Department of Economics and Management
Institute of Economics (ECON)
Statistical Methods and Econometrics
Dr. Sebastian Lerch

Seminar Report

Predictive Data Analytics

Written by **Tobias Biegert**

Matr. No. **2154836**
Information Systems

30. September 2022

Contents

List of Figures

List of Tables

Acronyms

CNT number of wildfires
BA burned area
MLP Multilayer Perceptron
CNN Convolutional Neural Network
RFR Random Forest Regressor
MSE Mean Squared Error

1 Introduction

Wildfires are an environmental hazard with significant impacts around the world, and their frequency and severity are expected to increase with global warming (Jones et al. 2020). This increase in fires creates a negative feedback loop through which naturally sequestered carbon is released back into the atmosphere, leading to further global warming (Pörtner et al. 2022). Wildfires in ecosystems where wildfire is uncommon or where non-native vegetation has encroached can have strong negative ecological impacts (Flannigan et al. 2006). Fires can also have serious consequences for human society, including health impacts from either direct harm or smoke, destruction of property, economic losses, and contamination of water and soil (Pörtner et al. 2022). Understanding and predicting the risk factors that contribute to wildfire occurrence, as well as their spatial and temporal distribution, is critical to managing wildfires.

In this work, we attempt to predict the occurance and size of wildfires, in a manner similar to the Data Challenge posed at the 2021 Extreme Value Analysis Conference (Thomas Opitz 2021). Unlike what is proposed in the Challenge, however, here we do not attempt to predict a probability distribution for the events, but rather the occurrence and size directly. The data is made available through the Challenge.

This report is divided into six chapters. Chapter 2 contains an explanation of the data set under consideration. In the third chapter, the two benchmark models are described and evaluated. Chapter 4 describes the implemented machine learning model architectures and the selection of the respective hyperparameters. The fifth chapter contains the evaluation and comparison of the different models. Finally, the last chapter summarizes the main results and gives suggestions for future research.

2 Data

2.1 Description

The dataset spans the time frame from 1993 to 2015 in monthly resolution, with only March through September available for each year, resulting in 161 months of available data. The area of the continental US is considered, meaning all states except Alaska and Hawaii are included in the data set. This landmass is divided into a grid with a $0.5° \times 0.5°$ resolution of longitude and latitude coordinates, which corresponds to approximately $55\,km$ by $55\,km$, yielding 3053 grid cells. The considered area and the applied subdivision can be seen in figure 1. Overall there are 563983 unique datapoints with no missing data for any of the variables. No erroneous data could be detected.

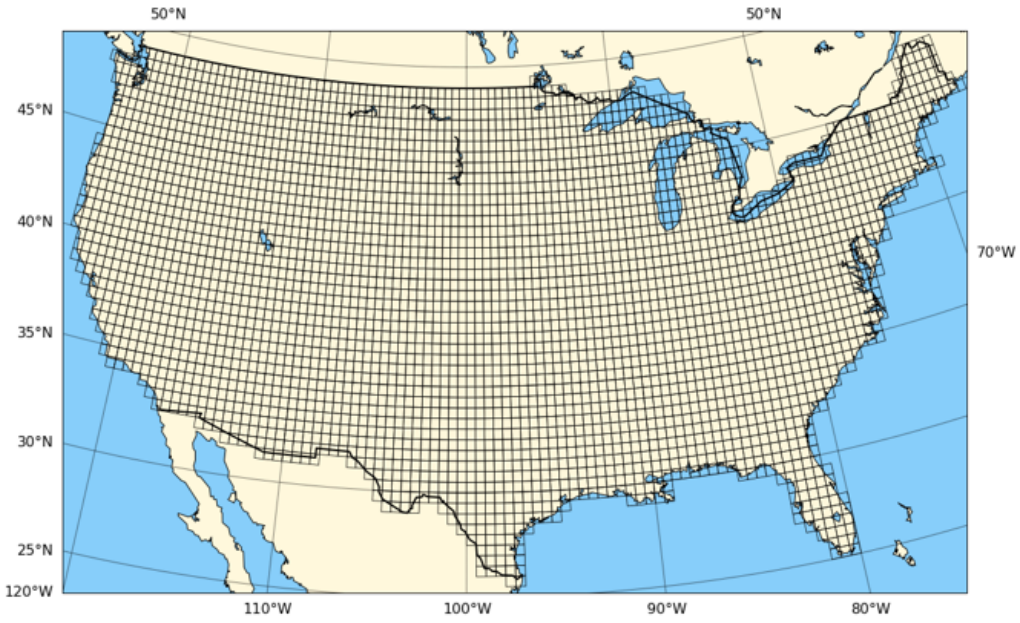


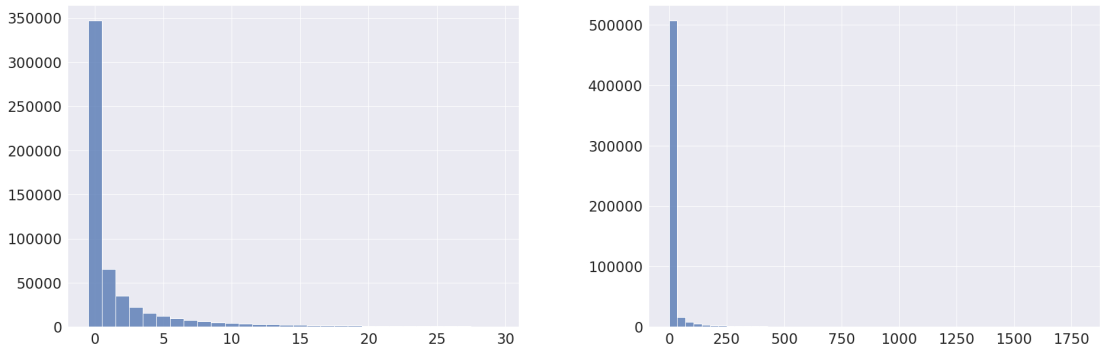
Figure 1: Subdivision of the considered area into 3503 grid cells.

The labels to be predicted are the number of wildfires (CNT) per grid cell and month, as well as the corresponding burned area (BA) in acres. We consider the number of spatially separated wildfire events as an observation related to the first aspect (CNT), and the total burned area of wildfires that occurred in that area as an observation related to the second aspect (BA). The distribution of the observed wildfires can be seen in figure 2.

As evident by the shown histograms for a large portion of the observations no wildfire occured in the given month, making the data very imbalanced.

The dataset contains 35 predictors to be used as input. These can be divided into three categories. The first category are meterological variables:

- 10m U-component of wind (zonal wind speed) $\left[\frac{m}{s}\right]$
- 10m V-component of wind (meridional wind speed) $\left[\frac{m}{s}\right]$



(a) CNT **(b)** BA

Figure 2: Histograms of both target variables. Only data up to the 99% quantile are shown for clarity, as there are some very high outliers.

- Dewpoint temperature (temperature at 2m from ground to which air must be cooled to become saturated with water vapor, such that condensation ensues) $[K]$
- Temperature $[K]$
- Potential evaporation (the amount of evaporation of water that would take place if a sufficient source of water were available) $[m]$
- Surface net solar radiation $\left[\frac{J}{m^2}\right]$
- Surface net thermal radiation $\left[\frac{J}{m^2}\right]$
- Pressure $[Pa]$
- Evaporation $[m]$
- Precipitation $[m]$

All meteorological predictors are obtained from ERA5 reanalysis data from the COPERNICUS Climate Data Service at a resolution of $0.1° \times 0.1°$ which are then scaled and averaged to the required resolution.

The variables of the second category categorize how the land of the corresponding cell is used:

- Cropland rainfed
- Cropland rainfed herbaceous cover
- Mosaic cropland
- Mosaic natural vegetation
- Tree broadleaved deciduous closed to open
- Tree broadleaved deciduous closed
- Tree needleleave evergreen closed to open
- Tree needleleave evergreen closed
- Tree mixed
- Mosaic tree and shrub

- Shrubland
- Grassland
- Sparse vegetation
- Tree cover flooded fresh or brakish water
- Shrub or herbaceous cover flooded
- Urban
- Bare areas
- Water

The land cover classification is produced by the European Union's COPERNICUS service for remote sensing in $300\,m$ spatial resolution. The original 38 classes are reduced to the 18 classes observed in non-negligible quantities in the continental United States. Since this data is at a finer resolution than the rest of the data in this study, it is aggregated to the required resolution as proportions of each grid cell.
The last category is made up of the remaining seven variables:

- Longitude coordinate of grid cell center
- Latitude coordinate of grid cell center
- Proportion of the grid cell overlapping the continental US
- Month of observation
- Year of observation
- Mean of altitude in grid cell $[m]$
- Standard devation of altitude in grid cell $[m]$

Variables related to altitude were made available by the Shuttle Radar Topography Mission at a $90\,m$ spatial resolution. An overview of all variables in tabular form is provided in table 1 appendix A.

Correlation between the predictors and the target variables is relatively low with the highest correlation in absolut terms between potential evaporation and CNT of -0.28. Additionally even the correlation between CNT and BA is only at 0.06 showing very little linear relationship between the features. Corellation matrices of the variables can be found in figures 13 and 14 in appendix B.

2.2 Pre-Processing

The month is transformed into two variables by the following transformation to reflect the cyclical nature of the seasons:

$$month_{sin} = \sin\left(2\pi \cdot \frac{month}{12}\right) \tag{2.1}$$

$$month_{cos} = \cos\left(2\pi \cdot \frac{month}{12}\right) \tag{2.2}$$

All variables that describe proportions (area and the landcover variables) and are therefore already restricted to the interval $[0, 1]$ are not normalized, since they are already in a perfect format for

the input of a learning machine. Additionally, the variable month will not be normalized either, since it has already been transformed according to 2.1 and 2.2.
All remaining variables are normalized using the following transformation:

$$x_{norm} = \frac{x - \bar{x}_{train}}{s_{train}} \tag{2.3}$$

where x_{norm} is the normalized variable, $\bar{x}_{train}$ is the mean of the variable values in the training data set, and s_{train} is its standard deviation in the training data.

2.3 Feature Engineering

Since the data on wind is only available split into U and V components, a new variable is formed from both to provide the model with the total wind speed as well. The new variable is calculated using the Pythagorean theorem:

$$\vec{v}_{wind} = \sqrt{u^2 + v^2} \tag{2.4}$$

where $\vec{v}_{wind}$ is the wind speed independent of direction and u and v are its directional components.

2.4 Data Split

The data was randomly split into training, validation, and test data sets, keeping one year's data together. Years were randomly split to avoid missing trends in the data due to, for example, climate change. This method resulted in the following split:

- Training: 1993, 1994, 1996, 1999, 2000, 2001, 2002, 2005, 2006, 2007, 2008, 2009, 2010, 2014, 2015
- Validation: 1995, 1997, 2003, 2012
- Testing: 1998, 2004, 2011, 2013

with approximately 65.2 % of data available for training and 17.4 % each for validation and testing.

3 Benchmark Model

To better assess the machine learning models that will be implemented later, two simple models are first experimented with to generate a benchmark.

3.1 Naïve Average

As the most naïve benchmark, we simply predict the average of the targets in the training and validation datasets $\overline{CNT}_{train}$ and $\overline{BA}_{train}$ for each data point. This approach results in an Mean Squared Error (MSE) of 38.6506 for CNT and 14137907.7077 for BA on the test data providing us with the most basic reference values the subsequent models have to beat.

3.2 Linear Regression

A second benchmark is generated using two linear regression models, one for each target. All predictors are used as input parameters. The models are trained on both training and validation data sets, since the validation data is not needed for early stopping or similar purposes for these models.

The trained models are then evaluated on the test data, yielding an MSE of 32.5829 ($R^2 = 0.1563$) for CNT and 11506993.1292 ($R^2 = 0.0043$) for BA. From this result we can already observe that predicting values for BA is very difficult, with the linear regression model performing poorer than the naïve prediction of the mean.

The large condition number of the exogenous matrix of 1330, calculated as the ratio of the largest to smallest singular value, indicates strong multicollinearity (Belsley et al. 2005, pp. 100-104). This can be explained as a result of the land use variables. The proportions representing these values sum up to 1 or nearly 1 (some rare categories were removed from the data set see section 2.1).

The addition of the wind speed parameter did not lead to any improvement of the predictions. Since this nonlinearly engineered variable should in theory add the most value to the linear model, but did not, it will not be included in the further course of this work.

4 Machine Learning Models

In this report three different machine learning models are implemented. Within this chapter, the structure of the different models is explained. The evaluation and comparison of the models can be found in chapter 5.

All models are trained in the cloud-based Jupyter notebook environment of Google Colab Pro. The hardware specifications of the Google Colab Pro runtime environment consist of two Intel Xeon CPUs running at 2.30 GHz, an NVIDIA Tesla P100 graphics card with 16 GB of GPU memory and 26.75 GB of available RAM.

MLP and CNN models are implemented using the open source deep learning library Keras (Chollet et al. 2015). The random forest models are created using the sklearn library (Pedregosa et al. 2011).

4.1 Multilayer Perceptron

A Multilayer Perceptron (MLP) is implemented as the first machine learning architecture. All tested MLP models are trained with the Adam optimization algorithm (Kingma & Ba 2014), with a learning rate of 0.001. A batch size of 1024 is used during training. Since the task is a regression problem, the MSE serves as the loss function needed to update the adjustable weights during training. The models are trained a maximum number of 100 epochs. However, training is terminated prematurely if there is no improvement in model predictions for 10 consecutive epochs. The parameterization according to the best previous epoch is then used. The training data is shuffled before each epoch.

Two separate models are trained for each of the two target variables. On the one hand, this allows the models to specialize in one task and, on the other hand, it ensures that the considerably higher loss values of BA do not dominate in training.

4.1.1 Baseline MLP

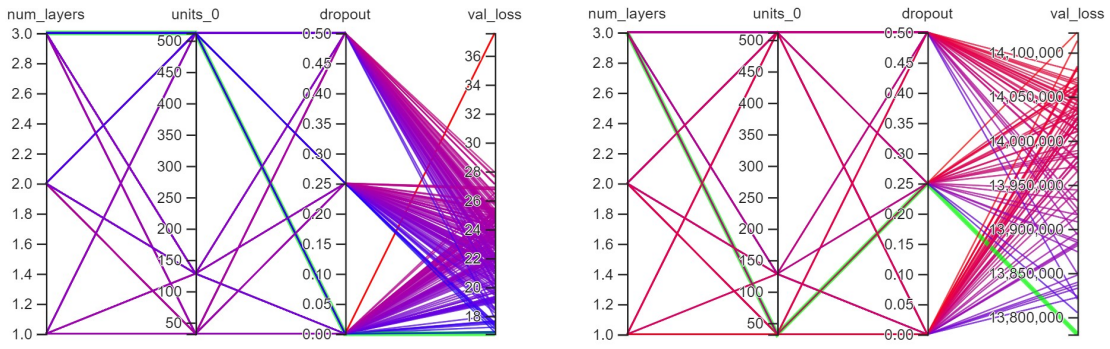
As a first step, a very simple MLP is implemented and trained. It consists of only one hidden layer with 32 neurons and one output neuron. Both layers use the ReLU activation function. Normally, ReLU is not applied to the output layer, but this is not problematic here, since only non-negative values have to be predicted. This simplistic implementation already outperforms the linear regression for predicting CNT with an MSE of 24.481 on the test dataset. However, when trying to predict BA, it fails the benchmark of the naive average, in the same way as linear regression.

4.1.2 Hyperparameter Optimization of MLP

In order to identify an as good as possible selection of hyperparameters with given computational constraints, a structured hyperparameter optimization is performed.

The following hyperparameters with associated search space are examined:

- number of hidden layers: $\{1, 2, 3\}$
- number of neurons per hidden layer: $\{32, 128, 512\}$
- dropout rate: $\{0, 0.25, 0.5\}$



(a) CNT **(b)** BA

Figure 3: A Visualiziation of both hyperparameter sweeps using TensorFLOW's Tensorboard. The number of neurons (units) per hidden layer can only be displayed for the first one, since this is the only layer that is always included. The green line marks the combination of parameters with the best performance.

The number of neurons can vary from layer to layer if there are multiple layers.

The search is performed with the Keras Tuner using the hyperband algorithm (Lisha Li et al. 2018), which performes a turnament-style sweep of the search space. Each permutation of the hyperparameters is trained for a few epochs, then evaluated and the worst are discarded, while the rest are trained for a few more epochs until only the best combination of parameters remains. The models that made it to the final round are trained for a maximum of 50 epochs before being compared.

After 270 trials and 72:29 minutes and 77:44 minutes of search time for CNT and BA, respectively, the following pairings proved to be the most successful:

- CNT
  - Number of hidden layer: 3
  - Neurons in first hidden layer: 512
  - Neurons in second hidden layer: 512
  - Neurons in third hidden layer: 32
  - Dropout rate: 0.0
- BA
  - number of hidden layer: 3
  - Neurons in first hidden layer: 32
  - Neurons in second hidden layer: 512
  - Neurons in third hidden layer: 512
  - dropout rate: 0.25

An overview of both sweeps can be viewed in Figure 3.

Since both models use the maximum number of layers, another model with 5 hidden layers is implemented and tested in each case. This architecture only leads to a better predictive power on the training data, but not on the test data, implying overfitting.

4.2 Convolutional Neural Network

As already described in chapter 2.1, the data is available in gridded structure. Additionally, wildfires and weather conditions of surrounding cells have an influence on each other. In order to recognize and learn from these spatial relationships, Convolutional Neural Network (CNN) models are implemented.

4.2.1 Data Preparation

A CNN requires three-dimensional data points in the form of $x \in \mathbb{R}^{H \times W \times C}$ as input where H and W are the height and width of the associated grid (normally the resolution of an image) and C is the number of channels. C would typically be 3 for a color image, but in our case the size of C is determined by the number of available input variables.

An intuitive approach to generate this data structure would be to pad the already given grid so that it is rectangular and then use the entire grid as input for the CNN. The vector of target variables of all grid cells would be the output. However, this approach would result in a data set of length 161. One data point for each month. Since this is too few data points for a deep learning model, a different approach is employed.

A grid is first created that rectangularly encloses the entire grid in Figure 1, the so-called position grid. In the available data set, all grid cells occur in the same order for each month. This is used to enter, with the help of the coordinates of each cell, its placement numbering within the month in the position grid. Those cells of the position grid that are not present in the data set (e.g. over the ocean) are assigned a -1.

The next step is to generate an 11×11 grid for each data point using the position grid, with the original cell of the data point as its center.

11 was chosen here because this was the largest expansion that the available memory could handle. One could split the data sets even further, create and save them individually, and then load them into memory during training using a loader. This would allow even greater expansion despite given RAM limitations. But since this approach already allows for an expansion in each direction of approximately 275 km and is mainly intended to test whether a spatial approach would be useful, only 5 grid cells in each direction are considered.

The respective variable vectors of the surrounding cells are now placed on this grid in the third dimension at the corresponding position. Eventually, this procedure results in a four-dimensional input data tensor in the form $X \in \mathbb{R}^{563983 \times 11 \times 11 \times 36}$ with which batch training can be executed. A schematic overview of this processing pipeline is shown in Figure 4.

4.2.2 Hyperparameter Optimization of CNN

A similar hyperparameter optimization as in chapter 4.1.2 is also performed for the CNN models. The dropout search space is identical to chapter 4.1.2. Likewise, between one and three layers are tested, only that these are now composed of convolutional layers. The kernel size was set to 3×3 for all layers with valid padding. As a consequence, the feature maps become increasingly smaller with each successive layer, which is why no pooling is applied to avoid downsampling too quickly. After the last layer, the generated feature map is flattened into a vector whose length depends on the number of convolutional layers implemented. Subsequently, a one-dimensional prediction is generated using a single output neuron, in the same way as for the MLP. The filter quantities of 32, 128 and 512 are tested for each layer.

The optimization algorithm, loss function, early stopping and batch size used are the same as those for the MLP, see chapter 4.1.

After another 270 attempts, but this time 389:51 minutes and 349:52 minutes of search time for CNT and BA, respectively, the following pairings emerged as the most effective within the search space:

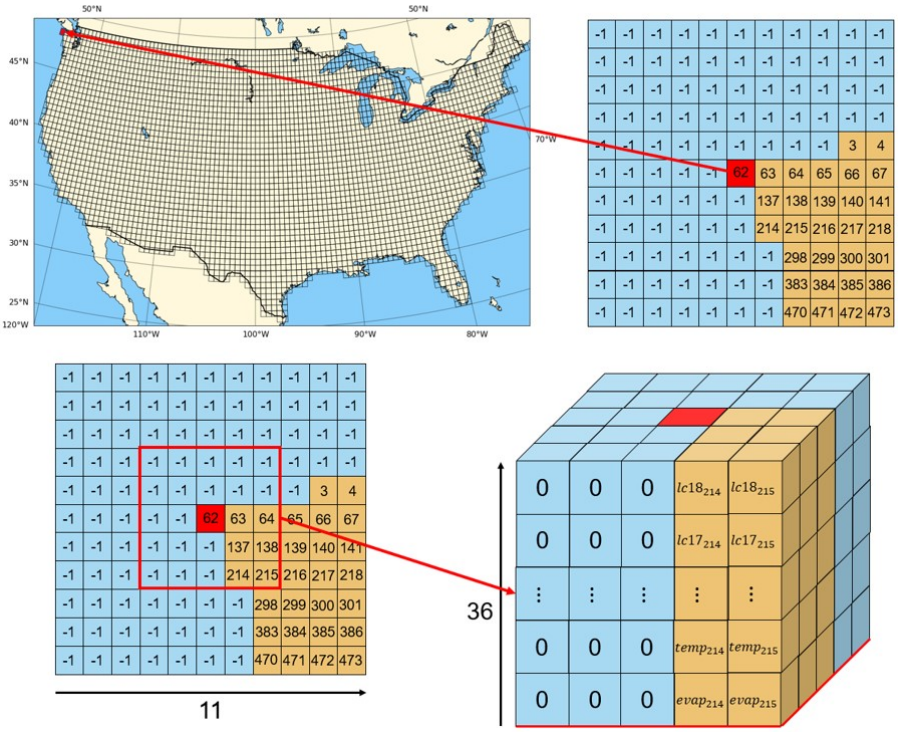
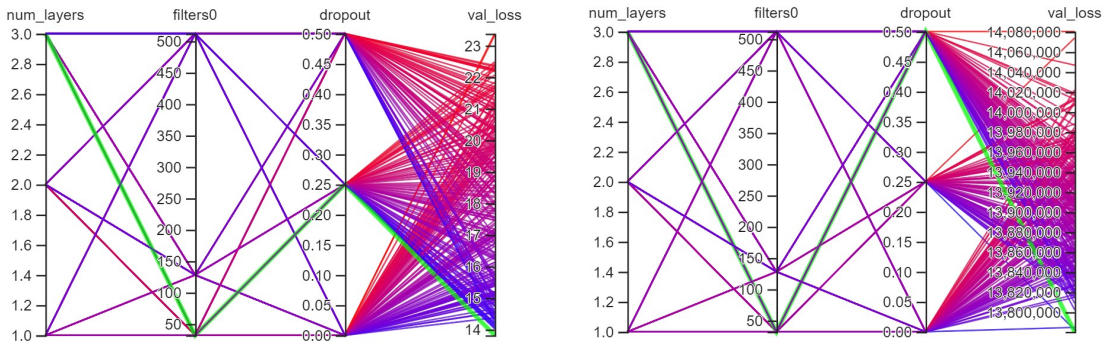


Figure 4: Schematic overview of the data preparation process for the CNN models. To ensure clarity in the illustration, the final step only shows the portion of the grid highlighted with the red square.

- CNT
  - number of convolutional layers: 3
  - number of filters in first convolutional layer: 32
  - number of filters in second convolutional layer: 128
  - number of filters in third convolutional layer: 512
  - dropout rate: 0.25
- BA
  - number of convolutional layers: 3
  - number of filters in first convolutional layer: 32
  - number of filters in second convolutional layer: 128
  - number of filters in third convolutional layer: 128
  - dropout rate: 0.5

The Tensorboard visualization of both sweeps can be viewed in Figure 5.

Once again, the optimized models use all available layers. Therefore, another CNN is implemented using another convolutional layer with 1024 and 518 filters for CNT and BA, respectively. These filter counts have been chosen as one power of 2 more than the number of filters in the previous convolutional layer. This results in a feature map of size 1×1. Just like the MLP, this deeper architecture leads to more overfitting without a clear improvement of the predictions on the test data.



(a) CNT **(b)** BA

Figure 5: A Visualiziation of both hyperparameter sweeps using TensorFLOW's Tensorboard. The number of filters per convolutional layer can only be displayed for the first one, since this is the only layer that is always included. The green line marks the combination of parameters with the best performance.

4.3 Rondom Forest Regressor

As a third machine learning architecture a Random Forest Regressor (RFR) is used. To identify matching hyperparameters, an exhaustive cross-validated gridsearch is conducted over the following search space:

- Number of trees: $\{10, 50, 100\}$
- Maximum depth of the trees: $\{5, 10, 15, 20\}$
- Whether bootstrap samples or the whole dataset is used when building trees.

The combined training and validation data are divided into five folds. A model for each combination of parameters is trained on four of the folds and evaluated on the one left out of training. This is done five times so that each data point was in the validation data once. This leads, with the 24 possible candidates, to 120 runs.

The model specifications with the lowest average MSE on the left out fold for the two different tasks are the following:

- CNT
  - Number of trees: 100
  - Maximum depth of trees: 20
  - Bootstrapping
- BA
  - Number of trees: 100
  - Maximum depth of trees: 5
  - Bootstrapping

where the RFR for CNT shows strong indications of overfitting with an MSE of 3.1341 on training and validation data and 16.0750 on the test dataset. This effect can be counteracted by limiting the maximum depth, if a slight loss in prediction accuracy on the test data is tolerated. An illustration of the relationship between performance on training and validation data on the one hand and on test data on the other hand is shown in Figure 6. In the following part of this work, a depth of 13 is used as it provides a balance between overfitting and predictive power.

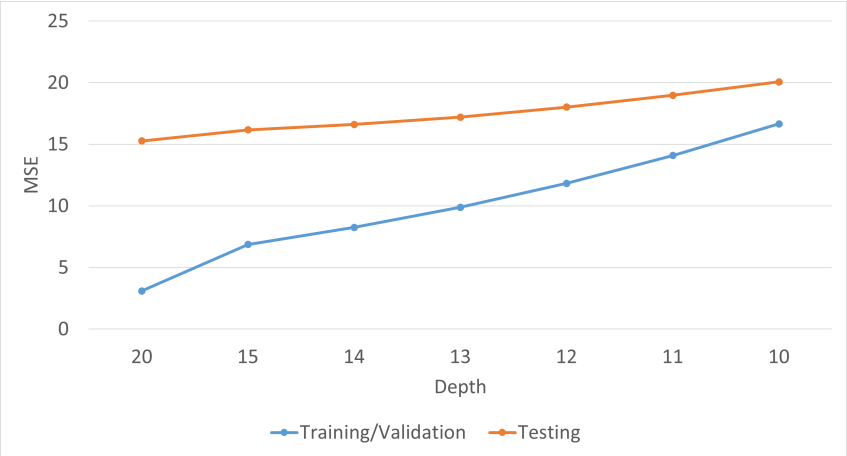


Figure 6: An illustration of the relationship of MSE on training/validation data and on test data for CNT that implicates overfitting. There is a break on the x-axis in between 20 and 15.

A similar pattern can be observed for BA. Again, overfitting decreases as tree depth decreases, while performance on the test data remains the same. However, the meaningfulness of this observation is limited here, as the predictive power of the RFR is not better than predicting the average. For the rest of this report, a depth of 5 is chosen. An equivalent overview of the relationship between tree depth and performance can be seen in Figure 15 in Appendix B.

Furthermore, it was also tested whether increasing the number of trees to 200 would lead to a better result. For both use cases, however, the predictions on the test data could not be significantly improved, which is why the respective RFRs with 100 trees are used in the remainder of this report.

4.4 Ensemble

As a final alternative, an ensemble of the best models of the three architectures implemented so far will be tested. For this, the average of the model predictions is computed and used as the prediction of the ensemble.

5 Results

In this chapter, the results of the different models are evaluated and discussed, and some further considerations regarding feature importance and quality of predictions are made.

5.1 CNT

For the prediction of CNT, the benchmarks can be clearly beaten. As the best individual model, the CNN emerged with an MSE on the test data of 16.6803. This indicates that a spatial view can add value and that the network was able to learn from these interrelationships between neighboring cells.

The only model that performs better than the CNN is the ensemble of models with an MSE on the test data of 16.0112. This demonstrates the effect of using a combination of models to obtain better predictions, even when using the best model together with two slightly less powerful ones.

Both MLP and RFR can significantly improve over the benchmarks as well with an MSE of 18.9558 and MSE of 17.1962, respectively.

The performance of all benchmarks and machine learning models can be viewed in Figure 7. The comparison of the error metric on the training dataset is not perfect because only the neural networks used a validation dataset for early stopping that is not directly trained on. The rest of the models use this data for training as well. However, the comparison of the test performances is based on the same data set for all models.

5.2 BA

Predicting values for BA has proven to be much more difficult. The best RFR was not able to produce better predictions than the benchmarks at an MSE of 11668276.0668 in testing.

The two neural networks were able to perform slightly better, with errors of 11440523.8580 and 11417953.6109 for MLP and CNN, respectively. Again, the CNN is able to produce slightly better predictions than the MLP.

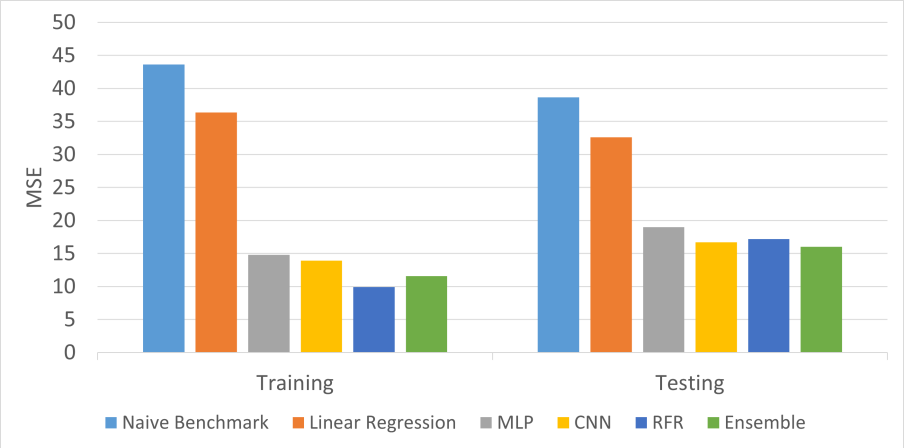


Figure 7: MSE of the different tested models on Training and Test datasets for CNT.

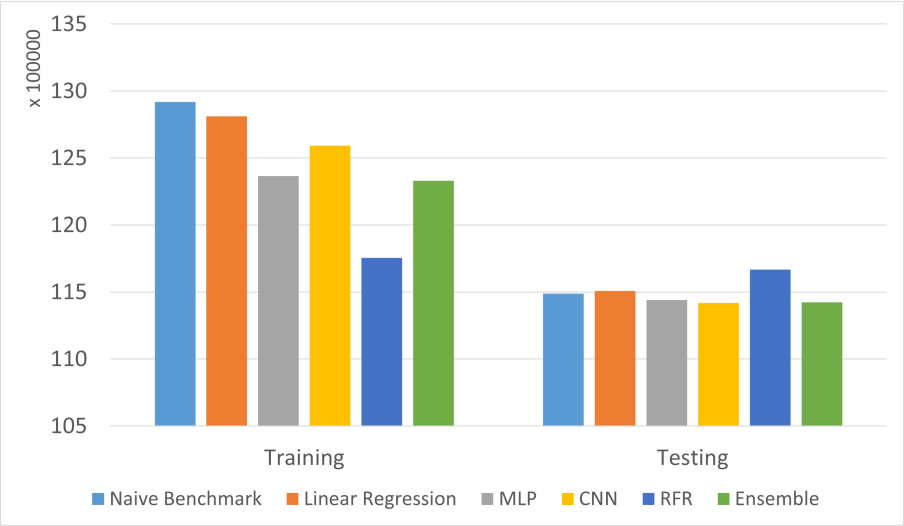


Figure 8: MSE of the different tested models on Training and Test datasets for BA. The y-axis does not start at 0 in this chart.

In this use case, the ensemble is not able to outperform the best individual architecture with an MSE of 11423795.5982.

A diagram of the error metrics of all models for BA is shown in Figure 8. It can be clearly observed that the more complex models do not lead to a meaningful improvement of the predictions compared to the simpler benchmarks.

5.3 Feature Importance in RFR

5.3.1 Impurity-Based Feature Importance

To examine the features for their importance to the RFR, we first compute the impurity-based feature importance. The importance of a feature is computed as the total reduction of the criterion brought by that feature. Here, we will concentrate on the feature importances for the prediction of CNT since this is the only RFR able to beat the benchmarks.

The features ranked by importance are shown in Figure 9. Potential evaporation, thermal radiation and the associated longitude are the most important features according to this metric. All of them make some intuitive sense for why they would be important. Potential evaporation, especially coupled with low actual evaporation might indicate dryness. Thermal radiation implies warm soil and there are more fires in the south than in the north. The following variable (lc_{16}) is code for urban areas. This can be explained with higher fire frequency where there is more population density.

5.3.2 Permutation Importance

Since impurity-based feature importance is computed on statistics derived from the training dataset, importance can also be high for characteristics that do not predict the outcome variable, if the model is able to use them to overfit. We have seen some amount of overfitting by the RFR model and therefore calculate the permutation importance (Breiman 2001) as an alternative.

In order to calculate the permutation importance first, a baseline metric is evaluated on the test dataset. Next, a feature column from the dataset is permuted and the metric is evaluated again. The permutation importance is the difference between the baseline metric and metric from permutating the feature column.

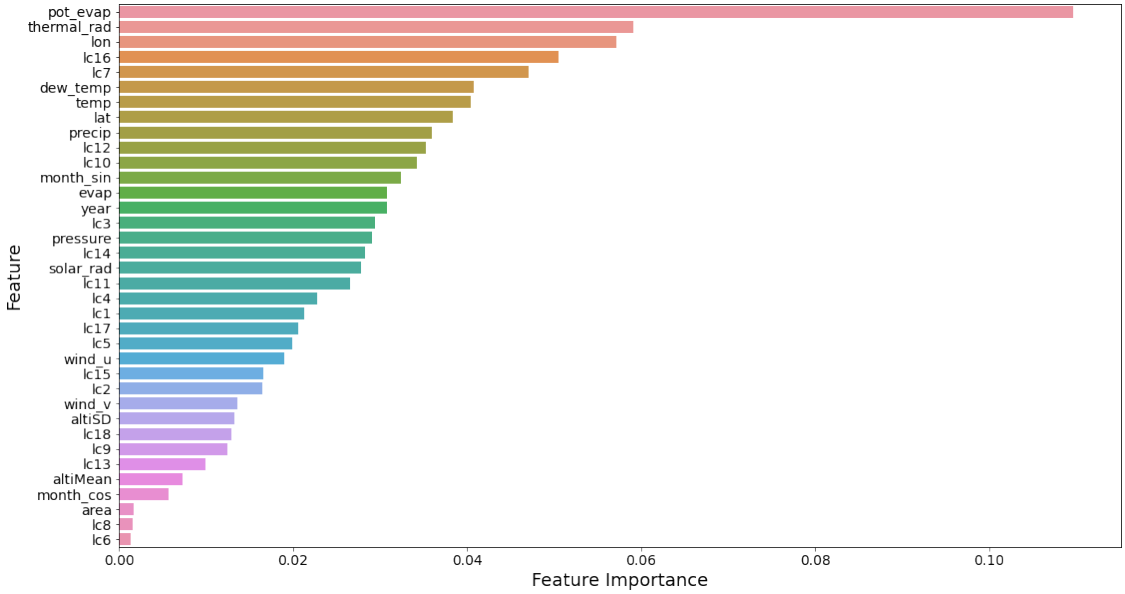


Figure 9: Each feature ranked by importance for the prediction of CNT.

The ranked permutation importances are displayed in Figure 10. Potential Evaporation and the urban landuse are very important features as well. But the most important feature according to this metric is the landuse variable "tree cover flooded fresh or brakish water" which presumably indicates high fire resistance.

5.4 Analysis of Predictions

To further examine the predictions of the models, scatterplots are considered. These contrast the predictions of the models with the observations, allowing patterns to be identified. Figure 11 shows the plots for the prediction of CNT of all four machine learning models.

Here we can see that all models systematically underestimate the number of wildfires when there is a high incidence. This effect can be observed more strongly in neural nets. However, it is also slightly present in the RFR predictions. Patterns of this type suggest that neural networks are failing to detect the very relationships in the data that lead to many fires, which would speak against their use in wildfire prevention. The better overall performance of the CNN can be explained by the fact that almost all data points contain very few wildfires. The CNN can provide better predictions there, even though it has difficulty predicting high values.

For the prediction of BA, we can observe similar, albeit much stronger, effects, shown in Figure 12. In this use case, the RFR has the most difficulty predicting higher values. All of the models rarely or never predict burned areas greater than 20000 acres, even though there are many instances with higher observed values in the dataset.

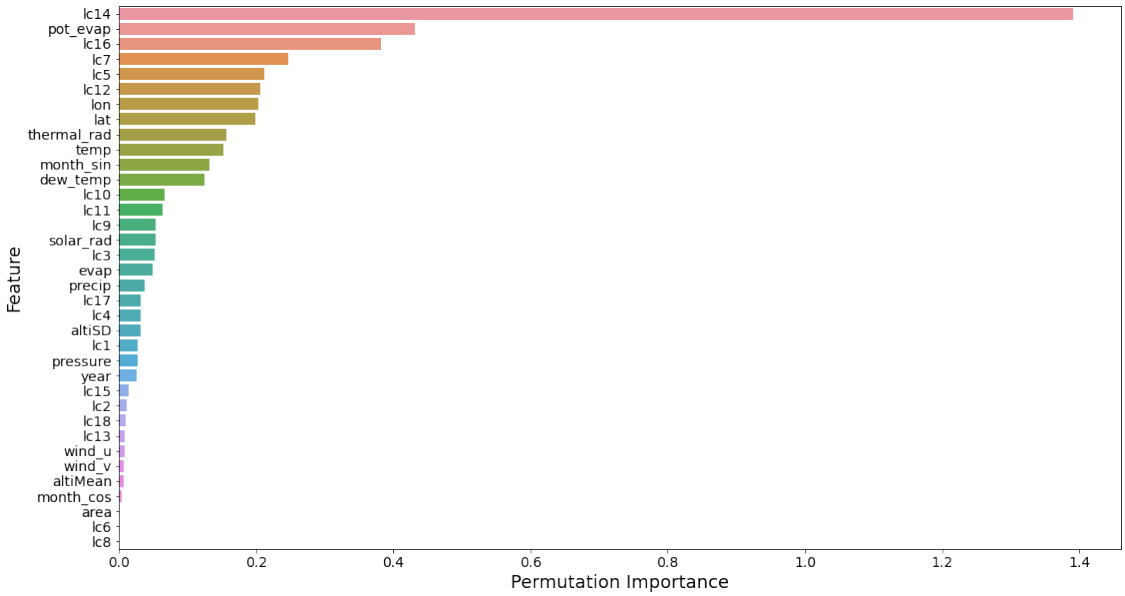
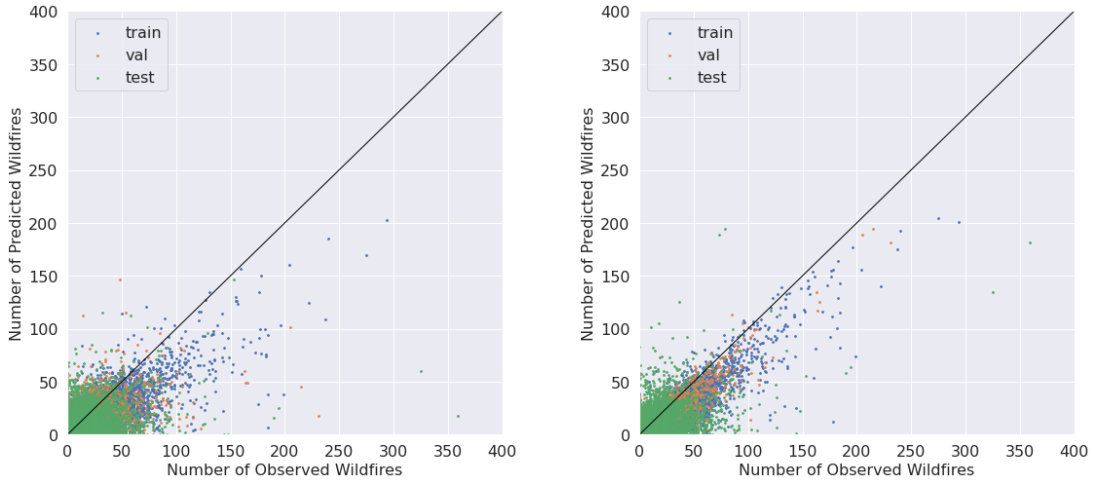
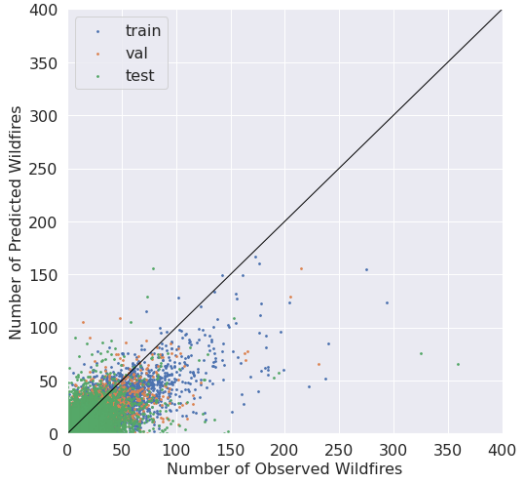


Figure 10: Each feature ranked by permutation importance for the prediction of CNT.

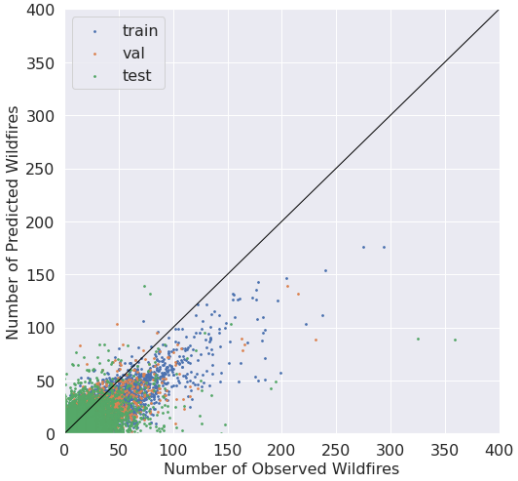


(a) MLP

(b) RFR

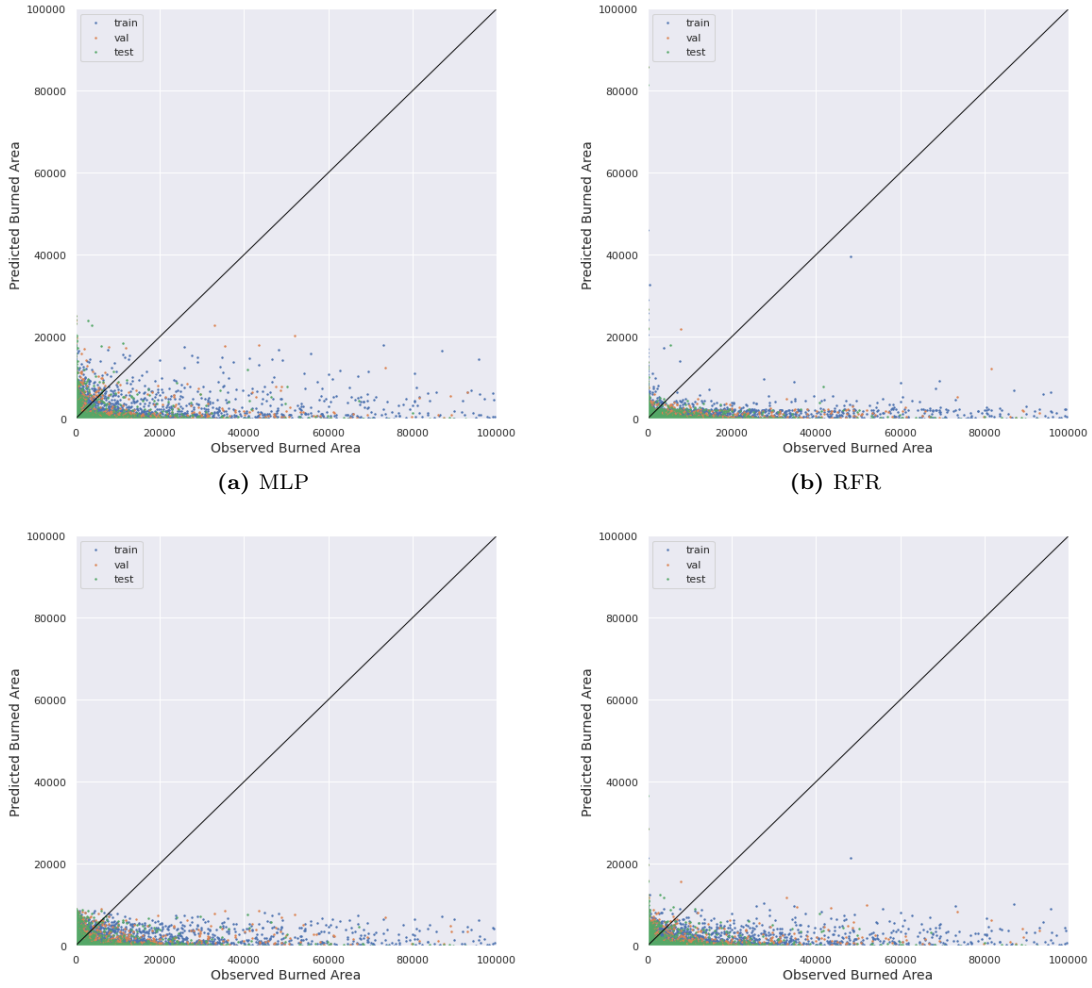


(c) CNN



(d) Ensemble

Figure 11: A comparison of observed CNT and predictions of each machine learning model. The black line indicates perfect predictions.



(a) MLP **(b)** RFR

(c) CNN **(d)** Ensemble

Figure 12: A comparison of observed BA and predictions of each machine learning model. The black line indicates perfect predictions.

6 Conclusion

6.1 Summary

The aim of this work was to provide an efficient way to predict the number of wildfires, as well as the area burned, in the continental US from 1993 to 2015. The focus was on the investigation of different machine learning models for this purpose.

To this end, two simple benchmarks were first generated against which the implemented machine learning models can be measured. Next, the three different architectures MLP, CNN and RFR were implemented and evaluated for the given tasks. Finally, an ensemble of the three models was formed and evaluated as well.

For the predictions of CNT, the ensemble of MLP, CNN and RFR can improve the individual models with an MSE of 16.0112 in testing. The stand-alone model with the best performance on the test data is the CNN, suggesting that there is added value in including spatial information.

The predictions for BA have proven to be much more difficult than CNT. Here, the CNN is the best model with an MSE of 11417953.6109 on the test data. However, this performance is only slightly better than that of the most simple benchmark.

6.2 Outlook

Since the CNN models have shown the best performance among the individual architectures, further experiments should be carried out with them. An increase of the considered surroundings of each cell, could lead to further improvements in this case. A possible approach for this has already been described in section 4.2.1.

As observed in section 5.3, the landuse variable "urban" plays an important role for the RFR models. To build on this observation, a population density variable could be introduced. The data needed for this would be available at a resolution of at least county level. These could be matched to the respective cells using the coordinate information and thus be included in the input.

Furthermore, a much wider range of different machine learning models could be implemented and tested building on this work.

In contrast to the approach here, one could also train models that predict both CNT and BA simultaneously. Possibly this could lead to useful synergy effects. However, the targets definitely need to be normalized here to avoid too much focus on BA, with its larger values.

In addition, it must be noted that no clear statement can be made about the statistical significance of the different error metrics of different model architectures. A comparison of the accuracy of the models with the help of a statistical hypothesis test should be part of further work.

In summary, machine learning models show great potential for predicting the number of wildfires, but could not prove effective for predicting burned area in the scope of this report.

Bibliography

Belsley, D. A., Kuh, E. & Welsch, R. E. (2005), *Regression diagnostics: Identifying influential data and sources of collinearity*, John Wiley & Sons.

Breiman, L. (2001), 'Random forests', *Machine learning* **45**(1), 5–32.

Chollet, F. et al. (2015), 'Keras'.

Flannigan, M. D., Amiro, B. D., Logan, K. A., Stocks, B. J. & Wotton, B. M. (2006), 'Forest fires and climate change in the 21st century', *Mitigation and adaptation strategies for global change* **11**(4), 847–859.

Jones, M. W., Smith, A., Betts, R., Canadell, J. G., Prentice, I. C. & Le Quéré, C. (2020), 'Climate change increases the risk of wildfires', *ScienceBrief Review* **116**, 117.

Kingma, D. P. & Ba, J. (2014), 'Adam: A method for stochastic optimization'.

Lisha Li, Kevin Jamieson, Giulia DeSalvo, Afshin Rostamizadeh & Ameet Talwalkar (2018), 'Hyperband: A novel bandit-based approach to hyperparameter optimization', *Journal of Machine Learning Research* **18**(185), 1–52.
URL: *http://jmlr.org/papers/v18/16-558.html*

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V. et al. (2011), 'Scikit-learn: Machine learning in python', *the Journal of machine Learning research* **12**, 2825–2830.

Pörtner, H.-O., Roberts, D. C., Adams, H., Adler, C., Aldunce, P., Ali, E., Begum, R. A., Betts, R., Kerr, R. B., Biesbroek, R. et al. (2022), 'Climate change 2022: Impacts, adaptation and vulnerability', *IPCC Sixth Assessment Report* .

Thomas Opitz (2021), 'Extreme value analysis 2021 conference: Data competition'.
URL: *https://www.maths.ed.ac.uk/school-of-mathematics/eva-2021/competitions/data-challengesec-9*

A Tables

Name	Description	Unit
lon	longitude coordinate of grid cell center	
lat	latitude coordinate of grid cell center	
area	proportion of the grid cell overlapping the continental US	
month	month of observation	
year	year of observation	
altiMean	mean of altitude in grid cell	$[m]$
altiSD	standard devation of altitude in grid cell	$[m]$
clim1	10m U-component of wind (zonal wind speed)	$\left[\frac{m}{s}\right]$
clim2	10m V-component of wind (meridional wind speed)	$\left[\frac{m}{s}\right]$
clim3	dewpoint temperature (temperature at 2m from ground to which air must be cooled to become saturated with water vapor, such that condensation ensues)	$[K]$
clim4	temperature	$[K]$
clim5	potential evaporation (the amount of evaporation of water that would take place if a sufficient source of water were available)	$[m]$
clim6	surface net solar radiation	$\left[\frac{J}{m^2}\right]$
clim7	surface net thermal radiation	$\left[\frac{J}{m^2}\right]$
clim8	pressure	$[Pa]$
clim9	evaporation	$[m]$
clim10	precipitation	$[m]$
lc1	cropland rainfed	
lc2	cropland rainfed herbaceous cover	
lc3	mosaic cropland	
lc4	mosaic natural vegetation	
lc5	tree broadleaved deciduous closed to open	
lc6	tree broadleaved deciduous closed	
lc7	tree needleleave evergreen closed to open	
lc8	tree needleleave evergreen closed	
lc9	tree mixed	
lc10	mosaic tree and shrub	
lc11	shrubland	
lc12	grassland	
lc13	sparse vegetation	
lc14	tree cover flooded fresh or brakish water	
lc15	shrub or herbaceous cover flooded	
lc16	urban	
lc17	bare areas	
lc18	water	

Table 1: All 35 predictor variables available in the original dataset.

B Figures

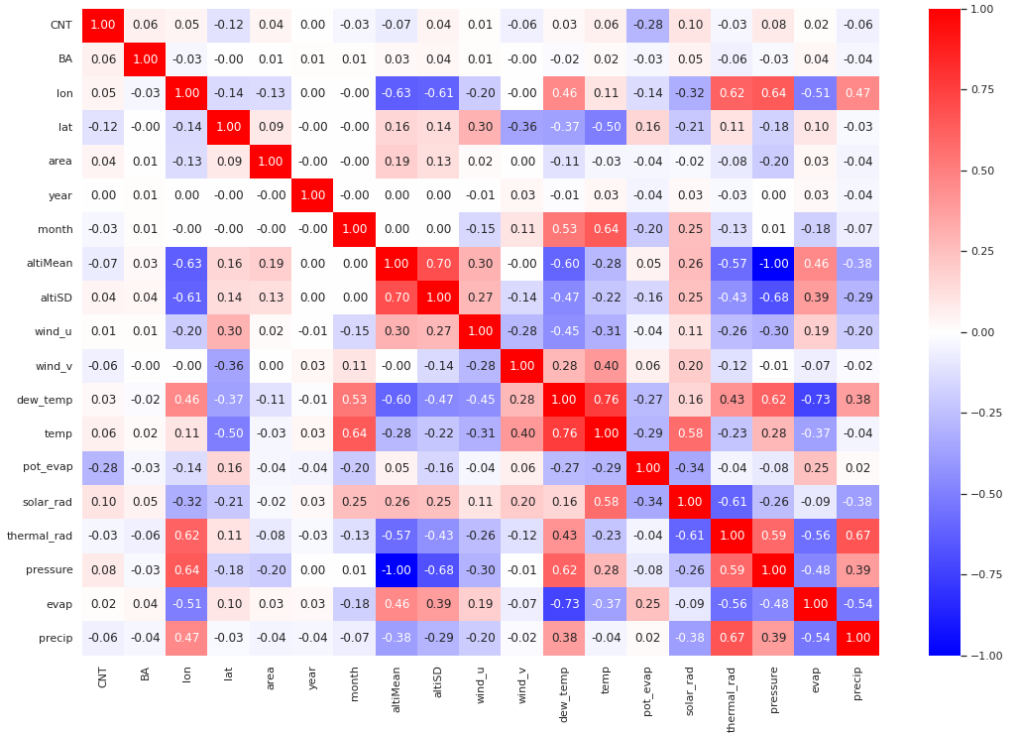


Figure 13: Correlation matrix of both targets with the non-landuse variables

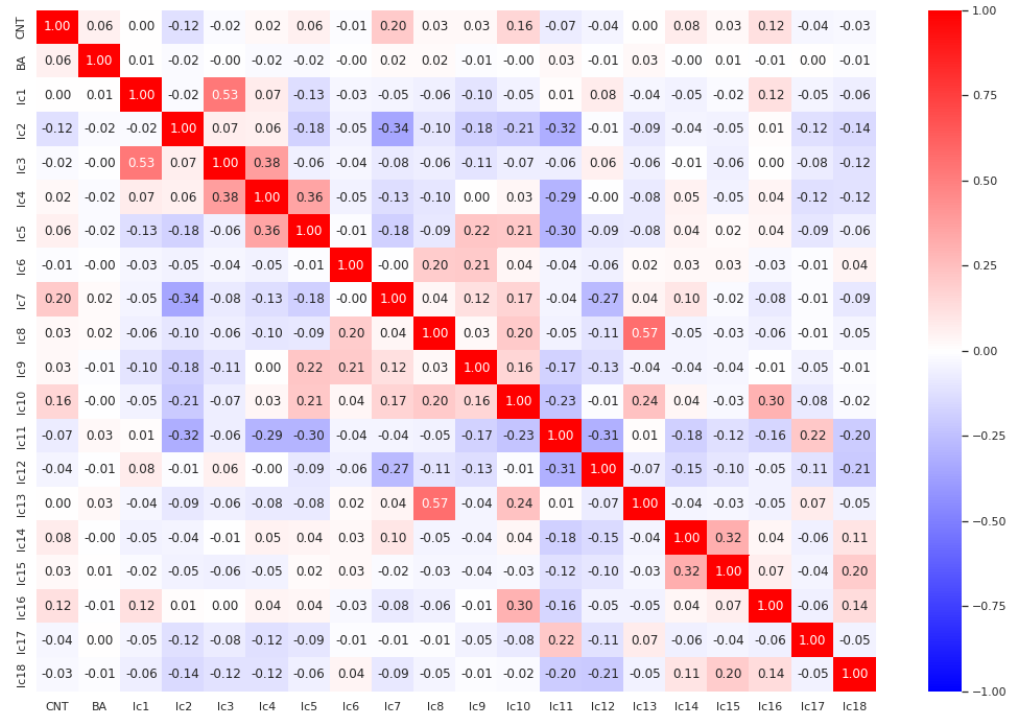


Figure 14: Correlation matrix of both targets with the landuse variables

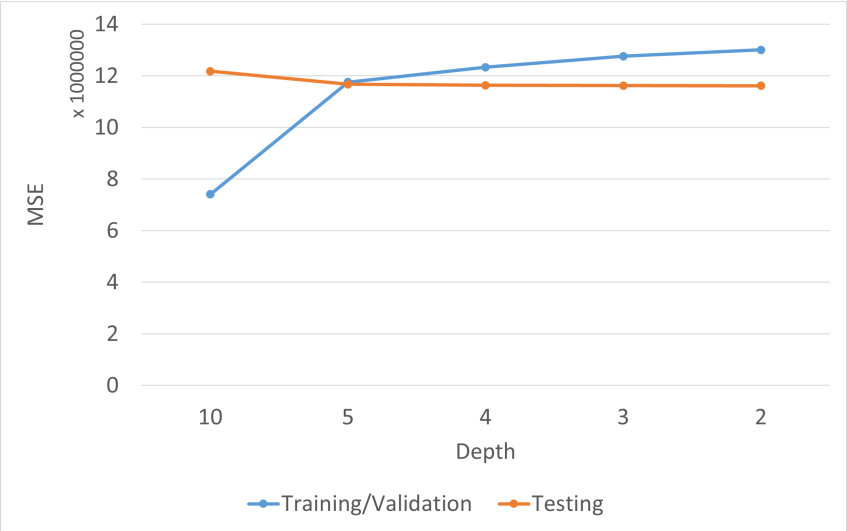


Figure 15: An illustration of the relationship of MSE on training/validation data and on test data for BA. There is a break on the x-axis in between 10 and 5.