Machine Learning Analyses of Meteor Data

Viswesh Krishna

Research Student, Centre for Fundamental Research and Creative Education.
Email: krishnaviswesh@cfrce.in
The objective of this paper is to analyse meteor data through statistical analysis and machine learning. Two examples are used to highlight the effectiveness of these tools in the analysis of meteor data: Outlier Detection and Feature Prediction. Outlier Detection consists of identifying meteor outbursts from the data using statistical methods. Feature Prediction involves predicting the Level, Shower Type and Next Occurrence of a predicted meteor shower with machine learning models. Code is available at [https://github.com/VisweshK/OutburstDetection-FeaturePrediction](https://github.com/VisweshK/OutburstDetection-FeaturePrediction).

1 Introduction

Meteor shower forecasting is today done through huge servers and supercomputers by plotting the exact trajectories of various objects in space and calculating the time to strike the atmosphere. However, this prediction is fraught with uncertainty due to varying levels of accuracy in the measurements of the position and velocity of the astronomical objects under consideration (Vaubil-lon, 2017). Further, it is impossible to predict other features of a meteor shower before hand as these do not correspond to any concrete mathematical formulae.

It is known that large datasets of meteor shower activity are uploaded by the International Meteor Organisation, NASA and other organisations. With the fast growth in Machine Learning and Data Analytics over the past few years, it is possible that new models could highlight previously unnoticed trends in meteor data. Indeed, as most of the meteor shower prediction is based on the past behaviour of the shower and parent bodies, it is clear that the application of these newly developed tools could provide insight into the relationship between meteor shower features.

Machine Learning works with data and processes it to discover patterns that can be later used to analyse new data. Machine learning usually relies on a specific representation of the data; a set of features that are understandable for a computer. For the current project, Supervised learning was applied. Supervised machine learning algorithms can apply what has been learned in the past to new data using labeled examples to predict future events.

Starting from the analysis of a known training dataset, the learning algorithm produces an inferred function to make predictions about the output values. The system is able to provide targets for any new input after sufficient training. The learning algorithm can also compare its output with the correct, intended output and find errors in order to modify the model accordingly. With the availability of 35 years of IMO meteor data, it is possible to build a supervised learning model for predicting future events. In this paper, we will explore some of the models which can be applied for supervised learning and their prediction ability.

2 General Preprocessing and Methods

The datasets were downloaded from the International Meteor Organisation (IMO) Visual Meteor DataBase (IMO, 1982), consisting of 35 years of meteor data. The columns (features) in the database were User ID, Start Date, End Date, Right Ascension, Declination, Teff, F, Lm, Shower Type, Method, Number, Latitude, Longitude and Elevation (of the observatory). The processing was first done on R. The MetFns package was developed by the IMO on R (Veljkovic, 2017) and allowed for the generation of new features such as Solar Longitude, and Zenithal Hourly Rates to enable better prediction.

2.1 Processing on R

The preprocessing of the data was first done on R as it enabled easily manipulation of the datasets and the MetFns package developed by the IMO could also be accessed here. The data was extracted from the CSV files, features were scaled and unnecessary indexing features were removed. Removal of duplicate entries was accomplished by using the differences in the Start Date, End Date, Right Ascension, Declination and Shower Type values.

2.2 Processing on Python

On Python, the data was extracted from the CSV file through the Pandas (McKinney, 2010) module. For scaling the data, the Standard Scaler and MinMaxScaler in the Preprocessing module of Scikit-Learn (Pedregosa et al., 2011) were utilised to normalize the features to advised values([-1,+1]) for the predictive models. Label Encoders from the Scikit-Learn module were used to index the various strings in the Shower data and convert them to integer values as string values could not be processed by certain machine learning models. Further, the latitude, longitude and elevation of an observatory were collapsed into a single index through another label encoder for ease of classification. The Keras (Chollet et al., 2015) and Scikit-Learn libraries on Python were used for building and evaluating the machine learning models. The standard training vs testing split for the supervised learning was set at 80-20:80% of the data was used to train the model and 20% was used to test the model.

3 Outlier Detection

3.1 Introduction

Outlier Detection involved the detection of meteor outbursts from the dataset through statistical analysis. Me-
teor outbursts could be thought of as irregular meteor showers with a high ZHR value and could be detected using customised outlier detection models. Detection of Meteor outbursts is difficult as they are generally highly aperiodic in nature and are thus difficult to study in detail. Further, it is likely that certain outbursts may be misclassified as periodic meteor showers. Thus, the detection, processing and extraction of meteor outbursts in the data was also extremely important for filtering the data.

3.2 Methods and Results

The outliers were handled as follows. First, the ZHR values for every data point were calculated. The meteor count of the individual months across all the years were plotted against each month to observe the distribution of the meteor count as seen in figure 1. As meteors occur in cycles (Jenniskens, 2006), there would be a clear clustering of the points towards some modal value. Any outlier months were identified by using the score function in the Outliers (Komsta, 2011) package on R. The score function incorporated various distributions such as the normal, t, chi-squared, IQR and MAD. The outlier points that were identified corresponded to certain showers from a particular month and year. For each outlier point, all showers from that month and year were extracted. Each month and year in this list should contain a number of meteor outbursts which caused the meteor count to deviate from the modal value. For every month and year on this list, showers with a large deviation of the ZHR value were classified as meteor outbursts and were filtered out of the main data frame. When the meteor counts for each month was plotted again, there is a clear clustering of the points towards a modal value as seen in figure 1.

4 Feature Prediction

4.1 Introduction

Feature Prediction involves predicting certain useful features of a meteor shower using machine learning models. The features that have been predicted are the Level, Shower Type and the Next Appearance of a meteor shower. The motivation behind predicting such features was to highlight the power of machine learning models in understanding trends in the meteor data.

4.2 Methods

For each feature given below, we first detail the processing steps, then describe the machine learning model implemented along with the reasoning behind the decision.

4.2.1 Level Prediction

The Level of the meteor shower was calculated from the Number value which represented the number of meteors observed during the shower. The Level was defined as Low, Medium or High by the process of Equi-Depth binning of the Number feature. The following features were used for Level prediction: Start Month, Right ascension, Declination, Shower Type, Observatory and Solar Longitude. Only features that would be present before the predicted meteor shower were applied to simulate real time testing of the model.

As Level prediction was a classification problem, we had to choose between the Decision Tree, K-Nearest Neighbours, Stochastic Gradient Descent, Support Vector Machines, Gaussian Process and Perceptron Classifiers. When a scoring algorithm was run on a small portion of the dataset, it was found that the Decision Tree Classifier and K-Nearest Neighbours had the least mean squared errors and were most suitable for the given data.

Decision Tree Classifiers

Decision Tree Classifiers were applied first to the data. The Decision Tree Classifier uses the training data to develop a Tree-like data structure and then classify the testing data based on this Tree. This model was ideal suited to the dataset because of its simplicity. The Decision Tree required simplified classification features and thus the Latitude, Longitude and Elevation features were combined into
the Observatory feature. Further, Shower types were also classified with an index to allow for simplicity in the Tree structure.

**K-Nearest Neighbour Classifiers**  
Level prediction was then done with the use of a K-Nearest Neighbours Classifier. This classifier first learns the distribution of the data in the N-D space by dividing it into clusters which are centred around specific points, corresponding to the level value of the shower. It then classifies the testing data based on their relative distance to the centre of the various clusters.

### 4.2.2 Shower Type Prediction

The Shower type data was present in the form of separate strings describing which type of shower it related to. By using the Label Encoder function from the Preprocessing module all the Shower types were indexed independently from 1 to 242 as there were 242 different meteor shower types in the data. The Label Encoder enabled us to later convert the indexes back to the Shower strings which could be then cross-referenced with the data. Only the Start Date was used for Shower type Prediction due to the shower type’s repetitive nature. The Shower Type prediction was once again treated as a Classification problem and was scored with various classifier models (see section 4.2.1). When a scoring algorithm was run on a small portion of the dataset, the Decision Tree was found to give us the least mean squared error and was most suitable to the data.

### 4.2.3 Meteor Forecasting

Meteor forecasting was done separately on the dataset consisting of the non-outbursts and the outbursts. This was implemented through an Long Short Term Memory (LSTM) network built on the Keras package in Python. The LSTM was advantageous as it kept track of one block of the data at a time thereby learning patterns within these blocks. This idea was specifically applicable to meteor forecasting as meteors occur in cycles which repeat periodically. By filtering the meteor outbursts out of the training set, the model could individually tackle the periodic meteor occurrences and then move on to the meteor outbursts dataset. The LSTM was used to predict the interval of time in hours between two successive meteor showers.

### 4.3 Analyses and Results

It was found that the binning of the Shower feature and combination of the latitude, longitude and elevation into an observatory index greatly increased our prediction rates across all models. The individual analyses and prediction rates for each of the above features are described sequentially below.

#### 4.3.1 Level Prediction

It was seen that binning of the Number value played an important role in the predictive capacity of the model. By using values that aligned with layman descriptions of the level of the meteor shower, the model worked with exceedingly high accuracy. The data had an asymptotic distribution of the frequency of the Number data with the majority of the entries clustering at the lower Number values (see figure 2). Due to this distribution, it was natural for the model to deviate towards predicting at the lowest bin value repeatedly. This behaviour was checked by ensuring equal error rates for prediction of all of the individual bins. It was seen that both the Decision Tree Classifier and the K-Nearest Neighbour Classifier had equal error rates for all bins, ensuring the model was truly learning the trends in the data.

![Figure 2 - Frequency Distribution of Number values](image)

Note the exponentially decreasing frequency as the meteor count value increases. This would cause the prediction model to stray towards predicting the lower meteor count values.

**Decision Tree Classifiers**  
When implemented, the Decision Tree Classifier had a prediction rate at 80% for predicting the Level of the meteor shower against a random prediction rate of 33% (see table 1). Out of the 20% error rate, 35% of the errors were seen when the actual Number value of the meteor shower was within ±3 of the binning range, which corresponded to a close prediction.

**K-Nearest Neighbour Classifiers**  
When implemented, the K-Nearest Neighbour Classifier had a prediction rate at 80% for predicting the Level of the shower against a random prediction rate of 33% (see table 1). Out of the 20% error rate, it was found that 37% of the errors were seen when the actual Number value of the meteor shower was within ±3 of the binning range which corresponded to a close prediction.

#### 4.3.2 Shower Type Prediction

The size of the Tree was customised as a large number of nodes had to be generated to account for the various
5 Conclusion

In this paper, we have first analysed the IMO Visual Meteor Database to predict various features of meteor showers using Machine Learning models. We also detect meteor outbursts with the help of customised outlier detection algorithms. Using the Keras and Scikit-Learn modules on Python, supervised learning models were built and applied to the dataset for predicting the level, shower type and next occurrence of a meteor shower. This could potentially be used for making initial predictions of upcoming meteor showers.

Acknowledgements

This work was carried out at the Center for Fundamental Research and Creative Education (CFRCE), Bangalore, India under the guidance of Dr. B S Ramachandra whom I wish to acknowledge. I would like to acknowledge the Director Ms. Pratiti B R for creating the highly charged research atmosphere at CFRCE. Further I would like to thank Abhiram Harithas and Jyothish Jose, my mentors, for their invaluable guidance on this project.

References