University of Novi Sad
Faculty of Sciences
Department of mathematics
and informatics

Master thesis

Pollen Allergy Symptoms Prediction using Machine Learning Algorithms

Author:
Nataša Diklić

Supervisor:
Oskar Marko, PhD

Novi Sad, September 2024

Abstract

The purpose of this thesis is to find and train the most adequate machine learning model for predicting pollen allergy symptoms, using Python programming language. As a major pandemic health problem for a human pollen allergies can interrupt people's everyday activities with sneezing, stuffy nose, watery eyes, cough, asthma and other eyes, nose and lungs problems. Two sources of data were used: Patient's Hayfever Diary (PHD) where users entered their symptoms and pollen measurements conducted by the Laboratory for Palynology at the Faculty of Sciences, University of Novi Sad. The thesis evaluates the performance of two approaches for the symptoms prediction: prediction of the intensity of overall symptoms (0-21) and prediction of the type of symptoms (eyes, nose or lungs symptoms). For the first approach, four regression models were evaluated: K-Nearest Neighbors Regressor, Random Forest Regressor, Gradient Boosting Regressor and XGB Regressor and for the second one five classification models were evaluated: K-Nearest Neighbors Classifier, Random Forest Classifier, Support Vector Machine Classifier, Gradient Boosting Classifier and XGB Classifier. All mentioned models were trained on both daily and cumulative pollen concentrations - cumulative n = 2, n = 3, n = 4, n = 5, n = 10 and n = 15, where n is the number of days. The best regression results were obtained with Random Forest Regressor for n = 15, while XGB Classifier showed the best performance for classification, also for n = 15. In-depth evaluation and exploration of the models was conducted for the user who used the pollen diary the most. Although in that case a smaller amount of data was used, the results obtained were highly satisfactory: accuracy score is equal to 0.83 for RF Classifier and mean absolute error is equal to 1.05 for RF Regressor. In addition, extensive data analysis was conducted in this thesis, revealing significant correlations, statistics and conclusions drawn from two distinct datasets.

Acknowledgements

I would like to express my sincere gratitude to my supervisor prof. Oskar Marko for support, directions and insightful comments. His knowledge and experience have encouraged me during this research.

I am also very grateful to Dr. Branko Šikoparija and Dr. Predrag Radišić for generously providing the pollen measurements data collected in the Laboratory for Palynology at the Faculty of Sciences, University of Novi Sad, which is funded by the City of Novi Sad (Department of Environmental Protection) and the Autonomous Province of Vojvodina (Provincial Secretariat for Urban Planning and Environmental Protection), as well as to Uwe Berger and Markus Berger for kindly supplying the data from the Patients' Hayfever Diary (PHD) application.

Additionaly, I would like to thank my friends for the wonderful times we shared during studies.

Finally, I would like to extend my deepest gratitude to my family for their unwavering love and support. I am especially thankful to my son, whose patience and understanding allowed me the space and time needed to complete this master's thesis. None of this would have been possible without them, and I dedicate this work to my family.

Contents

1 Introduction 7

2 Materials and Methods 13
2.1 Data . . . 13
2.1.1 Data Source and Data Description . . . 13
2.1.2 Data Analysis . . . 14
2.1.3 Data Pre-processing . . . 21
2.2 Methods . . . 25
2.2.1 K-Nearest Neighbors Algorithm . . . 26
2.2.2 Random Forest Algorithm . . . 29
2.2.3 Support Vector Machine Algorithm . . . 30
2.2.4 Gradient Boosting Algorithm . . . 32
2.2.5 XGBoost Algorithm . . . 33
2.2.6 Grid Search . . . 34
2.3 Evaluation Metrics . . . 36
2.3.1 Regression Evaluation Metrics . . . 36
2.3.2 Classification Evaluation Metrics . . . 37

3 Experimental Results 39
3.1 The Symptom Types Prediction . . . 39
3.1.1 Classification models evaluation on data set with repeated inputs . . . 39
3.1.2 Classification models evaluation on data set with unique inputs . . . 40
3.1.3 Classification models evaluation on data samples from user with the most entries . . . 41
3.2 The Overall Symptoms Prediction . . . 42
3.2.1 Regression models evaluation on data set with repeated inputs . . . 42
3.2.2 Regression models evaluation on data set with unique inputs . . . 43
3.2.3 Regression models evaluation on data samples from user with the most entries . . . 44

4 Discussion and Ideas for Future Work 45
4.1 Discussion . . . 45
4.2 Ideas for future work . . . 47

5 Conclusion 48

References 50

Biography 53

Appendix A 54

List of Figures

1 The seasons of pollen allergies 7
2 Machine Learning Methods 8
3 Features of pollen dataset 13
4 Features of PHD dataset 14
5 User's data entries 15
6 Data entries per year 15
7 Unique number of users per year 16
8 Correlation between symptoms 16
9 Overall Symptoms on month level 17
10 Percentage of use of Medical Treatment 17
11 Overall pollen concentration over the years 18
12 Overall symptoms data(blue graph) along pollen concentration measurements(green graph) for the top 10 users 20
13 Machine learning pipeline of the implementation 21
14 The most common pollen species and overall symptoms for a given month 23
15 Selected Features 23
16 Grouping samples by most intense type of symptoms 24
17 Grouping samples by average of overall symptoms 25
18 Cumulative concentration of Ambrosia for n = 2 26
19 KNN Algorithm working visualization for classification and regression 27
20 Random Forest Algorithm 30
21 Possible hyperplanes 31
22 Hyperplanes in 2D and 3D feature space 31
23 K-fold cross validation 35
24 Confusion Matrix 37
25 Prediction results with the best score highlighted in red - models evaluation on dataset with repeated inputs 39
26 Prediction results for user with the most entries 40
27 Prediction results with highlighted best scores for different n values, obtained after training and evaluating ML models using data set with grouped samples 40
28 Number of data entries from top 10 users 41
29 Prediction results with the best score highlighted in red for user with the most entries (classification) 41
30 Prediction results with the best scores highlighted in red - models evaluation on data set with repeated inputs 43

31 Prediction results with highlighted best scores for different n values, obtained after training and evaluating ML models using data set with grouped samples 44
32 Prediction results with the best scores highlighted in red for user with the most entries (regression) 45
33 Concentrations of different pollen species over the years 54
34 AMBR and URTI pollen concentrations alongside the symptoms of the top 10 users 55

1 Introduction

Pollen exposure is a major cause of respiratory allergies worldwide. Climate change, air pollution and urbanization could indirectly favour respiratory allergies, as increasing temperatures bring about earlier flowering and pollination periods and concomitantly overall shorter allergen-free seasons [1][2]. Pollen allergy can manifest itself as allergic rhinitis, allergic conjunctivitis and/or allergic bronchial asthma [3]. It affects millions of individuals worldwide, leading to symptoms which can significantly impact the quality of life, causing discomfort, reducing productivity and affecting daily activities. Accurate prediction of pollen allergy symptoms enables individuals to plan their activities accordingly, avoid exposure and take preventive medications. This can lead to a significant reduction in symptoms and an overall improvement in quality of life.

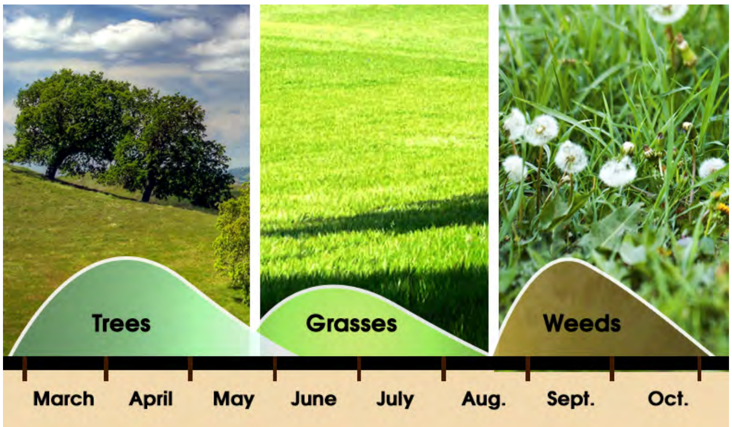


Figure 1: The seasons of pollen allergies [4]

The development of accurate predictive models for pollen allergy symptoms requires the integration of various scientific disciplines, including meteorology, biology and data science. This interdisciplinary approach can lead to advancements in research methodologies, data analysis and machine learning techniques. Machine learning algorithms, with their ability to handle large datasets and uncover complex patterns, present a promising approach to predicting pollen allergy symptoms. By leveraging historical data on pollen counts and reported symptoms, machine learning models can be trained to forecast symptom occurrence and intensity. This predictive capability can be harnessed through both regression and classification techniques. Regression algorithms can predict the intensity of symptoms on a continuous scale, while classification algorithms can categorize symptoms into distinct classes, such as eyes, nose, lungs or some of the combinations of these three groups of symptoms. Regression and classification algorithms are types of supervised machine learning algorithms, which are a subset of the broader field of machine learning.

Machine learning is a branch of artificial intelligence that allows systems to automatically learn from data and improve their performance over time without explicit programming. It involves creating algorithms capable of processing and analyzing large datasets, recognizing patterns, and making predictions or decisions based on the input data. Historically, machine learning has been influenced by biologically inspired models, with long-term objectives often focused on developing models and algorithms that can process information with the same efficiency as biological systems. The field also integrates many traditional statistical methodologies but emphasizes mathematical modeling and prediction. Today, machine learning is a central component of numerous areas within computer science and plays a critical role in large-scale data processing and analysis across various domains.

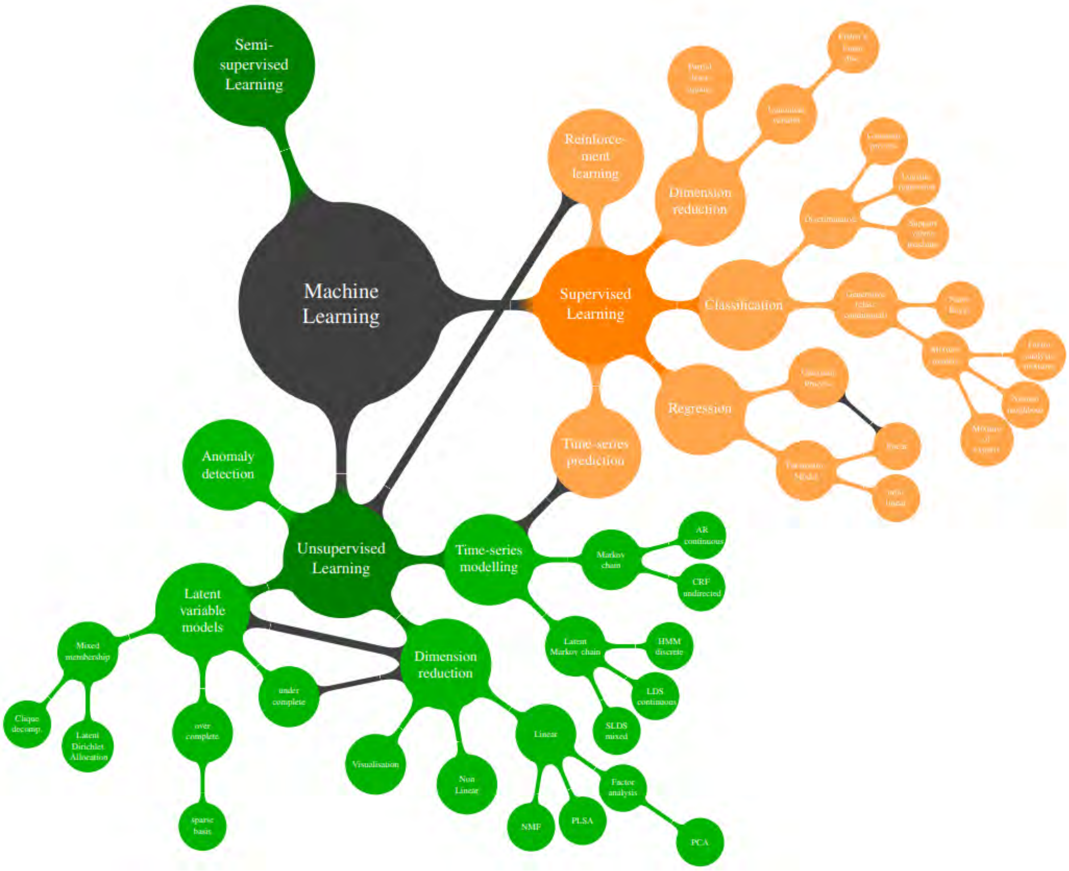


Figure 2: Machine Learning Methods [5]

Broadly speaking, the two main subfields of machine learning are *supervised learning* and *unsupervised learning.* In supervised learning, the primary focus is on making accurate predictions, while in unsupervised learning, the goal is to find concise representations or patterns within the data. In both approaches, the objective is to develop methods that generalize well to previously unseen data. Consequently, a distinction is made between the data used for training a model and the data used for evaluating the performance of the trained model.

Supervised Learning. Given a set of data $D = \{(x_n, y_n), n = 1, \dots, N\}$, the task is to learn the relationship between the input x and output y such that, when given a novel input x^*, the predicted output y^* is accurate. The pair (x^*, y^*) is not in D but is assumed to be generated by the same unknown process that generated D. To specify explicitly what accuracy means, one defines a loss function $L(y_{\text{pred}}, y_{\text{true}})$ or, conversely, a utility function $U = -L$. In supervised learning, our interest is in describing y conditioned on knowing x. From a probabilistic modelling perspective, we are therefore concerned primarily with the conditional distribution $p(y|x, D)$. The term 'supervised' indicates that there is a notional 'supervisor' specifying the output y for each input x in the available data D. The output is also called a 'label', particularly when discussing classification. [5]

Predicting tomorrow's pollen allergy symptom $y(T + 1)$ based on past observations $y(1), \dots, y(T)$ is a form of supervised learning. We have a collection of times and symptoms $D = \{(t, y(t)), t = 1, \dots, T\}$ where time t is the input and the symptom $y(t)$ is the output. If the output is one of a discrete number of possible classes, this is called a *classification problem.* If the output is continuous, this is called a *regression problem.*

Key algorithms in supervised learning, such as those utilized in this thesis, include Random Forest, Gradient Boosting, XGBoost, K-Nearest Neighbors, and Support Vector Machines. Each of these algorithms offers distinct advantages: Random Forest and mentioned boosting methods are ensemble techniques that aggregate multiple decision trees to enhance predictive accuracy, while K-Nearest Neighbors classifies data points based on the distance to nearby points. Support Vector Machines, on the other hand, are particularly well-suited for classification tasks in high-dimensional feature spaces.

Unsupervised Learning. Given a set of data $D = \{x_n, n = 1, \dots, N\}$, in unsupervised learning we aim to find a plausible compact description of the data. An objective is used to quantify the accuracy of the description. In unsupervised learning, there is no special prediction variable; thus, from a probabilistic perspective, we are interested in modeling the distribution $p(x)$. The likelihood of the model to generate the data is a popular measure of the accuracy of the description. [5]

Machine learning plays a crucial role in *aerobiology.* In addition to predicting pollen allergy symptoms, forecasting pollen concentrations is also of significant importance. By analyzing datasets that include historical pollen counts, meteorological data and environmental variables, machine learning algorithms can identify complex patterns and relationships that are not easily detectable using traditional methods.

The majority of pollen sensitization in Europe are caused by Betula(Birch) and Poaceae(Grass)[6]. Ambrosia is the second most important cause of seasonal asthma and rhinitis in many areas of its native distribution range (i.e. North America), and in the past decade, its clinical relevance has increased notably throughout Europe [7]. It is estimated that the number of allergic people in Europe will more than double by 2060 [8]. Hence, the ability to predict the variability of daily pollen concentrations for the most important allergenic pollen would be beneficial for a great number of pollen-sensitive individuals. *Pollen calendars* as predictive models for daily concentrations of airborne Ambrosia, Betula, and Poaceae pollen (pollen species used in this thesis) which are based on historical pollen data by calculating the mean or median pollen concentrations for specific dates over several years can provide reliable predictions for managing allergies. Increasing the number of calibration years generally enhances model performance, with four years being identified as the optimal period for the most significant improvement. However, calendar models are more accurate when using daily resolutions, as this better captures the variability in pollen exposure, which is crucial for individuals sensitive to pollen. Using advanced calendar models improves the prediction of pollen concentrations by lowering the normalized root mean square error (NRMSE) compared to standard models. Overall, pollen calendars have proven to be valuable tools for forecasting airborne pollen levels in the absence of meteorological data, offering reliable predictions that can assist in managing allergy symptoms effectively. [9]

Also, a model for predicting Ambrosia pollen emissions has been developed [10]. This model is based on a study conducted on the Pannonian Plain over three flowering seasons (2014-2016), involving the sampling of airborne pollen at different heights and temporal resolutions. The results demonstrated substantial variability in pollen production, with daily estimates ranging from 6.38 billion to 770 billion grains for the entire field. The weak correlations between pollen concentrations and meteorological parameters were found, indicating nonlinear relationships. High pollen concentrations were associated with temperatures between 20-24°C, while high humidity could delay or halt pollen emission. Additionally, the notable diurnal cycles in Ambrosia pollen release, with a significant morning peak and a secondary peak in the evening, are identified. These findings underline the need for further studies to refine emission models and explore the variability in pollen production across different regions.

Various techniques have been employed to forecast airborne Ambrosia pollen. One of them includes applying Deep Neural Networks and Ensemble Machine Learning methods: XGBoost, Random Forests and Bayesian Ridge Regression. The training data included twenty-four years of daily pollen concentration measurements together with the European Center for Medium-Range Weather Forecasts atmospheric weather and land surface reanalysis data from 1987 to 2011 is used to develop the machine learning predictive models. The last six years of the dataset from 2012 to 2017 is used to independently test the performance of models. The correlation coefficients between the estimated and actual pollen abundance for the independent validation datasets for the deep neural networks, random forest, extreme gradient boosting and Bayesian ridge were 0.82, 0.81, 0.81 and 0.75 respectively, showing that machine learning can be used to effectively forecast the concentrations of airborne pollen. [11]

In addition to predicting the concentration of Ambrosia, there is also investigations whether intermittent sampling can effectively replace continuous sampling for monitoring airborne pollen concentrations, particularly Ambrosia pollen [12]. Continuous long-term sampling, often considered the gold standard in aerobiology, is resource-intensive and may not always be feasible due to limitations in sampling media and equipment, especially in environments with high concentrations of airborne particles. Hourly pollen concentrations obtained by averaging 56, 28, 14 and 7 equidistantly distributed 1.07-min concentrations of Ambrosia airborne pollen were compared and the results showed that a majority of the information on trends and magnitudes of hourly pollen concentrations could be captured even with reduced sampling frequency. Although the absolute percentage error increased as the number of samples per hour decreased (averaging 10% for 28 samples, 20% for 14 samples, and 39% for 7 samples), these errors were considered acceptable given the strong correlations. The maximum observed error was 143% for the case of 7 samples per hour. [12]

Beyond the prediction of pollen from ragweed, birch, and grass, among the most prevalent and recognized pollen types in Europe, there exists a noteworthy study focused on predicting pollen concentrations of Oleaceae(olive)and Quercus Taxa(pedunculate oak). The study utilized the Gradient Boosting Regression technique to estimate pollen concentrations of both species, using daily meteorological and land surface data obtained from the European Center for Medium-Range Weather Forecasts. The method accurately predicted pollen concentrations, with an Index of Agreement (IoA) of 0.86 for Oleaceae and 0.78 for Quercus, despite the limited size of the dataset.[13]

Drawing from the same data sources as this thesis, namely the Patients Hayfever Diary and Pollen Concentration Data from the EAN database, several computational intelligence methods, such as Multi-layer Perceptron (MLP), Support Vector Regression (SVR), Least Squares Support Vector Regression (LS-SVR), K-Nearest Neighbors (kNN) and Multiple Linear Regression (MLR), are employed to develop personalized models for estimating overall symptoms based on pollen concentrations [14]. The focus was on users with a large amount of data records; the threshold was set to at least 100 data records, thus resulting in a sample of 102 distinct users. The root-mean-square error (RMSE), the correlation coefficient (r) and the index of agreement (d) have been used to validate the models. Results are presented as averages (and standard deviations) of the statistical indices used to evaluate model performance. LS-SVR has the highest d (0.79) and r (0.70), indicating that it generally has the best agreement with the actual data and the highest correlation between predicted and observed values among the models tested. It also has the lowest RMSE (1.92), suggesting it is the most accurate model for predicting the number of symptoms. kNN also performs well, with a relatively high d (0.80) and r (0.67) and a low RMSE (2.07). However, its performance is slightly inferior to LS-SVR. MLP has a moderate performance with d (0.76) and r (0.63) and an RMSE of 2.17. MLR and SVR show similar performance levels with d values of 0.74, r values around 0.61 - 0.64, and RMSE values around 2.11-2.26, indicating that they are less accurate than LS-SVR and kNN.

The etiopathogenesis of allergic diseases is multifactorial, as the immune system of each individual may react differently. Thus, the triggering of allergic symptoms in humans is a highly complex process that depends on several factors such as pollen concentrations, meteorological and chemical (i.e., air quality) weather conditions, people habits (outdoor activity, traveling, medication). Therefore, in some cases, the available data cannot produce accurate models. This was also demonstrated in [14] for the cases of two specific users ((users with ID number 80 and 85), with similar data records and maximum overall symptoms. The results of two user-specific models showed that the allergic symptoms indicated by User85 can be successfully modeled; however, the models were not as successful for User80. It is evident that the pattern of symptoms for both users is different and emphasizes the need for the production of user-specific models. It will also be demonstrated in this thesis that the prediction results for a single user (specifically, the user with the most entries) are superior to those obtained from training and evaluating models on data from all pollen diary users.

2 Materials and Methods

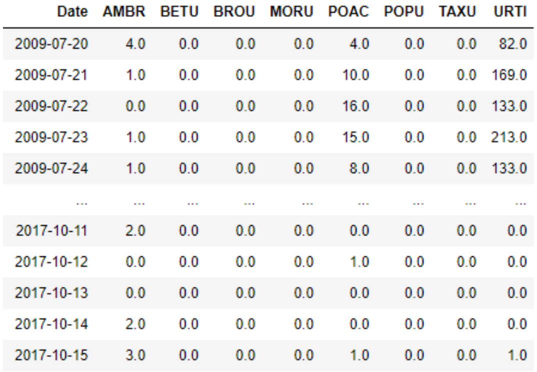
2.1 Data

Data is a critical component in the development and deployment of machine learning models. The quality, quantity and variety of data significantly influence the performance of the models.

2.1.1 Data Source and Data Description

Two sources of data were used in this research: Patients' Hayfever Diary (PHD) and pollen measurements data collected in the Laboratory for Palynology at the Faculty of Sciences, University of Novi Sad. The PHD is a web-based tool for people suffering from pollen allergy and asthma hosted by Austrian Pollen Information Service and the ORL Department at Vienna Medical University [15]. The data utilized in this research are not publicly available. The symptom data from the PHD database was obtained through collaboration with the ORL Department at Vienna Medical University. The pollen data included in this thesis cover the period from 2009 to 2017 for the region of Vojvodina. Also, symptom data corresponding to the same time period and region were extracted from the PHD database.

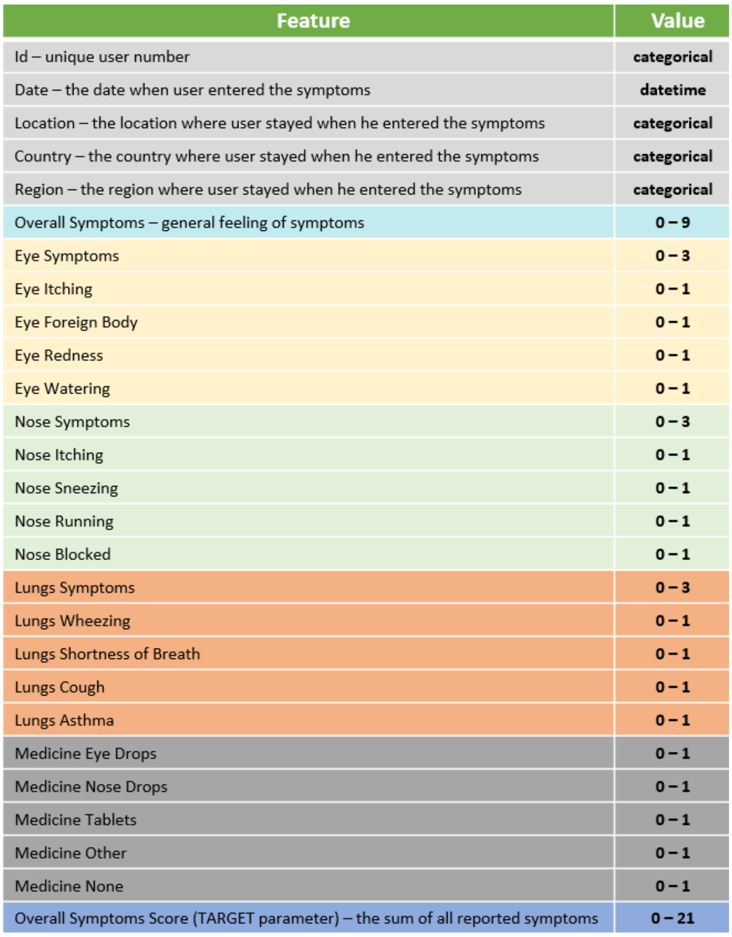
The first dataset ”$PHD - 2009 - 2017.csv$” contains 27 features and 33136 samples. The features represent information about people who use pollen diary application, their symptoms and medical treatments. The second dataset ”$ParticleCountDailyVojvodina2009 - 2017.xlsx$” contains 36 features which represent different pollen species and 3287 samples which represent daily pollen particle count for nine years. An excerpt of that dataset could be found in the Figure 3.



Date	AMBR	BETU	BROU	MORU	POAC	POPU	TAXU	URTI
2009-07-20	4.0	0.0	0.0	0.0	4.0	0.0	0.0	82.0
2009-07-21	1.0	0.0	0.0	0.0	10.0	0.0	0.0	169.0
2009-07-22	0.0	0.0	0.0	0.0	16.0	0.0	0.0	133.0
2009-07-23	1.0	0.0	0.0	0.0	15.0	0.0	0.0	213.0
2009-07-24	1.0	0.0	0.0	0.0	8.0	0.0	0.0	133.0
...	...	...	...	...	...	...	...	...
2017-10-11	2.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2017-10-12	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0
2017-10-13	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2017-10-14	2.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2017-10-15	3.0	0.0	0.0	0.0	1.0	0.0	0.0	1.0

Figure 3: Features of pollen dataset

All features from the PHD dataset and their values are displayed in Figure 4.



Feature	Value
Id – unique user number	categorical
Date – the date when user entered the symptoms	datetime
Location – the location where user stayed when he entered the symptoms	categorical
Country – the country where user stayed when he entered the symptoms	categorical
Region – the region where user stayed when he entered the symptoms	categorical
Overall Symptoms – general feeling of symptoms	0 – 9
Eye Symptoms	0 – 3
Eye Itching	0 – 1
Eye Foreign Body	0 – 1
Eye Redness	0 – 1
Eye Watering	0 – 1
Nose Symptoms	0 – 3
Nose Itching	0 – 1
Nose Sneezing	0 – 1
Nose Running	0 – 1
Nose Blocked	0 – 1
Lungs Symptoms	0 – 3
Lungs Wheezing	0 – 1
Lungs Shortness of Breath	0 – 1
Lungs Cough	0 – 1
Lungs Asthma	0 – 1
Medicine Eye Drops	0 – 1
Medicine Nose Drops	0 – 1
Medicine Tablets	0 – 1
Medicine Other	0 – 1
Medicine None	0 – 1
Overall Symptoms Score (TARGET parameter) – the sum of all reported symptoms	0 – 21

Figure 4: Features of the PHD dataset

2.1.2 Data Analysis

In order to uncover patterns, trends and insights that can inform model development and decision-making data analysis was performed before applying machine learning models. That process includes techniques such as statistical analysis, data visualization and exploratory data analysis to understand the underlying structure and relationships within the datasets.

As a first step, data entries in available PHD dataset for each user were counted and presented as histogram. It is shown in the Figure 5.

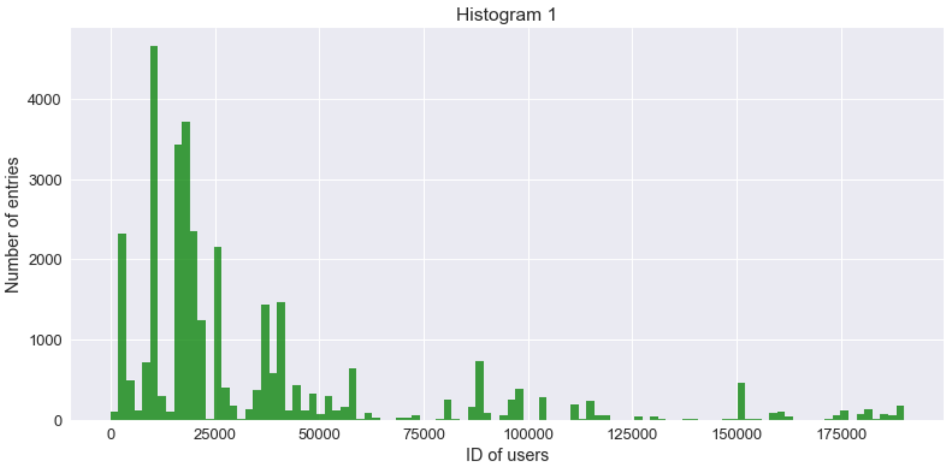


Figure 5: User's data entries

From the Figure 5 we can conclude that there are large oscillations regarding data entries. The pollen diary is not used equally by all users. There are some users who entered their symptoms, usage of the medications and the other information only when they had some symptoms. On the other hand, there are users who filled the diary on the daily level, regardless of whether they had symptoms or not. In order to see how constant diary usage and daily user's information are important for predictions the user with the most entries was found and the models are trained and evaluated on his data.

Additionally, entries per year and unique number of users per year were counted and presented graphically in the following figures.

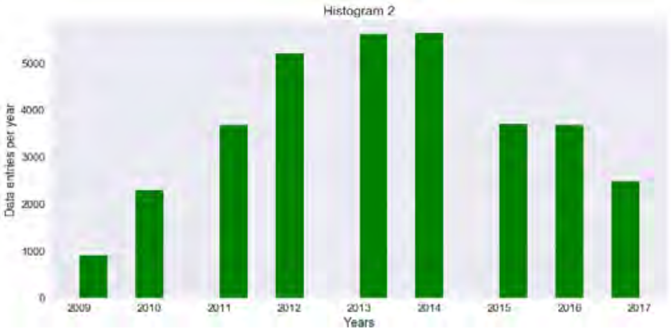


Figure 6: Data entries per year

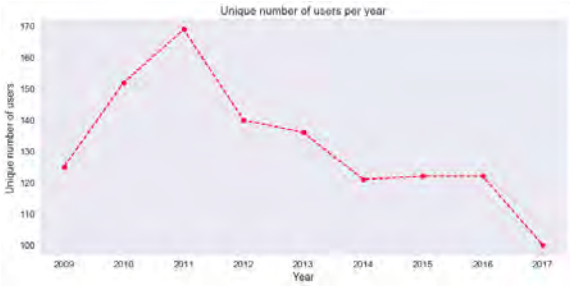
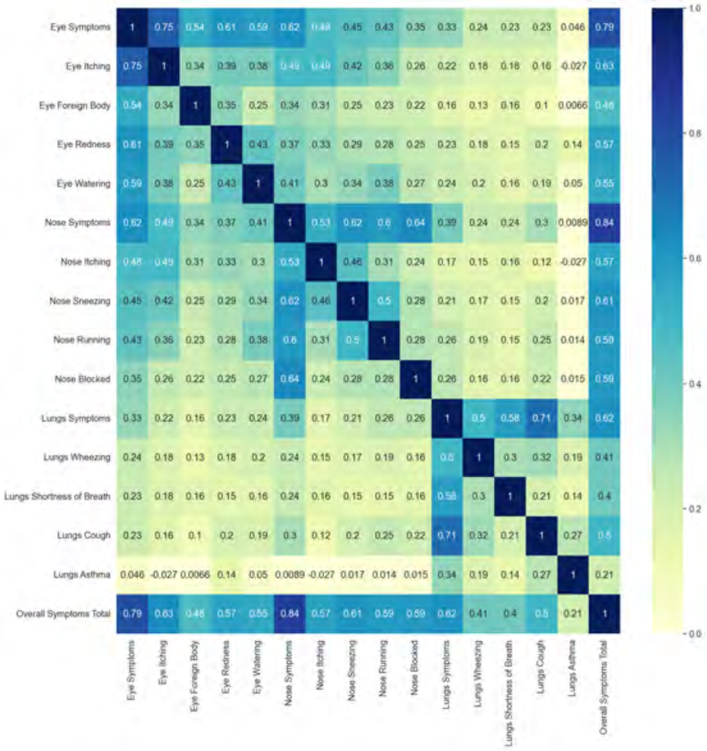


Figure 7: Unique number of users per year

The next step in data analysis was to measure correlations between different symptoms (Eye Itching, Eye Redness, Nose Sneezing, Lungs Cough,etc.) including also Overall Symptom Score. The pairwise correlations are presented with heatmap in the Figure 8, where we can see that overall symptoms are the most correlated with nose symptoms, excluding values on diagonal. It means that the value of overall symptoms can be predicted with high probability(0.84) based on knowledge of the value of nose symptoms.



	Eye Symptoms	Eye Itching	Eye Foreign Body	Eye Redness	Eye Watering	Nose Symptoms	Nose Itching	Nose Sneezing	Nose Running	Nose Blocked	Lungs Symptoms	Lungs Wheezing	Lungs Shortness of Breath	Lungs Cough	Lungs Asthma	Overall Symptoms Total
Eye Symptoms	1	0.75	0.54	0.61	0.59	0.62	0.48	0.45	0.43	0.35	0.33	0.24	0.23	0.23	0.046	0.79
Eye Itching	0.75	1	0.34	0.39	0.38	0.49	0.49	0.42	0.36	0.26	0.22	0.18	0.18	0.16	-0.027	0.63
Eye Foreign Body	0.54	0.34	1	0.35	0.25	0.34	0.31	0.25	0.23	0.22	0.16	0.13	0.16	0.1	0.0066	0.48
Eye Redness	0.61	0.39	0.35	1	0.43	0.37	0.33	0.29	0.28	0.25	0.23	0.18	0.15	0.2	0.14	0.57
Eye Watering	0.59	0.38	0.25	0.43	1	0.41	0.3	0.34	0.38	0.27	0.24	0.2	0.16	0.19	0.05	0.55
Nose Symptoms	0.62	0.49	0.34	0.37	0.41	1	0.53	0.62	0.6	0.64	0.39	0.24	0.24	0.3	0.0089	0.84
Nose Itching	0.48	0.49	0.31	0.33	0.3	0.53	1	0.46	0.31	0.24	0.17	0.15	0.16	0.12	-0.027	0.57
Nose Sneezing	0.45	0.42	0.25	0.29	0.34	0.62	0.46	1	0.5	0.28	0.21	0.17	0.15	0.2	0.017	0.61
Nose Running	0.43	0.36	0.23	0.28	0.38	0.6	0.31	0.5	1	0.28	0.26	0.19	0.15	0.25	0.014	0.59
Nose Blocked	0.35	0.26	0.22	0.25	0.27	0.64	0.24	0.28	0.28	1	0.26	0.16	0.16	0.22	0.015	0.59
Lungs Symptoms	0.33	0.22	0.16	0.23	0.24	0.39	0.17	0.21	0.26	0.26	1	0.5	0.58	0.71	0.34	0.62
Lungs Wheezing	0.24	0.18	0.13	0.18	0.2	0.24	0.15	0.17	0.19	0.16	0.5	1	0.3	0.32	0.19	0.41
Lungs Shortness of Breath	0.23	0.18	0.16	0.15	0.16	0.24	0.16	0.15	0.15	0.16	0.58	0.3	1	0.21	0.14	0.4
Lungs Cough	0.23	0.16	0.1	0.2	0.19	0.3	0.12	0.2	0.25	0.22	0.71	0.32	0.21	1	0.27	0.5
Lungs Asthma	0.046	-0.027	0.0066	0.14	0.05	0.0089	-0.027	0.017	0.014	0.015	0.34	0.19	0.14	0.27	1	0.21
Overall Symptoms Total	0.79	0.63	0.48	0.57	0.55	0.84	0.57	0.61	0.59	0.59	0.62	0.41	0.4	0.5	0.21	1

Figure 8: Correlation between symptoms

In order to have a clear visual summary of how symptoms vary throughout the year, highlighting differences in central tendency, spread and the presence of outliers, overall symptoms on monthly level are presented with boxplot which is shown in the Figure 9. We can see that the distribution of symptoms varies across the months. For instance, months like August and September have higher medians and a broader spread of symptom totals compared to other months. They have also more outliers indicating more variability or extreme values.

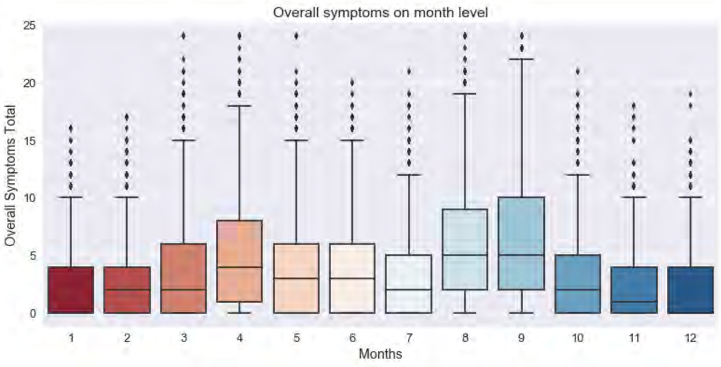


Figure 9: Overall Symptoms on the monthly level

In the following figure, counts of different medical treatments are presented with bar graph, where we can see that nose drops are the most used medicine for relieving the symptoms.

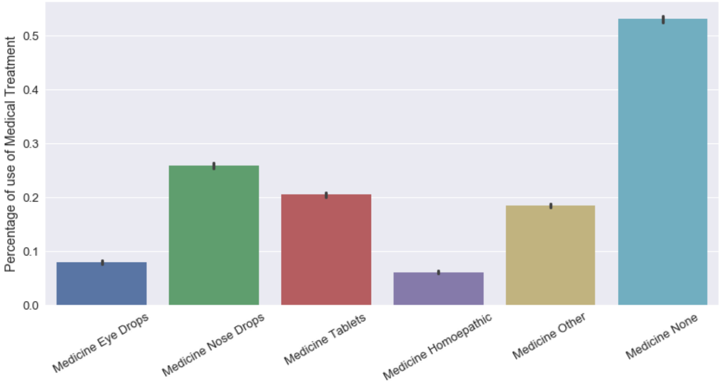


Figure 10: Percentage of use of Medical Treatment

The analysis of the second dataset was started with the presenting overall sums of daily measurements of different pollen species over the years, from 2009 to 2017. It is displayed in Figure 11, from which it can be concluded that the highest overall concentration of pollen is in the spring, and then in the fall where the peek is significantly smaller. A high overall pollen count does not always indicate a strong concentration of the specific pollen to which person is allergic so for this research it is important to find a correlation between pollen concentrations and user's symptoms, which will be done in some of the following data analysis steps.

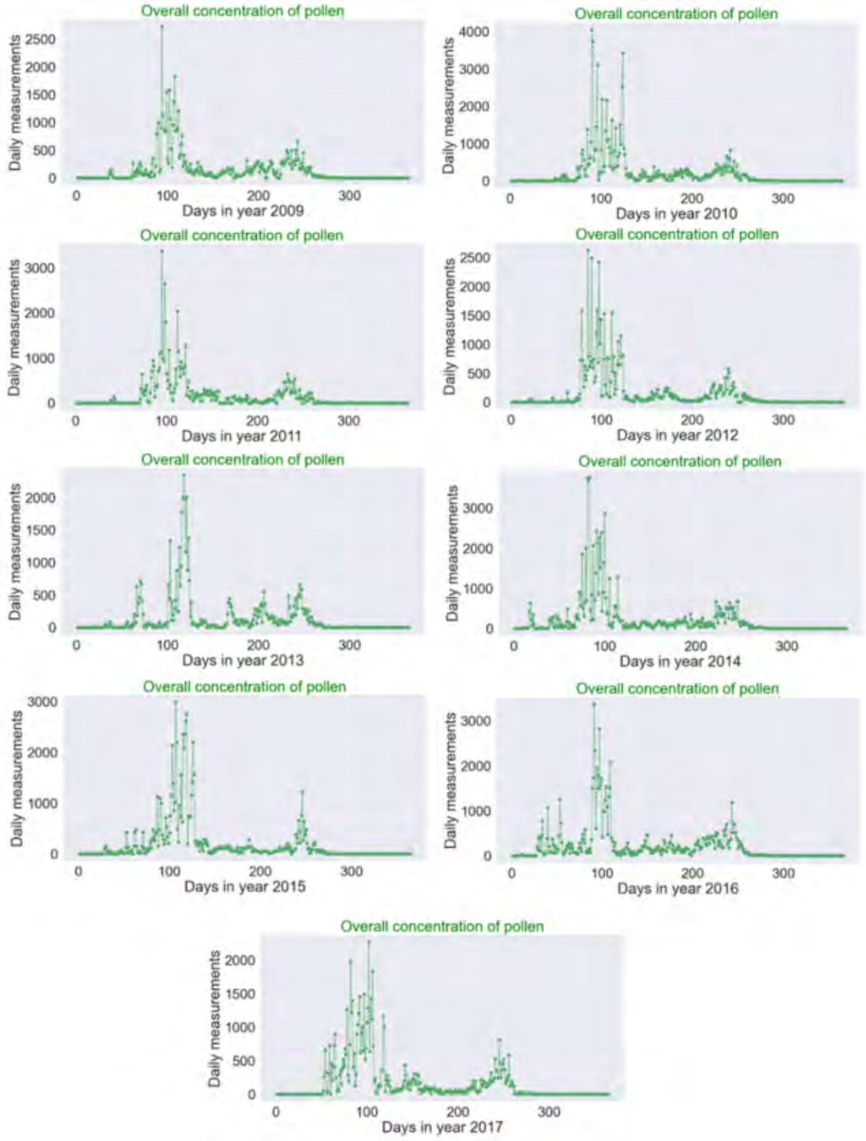


Figure 11: Overall pollen concentration over the years

The remaining data analysis tasks essential for this thesis focus on examining the relationship between the two datasets. Specifically, the connection between the data from PHD users and the concentrations of various pollen species will be analyzed. To identify correlations between symptoms and pollen concentrations, the top ten users based on the number of their records in the database were selected. For these users, their overall symptoms data alongside measurements of pollen concentrations were presented. For each of the top ten users, we also identified the year in which they used the PHD diary most frequently and graphically represented their symptoms and overall pollen concentrations for that year, as shown in Figure 12. In these graphics, symptoms are depicted in blue, while overall pollen concentrations are shown in green. From these ten graphics, we can observe that for User 1 (top left graphic), the peak in overall pollen concentration coincides with the peak in user-reported symptoms, both occurring in March and April. Additionally, it is evident that most of the top ten users experience the most intense symptoms in autumn. This observation highlights the importance of including pollen species that are predominant in the autumn (such as Ambrosia, represented by AMBR) as features in the training and validation of machine learning models for predicting symptoms.

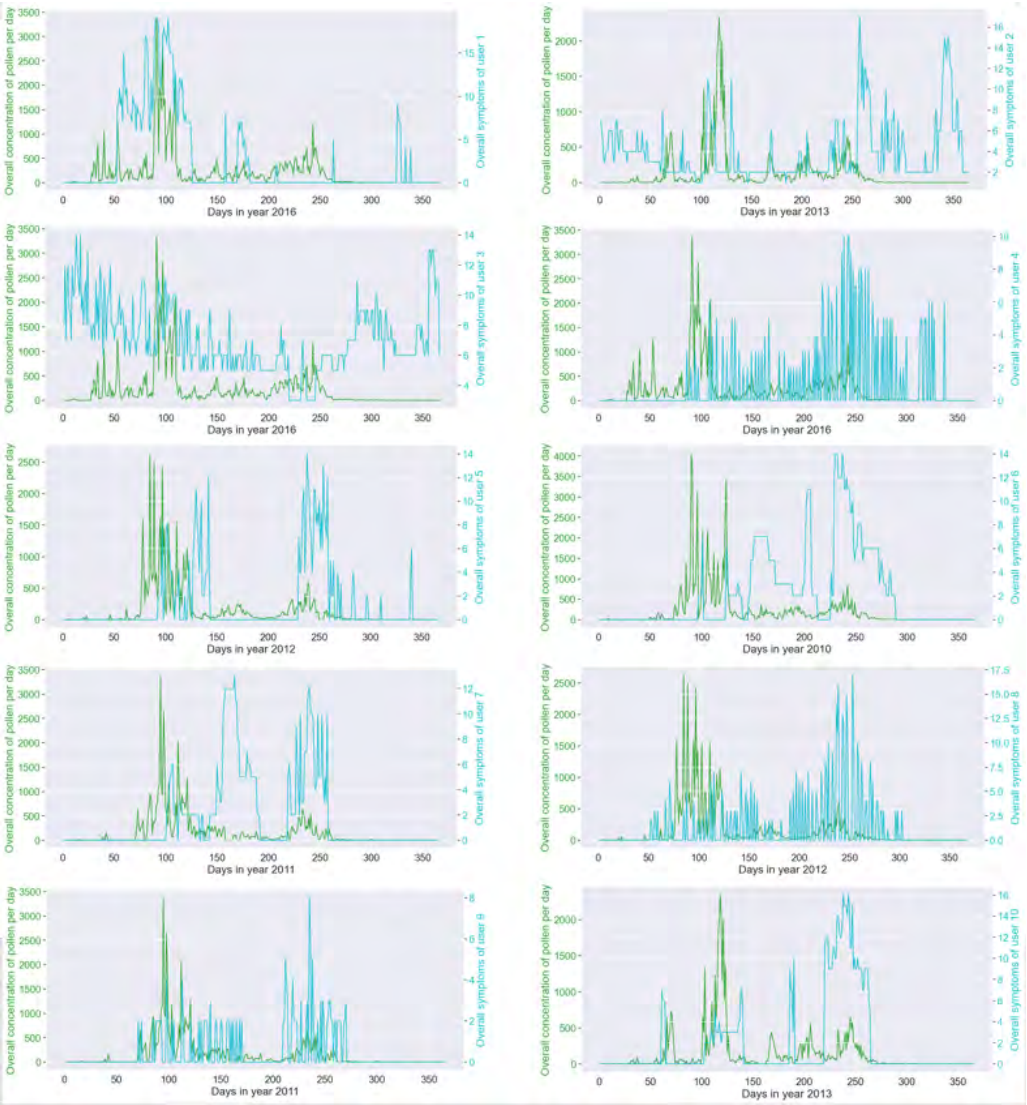


Figure 12: Overall symptoms data(blue graph) along pollen concentration measurements(green graph) for the top 10 users

2.1.3 Data Pre-processing

Data pre-processing is a critical step in the machine learning pipeline that involves improving data quality, selecting the features for the models, eliminating data issues such as missing values and make the data useful for machine learning purposes. The scheme of the machine learning process used in this research can be found in Figure 13.

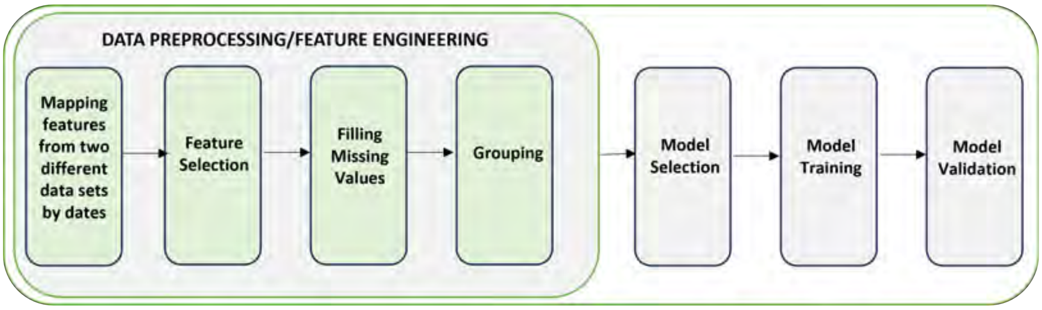

DATA PREPROCESSING/FEATURE ENGINEERING
Mapping features from two different data sets by dates
Feature Selection
Filling Missing Values
Grouping
Model Selection
Model Training
Model Validation

Figure 13: Machine learning pipeline of the implementation

• **Mapping features from two different data sets by dates.** In this step, the dates from the two data sets were firstly converted in the same format using library datetime and after that they are sorted properly. Since there is data for pollen concentration measurements for each day from year 2009 to 2017, but on the other hand there are no symptoms recorded by PHD diary user's for all of those days we only selected samples from pollen concentrations data set for which we have symptoms.

• **Feature Selection.** Feature selection is a very important step in machine learning because it can hugely impact the performance of the model. Features used for training the machine learning model have a huge influence on the performance the model can achieve, so in this phase we wanted to find the features i.e. the pollen species that are the most correlated with the symptoms. To find those features the Pearson's correlation coefficient was the first try in this implementation. The Pearson's correlation coefficient measures the strength of the linear relationship between two variables. It is calculated as the covariance of the two variables divided by the product of the standard deviation of each data sample. It is a normalized measurement of the covariance. Because of that, the value of the Pearson's correlation coefficient is always between -1 and 1 representing the limits of correlation from a full negative correlation to a full positive correlation. [16] The formula for calculating Pearson's correlation coefficient is:

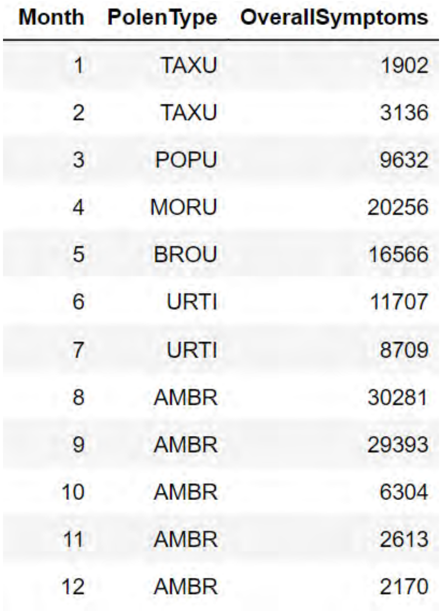
$$r = \frac{\sum(X-\bar{X})(Y-\bar{Y})}{\sqrt{\sum(X-\bar{X})^2(Y-\bar{Y})^2}}$$

After calculating Pearson's correlation coefficient between different pollen species and symptoms, the highest coefficient value, r = 0.27, was obtained for Ambrosia. For other pollen species, the coefficient was mostly positive, but significantly lower than coefficient for Ambrosia. A Pearson correlation of 0.27 suggests a weak linear relationship but does not rule out the possibility of a strong non-linear relationship. In such cases, using Spearman's rank correlation might be more appropriate, so it was used as another means to find pollen species that causes the most symptoms.
The Spearman's correlation coefficient measures the strength of the relationship between two variables which does not have to be strictly linear. While pearsons's correlation calculates the coefficient using covariance and standard deviations, spearman's correlation calculates the same but using rank variables. Instead of raw data x and y, spearman's correlation coefficient uses ranked data r_x and r_y:

$$\rho_{r_x,r_y} = \frac{cov(r_x, r_y)}{\sigma_{rx}\sigma_{ry}}$$

where $cov(r_x, r_y)$ is the of ranked data r_x and r_y, while σ_{rx} and σ_{ry} are the standard deviations of r_x and r_y. For all these values of standard deviation, centered and ranked values, covariance, and correlation, we use functions from the scipy library.[17] After calculating Spearman's correlation coefficient between different pollen species and symptoms, the highest coefficient value, $\rho = 0.19$ and p-value = 0, was obtained again for Ambrosia. A Spearman correlation coefficient of 0.19 with a p-value of 0 suggests a weak but statistically significant monotonic relationship between the variables. While the correlation is weak, the statistical significance underscores the presence of a genuine relationship in the data, meriting further investigation or contextual analysis.

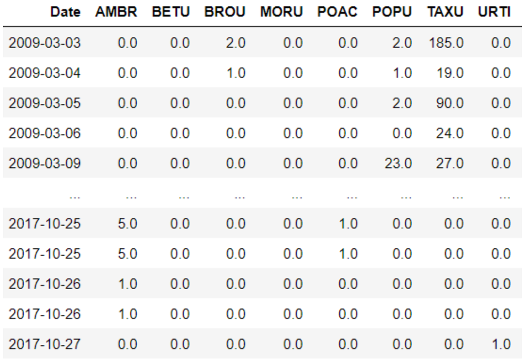
After the previous analyses, we can conclude that Ambrosia concentrations should be definitely included in the features combination for model training, but additional investigation is needed in order to select other features as well, so the next step was to find the pollen species with the highest concentration for a given month. Figure 14 shows the most common type of pollen for a given month and the overall symptoms for that month.



Month	PolenType	OverallSymptoms
1	TAXU	1902
2	TAXU	3136
3	POPU	9632
4	MORU	20256
5	BROU	16566
6	URTI	11707
7	URTI	8709
8	AMBR	30281
9	AMBR	29393
10	AMBR	6304
11	AMBR	2613
12	AMBR	2170

Figure 14: The most common pollen species and overall symptoms for a given month

Finally, the pollen species that were most common for a certain month and two more well-known types of pollen, grass (POAC - Poaceae) and birch (BETU - Betula), will be used as a feature combination for training the machine learning models, which can be found in Figure 15 .

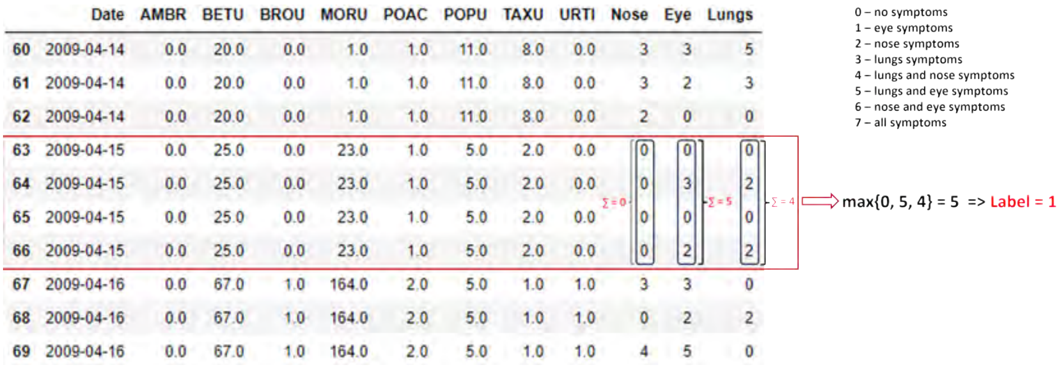


Date	AMBR	BETU	BROU	MORU	POAC	POPU	TAXU	URTI
2009-03-03	0.0	0.0	2.0	0.0	0.0	2.0	185.0	0.0
2009-03-04	0.0	0.0	1.0	0.0	0.0	1.0	19.0	0.0
2009-03-05	0.0	0.0	0.0	0.0	0.0	2.0	90.0	0.0
2009-03-06	0.0	0.0	0.0	0.0	0.0	0.0	24.0	0.0
2009-03-09	0.0	0.0	0.0	0.0	0.0	23.0	27.0	0.0
...	...	...	...	...	...	...	...	...
2017-10-25	5.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0
2017-10-25	5.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0
2017-10-26	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2017-10-26	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2017-10-27	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0

Figure 15: Selected Features

• **Filling the Missing Values.** The missing values were filled with zeros in this implementation. Most of the NaN values were in December and some of them were in January when pollen concentration measurements are usually 0.

• **Grouping.** There are the samples with the same dates and the same pollen concentration measurements but the different symptoms entered by different pollen diary user's. It is the main reason why the models can get confused when making predictions. In order to see how multiple same inputs with a lot of different targets affect model performance, we evaluate the models on both multiple inputs(same dates and the same pollen concentrations) and the input grouped in the way explained in the following figure:



	Date	AMBR	BETU	BROU	MORU	POAC	POPU	TAXU	URTI	Nose	Eye	Lungs
60	2009-04-14	0.0	20.0	0.0	1.0	1.0	11.0	8.0	0.0	3	3	5
61	2009-04-14	0.0	20.0	0.0	1.0	1.0	11.0	8.0	0.0	3	2	3
62	2009-04-14	0.0	20.0	0.0	1.0	1.0	11.0	8.0	0.0	2	0	0
63	2009-04-15	0.0	25.0	0.0	23.0	1.0	5.0	2.0	0.0	0	0	0
64	2009-04-15	0.0	25.0	0.0	23.0	1.0	5.0	2.0	0.0	0	3	2
65	2009-04-15	0.0	25.0	0.0	23.0	1.0	5.0	2.0	0.0	0	0	0
66	2009-04-15	0.0	25.0	0.0	23.0	1.0	5.0	2.0	0.0	0	2	2
67	2009-04-16	0.0	67.0	1.0	164.0	2.0	5.0	1.0	1.0	3	3	0
68	2009-04-16	0.0	67.0	1.0	164.0	2.0	5.0	1.0	1.0	0	2	2
69	2009-04-16	0.0	67.0	1.0	164.0	2.0	5.0	1.0	1.0	4	5	0

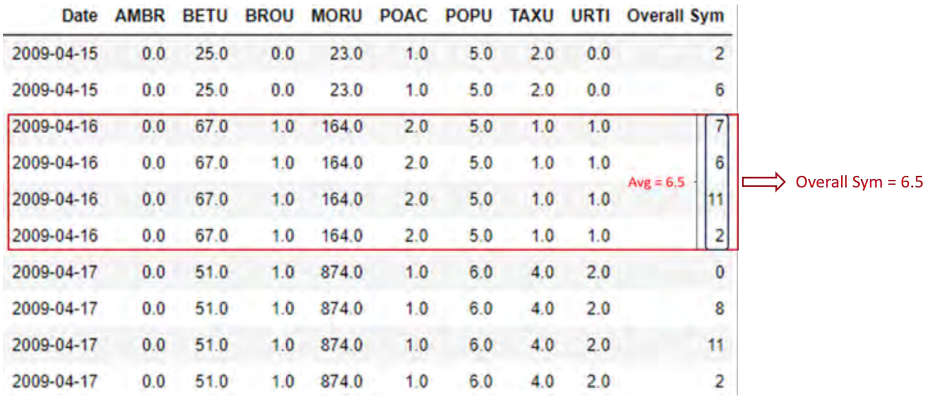
0 – no symptoms
1 – eye symptoms
2 – nose symptoms
3 – lungs symptoms
4 – lungs and nose symptoms
5 – lungs and eye symptoms
6 – nose and eye symptoms
7 – all symptoms

Figure 16: Grouping samples by most intense type of symptoms

Figure 16 explains how the samples are grouped for classification problem where we need to predict one of the eight labels:

- 0 - no symptoms,
- 1 - eye symptoms,
- 2 - nose symptoms,
- 3 - lungs symptoms,
- 4 - lungs and nose symptoms,
- 5 - lungs and eye symptoms,
- 6 - nose and eye symptoms,
- 7 - all symptoms.

Grouping for regression problem where we want to predict the intensity of overall symptoms is shown in the Figure 17.



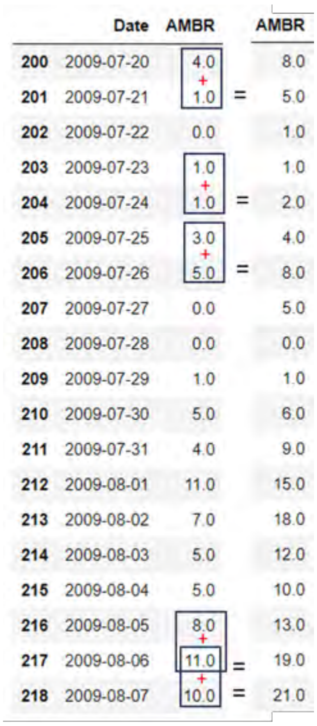
Date	AMBR	BETU	BROU	MORU	POAC	POPU	TAXU	URTI	Overall Sym
2009-04-15	0.0	25.0	0.0	23.0	1.0	5.0	2.0	0.0	2
2009-04-15	0.0	25.0	0.0	23.0	1.0	5.0	2.0	0.0	6
2009-04-16	0.0	67.0	1.0	164.0	2.0	5.0	1.0	1.0	7
2009-04-16	0.0	67.0	1.0	164.0	2.0	5.0	1.0	1.0	6
2009-04-16	0.0	67.0	1.0	164.0	2.0	5.0	1.0	1.0	11
2009-04-16	0.0	67.0	1.0	164.0	2.0	5.0	1.0	1.0	2
2009-04-17	0.0	51.0	1.0	874.0	1.0	6.0	4.0	2.0	0
2009-04-17	0.0	51.0	1.0	874.0	1.0	6.0	4.0	2.0	8
2009-04-17	0.0	51.0	1.0	874.0	1.0	6.0	4.0	2.0	11
2009-04-17	0.0	51.0	1.0	874.0	1.0	6.0	4.0	2.0	2

Figure 17: Grouping samples by average of overall symptoms

2.2 Methods

The thesis evaluates the performance of two approaches for the symptoms prediction: prediction of the intensity of overall symptoms (0-21) and prediction of the type of symptoms (eyes, nose, lungs symptoms or some of the combinations of those symptoms). For the first approach, four regression models were evaluated: K-Nearest Neighbors Regressor, Random Forest Regressor, Gradient Boosting Regressor and XGB Regressor and for the second one five classification models were evaluated: K-Nearest Neighbors Classifier, Random Forest Classifier, Support Vector Machine Classifier, Gradient Boosting Classifier and XGB Classifier. All mentioned models were trained for both daily and cumulative pollen concentrations - cumulative n = 2, n = 3, n = 4, n = 5, n = 10 and n = 15, where n is the number of days.

The functions for summing current pollen concentrations with concentrations for previous n days, where different values of n were used: cumulative n = 2, n = 3, n = 4, n = 5, n = 10 and n = 15, were implemented. As an example visual representation of function for cumulative n = 2 is showed in the following figure:



	Date	AMBR	AMBR
200	2009-07-20	4.0	8.0
201	2009-07-21	1.0	5.0
202	2009-07-22	0.0	1.0
203	2009-07-23	1.0	1.0
204	2009-07-24	1.0	2.0
205	2009-07-25	3.0	4.0
206	2009-07-26	5.0	8.0
207	2009-07-27	0.0	5.0
208	2009-07-28	0.0	0.0
209	2009-07-29	1.0	1.0
210	2009-07-30	5.0	6.0
211	2009-07-31	4.0	9.0
212	2009-08-01	11.0	15.0
213	2009-08-02	7.0	18.0
214	2009-08-03	5.0	12.0
215	2009-08-04	5.0	10.0
216	2009-08-05	8.0	13.0
217	2009-08-06	11.0	19.0
218	2009-08-07	10.0	21.0

Figure 18: Cumulative concentration of Ambrosia for n = 2

The other functions work on the same principle, only for a higher value of the parameter n which indicates a number of days.

The implementation of this thesis is coded in Python programming language, using Jupyter Notebook. A Jupyter Notebook is an open source web application used for all sorts of data science tasks such as exploratory data analysis (EDA), data cleaning and transformation, data visualization, statistical modeling, machine learning and deep learning. The most important used libraries are Scikit-learn - a simple and efficient tool used in predictive data analytics, Pandas - a library used for analyzing, cleaning and manipulating data, Matplotlib - a tool used for performing different visualizations and Datetime - a library used for manipulating dates and mapping two data sets by dates as one of the the most important step in this research.

2.2.1 K-Nearest Neighbors Algorithm

K-Nearest Neighbors (KNN) is a non-parametric, simple yet powerful supervised algorithm that can be used for both regression and classification tasks. This works by finding K nearest neighbors to the new, unlabeled data and making a prediction of the value or class that the new data point belongs to.

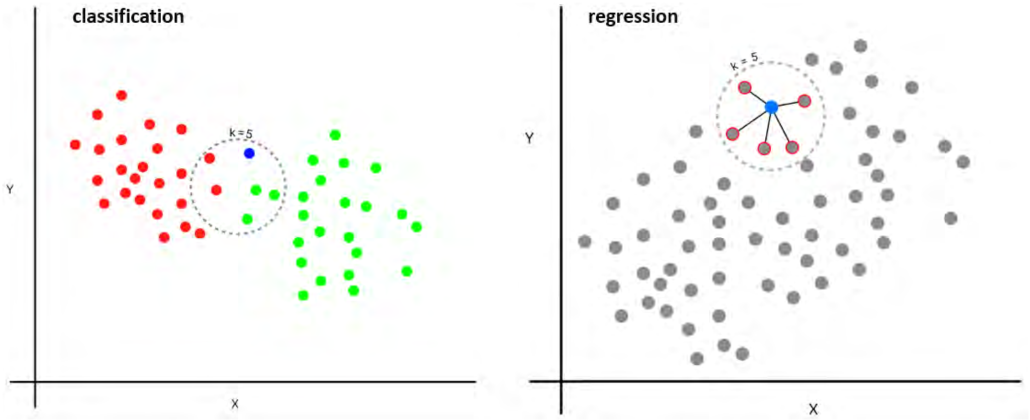


Figure 19: KNN Algorithm working visualization for classification and regression

Visually observing classification from Figure 19, there are two classes, red and green. When there is a new data point (blue), and K = 5, we can see that the blue point has 3 green neighbors and 2 red neighbors; this says that the blue point is classified as the green class as the majority voting is 3. Similarly, when the K value changes, the number of neighbors increases, and the new data point is classified into its corresponding majority voting class. KNN regressor is quite different from the classifier. As in a regressor, the dependent variable is continuous, it is scattered throughout the coordinate plane. When there is a new data point, the number of neighbors (K) is found by any of the distance metrics. After finding the neighbors, the predicted value of the new data point is the average of all the neighbor's values combined. [18]

Step-by-Step explanation of how KNN works is discussed below:

Step 1: Calculating distance

In order to measure the similarity between target and training data points the first step is to calculate the distance between the new point and each training point. There are various methods for calculating this distance, of which the most commonly known methods are – Euclidean, Manhattan (for continuous), Minkowski and Hamming distance (for categorical).

Euclidean distance is the most widely used distance metric in KNN, and this is the default distance metric for SKlearn library in Python. This is the straight line distance between two data points in the Euclidean space, calculated with the square root of the sum of squares of data points. The formula for Euclidean distance between two points $\mathbf{x} = (x_1, x_2, \ldots, x_n)$ and

$\mathbf{y} = (y_1, y_2, \ldots, y_n)$ is given by:

$$d = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$$

Manhattan distance, also known as taxicab distance, is a measure of the distance between two points. It is named after a grid-like layout of Manhattan, where the distance between the two points is the shortest path a taxi could take. It is the sum of the absolute difference of the coordinates. The formula for Manhattan distance between two points $\mathbf{x} = (x_1, x_2, \ldots, x_n)$ and $\mathbf{y} = (y_1, y_2, \ldots, y_n)$ is calculated as follows,

$$d = \sum_{i=1}^{n} |x_i - y_i|$$

Minkowski distance is used to measure the distance between the points in multidimensional space. This metric generalizes the Manhattan and Euclidean distance metrics. This is computed as the pth root of the sum of absolute difference raised to the power of p. We can manipulate the above formula to give us different distance metrics like:

- if p = 1, we get the Manhattan distance
- if p = 2, we get Euclidean distance.

There are a few conditions that the distance metric must satisfy:

- Non-negativity: The distance between any two points cannot be negative.
- Identity: The distance between a point and itself is zero.
- Symmetry: The distance between two points x and y should be the same as the distance between y and x.
- Triangle Inequality: The distance between two points x and y should always be less than or equal to the sum of the distances between x and y, and between y and z. [18]

The formula for Minkowski distance between two points $\mathbf{x} = (x_1, x_2, \ldots, x_n)$ and $\mathbf{y} = (y_1, y_2, \ldots, y_n)$ is,

$$d = \|\mathbf{x} - \mathbf{y}\|_p = \left(\sum_{i=1}^{n} |x_i - y_i|^p \right)^{\frac{1}{p}}$$

Hamming distance is used for categorical variables. The Hamming distance between two strings of equal length is the number of positions at which the corresponding symbols are different. For two points $\mathbf{x} = (x_1, x_2, \ldots, x_n)$ and $\mathbf{y} = (y_1, y_2, \ldots, y_n)$ is given by:

$$d_H(\mathbf{x}, \mathbf{x}) = \sum_{i=1}^{n} \delta(x_i, x_i)$$

where

$$\delta(x_i, y_i) = \begin{cases} 1, & \text{if } x_i \neq y_i \\ 0, & \text{if } x_i = y_i \end{cases}$$

Step 2: Selecting the optimal value of K
K represents the number of nearest neighbors that needs to be considered while making prediction. It is important to note that the choice of the k value depends on the data set and the problem. A smaller k value can lead to overfitting, while a larger value of k can lead to underfitting. Therefore, it is recommended to experiment with different values of k to find the optimal value for a specific data set.
Step 3: Finding Nearest Neighbors
The k data points with the smallest distances to the target point are the nearest neighbors.
Step 4: Voting for Classification or Taking Average for Regression
In the classification problem, the class labels of K-nearest neighbors are determined by performing majority voting. The class with the most occurrences among the neighbors becomes the predicted class for the target data point. In the regression problem, the class label is calculated by taking average of the target values of K nearest neighbors. The calculated average value becomes the predicted output for the target data point.

2.2.2 Random Forest Algorithm

A Random Forest is an ensemble technique for both regression and classification with the use of multiple decision trees using bootstrap and aggregation. Random forest consists of a large number of decision trees operating as an ensemble. The ensemble of learner is built using the same learning algorithm but train each learner on a different randomly chosen data sets. That is bootstrap. In the case of a classification problem, the final output is chosen as majority voting classifier. On the other hand, for regression problem, the final output is the mean of all the outputs. That technique is called aggregation. Bootstrap and aggregation are known as bootstrap aggregating or bagging. The working scheme of random forest regression can be found in Figure 20.[19][20]

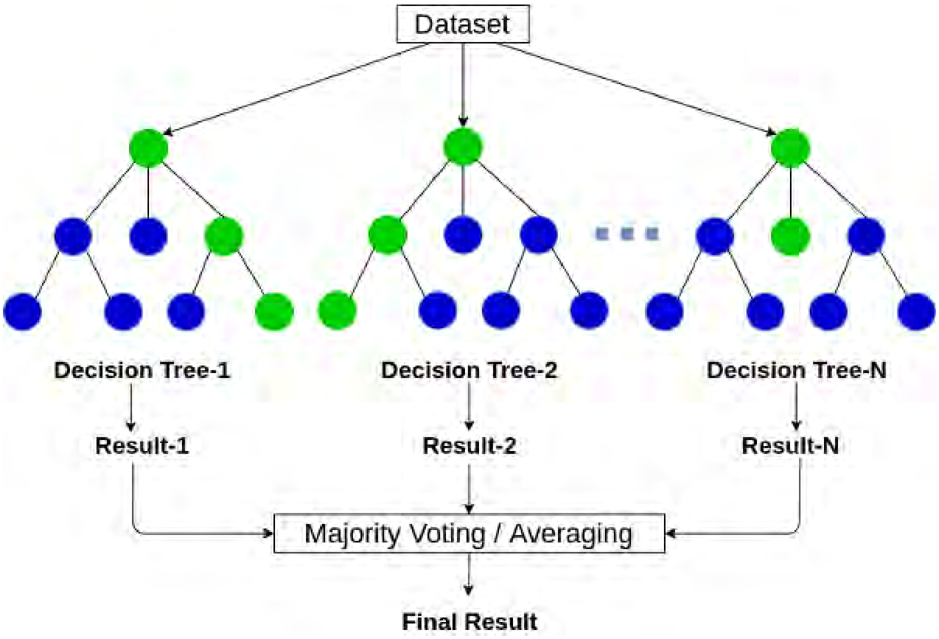


Figure 20: Random Forest Algorithm

Source: Random Forest Algorithm Explained [21]

2.2.3 Support Vector Machine Algorithm

Support vector machines are a set of supervised learning methods used for both, classification and regression. It is one of the most popular machine learning algorithms and very powerful. The objective of the support vector machine algorithm is to find a hyperplane in an n-dimensional space, where n is the number of features, that distinctly classifies the data points. In Figure 21, two classes of data points can be seen, blue circles, and orange triangles. There can be seen many possible hyperplanes that can be chosen to separate two classes of data points. These possible hyperplanes are presented with green lines. Hyperplanes are decision boundaries that are used to predict the continuous output. The goal is to find a hyperplane with a maximum margin which represent the maximum distance between data points of both classes. What would be the maximum margin and which hyperplane is optimal is shown in Figure 21. Support vectors are data points that are closer to the hyperplane and condition the position and orientation of the hyperplane. They are represented in the right picture with a colored circle and squares. Also, they are important for building support vector machines because they maximize the margin of the classifiers.[22][23]

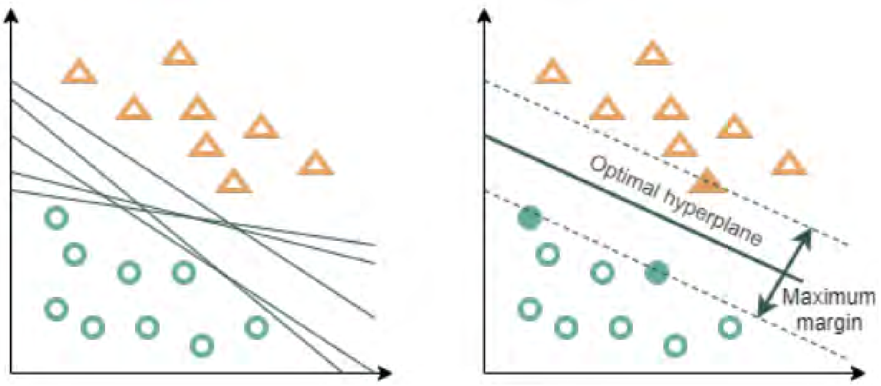


Figure 21: Possible hyperplanes

Source: Introduction to Support Vector Machines [24]

The number of features conditions the dimension of the hyperplane. So if the number of features is two, there is 2D space and the hyperplane is a line, while in 3D space there is a two-dimensional plane for a hyperplane. The examples of hyperplanes for both spaces are in Figure 22.

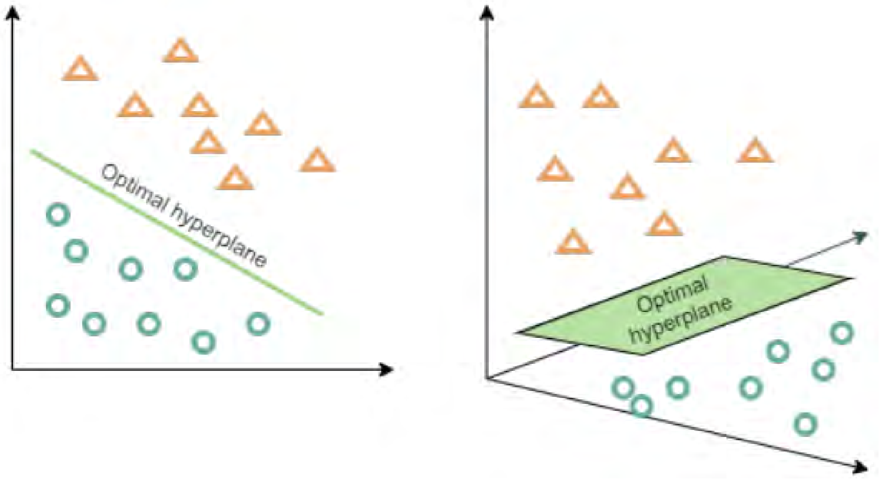


Figure 22: Hyperplanes in 2D and 3D feature space

Source: Introduction to Support Vector Machines [24]

2.2.4 Gradient Boosting Algorithm

Gradient Boosting is an supervised machine learning algorithm used for classification and regression problems. It is an ensemble technique which uses multiple weak learners to produce a strong model for regression and classification. The algorithm relies on the intuition that the best possible next model , when combined with the previous models, minimizes the overall prediction errors. The key idea is to set the target outcomes from the previous models to the next model in order to minimize the errors.
Step-by-Step explanation of how Gradient Boosting Algorithm works is discussed below:
Step 1: Initialize model with the constant value $\mathbf{F_0}$:

$$F_0(x) = \arg\min_{\gamma} \sum_{i=1}^{N} L(y_i, \gamma) \tag{1}$$

Where $L(y_i, \gamma)$ is the loss function, and γ is the initial prediction.

Step 2: For each iteration m $= 1, 2, \ldots, M$
• **Compute the Pseudo-Residuals**:

$$r_i^{(m)} = -\left[\frac{\partial L(y_i, F(x_i))}{\partial F(x_i)}\right]_{F(x)=F_{m-1}(x)} \tag{2}$$

Here, $r_i^{(m)}$ is the pseudo-residual for instance i at iteration m.

• **Fit a Weak Learner**:
Train a weak learner $h_m(x)$ to the pseudo-residuals:

$$h_m(x) = \arg\min_{h} \sum_{i=1}^{N} \left(r_i^{(m)} - h(x_i)\right)^2 \tag{3}$$

• **Compute the Step Size:**
Find the optimal step size γ_m:

$$\gamma_m = \arg\min_{\gamma} \sum_{i=1}^{N} L(y_i, F_{m-1}(x_i) + \gamma h_m(x_i)) \tag{4}$$

• **Update the Model:**
Update the model with the new learner and step size:

$$F_m(x) = F_{m-1}(x) + \gamma_m h_m(x) \tag{5}$$

Step 3: The Final Model:

$$F_M(x) = \sum_{m=0}^{M} \gamma_m h_m(x) \tag{6}$$

In conclusion, Gradient Boosting is a highly effective and versatile algorithm that excels in capturing complex relationships within data. Its ability to iteratively improve model performance through sequential learning makes it a preferred choice for many machine learning practitioners. However, the trade-off between accuracy and computational complexity requires careful consideration and expertise in hyperparameter tuning. [25] [26] [27]

2.2.5 XGBoost Algorithm

XgBoost or eXtreme Gradient Boosting is a decision tree-based machine learning algorithm which is using gradient boosting. The impact of XGboost has been recognized in many machine learning and data mining challenges. One of them is machine learning competition site Kaggle, where in 2015 among the 29 challenge winning solutions, 17 of them were using XGboost. This machine learning algorithm can be used for both regression and classification. Elements of gradient boosting are loss function which needs to be optimized, a weak learner which needs to make predictions, and an additive model which need to add weak learners to minimize the loss function. A loss function is used to evaluate how well we can predict the value. It must be differentiable. Usually, different loss functions are used for regression and for classification. For loss function for classification may be used logarithmic loss function, while for regression is most common used squared error. In gradient boosting decision trees are used as the weak learner. When gradient boosting is used for regression we start with a leaf that is the average value of the variable we want to predict. Regarding additive model, we add a tree based on the residuals, the difference be-tween the observed values and the predicted values and we scale the tree's contribution to the final prediction with a learning rate. The learning rate is usually a value between 0 and 1. We use learning rate to avoid overfitting of the model. Then we add another tree based on the new residuals and we keep adding trees based on the errors made by the previous tree. Gradient boost continues to build trees in this fashion until it has made the number of trees we asked for, or additional trees fail to improve the fit.[28][29]

2.2.6 Grid Search

Grid Search acts as a valuable tool for identifying the optimal parameters for a machine learning model. Instead of manually testing various combinations of parameters, Grid Search systematically explores a predefined set of parameter values, effectively creating a grid of possible configurations. By evaluating the model's performance across the grid, Grid Search helps identify the best parameter combination that optimizes the model's performance, making the tuning process much more efficient and less prone to human error.
Step-by-Step explanation of how Grid Search works is discussed below:

Step 1: Define a hyperparameter grid:
A hyperparameter grid is defined using a python dictionary which contains configuration for the model we are targeting for tuning. As an example SVM algorithm used for predicting the type of the symptoms will be considered. SVM takes three parameters named **C**, **kernel** and **gamma**, so in this implementation we defined nine different values of C (*'C' : [0.001, 0.01, 0.1, 0.5, 1, 5, 10, 100, 1000]*), three different values of kernel (*'kernel': ['linear', 'rbf', 'poly']*) and seven different values of gamma (*'gamma' : [0.001, 0.01, 0.1, 0.5, 1, 10, 100]*) in an array. Of course, we could define more if we wanted. C is known as the regularization parameter or the cost parameter, which controls the trade-off between maximizing the margin (distance between the decision boundary and the data points) and minimizing the classification error on the training data. Kernel is used to specify the type of kernel function to be used when transforming the input data into a higher-dimensional space. The parameter gamma plays a crucial role in defining the behavior of the decision boundary. It can be seen as the inverse of the radius of influence of samples selected by the model as support vectors. Intuitively, a low gamma value means that the influence of a single training example reaches far, affecting a larger region of the feature space. Conversely, a high gamma value means that the influence is close, affecting only the region near the training example.
Step 2: Model Training and Evaluation:
Grid Search typically uses cross-validation to evaluate model performance. We split our data set into multiple subsets (folds), trained the model on some of them, and evaluated it on others. This helps ensure that the model's performance is robust and not just tailored to the training data. Grid Search accepts several arguments

$$gs = GridSearchCV(SVC(), parameters, cv = 10, scoring =' accuracy').$$

The first argument is the model which we want to evaluate. The second argument is the grid configuration we made earlier using python dictionary. The **cv** argument accepts integers and represents number of folds for K-fold cross-validation. K-fold cross-validation (Figure 23) is an iterative process that divides the train data into k partitions. Each iteration keeps one partition for testing and the remaining k-1 partitions for training the model. The next iteration will set the next partition as test data and the remaining k-1 as train data and so on. In each iteration, it will record the performance of the model and at the end give the average of all the performance. Thus, it is also a time-consuming process.

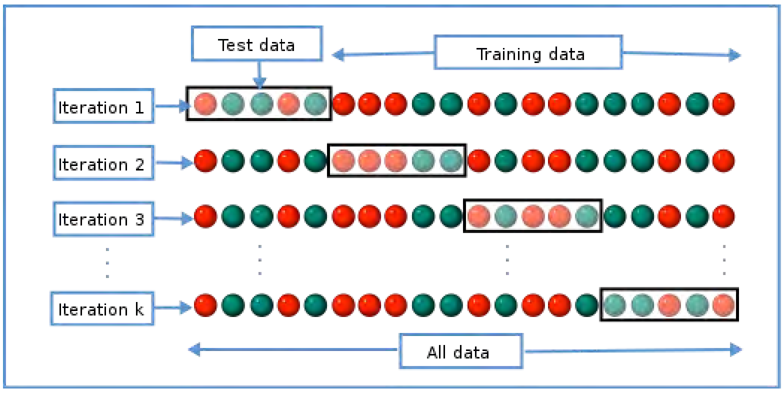


Figure 23: K-fold cross validation

Scoring represents the strategy employed to evaluate the model. After the splitting the data into train and test set GridSearchCV() object can be fitted:

$$gs.fit(X\_train, y\_train).$$

Step 3: Get the best scores:
This step requires to access the *best_params_* attribute from the processed GridSearchCV object

$$best\_params = gs.best\_params\_$$

$$best\_score = gs.best\_score\_.$$

The *best_score_* attribute gives us a float value which represent the best accuracy score in this scenario.

Step 4: Final model training:
In this step the model was trained again, but this time using the parameter values which got the highest scores.

$$\begin{aligned} final\_model = SVC(C = best\_params['C'], \\ kernel = best\_params['kernel'], \\ gamma = best\_params['gamma']) \end{aligned}$$

$$final\_model.fit(X\_train, y\_train)$$

Finally, we can conclude that by evaluating all possible combinations of hyperparameters, Grid Search ensures that the model achieves the best possible performance based on the chosen evaluation metric. However, it is computationally expensive, especially with high-dimensional parameter grids and large datasets. Despite this, Grid Search remains a widely used approach due to its straightforward implementation and ability to enhance model performance significantly. For efficient hyperparameter tuning, Grid Search can be complemented with more advanced techniques like Random Search or Bayesian Optimization, particularly when computational resources are limited. [32] [33]

2.3 Evaluation Metrics

As the goal of this thesis is to build and deploy a well-generalized model for predicting pollen allergy symptoms, model performance need to be measured. Because of that, it is required to evaluate the model on different metrics which helps us to optimize the performance and efficiency of the model. Regression refers to predictive modeling problems that involve predicting a numeric value. It is different from classification that involves predicting a class label. Two most common metrics, Mean Absolute Error and Root Mean Squared Error, were used for regression problem where the intensity of overall symptoms need to be predicted. For classification problem, where the goal is to predict the type of the symptom, Accuracy, F1 Score, Recall and Precision metrics were used. Sckit-learn library provide functions for calculating these metrics.

2.3.1 Regression Evaluation Metrics

- **Mean Absolute Error** represents the average of the difference between observed and predicted values.[30] It calculates how far the predictions are from the observed values but do not give the direction of the error. Mathematically, it is represented as:

$$MAE = \frac{1}{n}\sum_{i=1}^{n} |Yi - \widehat{Y_i}| \tag{7}$$

where Y_i are observed and $\widehat{Y_i}$ are predicted values.

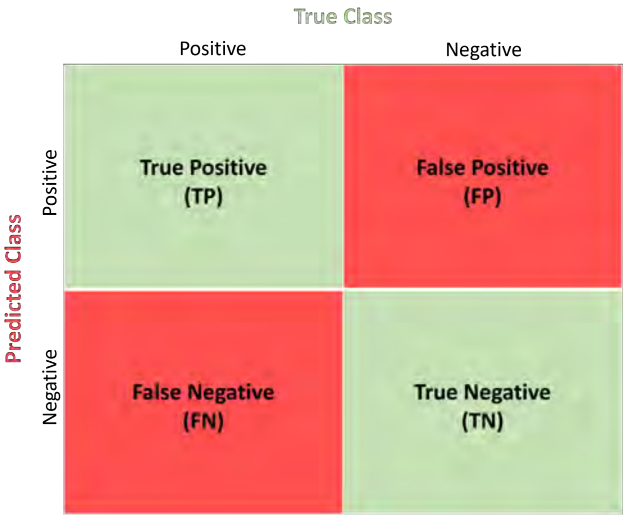
- **Root Mean Squared Error** is very similar to mean absolute error, but it represents the root of the average of the square of the difference between observed and predicted values. Mathematical formula for it is:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(Y_i - \widehat{Y_i})^2} \tag{8}$$

where Y_i are observed and $\widehat{Y_i}$ are predicted values.

2.3.2 Classification Evaluation Metrics

An important tool for understanding model performance in classification tasks is the confusion matrix. It is a matrix whose elements represent number of correctly or incorrectly classified data points (Figure 24) .



		True Class	
		Positive	Negative
Predicted Class	Positive	True Positive (TP)	False Positive (FP)
	Negative	False Negative (FN)	True Negative (TN)

Figure 24: Confusion Matrix

Based on values of this matrix, and the priorities of classification, different metrics have been developed [31]. During training and testing, the following metrics were monitored:

- **Accuracy** is a fundamental metric in classification, providing a straightforward measure of how well a model performs its intended task. It represents the ratio of correctly predicted instances to the total number of instances in the data set. The formula is:

$$Accuracy = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{FP} + \mathrm{FN} + \mathrm{TN}} \tag{9}$$

- **Precision** is used to measure the positive patterns that are correctly predicted from the total predicted patterns in a positive class, with formula:

$$Precision = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} \tag{10}$$

- **Recall**, also known as Sensitivity or True Positive Rate, is defined as the ratio of true positive predictions to the total number of actual positive cases.

$$Recall = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \tag{11}$$

- **F1 Score**, also known as Dice loss, represents the harmonic mean between recall and precision values. As precision grows recall usually declines, and vice versa, meaning that high F1 score indicates good balance between the two. The formula is:

$$F1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} = \frac{2 \times \mathrm{TP}}{2 \times \mathrm{TP} + \mathrm{FP} + \mathrm{FN}} \tag{12}$$

3 Experimental Results

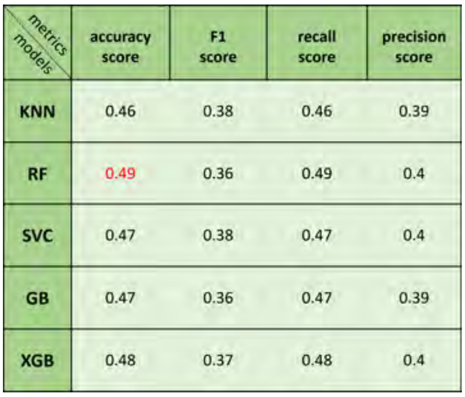
In this section the results obtained using different classification and regression machine learning algorithms are represented.

3.1 The Symptom Types Prediction

Five machine learning algorithms, K-Nearest Neighbors Classifier, Random Forest Classifier, Support Vector Machine Classifier, Gradient Boosting Classifier and XGB Classifier, were used to predict the type of symptoms caused by pollen allergy. All of them were evaluated using different evaluation metrics and obtained results were compared.

3.1.1 Classification models evaluation on data set with repeated inputs

Since there are the samples in our data set with the same dates and the same pollen concentration measurements but the different symptoms entered by different pollen diary user's, machine learning models can get confused when making predictions. In order to see how repeated inputs with different targets affect model performance, we evaluate the models without modifications on our data set using grouping showed in the Figure 16. The results are showed in the Figure 25. Although the Random Forest model achieved the highest accuracy score of 0.49 (highlighted in red), indicating it is the most accurate among the five models, this result is still not satisfactory as nearly 50% of the labels are misclassified.

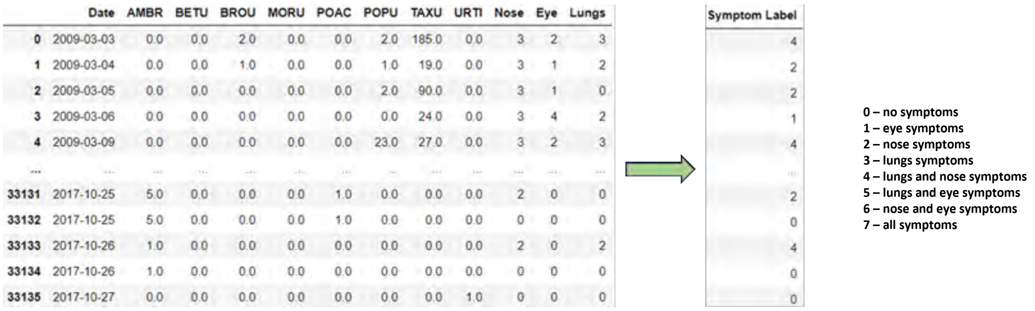


metrics models	accuracy score	F1 score	recall score	precision score
KNN	0.46	0.38	0.46	0.39
RF	0.49	0.36	0.49	0.4
SVC	0.47	0.38	0.47	0.4
GB	0.47	0.36	0.47	0.39
XGB	0.48	0.37	0.48	0.4

Figure 25: Prediction results with the best score highlighted in red - models evaluation on dataset with repeated inputs

3.1.2 Classification models evaluation on data set with unique inputs

After performed grouping showed in the Figure 16 mentioned classification machine learning models were trained and evaluated for both daily and cumulative pollen concentrations. The new modified data set with the daily pollen concentrations can be found in the Figure 26.



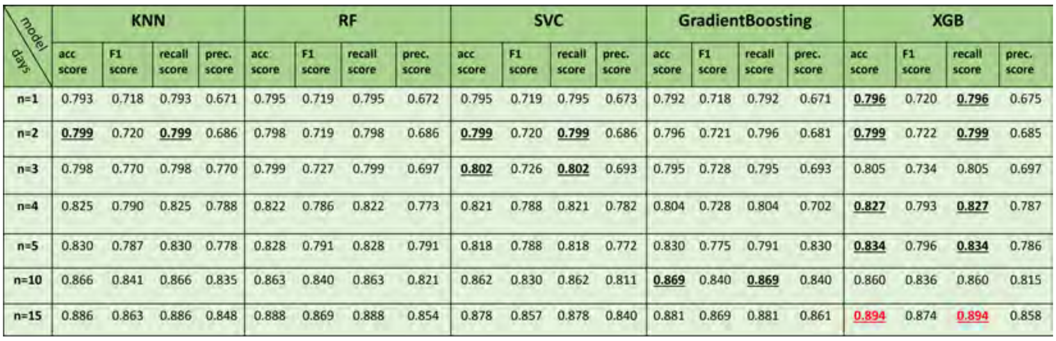
	Date	AMBR	BETU	BROU	MORU	POAC	POPU	TAXU	URTI	Nose	Eye	Lungs
0	2009-03-03	0.0	0.0	2.0	0.0	0.0	2.0	185.0	0.0	3	2	3
1	2009-03-04	0.0	0.0	1.0	0.0	0.0	1.0	19.0	0.0	3	1	2
2	2009-03-05	0.0	0.0	0.0	0.0	0.0	2.0	90.0	0.0	3	1	0
3	2009-03-06	0.0	0.0	0.0	0.0	0.0	0.0	24.0	0.0	3	4	2
4	2009-03-09	0.0	0.0	0.0	0.0	0.0	23.0	27.0	0.0	3	2	3
...	...	...	...	...	...	...	...	...	...	...	...	...
33131	2017-10-25	5.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	2	0	0
33132	2017-10-25	5.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0	0	0
33133	2017-10-26	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2	0	2
33134	2017-10-26	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0	0	0
33135	2017-10-27	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0	0	0

Symptom Label
4
2
2
1
4
...
2
0
4
0
0

0 – no symptoms
1 – eye symptoms
2 – nose symptoms
3 – lungs symptoms
4 – lungs and nose symptoms
5 – lungs and eye symptoms
6 – nose and eye symptoms
7 – all symptoms

Figure 26: Prediction results for user with the most entries

After training and evaluating machine learning models on daily and cumulative pollen concentrations the results showed in the Figure were obtained.



model / days	KNN				RF				SVC				GradientBoosting				XGB			
	acc score	F1 score	recall score	prec. score	acc score	F1 score	recall score	prec. score	acc score	F1 score	recall score	prec. score	acc score	F1 score	recall score	prec. score	acc score	F1 score	recall score	prec. score
n=1	0.793	0.718	0.793	0.671	0.795	0.719	0.795	0.672	0.795	0.719	0.795	0.673	0.792	0.718	0.792	0.671	**0.796**	0.720	**0.796**	0.675
n=2	**0.799**	0.720	**0.799**	0.686	0.798	0.719	0.798	0.686	**0.799**	0.720	**0.799**	0.686	0.796	0.721	0.796	0.681	**0.799**	0.722	**0.799**	0.685
n=3	0.798	0.770	0.798	0.770	0.799	0.727	0.799	0.697	**0.802**	0.726	**0.802**	0.693	0.795	0.728	0.795	0.693	0.805	0.734	0.805	0.697
n=4	0.825	0.790	0.825	0.788	0.822	0.786	0.822	0.773	0.821	0.788	0.821	0.782	0.804	0.728	0.804	0.702	**0.827**	0.793	**0.827**	0.787
n=5	0.830	0.787	0.830	0.778	0.828	0.791	0.828	0.791	0.818	0.788	0.818	0.772	0.830	0.775	0.791	0.830	**0.834**	0.796	**0.834**	0.786
n=10	0.866	0.841	0.866	0.835	0.863	0.840	0.863	0.821	0.862	0.830	0.862	0.811	**0.869**	0.840	**0.869**	0.840	0.860	0.836	0.860	0.815
n=15	0.886	0.863	0.886	0.848	0.888	0.869	0.888	0.854	0.878	0.857	0.878	0.840	0.881	0.869	0.881	0.861	**0.894**	0.874	**0.894**	0.858

Figure 27: Prediction results with highlighted best scores for different n values, obtained after training and evaluating ML models using data set with grouped samples

From the previous figure it can be seen that XGB Classifier achieved the highest accuracy score of **0.894** (marked in red) for n = 15, that indicates the highest proportion of correct predictions compared to others, and equally good recall score which indicates that the effectively captured positive instances and have a low rate of false negatives. The XGB parameters of best score:

$\{'learning\_rate' : 0.2, 'max\_depth' : 5, 'n\_estimators' : 100\}$
were obtained using GridSearchCV.

3.1.3 Classification models evaluation on data samples from user with the most entries

We have already seen that data entered by a lot of different users significantly affects models performance so top 1 user with the most entries was found and models were trained and evaluated on his data samples. A pollen diary user with the most entries has ID number **16796** and **2259** entries, which can be seen in the following histogram that represents number of data entries from top 10 users.

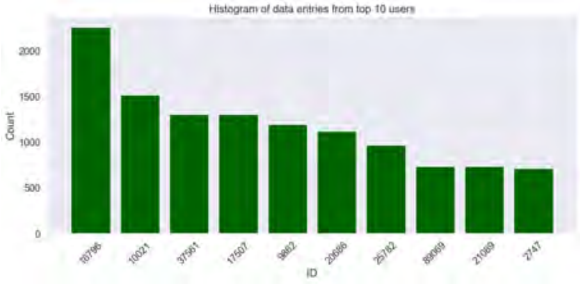
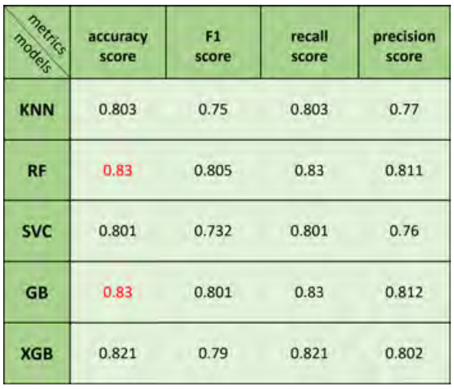


Figure 28: Number of data entries from top 10 users

After evaluating the classification models on the daily pollen concentrations and the symptoms only from user with the most entries the results from the Figure 29 are obtained.



metrics / models	accuracy score	F1 score	recall score	precision score
KNN	0.803	0.75	0.803	0.77
RF	0.83	0.805	0.83	0.811
SVC	0.801	0.732	0.801	0.76
GB	0.83	0.801	0.83	0.812
XGB	0.821	0.79	0.821	0.802

Figure 29: Prediction results with the best score highlighted in red for user with the most entries (classification)

From the previous figure, it can be concluded that Random Forest and Gradient Boosting, both achieving an **accuracy score of 0.83** (marked in red), are the top performers, demonstrating strong performance across all metrics. Both models show balanced performance in accuracy, precision, recall, and F1 score, making them suitable for applications where class balance and correct predictions are critical. Random Forest and Gradient Boosting parameters of best score, obtained using GridSearchCV, are:

$$\begin{aligned} Random\ Forest - \{'criterion' :' gini',' max\_depth' : 7, \\ 'max\_features' : 3,' n\_estimators' : 15\} \end{aligned}$$

$$\begin{aligned} Gradient\ Boosing - \{'n\_estimators' : 200,' max\_depth' : 5, \\ 'learning\_rate' : 0.01\} \end{aligned}$$

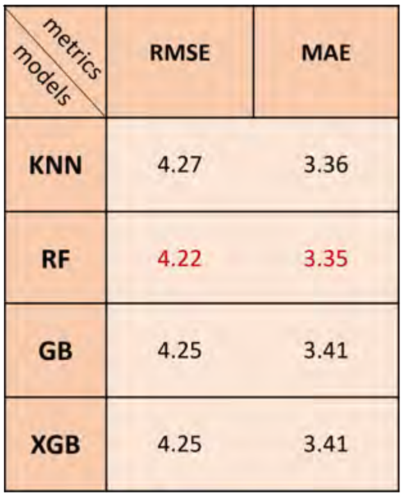
Additionally, the data set which is smaller then initial one with the data from all users, yielded better results because it does not contain samples with the same inputs (pollen concentrations) and differing targets (user symptoms), leading to more reliable and consistent modeling.

3.2 The Overall Symptoms Prediction

Four machine learning algorithms, K-Nearest Neighbors Regressor, Random Forest Regressor, Gradient Boosting Regressor and XGB Regressor, were employed to predict the overall symptom intensity caused by pollen allergy, ranging from 0 to 21. All those regression models were evaluated using Mean Absolute Error and Root Mean Squared Error evaluation metrics and obtained results were compared.

3.2.1 Regression models evaluation on data set with repeated inputs

Similar to the classification problem, the worst results after training and evaluating the regression models were obtained using an initial data set containing many samples with the same inputs and different targets. The grouping showed in the Figure 17 was not performed here. The prediction results can be found in the Figure 30.



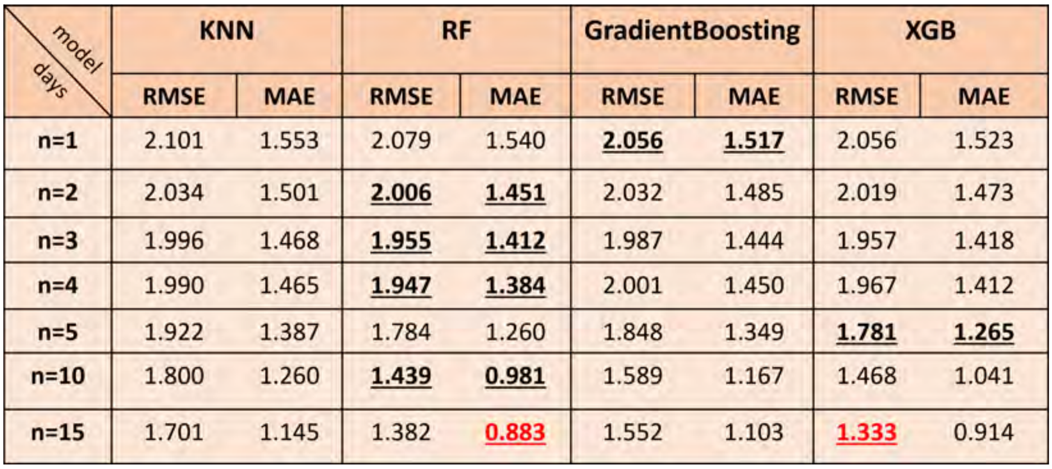
metrics models	RMSE	MAE
KNN	4.27	3.36
RF	4.22	3.35
GB	4.25	3.41
XGB	4.25	3.41

Figure 30: Prediction results with the best scores highlighted in red - models evaluation on data set with repeated inputs

Random Forest Regressor gave us the best results: **MAE = 4.22** and **RMSE = 3.35**. Given that the range of the prediction parameter is 21, an MAE of 4.22 is about 20% of the total range, which could be considered relatively high. RMSE gives more weight to larger errors due to squaring the differences before averaging. An RMSE of 3.35 is about 16% of the total range. This is slightly better than the MAE in terms of error magnitude, but still relatively high compared to the range.

3.2.2 Regression models evaluation on data set with unique inputs

After performed grouping showed in the Figure 17 mentioned regression machine learning models were trained and evaluated for both daily and cumulative pollen concentrations. In this section, the objective is to predict the mean of overall symptoms, as the samples are categorized based on that average. The same cumulative pollen concentrations as in section 3.1.2 were used and results are shown in Figure 31.



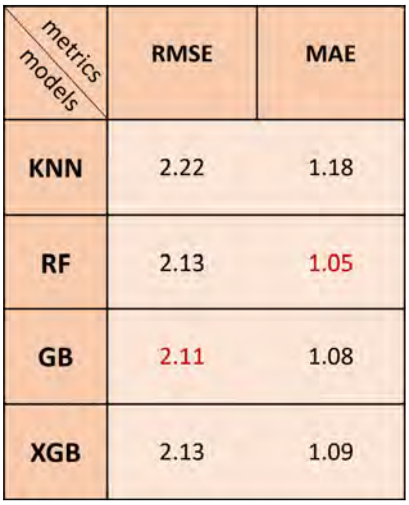
model days	KNN		RF		GradientBoosting		XGB	
	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE
n=1	2.101	1.553	2.079	1.540	**2.056**	**1.517**	2.056	1.523
n=2	2.034	1.501	**2.006**	**1.451**	2.032	1.485	2.019	1.473
n=3	1.996	1.468	**1.955**	**1.412**	1.987	1.444	1.957	1.418
n=4	1.990	1.465	**1.947**	**1.384**	2.001	1.450	1.967	1.412
n=5	1.922	1.387	1.784	1.260	1.848	1.349	**1.781**	**1.265**
n=10	1.800	1.260	**1.439**	**0.981**	1.589	1.167	1.468	1.041
n=15	1.701	1.145	1.382	**0.883**	1.552	1.103	**1.333**	0.914

Figure 31: Prediction results with highlighted best scores for different n values, obtained after training and evaluating ML models using data set with grouped samples

The previous table provides a comparison of four different machine learning models across various cumulative pollen concentrations (n = 1, 2, 3, 4, 5, 10, and 15 days). Lower RMSE and MAE values indicate better model performance, as they reflect smaller errors in prediction. K-Nearest Neighbors showed consistent improvement in both RMSE and MAE as the cumulative concentration increases, indicating that it benefits from more data points. Random Forest exhibited significant improvement with increasing n, particularly n=15 and parameters of best score: $\{'max\_depth' : 10,' max\_features' : 3,' n\_estimators' : 100\}$, where it has the lowest **MAE = 0.883** (marked in red) across all models. Gradient Boosting performed well overall, with competitive RMSE and MAE values, particularly excelling for daily pollen concentrations (n = 1). Extreme Gradient Boosting demonstrated strong performance, achieving the lowest **RMSE = 1.333** (marked in red) with optimal parameters: $\{'learning\_rate' : 0.1,' max\_depth' : 7,' n\_estimators' : 200\}$, for n=15.

3.2.3 Regression models evaluation on data samples from user with the most entries

The mentioned regression models were also trained and evaluated on data samples from pollen diary user with the most entries, with ID number 16796 (Figure28), and daily pollen concentrations (n = 1). The results, represented with the Figure 32, were obtained.



metrics models	RMSE	MAE
KNN	2.22	1.18
RF	2.13	1.05
GB	2.11	1.08
XGB	2.13	1.09

Figure 32: Prediction results with the best scores highlighted in red for user with the most entries (regression)

From the previous figure it can be seen that Gradient Boosting with an MAE of **2.11** is the best for Minimizing Average Error(MAE) and is Random Forest with an RMSE of **1.05** is the best for Minimizing Larger Errors(RMSE). K-Nearest Neighbors showed the least favorable performance in both metrics, while Random Forest and Gradient Boosting are the top contenders depending on whether minimizing average or larger errors is more critical. Comparing the prediction results for both, all pollen diary users and top user with the most entries (Figure 30 and Figure 32), we can conclude that the models achieved better results with the data set smaller then initial one with the data samples from all users, because it lacked samples that had identical inputs (pollen concentrations) but different targets (user symptoms).

4 Discussion and Ideas for Future Work

The results obtained in this thesis highlight the potential of machine learning models in predicting pollen allergy symptoms, yet several avenues for improvement and future research remain.

4.1 Discussion

Several observations and conclusions can be drawn from the results analysis:

4.1 Discussion

Impact of Repeated Inputs on Model Performance. When training and evaluating classification and regression models on datasets containing repeated inputs (i.e., identical dates and pollen concentrations but different symptom records from different users), the models generally performed poorly. This is evident from the lower accuracy scores and high error rates. For example, the Random Forest classifier, which achieved the highest accuracy of only 0.49 on the dataset with repeated inputs, indicates that nearly half of the predictions were incorrect. Similarly, regression models trained on repeated inputs showed suboptimal performance, with relatively high Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) values. These findings suggest that the presence of identical inputs with varying target outputs confuses the models, reducing their ability to learn consistent patterns and make accurate predictions.

Effectiveness of Grouped and Unique Input Datasets. In contrast, models trained and evaluated on datasets with unique inputs demonstrated significant improvements in performance. For classification models, the XGB classifier achieved an accuracy score of 0.894, showing a strong ability to predict symptom types accurately when trained on a refined dataset. Similarly, regression models evaluated on datasets with unique inputs yielded better results, as indicated by lower MAE and RMSE values. The K-Nearest Neighbors regressor, Random Forest regressor, and Extreme Gradient Boosting regressor all showed improved performance with increasing cumulative pollen concentrations (n = 1 to 15 days), suggesting that aggregating data over a period helps in capturing underlying patterns more effectively.

Model Performance for Individual Users with the Most Entries. The analysis also revealed that machine learning models trained on data from the user with the most entries (User ID 16796) performed better than those trained on the aggregated data from all users. Both Random Forest and Gradient Boosting classifiers achieved high accuracy scores of 0.83, highlighting their ability to handle individual-specific variations in symptom presentation effectively. For regression tasks, the Gradient Boosting and Random Forest regressors showed the lowest errors, with MAE of 2.11 and RMSE of 1.05, respectively, further demonstrating that user-specific models can be more reliable and consistent. These findings emphasize the need for personalized models that can account for individual variability in allergic responses to pollen exposure.

Comparison of Model Performance Across Different Algorithms. Across different machine learning algorithms, the performance varied significantly. For classification tasks, ensemble methods such as Random Forest, Gradient Boosting, and XGB classifiers outperformed simpler models like K-Nearest Neighbors, demonstrating the strength of these models in capturing complex relationships in the data. In regression tasks, similar trends were observed, with ensemble methods like Random Forest and Gradient Boosting showing superior performance, particularly when evaluated on datasets with unique inputs or individual user data. These results suggest that ensemble methods are more effective in handling the complexities and variability inherent in allergic symptom prediction.

4.2 Ideas for future work

Data Quality and Diversity. One of the primary limitations encountered during this study was the presence of identical input samples (pollen concentrations) associated with different symptom labels across various users. This inconsistency affected the performance of the models, leading to less accurate predictions. Future work should focus on obtaining more precise and consistent data. This could be achieved by incorporating additional contextual data, such as weather conditions, pollution levels, or more details about user, to improve the robustness of the predictions.

Model Enhancements. While Random Forest and Extreme Gradient Boosting emerged as the top-performing models, there is still room for optimization. Future efforts could involve exploring more advanced ensemble techniques, such as stacking or blending multiple models to leverage their strengths. Additionally, hyperparameter tuning could be further refined by implementing more sophisticated search techniques like Bayesian optimization or genetic algorithms, which may yield better parameter sets than those obtained through GridSearchCV.

Feature Engineering and Selection. The current study primarily focused on daily and cumulative pollen concentrations as input features. Future research could benefit from incorporating additional features, such as historical symptom trends or time-lagged pollen data, which might capture latent patterns and improve model predictions. Additionally, feature selection methods could be applied to identify the most relevant features, potentially reducing model complexity and improving interpretability.

Temporal Modeling and Longitudinal Analysis. Given the temporal nature of allergy symptoms and pollen concentrations, future work could explore time series modeling approaches, such as recurrent neural networks (RNNs) or long short-term memory (LSTM) networks, which are specifically designed to handle sequential data. These models could capture temporal dependencies and improve the prediction of symptom onset or progression over time.

Personalized Prediction Models. While this study explored predictions for a single user with the most entries, future work could expand this approach by developing personalized models for other users or user segments. By tailoring models to individual or group-specific data, the predictions could become more accurate and relevant, thus enhancing their practical utility.

Real-Time Implementation and Application. The ultimate goal of this research is to create a tool that can be used in real-time to help individuals manage their allergy symptoms. Future work could focus on developing a user-friendly application that integrates these machine learning models, allowing users to input current conditions and receive immediate predictions. Such an application could also continuously learn and adapt to new data, further improving prediction accuracy over time.

Broader Data Integration. Integrating data from other geographical regions or incorporating pollen data from other networks could expand the applicability of the models developed in this thesis. This could enable a more comprehensive understanding of pollen allergies on a global scale and enhance the generalizability of the models.

By addressing these areas of improvement, future work can build on the foundation laid in this thesis, further refining the models and enhancing their predictive capabilities, ultimately contributing to better management of pollen allergies and improved quality of life for sufferers.

5 Conclusion

This thesis successfully identified and trained the most suitable machine learning models for predicting pollen allergy symptoms, demonstrating various predictive capabilities in both regression and classification tasks. By leveraging data from the Patient's Hayfever Diary (PHD) and the Laboratory for Palynology at the Faculty of Sciences in Novi Sad, the study evaluated multiple models across different time frames.

For symptom type classification, the XGB Classifier consistently outperformed other models, achieving the highest accuracy and recall scores, particularly when using cumulative pollen concentrations over multiple days. This indicates that the XGB Classifier is highly effective in capturing the relationship between pollen concentrations and the type of symptoms experienced by users.

In the regression tasks, the Random Forest Regressor and Gradient Boosting Regressor emerged as the most reliable models, with the Random Forest Regressor showing particularly strong performance when trained on cumulative pollen data over a 15-day period. These models were able to predict overall symptom intensity with relatively low error rates, indicating their suitability for forecasting the severity of pollen allergy symptoms based on environmental data.

Interestingly, the evaluation of models on data from the user with the most entries revealed that the absence of conflicting samples (where identical pollen concentrations were associated with different symptoms) led to significantly improved model performance. Even with a smaller, user-specific dataset, the models maintained high accuracy and low error rates, underscoring the robustness and reliability of the chosen approaches.

Furthermore, extensive data analysis provided valuable insights, revealing important correlations and statistical patterns that enhance our understanding of pollen allergy symptoms. These findings contribute to the broader goal of improving daily life for individuals affected by pollen allergies through more accurate symptom prediction.

References

[1] D'Amato, G., Vitale, C., De Martino, A., et al., 2015. *Effects on asthma and respiratory allergy of climate change and air pollution*

[2] Schiavoni, G., D'Amato, G., Afferni, C., 2017. *The dangerous liaison between pollens and pollution in respiratory allergy*

[3] Erbas, B., Jazayeri, M., Lambert, K.A., Katelaris, C.H., Prendergast, L.A., Tham, R., Parrodi, M.J., Davies, J., Newbigin, E., Abramson, M.J., Dharmage, S.C., 2018. *Outdoor pollen is a trigger of child and adolescent asthma emergency department presentations: a systematic review and meta-analysis*

[4] Pollen Allergies: Symptoms and Natural Support Strategies

[5] David Barber, *Bayesian Reasoning and Machine Learning, 2017*

[6] Bousquet, P.-J., Chinn, S., Janson, C., Kogevinas, M., Burney, P., Jarvis, D. (2007). *Geographical variation in the prevalence of positive skin tests to environmental aeroallergens in the European Community Respiratory Health Survey I*

[7] Smith, M., Cecchi, L., Skjoth, C. A., Karrer, G., Sikoparija, B. (2013). *Common ragweed: A threat to environmental health in Europe*

[8] Lake, I. R., Jones, N. R., Agnew, M., Goodess, C. M., Giorgi, F., Hamaoui-Laguel, L., et al. (2017). *Climate change and future pollen allergy in Europe*

[9] B. Sikoparija, O. Marko, M. Panic, D. Jakovetic, P. Radisic. (2018). *How to prepare a pollen calendar for forecasting daily pollen concentrations of Ambrosia, Betula and Poaceae?*

[10] Sikoparija, B., Mimic, G., Panic, M., Marko, O., Radisic, P., Pejak-Sikoparija, T., Pauling, A. (2018). *High temporal resolution of airborne Ambrosia pollen measurements above the source reveals emission characteristics*

[11] Zewdie, G. K., Lary, D. J., Levetin, E., Garuma, G. F. (2019). *Applying Deep Neural Networks and Ensemble Machine Learning Methods to Forecast Airborne Ambrosia Pollen*

[12] Sikoparija, B.; Mimic, G.; Matavulj, P.; Panic, M.; Simovic, I.; Brdar, S. . (2019). *Short communication: Do we need continuous sampling to capture variability of hourly pollen concentrations?*

[13] S. Papadogiannaki, S. Kontos, D. Parliari, D. Melas. (2023). *Machine Learning Regression to Predict Pollen Concentrations of Oleaceae and Quercus Taxa in Thessaloniki, Greece*

[14] Voukantsis, D., Karatzas, K., Jaeger, S., Berger, U., Smith, M. (2012). *Analysis and forecasting of airborne pollen–induced symptoms with the aid of computational intelligence methods*

[15] Patient's Hayfever Diary
https://www.pollendiary.com/Phd/

[16] sklearn.preprocessing.StandardScaler
https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html

[17] How to Calculate Correlation Between Variables in Python
https://machinelearningmastery.com/how-to-use-correlation-to-understand-the-relationship-between-variables/

[18] Understanding K-Nearest Neighbors: A Simple Approach to Classification and Regression
https://pub.towardsai.net/understanding-k-nearest-neighbors-a-simple-approach-to-classification-and-regression-e4b30b37f151

[19] Random Forest Regression
https://medium.com/swlh/random-forest-and-its-implementation-71824ced454f

[20] Random Forest Regression in Python
https://www.geeksforgeeks.org/random-forest-regression-in-python/

[21] Random Forest Algorithm Explained
https://anasbrital98.github.io/blog/2021/Random-Forest/

[22] Boser, Guyon, Vapnik, *Support Vector Machines, 1992.*

[23] Unlocking the True Power of Support Vector Regression
https://towardsdatascience.com/unlocking-the-true-power-of-support-vector-regression-847fd123a4a0

[24] Introduction to Support Vector Machines https://www.theaidream.com/post/introduction-to-support-vector-machines-svm

[25] Introduction to the Gradient Boosting Algorithm https://medium.com/analytics-vidhya/introduction-to-the-gradient-boosting-algorithm-c25c653f826b

[26] All You Need to Know about Gradient Boosting Algorithm - Part 1. Regression https://towardsdatascience.com/all-you-need-to-know-about-gradient-boosting-algorithm-part-1-regression-2520a34a502

[27] Gradient Boosting Algorithm: A Complete Guide for Beginners https://www.analyticsvidhya.com/blog/2021/09/gradient-boosting-algorithm-a-complete-guide-for-beginners/

[28] Tianqi Chen, Carlos Guestrin, XGBoost: *A Scalable Tree Boosting System, 2016*

[29] A Gentle Introduction to the Gradient Boosting Algorithm for Machine Learning https://machinelearningmastery.com/gentle-introduction-gradient-boosting-algorithm-machine-learning/

[30] Metrics To Evaluate Machine Learning Algorithms in Python https://machinelearningmastery.com/metrics-evaluate-machine-learning-algorithms-python/

[31] M. Hossin and M. N. Sulaiman. A review on evaluation metrics for data classification evaluations. *International journal of data mining and knowledge management process, 2015.*

[32] Using Grid Search For Hyper-Parameter Tuning https://medium.com/@hammad.ai/using-grid-search-for-hyper-parameter-tuning-bad6756324cc

[33] Tune Hyperparameters with GridSearchCV https://www.analyticsvidhya.com/blog/2021/06/tune-hyperparameters-with-gridsearchcv/

Biography

Nataša Diklić was born on the 10th of November 1994 in Prnjavor, Republic of Srpska, Bosnia and Herzegovina. She received her Bachelor's degree in Applied Mathematics in 2017 from the Faculty of Sciences, University of Novi Sad, Serbia and she continued her Master studies in the field of Data Science at the same faculty. At the end of her studies, Nataša attended a six-week internship at the Continental Automotive, where she is still employed as a Software Engineer and Scrum Master in the team dealing with highly automated driving.

Appendix A

Figure 33 shows concentrations of different pollen species over the years.

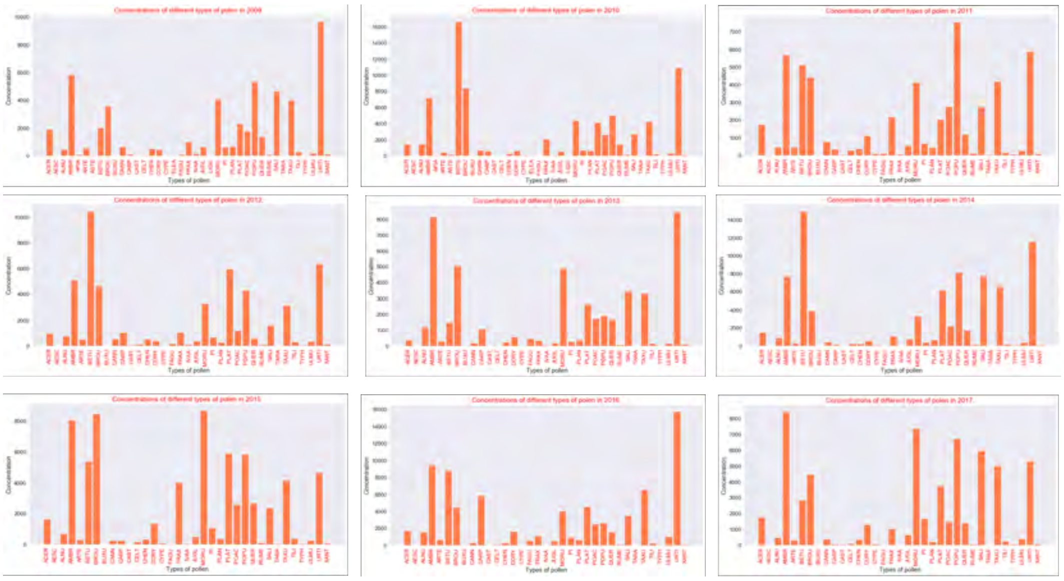
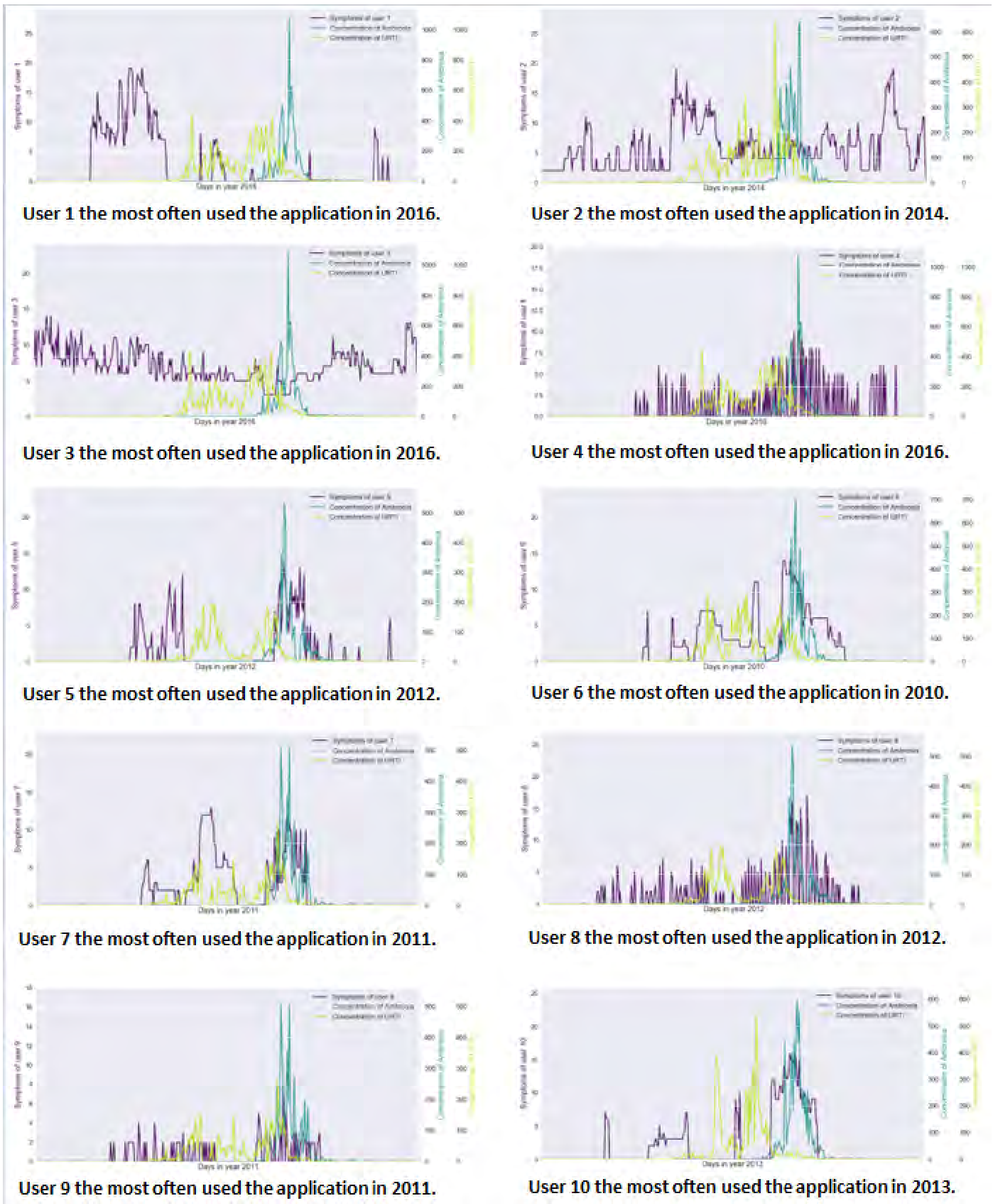


Figure 33: Concentrations of different pollen species over the years

Given that AMBR (Ambrosia - Ragweed) and URTI (Urticaceae - Nettles) consistently rank among the top 5 pollen concentrations each year, as shown in the previous figure, these two pollen species were specifically compared with the symptoms reported by the top 10 pollen diary users. The graphics depicting ragweed and nettle pollen concentrations alongside the symptoms of the top 10 users are presented in Figure 34.



User 1 the most often used the application in 2016.

User 2 the most often used the application in 2014.

User 3 the most often used the application in 2016.

User 4 the most often used the application in 2016.

User 5 the most often used the application in 2012.

User 6 the most often used the application in 2010.

User 7 the most often used the application in 2011.

User 8 the most often used the application in 2012.

User 9 the most often used the application in 2011.

User 10 the most often used the application in 2013.

Figure 34: AMBR and URTI pollen concentrations alongside the symptoms of the top 10 users