, ARIMA, and other regression models, combined with XAI, ensured that predictions are not only just accurate but also interpretable, making this study an exemplary case of how AI can be made transparent and accountable, especially in domains having significant social impact.

This paper is structured as follows: Section 3 reviews related works on incident prediction and explainable AI (XAI). Section 4 is details on methodology that are used in this study, including a comprehensive explanation of the data and regression models that are used in this study. The results of the experiments, along with their discussion, are presented in Section 5. Section 6 concludes the study and section 7 outlines future research directions. Lastly, Section 8 explains contributions of this study.

## 3. Related Work

For this study, a literature review was conducted, focusing on the previous general concept of incident prediction since there is no prior research specifically on prediction of the number of fire incidents. Additionally, literature review on Explainable AI (XAI) was conducted. This section is divided into two parts: *3.1. Prediction of incident number*, and *3.2. Explainable AI (XAI)*, to provide a comprehensive understanding.

### *3.1. Prediction of incident number*

The "Using Machine Learning to Predict the Impact of Incidents on the M50 Motorway in Ireland"[3] investigated traffic incident duration prediction on the M50 using machine learning techniques like SVM and ANN. The research employed various machine learning methods, with Support Vector Machines (SVM) excelling in short to medium incident predictions, while Artificial Neural Networks (ANN) showed promise for longer incidents. This methodical approach emphasized the role of tailored algorithms in specific prediction scenarios. However broader range of models for predicting the incidents like Prophet and ARIMA was not employed, which are more suited for complex time series data. This research can be needed to employ a more diversified approach, potentially offering greater adaptability and precision in unpredictable scenarios. Competitively, the current study on predicting the number of fire incidents in Dublin used varied models and demonstrates a methodology for handling complex, variable-rich datasets.

Additionally, the "Using Linear Regression to Forecast Future Trends in Crime of Bangladesh"[4] served as a valuable reference in the context of predictive analytics using linear models. This study focused on employing linear regression to predict various crime trends in Bangladesh, offering insights for police and law enforcement agencies. Using data from the Bangladesh police website, the study applied linear regression to forecast crimes such as robbery, murder, and theft. This approach highlighted the potential of data mining in crime analysis, especially in areas with high crime rates. While effective in its context, the study's reliance on a single linear regression model is a notable limitation, especially when dealing with complex time series data. In contrast, the current study on predicting the number of fire incidents in Dublin employed a more varied methodology, including models like Prophet, ARIMA, linear regression and so on. This diversity in modeling not only allowed for capturing non-linear patterns but also emphasizes the need for a comprehensive analytical approach in time series forecasting.

### *3.2. Explainable AI (XAI)*

As seen in Figure 1[5], explainable AI (XAI) is currently a trending part in the AI field, aiming to make AI decision-making processes transparent, understandable, and accountable. The trend is towards developing AI systems that provide insights into their reasoning, ensuring that AI-driven decisions can be trusted and fair, especially in critical industries such as healthcare, finance, autonomous driving and so on. There is an emerging need for understanding how such decisions are furnished by AI methods[6]. “Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI”[5] is a comprehensive review of Explainable Artificial Intelligence (XAI), focusing on its concepts, taxonomies, opportunities, and challenges toward responsible AI. It covers topics such as the need for explainability in machine learning models, different levels of transparency, post-hoc explainability techniques, transparent machine learning models like linear/logistic regression, decision trees, K-Nearest Neighbors, and others. The research also addressed the intersection of XAI with AI principles such as fairness, privacy, and data fusion, ultimately aiming to establish guidelines for implementing responsible AI.

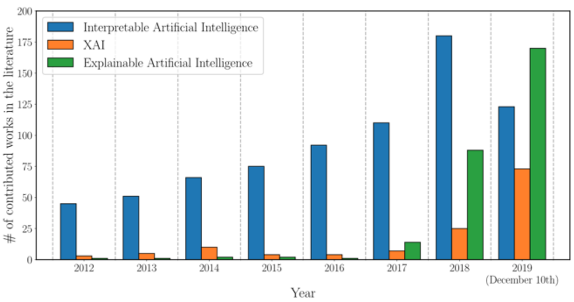



Figure 1. Evolution of the number of total publications whose title, abstract, and/or keywords refer to the field of XAI during the last years. Data retrieved from Scopus® (December 10th, 2019) by using the search terms indicated in the legend when querying this database. It is interesting to note the latent need for interpretable AI models over time (which conforms to intuition, as interpretability is a requirement in many scenarios), yet it has not been until 2017 when the interest in techniques to explain AI models has permeated throughout the research community.

1) Black Box Models

As seen in Figure 2[7], Black Box Models refer to systems where the internal workings are not visible or understandable to the user. In machine learning, this often applies to complex models like deep neural networks, where it's difficult to interpret how the model arrives at a particular decision or prediction.

2) Gray Box Models

As seen in Figure 2[7], Gray Box Models refer to systems that is between black box and white box models. These systems provide some level of insight into their internal workings or decision-making process, but not as clearly or completely as white box models. Gray box models strike a balance by providing a degree of interpretability while managing complex data that might be too intricate for a fully transparent white box model.

3) White Box Models

As seen in Figure 2[7], White box models are transparent, and their internal logic is understandable. Some of Examples are decision trees or linear regression, where the decision-making process is clear and can be easily followed.

The primary goal of XAI is to obtain human interpretable models, especially in critical industries such as healthcare, finance, autonomous driving and so on, since domain specialists need help solving problems more effectively, but they also want to be provided with meaningful output to understand and trust those solutions.[7]

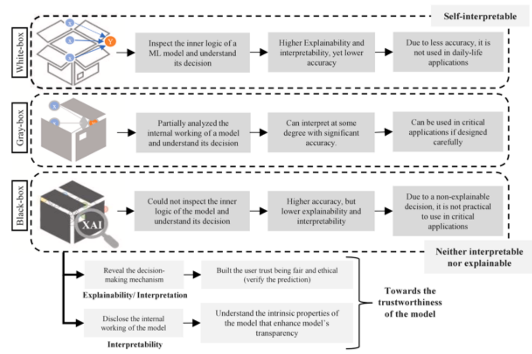



Figure 2. A comparison of white-box, gray-box, and black-box models. On the one hand, white-box models are interpretable by design thus making their outputs easier to understand but less accurate. In addition, gray-box models yield a good interpretability-accuracy tradeoff. On the other hand, black-box models are more accurate but less interpretable. More complex XAI techniques are required for creating trustworthy models.

## 4. Methodology

As seen in Figure 3, this study followed machine learning pipeline to analyze fire incident data in Dublin and conducted comparative analysis with seven models. The models used were Prophet, ARIMA, simple linear regression, polynomial regression, support vector regression, decision tree regression, and random forest regression. The model's performance was evaluated using metrics such as Mean Squared Error (MSE), Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE) and R-Squared scores. The rest of this section will provide a detailed explanation of how each step was applied to the study.

### *4.1. Problem formulation*

In recent years, Dublin has experienced a significant number of fire incidents, leading to serious risks to

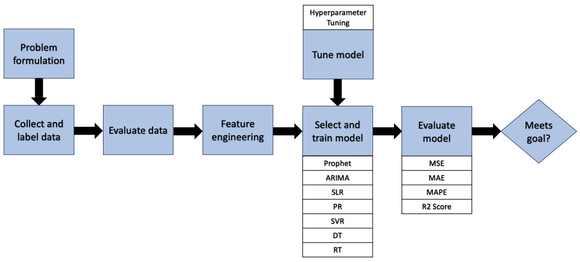



Figure 3. Machine Learning Pipeline on this study

public safety and property. The increasing urban density and complexities in the Dublin have made fire incidents more unpredictable and hazardous. Addressing this challenge requires a proactive approach to fire management and emergency response. The machine learning problem for this study is predicting the number of fire incidents in Dublin. The goal is to identify the most accurate and reliable regression model using historical fire incident data in Dublin. This identifies key factors that contribute to the frequency of fire incident, providing actionable insights for Dublin Fire Brigade and Dublin city.

### *4.2. Collect and label data*

The data for this study has been sourced from the Smart Dublin (https://data.smartdublin.ie/dataset/fire-brigade-and-ambulance)[8], which provides a comprehensive dataset of fire annual incident activity logs. This open data initiative is part of Dublin's commitment to transparency and public safety. The dataset encompasses detailed logs of annual incident activities recorded by the Dublin Fire Brigade (DF) services. It includes critical data features such as:

- ID: A unique identifier for each incident.
- Date: The date when the incident occurred.
- Station Name: The name of the fire station that responded to the incident.
- Description: A brief description of the incident.
- TOC: Time when the call is received in the control center.
- ORD: Time when the vehicle is mobilized to the incident.
- MOB: Time when the vehicle starts moving towards the incident.
- IA: Time when the vehicle arrives at the incident site.
- LS: Time when the ambulance is leaving the scene for the hospital.
- AH: Time when the ambulance arrives at the hospital.
- MAV: Time when the vehicle is mobile and available (heading back to the station).
- CD: Time when the vehicle is closing down (back at the station).

### *4.3. Evaluate data*

The dataset comprises 85,813 entries spread across 12 features, with each entry uniquely identified by an integer ID. Key features include 'Date', 'Station Name', 'Description', 'TOC', 'ORD', 'MOB', 'IA', 'LS', 'AH', 'MAV', and 'CD'. Notably, the dataset is a mix of integer, object, and float data types. While most columns are complete with non-null values, 'MOB' and 'IA' have some missing entries, and 'LS' and 'AH' are entirely null, indicating a lack of data in these fields. Through exploring data, the number of fire incidents in Dublin by fire station is like Figure 4.

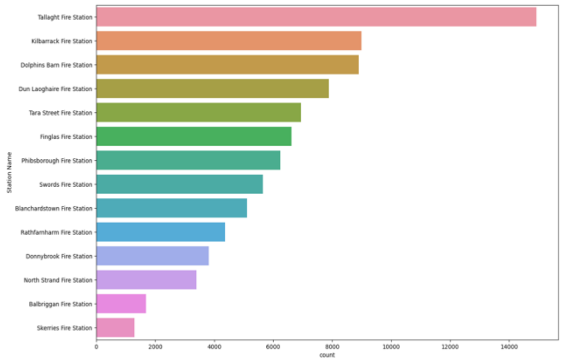



Figure 4. The number of fire incidents in Dublin by fire station.

### *4.4. Feature engineering*

1) Handling Missing Values and Dataset Column Restructuring
   For this study, I retained the 'Date' column and removed all other columns that are not relevant to the analysis. This step simplifies the dataset and focuses on the most critical information required for our predictive model.

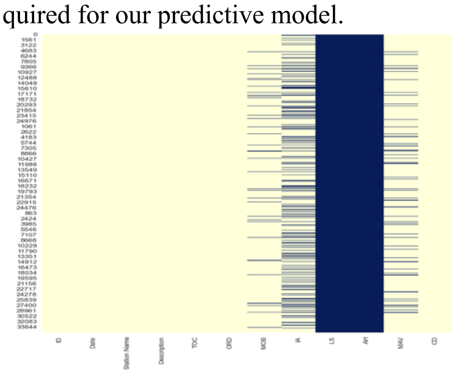


Figure 5. Missing values (Navy) in dataset.

2) Date Column Formatting

To convert the 'Date' column into a datetime object. This conversion is crucial for time series analysis as it enables the model to accurately interpret the date information.

```
dublin_df_processed.Date = pd.to_datetime(dublin_df_processed.Date, format='%d/%m/%Y')
dublin_df_processed.index = pd.DatetimeIndex(dublin_df_processed.Date)
```

Figure 6. Code for converting the 'Date' into datetime object.

3) Extracting Information from the 'Date' Field

As seen in Figure 7, 'Date' field was harnessed to extract valuable time-related features such as the day of the week, month, and year. These new features were pivotal in our analysis, providing deeper insights into temporal patterns and trends in fire incidents.

| Date | Count |
|---|---|
| 2016-01-01 | 31 |
| 2016-01-02 | 32 |
| 2016-01-03 | 28 |
| 2016-01-04 | 29 |
| 2016-01-05 | 21 |
| ... | ... |

↓

| Date | dayofweek | quarter | month | year | dayofyear | dayofmonth | weekofyear | weekday | season | Count |
|---|---|---|---|---|---|---|---|---|---|---|
| 2016-01-01 | 4 | 1 | 1 | 2016 | 1 | 1 | 53 | Friday | Winter | 31 |
| 2016-01-02 | 5 | 1 | 1 | 2016 | 2 | 2 | 53 | Saturday | Winter | 32 |
| 2016-01-03 | 6 | 1 | 1 | 2016 | 3 | 3 | 53 | Sunday | Winter | 28 |
| 2016-01-04 | 0 | 1 | 1 | 2016 | 4 | 4 | 1 | Monday | Winter | 29 |
| 2016-01-05 | 1 | 1 | 1 | 2016 | 5 | 5 | 1 | Tuesday | Winter | 21 |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |

Figure 7. Extracting information from the 'Date' Field.

Convert 'Date' fields to datetime objects to accurately interpret date information that is important for time series analysis. Analyze temporal patterns and trends in fire events by extracting time-related characteristics such as day, month, and year from the 'Date' field.

4) Outlier Handling Methods

Yearly outliers can be identified as Figure 8, and while it is possible to address them, due to the nature of time series data, values that appear as outliers might be part of the pattern. In fact, in this study, removing outliers resulted in a decrease in model performance. This indicates that values identified as outliers are also part of the time series pattern.

### *4.5. Select and train model*

When analyzing a dataset that records daily fire counts over a period of seven years, it is crucial to consider the time series characteristics of the data. In such

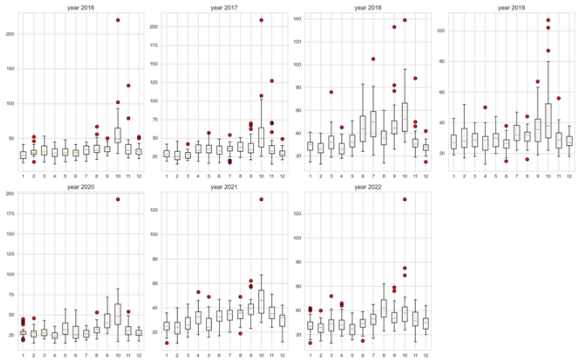



Figure 8. Yearly outlier.

data, there may be patterns, seasonality, and trends that occur over time and can influence future values. Therefore, randomly splitting the data into training and test sets, as typically done in standard regression analysis, is not appropriate. If the data exhibits patterns related to time, then time series analysis should be considered. Otherwise, if there is no such temporal relationship, general regression analysis should be contemplated. The choice of the model should be determined based on the characteristics of the data and the study's objectives. To select an appropriate regression model, it is essential to first understand the distinction between "time series analysis" and "general regression analysis."

1) Analysis Type

i) Time Series Analysis

Time series analysis involves analyzing data points measured in sequential order over time. A time series data set must be determined to be stationary and non-random to allow prediction[9]. It assumes that the data points are arranged in time order and that previous observations can influence future values. Time series data exhibit patterns over time, where data from earlier time points can influence data at later time points. Many time series data show seasonality (e.g., annual, monthly, weekly patterns) or trends (e.g., tendencies to increase or decrease over time). Time Series Specific Models are Prophet and ARIMA. As seen in Figure 9, in this dataset, Seasonality can be observed in time period of monthly and quarterly.

ii) General Regression Analysis

General regression analysis models and predicts the relationship between one or more independent variables and a dependent variable. It attempts to represent the relationship between data points with a mathematical function. Regression analysis identifies relationships between variables, analyzing how a change in one variable impact another. There are various forms

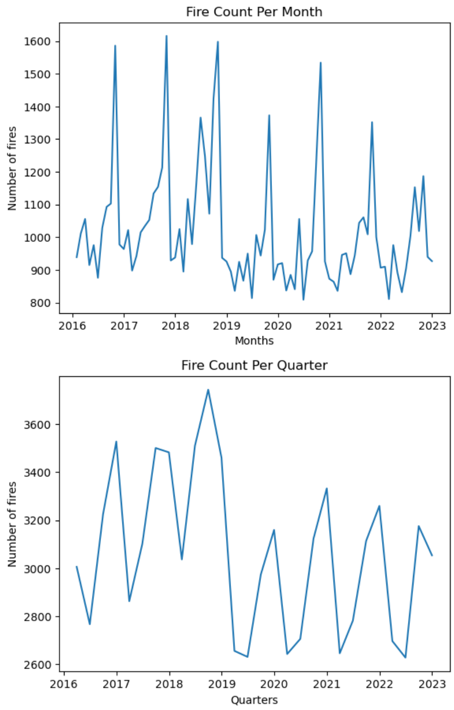



Figure 9. Seasonality in monthly and quarterly dataset

of regression models, including linear regression, polynomial regression, and logistic regression, chosen based on the complexity of the data and the purpose of the analysis. In general regression analysis, it is assumed that each observation is independent of others.

Time series emphasize dependency over time, whereas in general regression analysis, each data point is considered independent. Time series data must be ordered in time, whereas general regression analysis does not have this constraint. Time series analysis is mainly used for understanding and predicting patterns in data over time, while general regression analysis focuses on identifying and explaining relationships between variables.

1) Proposed Models

i) Prophet: Developed by Facebook, Prophet is designed for forecasting time series data. It works well with daily observations that display patterns on different time scales such as holidays, weekly, and yearly seasonality. It's robust to missing data and shifts in the trend, and typically handles outliers well. Prophet generally outperforms ARIMA[10].

ii) ARIMA (AutoRegressive Integrated Moving Average): This is a popular statistical method for time series forecasting. The ARIMA model includes autoregressive (AR) model, moving average (MA) model, and seasonal autoregressive integrated moving average model[11].

iii) Simple Linear Regression: This is a basic form of regression analysis where the relationship between two variables is modeled with a linear equation. It's used when we want to predict the value of a variable based on the value of another variable.

iv) Polynomial Regression: An extension of linear regression where the relationship between the independent variable x and the dependent variable y is modeled as an nth degree polynomial. It's useful for describing curvilinear relationships.

v) Support Vector Regression (SVR): Derived from Support Vector Machines (SVM), SVR is used for regression problems. It attempts to fit the best line within a threshold value, where the best line is the line that has the maximum number of points.

vi) Decision Tree Regression: This method uses a decision tree to model the relationship between the features and the target variable. The tree splits the data into subsets based on the value of the input features, with the goal of reducing variance within each subset.

vii) Random Forest Regression: An ensemble learning method that operates by constructing multiple decision trees during training and outputting the mean prediction of the individual trees. It's particularly effective for dealing with non-linear and complex data due to its ability to capture interactions among features.

2) Split the dataset to Train set and Test set

When predicting with time series data, it is important to split the data while maintaining the time order. In this case, the last 20% of the dataset was designated as the test set. By doing so, the model learns from data that occurred in real chronological order, allowing for more accurate future fire count prediction. More specifically:

i) Data Sorting: The dataset is sorted in chronological order. This helps the model to recognize and learn patterns in the data as they occur over time.

ii) Training and Test Set Split: As seen in Figure 10, the first 80% of the entire dataset is used as the training set, and the last 20% as the test set. This method is crucial in preserving the continuity and temporal structure of time series data.

iii) Considering Time Series Characteristics: During the training process, the model learns variations in the data over time, including seasonal factors and long-term trends. This is essential for performance evaluation on the test set and future predictions.

iv) Validity of Future Predictions: By composing the test set of the most recent data in chronological order, the model's performance in real world conditions can be better evaluated. This increases the reliability of the model's application in real world settings.

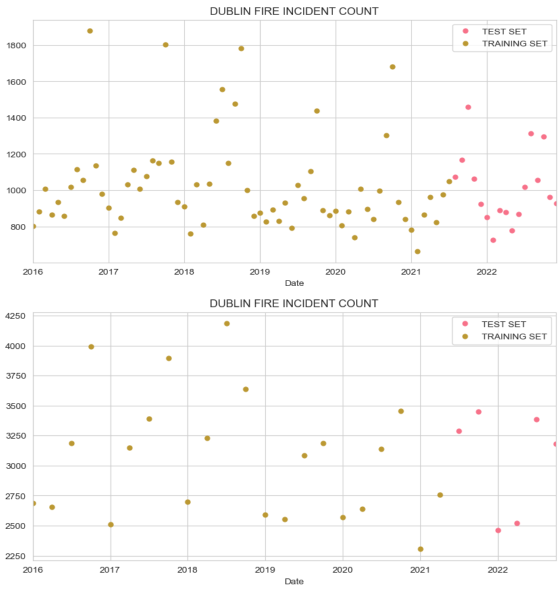


Figure 10. Training and test set of monthly and quarterly data.

*4.6. Evaluate model*

The Prophet model was chosen because of its suitability and high performance for time series data. The model was trained on daily, monthly, and quarterly datasets and evaluated using Mean Squared Error (MSE), Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE) and R-Squared scores.

1) Mean Squared Error (MSE): MSE measures the average of the squares of the errors, that is the average squared difference between the estimated values and the actual value. MSE is always non-negative, and a value of 0 indicates perfect predictions. It penalizes larger errors more than smaller ones, due to the squaring of each term. Commonly It is used in regression problems; however, because it squares the errors, it can give more weight to outliers.

$$MSE = \frac{1}{m}\sum_{i=1}^{m}(X_i - Y_i)^2$$

2) Mean Absolute Error (MAE): MAE measures the average magnitude of the errors in a set of predictions, without considering their direction. Like MSE, MAE is non-negative, and a value of 0 indicates no error. It is less sensitive to outliers than MSE, as it does not square the error terms. Often It is used in regression analysis, particularly when you want to avoid the larger error penalization characteristic of MSE.

$$MAE = \frac{1}{m}\sum_{i=1}^{m}|X_i - Y_i|$$

3) Mean Absolute Percentage Error (MAPE): MAPE measures the average of the absolute percentage errors of predictions. It expresses accuracy as a percentage, which can be easier to interpret. However, it can be problematic for values close to zero and can lead to infinite or undefined values. It is useful when you want to express the prediction accuracy in terms of percentage error, often used in forecasting and financial analysis.

$$MAPE = \frac{1}{m}\sum_{i=1}^{m}\left|\frac{Y_i - X_i}{Y_i}\right|$$

4) R-squared ($R^2$ Score): R-squared, also known as the coefficient of determination, measures the proportion of the variance in the dependent variable that is predictable from the independent variables. $R^2$ ranges from 0 to 1, with higher values indicating a better fit. A value of 1 implies that the model explains all the variability of the response data around its mean. The coefficient of determination can be interpreted as the proportion of the variance in the dependent variable that is predictable from the independent variables[12]. It is widely used in regression analysis to gauge the explanatory power of the model.

$$R^2 = 1 - \frac{SSE}{SST} = 1 - \frac{\sum(Y_i - \hat{Y}_i)^2}{\sum(Y_i - \bar{Y}_i)^2}$$

The result is like table on the Figure 11. That shows the result with comparing different forecasting models' performance metrics on monthly and quarterly data. It shows the R-Squared, Mean Squared Error (MSE), Mean Absolute Error (MAE), and Mean Absolute Percentage Error (MAPE) for models like Prophet,

| No | Model | Quarterly | | | |
|---|---|---|---|---|---|
| | | R-Squared | MSE | MAE | MAPE |
| 1 | **Prophet** | **0.91** | **115.70** | **104.33** | **3.58** |
| 2 | Arima | 0.74 | 204.35 | 189.97 | 6.49 |
| 3 | Linear regression | -0.26 | 450.41 | 439.02 | 14.32 |
| 4 | Polynominal regression | -5.81 | 1046.88 | 939.66 | 29.52 |
| 5 | SVR | -0.32 | 461.53 | 442.55 | 14.30 |
| 6 | Decision Tree regression | -0.52 | 495.49 | 466.83 | 14.87 |
| 7 | Random Forest regression | -0.90 | 553.41 | 496.96 | 15.47 |

| No | Model | Monthly | | | |
|---|---|---|---|---|---|
| | | R-Squared | MSE | MAE | MAPE |
| 1 | **Prophet** | **0.43** | **145.78** | **145.78** | **10.74** |
| 2 | Arima | 0.00 | 193.37 | 156.69 | 15.66 |
| 3 | Linear regression | -0.11 | 204.12 | 153.66 | 14.23 |
| 4 | Polynominal regression | -2.07 | 339.71 | 279.20 | 25.26 |
| 5 | SVR | -0.06 | 200.01 | 158.22 | 15.16 |
| 6 | Decision Tree regression | -0.03 | 196.76 | 162.17 | 16.58 |
| 7 | Random Forest regression | 0.00 | 193.74 | 156.86 | 15.60 |

Figure 11. R-Squared, MSE, MAE, MAPE table.

ARIMA, Linear Regression, Polynomial Regression, SVR (Support Vector Regression), Decision Tree Regression, and Random Forest Regression. For monthly data, the Prophet model performs best with an R-Squared of 0.43, indicating a moderate fit. ARIMA and the regression models exhibit low to negative R-Squared values, suggesting poor fits to the monthly data. Quarterly data show better R-Squared value particularly for Prophet (0.91) and ARIMA (0.74), indicating a better fit. The Prophet model again has the lowest MAPE, suggesting it has the best predictive accuracy for both monthly and quarterly data.

Additionally, As seen in Figure 12, in general regression models such as linear regression, SVR, decision tree regression, and random forest regression, issues of both overfitting and underfitting were encountered. Through this, it became evident that these general regression models are not well-suited for handling time series data.

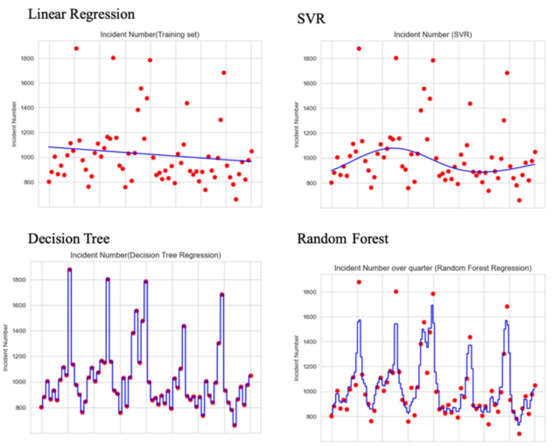



Figure 12. Underfitting (Linear Regression and SVR), Overfitting (Decision Tree and Random Forest)

*4.7. Tune model*

Model tuning refers to the process of adjusting or optimizing a machine learning or statistical model's settings to improve its performance. Especially, Prophet is a model used for predicting time series data, primarily focusing on forecasting trends and seasonality patterns in the data. To tune the Prophet model, there are following steps like below:

1) changepoint_prior_scale: This hyperparameter is used to adjust the prior scale for changepoints in the Prophet model. Changepoints represent the points in the data where trends change. By tuning changepoint_prior_scale, the model can be made sensitive to detecting changes in trends. This helps the model fit the data's trends more appropriately.

2) seasonality_prior_scale: This hyperparameter is used to adjust the prior scale for seasonality components in the Prophet model. Seasonality refers to the periodic patterns in the data, and by adjusting seasonality_prior_scale, we can control how strongly the model accounts for seasonality patterns. This enables the model to predict seasonality in the data more effectively.

3) holidays_prior_scale: This hyperparameter is used to adjust the prior scale for holiday effects in the Prophet model. Prophet considers holiday data when making predictions. By tuning holidays_prior_scale, we can control the importance given to holiday effects in the model, allowing it to better incorporate holiday related information.

4) seasonality_mode: This hyperparameter is used to set the seasonality mode for the Prophet model. We can choose between 'additive' or 'multiplicative' modes. 'Additive' mode implies that seasonality is added linearly, while 'multiplicative' mode implies that seasonality is multiplied. This choice determines how the model handles seasonality in its predictions.

```
param_space = {
    'changepoint_prior_scale': hp.uniform('changepoint_prior_scale', 0.001, 0.5),
    'seasonality_prior_scale': hp.uniform('seasonality_prior_scale', 0.01, 10),
    'holidays_prior_scale': hp.uniform('holidays_prior_scale', 0.01, 10),
    'seasonality_mode': hp.choice('seasonality_mode', ['additive', 'multiplicative'])
}
```

```
def objective(params):
    m = Prophet(
        changepoint_prior_scale=params['changepoint_prior_scale'],
        seasonality_prior_scale=params['seasonality_prior_scale'],
        holidays_prior_scale=params['holidays_prior_scale'],
        seasonality_mode=params['seasonality_mode']
    )
    m.fit(fire_train_prophet)
    future = m.make_future_dataframe(periods=365)
    forecast = m.predict(fire_test_prophet)

    mse = mean_squared_error(fire_test_prophet['y'], forecast['yhat'][:len(fire_test_prophet)])
    return {'loss': mse, 'status': STATUS_OK}
```

```
trials = Trials()
best = fmin(fn=objective,
            space=param_space,
            algo=tpe.suggest,
            max_evals=100,
            trials=trials)
```

```
best
```
```
{'changepoint_prior_scale': 0.020717217107150774,
 'holidays_prior_scale': 4.913828162255071,
 'seasonality_mode': 1,
 'seasonality_prior_scale': 0.035591724630509525}
```
```
opt_changepoint_prior_scale = best['changepoint_prior_scale']
opt_seasonality_prior_scale = best['seasonality_prior_scale']
opt_holidays_prior_scale = best['holidays_prior_scale']
opt_seasonality_mode = ['additive', 'multiplicative'][best['seasonality_mode']]
```
```
model = Prophet(
    changepoint_prior_scale=opt_changepoint_prior_scale,
    seasonality_prior_scale=opt_seasonality_prior_scale,
    holidays_prior_scale=opt_holidays_prior_scale,
    seasonality_mode=opt_seasonality_mode
)
```
```
model.fit(fire_train_prophet)
```
```
17:11:15 - cmdstanpy - INFO - Chain [1] start processing
17:11:15 - cmdstanpy - INFO - Chain [1] done processing
<prophet.forecaster.Prophet at 0x13731b890>
```

Figure 13. Code for tuning the hyperparameters.

The code in Figure 13. represents the process of hyperparameter optimization for forecasting time series data using the Prophet model. It defines search ranges for hyperparameters like changepoint_prior_scale, seasonality_prior_scale, holidays_prior_scale, and seasonality_mode, and finds the optimal values. The fmin function from the Hyperopt library was used to minimize the objective function, which trained the Prophet model and calculated the Mean Squared Error (MSE) on the test data. Finally, the optimized hyperparameters are used to create and fit a Prophet model to the training data. This process was utilized to improve the accuracy of time series predictions as Figure 14.

| No | Tuning Model | Monthly | | | | Quarterly | | | |
|---|---|---|---|---|---|---|---|---|---|
| | Before / After | R-Squared | MSE | MAE | MAPE | R-Squared | MSE | MAE | MAPE |
| 1 | Before Tuning | 0.43 | 145.78 | 145.78 | 10.74 | 0.91 | 115.70 | 104.33 | 3.58 |
| 2 | After Tuning | 0.65 | 114.23 | 79.05 | 7.56 | 0.92 | 114.05 | 103.42 | 3.54 |

Figure 14. Improved score after tuning the model.

## 5. Evaluation / Experiments

In this section, we will explore how the selected Prophet model, chosen as the most suitable model, generates prediction values and the underlying mechanism behind its predictions. We will analyze this process from the perspective of Explainable AI (XAI) to provide an interpretation.

### *5.1. Trend*

The trend component captures the long-term progression of the data. As seen in Figure 15, the linear decreasing trend in the chart suggests that the model has detected a long-term downward trend in the variable over time. This component is modelled using a piecewise linear regression model, which allows for flexibility in fitting the trend to the data[13].

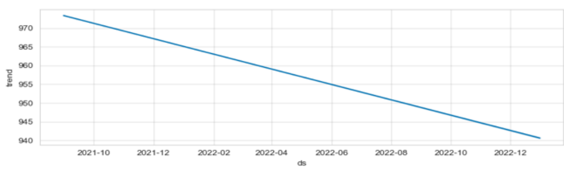



Figure 15. Quarterly data trend

### *5.2. Seasonality*

The seasonality represents regular patterns of change, such as yearly or weekly fluctuations. The variations in the lower chart show how the metric changes at certain times of the year, and the prediction model uses this pattern to estimate future seasonal variations[14].

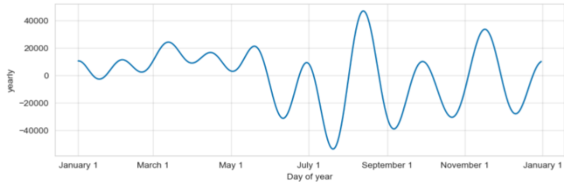



Figure 16. Quarterly data seasonality

### *5.3. Prediction*

The above-mentioned trend and seasonality graphs are components of the Prophet model's output, demonstrating how the model has analyzed the data to generate its predictions. Prophet models time series data by identifying and incorporating trends, seasonality, and holiday effects. The trend graph showed the trend over time, representing the overall direction that the model predicts the data is heading. The seasonality graph illustrated the yearly seasonality, which shows repeating patterns over the course of a year. Thus, the predictions were made based on the analysis. The final forecast was a combination of these components as seen in Figure 17.

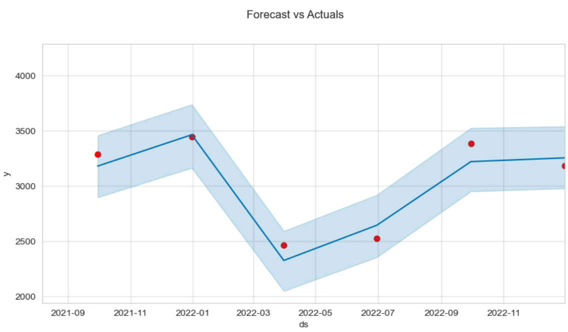
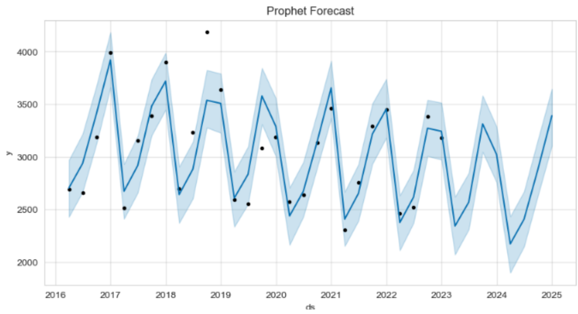



Figure 17. Quarterly data Predictions (Blue line) and Target values (Red points)

Lastly, I have predicted the Dublin fire incidents for the next 2 years, which means for the years 2023 and 2024, on a monthly and quarterly basis using the method described above. It's important to note that the existing dataset only extends up to 2022, which is why

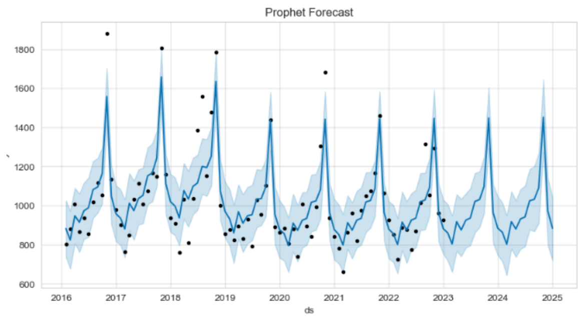
I made the predictions for 2023 and 2024. The related graphs and prediction tables are as Figure 18 and 19. However, it is important to mention that the evaluation scores for the monthly and quarterly prediction models are different, and it should be remembered that the quarterly prediction model showed relatively higher evaluation scores, as mentioned earlier.




| Date | 31/03/2023 | 30/06/2023 | 30/09/2023 | 31/12/2023 |
|---|---|---|---|---|
| Count | 2,343 | 2,569 | 3,313 | 3,023 |

| Date | 31/03/2024 | 30/06/2024 | 30/09/2024 | 31/12/2024 |
|---|---|---|---|---|
| Count | 2,174 | 2,408 | 2,882 | 3,390 |

Figure 18. Quarterly prediction for 2023 – 2024 graph and table




| Date | 31/01/2023 | 28/02/2023 | 31/03/2023 | 30/04/2023 | 31/05/2023 | 30/06/2023 |
|---|---|---|---|---|---|---|
| Count | 859 | 803 | 918 | 875 | 925 | 936 |

| Date | 31/01/2024 | 29/02/2024 | 31/03/2024 | 30/04/2024 | 31/05/2024 | 30/06/2024 |
|---|---|---|---|---|---|---|
| Count | 863 | 802 | 917 | 880 | 932 | 941 |

| Date | 31/07/2023 | 31/08/2023 | 30/09/2023 | 31/10/2023 | 30/11/2023 | 31/12/2023 |
|---|---|---|---|---|---|---|
| Count | 1,021 | 1,030 | 1,097 | 1,448 | 963 | 884 |

| Date | 31/07/2024 | 31/08/2024 | 30/09/2024 | 31/10/2024 | 30/11/2024 | 31/12/2024 |
|---|---|---|---|---|---|---|
| Count | 1,024 | 1,032 | 1,090 | 1,452 | 972 | 884 |

Figure 19. Monthly prediction for 2023 – 2024 graph and table

## 6. Conclusion

This study conducted a comparative analysis of seven machine learning models for predicting the number of fire incidents in Dublin. The performance of each model was measured using Mean Squared Error (MSE), Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE) and R-Squared scores. Among these, the Prophet model demonstrated the best performance. The study emphasized technical transparency by applying the principles of Explainable AI (XAI), focusing on data transparency, model explainability, and visualizations. This approach contributes to a better understanding and trust in AI models, providing significant insights for various applications related to fire incident predictions.

## 7. Future Work

In terms of future work, the current study is centered around analyzing the number of fire incidents in Dublin using time series data. I have identified two key areas that I can delve into for further research:

1) Time Series Analysis combined with other datasets : To deepen the analysis, I intend to combine the existing dataset with weather-related data. This will allow me to investigate the connections between different weather variables and the incidence of fire events. Such an approach may reveal valuable insights into how particular weather conditions or patterns could affect the probability of fire incidents. Furthermore, I can utilize the Dublin population dataset to explore the correlation between fire incidents and changes in the population over time. This exploration may provide insights into whether population dynamics have any influence on the frequency of fire incidents.

2) Geospatial Analysis: Another part of the study that I aim to pursue is a more localized examination of fire incidents within specific areas of Dublin. This geospatial analysis can provide highly specific insights that are valuable for both the city of Dublin and the Dublin Fire Brigade. For example, I plan to determine which areas in Dublin would benefit most from the construction of additional fire stations based on incident density. Additionally, I can assess which fire stations may require additional resources or personnel to efficiently respond to incidents in their respective areas. By focusing on these specific geographic regions, I expect to be able to answer critical questions related to resource allocation, station optimization, and overall incident management strategies.

In summary, the future work will involve a comprehensive exploration of time series data, weather-related factors, population dynamics, and geospatial analysis to gain a deeper understanding of fire incidents in Dublin. This study has the potential to provide valuable insights that can inform decision-making processes, enhance public safety, and optimize the allocation of resources for the city and its fire services.

## 8. Contributions

In this section, I have summarized the findings and answers to the research questions:

### *8.1. Which regression model has the best performance for predicting time series data?*

For the first research question, it was determined that time series models outperform general regression models for predicting time series data. Among the time series models evaluated, the Prophet model emerged as the best-performing one in terms of prediction accuracy and reliability.

### *8.2. What time period exhibits the most distinct seasonality for prediction?*

Regarding the second research question, an in-depth exploration of the dataset revealed that the quarterly time period exhibited the most distinct and prominent seasonality patterns. This insight is crucial for understanding the underlying patterns in the data and making accurate forecasts.

### *8.3. How can the AI project be made explainable for users?*

To address the last research question and ensure that the AI project is explainable to users, I have adopted following strategies:

1) Data Transparency: Begin by providing transparency about the data sources and preprocessing steps. Explain where the data comes from, how it's collected, and any data transformations applied. Users should have a clear understanding of the data used for predictions.

2) Model Explainability: To enhance the interpretability of the Prophet model, I focused on making it more transparent. I explained the model by showcasing trend and seasonality graphs, which illustrated the prediction process. This approach aimed to improve the model's explainability, enabling users to understand how it works without relying on complex analysis techniques.

3) Visualizations: Visual aids can be powerful in conveying how the model works. You can create visualizations that show the historical data, observed patterns, and the model's predictions. Highlight key trends, seasonality, and any anomalies that the model has captured.

In summary, the study has not only provided insights into the choice of regression models and seasonal patterns but also emphasized the importance of explainability in making AI-driven projects accessible and trustworthy for end-users.

## 9. References


[1] V. Vakkuri, K.-K. Kemell, J. Kultanen, and P. Abrahamsson, 'The Current State of Industrial Practice in Artificial Intelligence Ethics', *IEEE Software*, vol. 37, no. 4, pp. 50–57, Jul. 2020, doi: 10.1109/MS.2020.2985621.

[2] M. Langer *et al.*, 'What do we want from Explainable Artificial Intelligence (XAI)? – A stakeholder perspective on XAI and a conceptual model guiding interdisciplinary XAI research', *Artificial Intelligence*, vol. 296, p. 103473, Jul. 2021, doi: 10.1016/j.artint.2021.103473.

[3] L. Yang, R. Corbally, and A. Malekjafarian, *Using Machine Learning to Predict the Impact of Incidents on the M50 Motorway in Ireland*. 2022.

[4] Md. A. Awal, J. Rabbi, Sk. I. Hossain, and M. M. A. Hashem, 'Using linear regression to forecast future trends in crime of Bangladesh', in *2016 5th International Conference on Informatics, Electronics and Vision (ICIEV)*, May 2016, pp. 333–338. doi: 10.1109/ICIEV.2016.7760021.

[5] A. Barredo Arrieta *et al.*, 'Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI', *Information Fusion*, vol. 58, pp. 82–115, Jun. 2020, doi: 10.1016/j.inffus.2019.12.012.

[6] B. Goodman and S. Flaxman, 'European Union Regulations on Algorithmic Decision Making and a "Right to Explanation"', *AI Magazine*, vol. 38, no. 3, pp. 50–57, 2017, doi: 10.1609/aimag.v38i3.2741.

[7] S. Ali *et al.*, 'Explainable Artificial Intelligence (XAI): What we know and what is left to attain Trustworthy Artificial Intelligence', *Information Fusion*, vol. 99, p. 101805, Nov. 2023, doi: 10.1016/j.inffus.2023.101805.

[8] ‘Fire Brigade and Ambulance Call Outs DCC - Dataset - data.smartdublin.ie’. Accessed: Dec. 16, 2023. [Online]. Available: https://data.smartdublin.ie/dataset/fire-brigade-and-ambulance
[9] N. K. Rathlev *et al.*, ‘Time Series Analysis of Variables Associated With Daily Mean Emergency Department Length of Stay’, *Annals of Emergency Medicine*, vol. 49, no. 3, pp. 265–271, Mar. 2007, doi: 10.1016/j.annemergmed.2006.11.007.
[10] C. B. Aditya Satrio, W. Darmawan, B. U. Nadia, and N. Hanafiah, ‘Time series analysis and forecasting of coronavirus disease in Indonesia using ARIMA model and PROPHET’, *Procedia Computer Science*, vol. 179, pp. 524–532, Jan. 2021, doi: 10.1016/j.procs.2021.01.036.
[11] ‘Forecasting of demand using ARIMA model - Jamal Fattah, Latifa Ezzine, Zineb Aman, Haj El Moussami, Abdeslam Lachhab, 2018’. Accessed: Dec. 18, 2023. [Online]. Available: https://journals.sagepub.com/doi/full/10.1177/1847979018808673
[12] D. Chicco, M. J. Warrens, and G. Jurman, ‘The coefficient of determination R-squared is more informative than SMAPE, MAE, MAPE, MSE and RMSE in regression analysis evaluation’, *PeerJ Comput. Sci.*, vol. 7, p. e623, Jul. 2021, doi: 10.7717/peerj-cs.623.
[13] P. Khare, ‘Understanding FB Prophet: A Time Series Forecasting Algorithm’, ILLUMINATION. Accessed: Dec. 18, 2023. [Online]. Available: https://medium.com/illumination/understanding-fb-prophet-a-time-series-forecasting-algorithm-c998bc52ca10
[14] ‘Time Series Analysis and Forecasting with FB Prophet Python’, Model Differently. Accessed: Dec. 18, 2023. [Online]. Available: https://www.modeldifferently.com/en/2022/04/analisis_prediccion_ts_prophet/