Data Mining Project HWS 2021

presented by
Group 14 (ThundERA), that is
Annika Herbert (1820576)
Oghenekeno Omogha (1725331)
Aleksandro Vogli (1820382)
Hoyoon Lee ()
Sven Oerther (1417750)

submitted to the
Data and Web Science Group
Prof. Dr. Tobias Weller
University of Mannheim

December 23, 2021

Introduction

Application area and goals (Business Understanding)

Meteorological forecasts - Hardly questioned but indispensable in everyday life. We rely on the experts who had to study the field for years to approximate its complexity. But even experts can't simply look to the sky and predict the weather over the long term. Only by using supercomputers to analyze the amount of data needed to make predictions will they be able to identify complex weather movements in the future.

Recently, a natural disaster caused by extreme weather occurred in the German state of Rhineland-Palatinate, where numerous people died to and buildings as well as infrastructures were completely destroyed within hours [1]. Now, we ask ourselves: is it possible to study such weather abnormalities using ML and find statistical regularities? Therefore, we strive to support traditional forecasting methods by using data mining to determine the severity of thunderstorms and other extreme weather (e.g. heavy rain).

Structure and Size of the Data Set

Structure and size of the data set (Data Understanding)

For this project, we have used the ERA5 dataset which contains hourly weather data on single levels from 1979 to the present. The data was provided by grids of 0.25 degrees latitude and longitude each. Mainly, it consisted of four subsets: hourly and monthly products, both on pressure levels (upper airfields) and single levels (atmospheric, ocean-wave, and land surface quantities). While the full data is archived in the ECMWF data archive, we utilized a subset of the full dataset which we accessed via the C3S Climate Data Store. We conducted initial research to figure out meaningful features that should be analyzed. The parameters were archived in netcdf4 format, and the analysis was done with three different dimensions of the time axis, location information, and the feature itself. The export of data was done in an array per dimension, and it gave the features as numerical and categorical data. As the objective of this project is to forecast thunderstorms and floods that can happen in Mannheim, we selected the grid that covers the area of Mannheim. Therefore, while exporting the dataset, we chose the coordinates of the north: 49.65, south: 49.4, east: 8.55, and west: 8.3. By setting the longitude and latitude, it was possible to obtain specific historical data regarding Mannheim's weather situation. Moreover, to work with a sufficient amount of information, we got 10 years' hourly weather data for each day. Since we selected

a large amount of data containing information about 87,600 hours of weather, it resulted in challenges to deal with the huge and multidimensional dataset. Overcoming the challenges, we adopted the mean and variance values of each day. In the dataset, the indication of the probability of thunderstorm occurrence, “Total totals index”, was provided as categorical data that was presented as “Thunderstorms Not likely”, “Thunderstorms likely”, “Isolated Severe Thunderstorms”, “Scattered severe thunderstorms more likely”, and “widely Scattered Severe Thunderstorms”. In addition, the dataset also contains numerical data such as the parameter showing the transfer of heat between surface and atmosphere called “Surface sensible heat flux” and temperature of the air at 2m above the surface named “2m dewpoint temperature”. The diagrams given below show the distribution of variables. On the left, one of the features’ exemplary distributions is provided. Most of features looked like this and had normal distributions. On the top right corner, the distribution of target variable is described, and the lower right depicts the distribution of binned target variable according to expert knowledge.

Preprocessing

Preprocessing for classification

Educated Feature Preselection: Due to the size of the dataset a pre-selection of attributes was absolutely necessary. Some features such as ”ocean waves” could be ignored by principle since Mannheim has no ocean nearby. Furthermore we researched possible correlations between the available features and our target. The knowledge was gathered and left us with a list of possibly meaningful features. To determine the actual usefullness of these features we identified a specific time frame in the data that has a high variance in our target variable. For this time frame we obtained all features and conducted a correlation analysis. Our expectation: if the correlation holds for this reduced time frame of high target variance (similar to the target variance for the whole time frame of 10 years) it is applicable for the whole time frame.

Data Exploration: As mentioned above features are all numerical and approximately normally distributed. We do not have any missing values in our data. Quaility and validity check lies outside our expertise, it could only be performed for few features where we were able to infer about (e.g. Geopotential is a constant etc.) otherwise we were forced to rely on the quality control of the responsible insitutions for which they provide documentation (see [2]).

First Aggregation: For each feature, we receive four point values from the grid. These values are quite similar since they are taken in a 0,25° degree lon-lat grid around Mannheim. In order to obtain a single meaningful feature instead of a

feature vector, we decide to use the mean.

Handling of time series: Since we are looking to solve a forecasting problem we need to be aware of information leak. We solve this problem by getting rid of the temporal coupling of the data. To do that we decide to take a measure over the span of a day and predict the target variable only for the next day. We compute a second step of aggregation: taking the mean, variance, minimum and maximum of the measurements that were recorded over the span of a day. These aggreagations are substituted for the actual hourly measures.

Feature Engineering and generation We had multiple hypotheses about the expression and generation of features.
Hypothesis 1: The relation of high to low cloud cover has relevance for thunderstorm prediction. We explored these features and their releation to each other together with their correlation to the target variable. The hypothesis did not hold.
Hypothesis 2: Some feature expressions can be boosted with basic arithmetics. We aggregated all features and check again for relevance. Also we applied the logarithm, a second order polynom and the square root to all features while taking into account the range and domain of these functions. The hypothesis did hold, but the "boost" in feature expression was barely significant. Still, we kept those aggregated samples in the final dataframe for further evaluation through the algorithm performance measure.

Further preprocessing: Binning of the target variable was performed as described above. The individual samples (x_i) are unique. Through all of these feature generation and selection processes, we attempted to maximize the correlation coefficient to the target variables. Outliers were handled by normalizing the data. When computing z-scores we introduced the inductive bias that all features are normally distributed and of equal importance. The distribution has been checked, the assumption of equal importance holds until further feature selection is performed through a different measure.

Building the final dataset: We built multiple feature subsets that we wanted to try on our classifiers. One feature subset where only the features with the best representation of a feature is present. E.g. we selected a feature based on highest correlation coefficient and remove all other aggregated representations of this same feature from the dataset. Our aggregations are non-linear, so we did not expect to run into problems with possible linearly dependent features (we avoid unstable weights for logistic regression by having linearly independent features for example). Therefore, we wanted to try all features and the described subset on our ML algorithms.

Deviations of preprocessing for regression

Handling Outliers We have used the inbuilt seaborn boxplot to check for outliers and with the help of z-score (threshold 2.4) we removed outliers. After removing outliers in the train dataset, we see a reduction in the size of split dataset from 2555 to 1823 values. It is important to mention that removal of outliers did not yield good results.

Further preprocessing

- Manual train test split (first 8 years for training and 2 last years for testing)
- Train test split function
- T-3 features
- Aggregated features
- Best correlated features based on the correlation coefficient

Data Mining

Classification approaches

For **classification** we have tried out six different ML approaches: Logistic Regression, Decision Tree, Random Forest, Multi-Level Perceptron (feedforward Neural Network), Naive Bayes, k-Nearest Neighbor Classifier.

Baseline: We have the choice of two different baselines, we can use the trivial approach and use the relative frequencies of the classes as baseline (approximate 60/40 split of majority to minority class). Secondary we can use the un-optimized plain algorithms as baseline. This is a valid approach since we expect that mutliple different processes like balancing the data or hyperparameter optimization significantly improve on the initial results. We take one performance measure, consisting of accuracy, precision, recall, f1-score on the unoptimized algorithms. Of the plain classifiers logistic regression performed best.

Splits and Validation: In the very beginning we split off 30% of the data as a held out test set, we made sure to stratify the split for the target labels. This test set is to be used on the final optimized models for performance evaluation. For validation of our indivdual components we used k-fold cross validation on the reamaining 70%. Because of limited computational resources, we resorted to using 5-folds. In order to correctly score each model on each pipeline configuration, we

wrote a dedicated function that scores the specified performance measure (accuracy, precision, recall and f1-score) on an entire pipeline for each algorithm. This function is now used to evaluate different configurations (e.g. usage of different sampling techniques) for all classifiers.

Configurations and interpretation of results

Balancing: We applied four different ways of balancing. Between oversampling, undersampling, smote and a mixture of oversampling and undersampling, using only undersampling performed overall best on all classifiers. We attribute this to an overabundance of "good weather" samples which are very similar and have no variance in feature expression, we do not loose any information if we undersample in this case. For the next pipeline evaluations we resorted to using undersampling.

Feature Selection: When scoring our k-best subset on all classifiers, we realize that reducing the amount of features decreases all our measured scores. We decided to only use the full dataset. This is to be expected from classifiers like Random Forest that have their own form of randomized feature selection per tree but is surprising for the MLP classifier. This classifier is known to perform better on limited subset of features.

Random Forest: The Random Forest classifier uses an ensemble learning method for classification; thus, we used sklearn's ensemble RandomForestClassifier function. In this supervised machine learning algorithm, the classifier combines multiple decision trees that form a forest of trees.In this project we have used default parameters of the classifier for training the model. Similarly, as in decision trees, the best parameters for splitting the trees will be recorded after the parameters are optimized. In measuring the performance of the model for classifying severity and no severity of thunderstorm, accuracy and recall scores are 71% and 72% respectively. The achieved scores for the precision and f1 are lower, achieving 62% and 66% respectively.

Hyperparameter Optimization

In this project, we have adopted the HalvingGridSearchCV hyper-parameter tuning method to reduce the run-time in hyper-parameter tuning. This technique helped us to tune parameters model for best possible combination of the hyperparameter values. The results of the hyper-parameter tuning were then applied on the test data set and then evaluated further. For the Decision Trees, the following hyper-parameter were selected: criterion, random state, class weight, and splitter. Hyper-parameter optimization resulted in the following options: class weight: balanced,

Classifiers	Accuracy	F1	Precision	Recall
GaussianNB	0.685967	0.664385	0.588656	0.763972
LogisticRegression	0.711383	0.677971	0.622216	0.745694
DecisionTreeClassifier	0.645695	0.595114	0.556723	0.640187
KNeighborsClassifier	0.689497	0.656639	0.598055	0.728464
RandomForestClassifier	0.710993	0.669619	0.627068	0.719838
MLPClassifier	0.725864	0.689091	0.640626	0.746683

Table 1: Comparison of classifier on no Parameters

criterion: Gini, and splitter: random. With this a slight increase of the results could be achieved with the following results: accuracy: 0.68, precision: 0.67, recall:0.68, and F1: 0.67. These are, compared to the other algorithms, the lowest scores of all trained algorithms.
In the Random Forest model, we used following hyper-parameter: Number of Trees to be constructed for the decision forest, number of features to be selected at random, and minimum splits for a child node. Since, large max features can result in high model performance but issues of under-fitting and over-fitting can arise[6]. We used auto and sqrt parameter, which places no limitations on the number of features that can be re-sampled and also takes the square root of the total number of features. We will also measure the quality of the split on gini and entropy. We have opted for 1000 n_estimators to control the number of trees in the classifier. Our best parameters turned out to be max features = [auto] n_estimators = [1000] and min sample split of 4 and gini. With this we achieved the best scores in comparison to all models with accuracy: 0.77, precision: 0.76, recall:0.77, and F1: 0.76.
Similarly in Logistic Regression, we used HalvingGridSearchCV with the "liblinear" solver for classification with different C values which induce regularization in the model. We observed that by lowering the value the C it induces stronger regularization and therefore saw an overall increase in F1 and accuracy scores.

Regression

Extreme weather conditions and natural disasters are incidents that continue to plague the earth. One of the common natural disasters is flooding which is a result of extreme rainfall. A part of our goal in this project is to predict the precipitation of rainfall for a 24-hour period. Our dataset from ERA has provided us with required feature data to carry out this prediction. Using information provided in the overview of the dataset and due research, the "Mean total precipitation" is used as a

target variable for this task. We have adjusted this variable to "Total precipitation" by taken the mean values over a 24-hour period to give better information about the total amount of precipitation during 24 hours. Considering that we are using a continuous target variable, the chosen approach is regression.

Feature selection To select the best features in order to achieve the best results we used p-value score with alpha=0.05. Regression is a supervised learning approach, hence in our approach to achieve best results we used LinearRegression(), Ridge(), XGBRegressor(), Lasso(alpha=0.1) and DecisionTreeRegressor(max_depth =2).

	R2	**Neg.MAE**
Linear Regression	0.83981673	-9.86E-06
Ridge	0.8398288	-9.86E-06
XGBRegressor	0.17199737	-2.27E-05
Lasso	-0.0023884	-2.53E-05
Decision Tree Regressor	0.58643456	-1.62E-05

Table 2: Comparison of model performance

Hyperparameter tuning This step is performed by using Cross validation and Grid search. The hyper parameter tuned are solver in Ridge() and total number of features using RFE models.

Evaluation

To find the best model 10-fold cross validation is used over the train subset. The reason of doing this is to find the best model and then test it on unseen data. Firstly, we have used the train test split function (80:20) and the target variable is stratified.

	auto	**svdc**	**cholesky**	**lsqr**	**sparse_cg**	**sag**	**saga**
R2	0.8398	0.8398	0.8398	0.8396	0.8398	0.8398	0.8398
Negative MAE	-9.862	-9.862	-9.8624	-9.8706	-9.8624	-9.8629	-9.8636

Table 3: Solver Score

Model analysis

Considering that the best model was LinearRegression() by using T-3 features we tried to remove outliers, polynomial regression and standardization. Firstly we

used scattered plots to have a better idea how the features are related to our target variable. After plotting them we assumed that using just a linear regression would be perfectly fine. The best scores are the models that include outliers and do not include polynomial regression. A problem that our model faced was multicollinearity. This may occur due to the T-3 approach we are using. Furthermore we used VIF-factor to solve our problem by deleting features with high VIF-factor score. But it leaded to a decrease of scores.

Over/Under fitting

By testing the final model on the test dataset we had the following results our model does not suffer from over/under fitting because the train and test results are really close.

	Test Score	**Train Score**
R2	0.844	0.839

Table 4: Train and Test scores

After the best model is found, which is LinearRegression(),we modified the target variable to Total Precipitation in mm (24hours) to make our model more understandable.

RMSE	**R2**
0.3051	0.844

Table 5: Final Results

Evaluation and Results

Exemplary analysis: We decided to interpret the weights of the logistic regression classifier. The logistic regression introduces the inductive bias that the log odds on the class label y are a linear function of x. This assumption seemed to hold. The classifier had a higher accuracy than our majority class frequency, thus we can assume the learned weights represent the data in some way.
Interpreting the weights: We obtained the coefficients (weights) from the logistic regression classifier. Noting that the weights are very small and look almost uniform. For the amount of features we couln't really discern any large or small outlier weights. This means that our features have approximately had the same influence on the decision. Climate is complex, so it is possible that a multitude of different

factors are mutually dependent. For our decision, we assumed that the features either depend on each other or directly influence the target to the same degree and thus needed to be taken into account together for a class decision. The fact that our performance measure worsens with fewer features supports this interpretation and is the result of the complexity of our business case.

Feature importance: Feature Importance could not be obtained from the weights of logistic regression, so we decided to plot the most important features according to the random forest, our best performing model.

Classification result

The best performing model for our **classification** task was the Random Forest classifier with 6% improved accuracy from 71% to 77%, precision score of 68% (4% increase), recall of 81% (9% increase) and f1 score of 74%. The classifier has also performed best with the random under-sampling technique and entropy criterion.

	Accuracy	f1	precision	recall
LogisticRegression	0.76	0.755	0.76	0.77
DecisionTreeClassifier	0.68	0.675	0.675	0.68
KNeighborsClassifier	0.7	0.69	0.69	0.69
RandomForestClassifier	0.77	0.765	0.765	0.775
MLPClassifier	0.74	0.74	0.745	0.75

Figure 1: Final result of classifcation

Conclusion

For the given problem which was to save lives and property from thunderstorms and floods, we decided to use the regression and classification together. To initiate the implementation, we eliminated temporal coupling and simplified the multidimensional data to make usage in our approach. Among the models we had utilized, the random forest model is identified as the best approach because the results were noticeably better than the baseline model. Currently, weather forecasting is done

using supercomputers to model the physical properties of the weather (pressure fields, cloud movements, and such). Supercomputers are necessary to have an accurate representation/model of the inner workings and physics in a weather system. Our research did look at the possibility to forecast only individual independent phenomena in a single location using ML to find statistical patterns.

The correct next step would be to try using ML and find a way to model the whole weather system for a bigger area as it is done with supercomputers. For this, in the current literature, Graph Neural Networks are explored. They can model patterns and propagate change over an area where different locations are represented as nodes. This can merge the current forecasting approach (where a system is modeled using physics) and ML which allows learning and finding patterns in existing data. The big plus with the ML approach is that we can save computation power and resources (which are immense for supercomputers) by initially training only once. We can use the knowledge gained by the current modeling techniques to design a good Graph Neural Network which has the potential to improve on the current state of the art. To conclude, this project has significantly contributed to deepening the understanding of data mining and ML as we could have a chance to apply and test in solving a real-life problem.

Bibliography

[1] Bundeszentrale für politische Bildung. Jahrhunderthochwasser 2021 in deutschland. 07.28.2021.

[2] Hans Hersbach, Bill Bell, Paul Berrisford, Shoji Hirahara, András Horányi, Joaquín Muñoz-Sabater, Julien Nicolas, Carole Peubey, Raluca Radu, Dinand Schepers, Adrian Simmons, Cornel Soci, Saleh Abdalla, Xavier Abellan, Gianpaolo Balsamo, Peter Bechtold, Gionata Biavati, Jean Bidlot, Massimo Bonavita, Giovanna Chiara, Per Dahlgren, Dick Dee, Michail Diamantakis, Rossana Dragani, Johannes Flemming, Richard Forbes, Manuel Fuentes, Alan Geer, Leo Haimberger, Sean Healy, Robin J. Hogan, Elías Hólm, Marta Janisková, Sarah Keeley, Patrick Laloyaux, Philippe Lopez, Cristina Lupu, Gabor Radnoti, Patricia Rosnay, Iryna Rozum, Freja Vamborg, Sebastien Villaume, and Jean-Noël Thépaut. The ERA5 global reanalysis. *Quarterly Journal of the Royal Meteorological Society*, 146(730):1999–2049, July 2020.

Ehrenwörtliche Erklärung

On this project and subsequent report worked jointly:

Annika Herbert (1820576)
Oghenekeno Omogha (1725331)
Aleksandro Vogli ()
Hoyoon Lee ()
Sven Oerther (1417750)

Mannheim, den 23.12.2021