CEE 498 Project 10: Water Withdrawal Prediction

This manuscript (permalink) was automatically generated from gooood-night/Project10_Water_Withdrawal_Prediction@b8740cc on December 3, 2020.

Authors

- **Kathryn Grace Gunderson**

  Civil and Environmental Engineering, University of Illinois at Urbana-Champaign

- **Yanan Chen**
  · gooood-night
  Civil and Environmental Engineering, University of Illinois at Urbana-Champaign

Abstract

1. Introduction

1.1 Background

Water is a kind of crucial and finite resources. Only one percent of the water in the world is readily accessible for human use. With the increase of population and the development of society and economics, many countries are faced with water scarcity. In recent years, climate change and land use changes have affected hydrological processes, bringing high uncertainty to precipitation and river flow. Consequently, the water supply systems are under huge pressure. In this case, improving the accuracy of water consumption prediction is of vital importance, which is not only necessary to design new water facilities, but also essential to optimize the operation and management of the existing water supply systems.

Water consumption is affected many factors, such as water availability, climate, demographics, economics, etc. Machine learning methods have the ability to capture the correlations between independent variables and the non-linear relationships between features and the target variables, so ML methods have been widely applied to complex water consumption modeling and water demand prediction. Villarin and Rodriguez-Galiano (2019) applied two tree-based machine learning methods, including classification and regression trees (CART) and random forest (RF), to the water demand modeling in the city of Seville, Spain at the census tract level. They choose 16 variables that represent the sociodemographic and urban building characteristics of urban area as the independent variables to predict water consumption per capita. The results show that RF model performs better than CART model.

1.2 Objective

The goal of the project is to use machine learning algorithms to predict water withdrawal per capital in the world. Compared with previous, our project will use worldwide data at national scale rather than the data from one city or one country, and we will involve more independent variables such as precipitation, renewable water resources, GDP per capital in order to improve the prediction results. Meanwhile, we plan to use two machine learning algorithms: Neural networks and Random Forest to predict the target variable respectively and compare the model performances. We hope to find a ML method that produces more accurate predictions of water withdrawal, which allows improvement of water resources planning and management.

2. Data

2.1 Raw Data

The raw dataset was obtained from FAO's Global Information System on Water and Agriculture (AQUASTAT). AQUASTAT provides free access to over 180 variables and indicators by country from 1960s, which are mainly related to water resources, water uses and agricultural water management. In this project, we intend to predict annual total water withdrawal per capita and select 13 variables as features based on previous studies, which provide information about geography, economic, climate, water resources utilities, etc. So, the raw dataset used includes 1 dependent variable and 13 features that span 200 countries across 8 different five-year periods from 1978 to 2017.

2.2 Exploratory Data Analysis

This project aims at predicting water withdrawal per capita. Water withdrawal per capita is affected many factors, such as water availability, climate, demographics, economics, etc. Let's first look at the water withdrawal per capita.

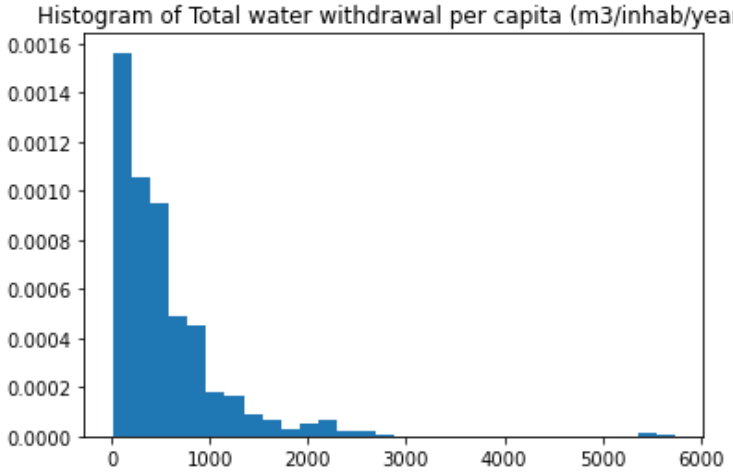

Figure 1: Histogram of the target variable

As shown in Figure 1, the distribution of total water withdrawal is not symmetric. For 75% entries, the annual total water withdrawal per capita is less than 704.7 m3/inhab/year, while the maximum annual total water withdrawal per capita is 5739 m3/inhab/year. The difference between countries is huge. Let us closely look at the average annual total water withdrawal per capita during 2013-2017 in each country.

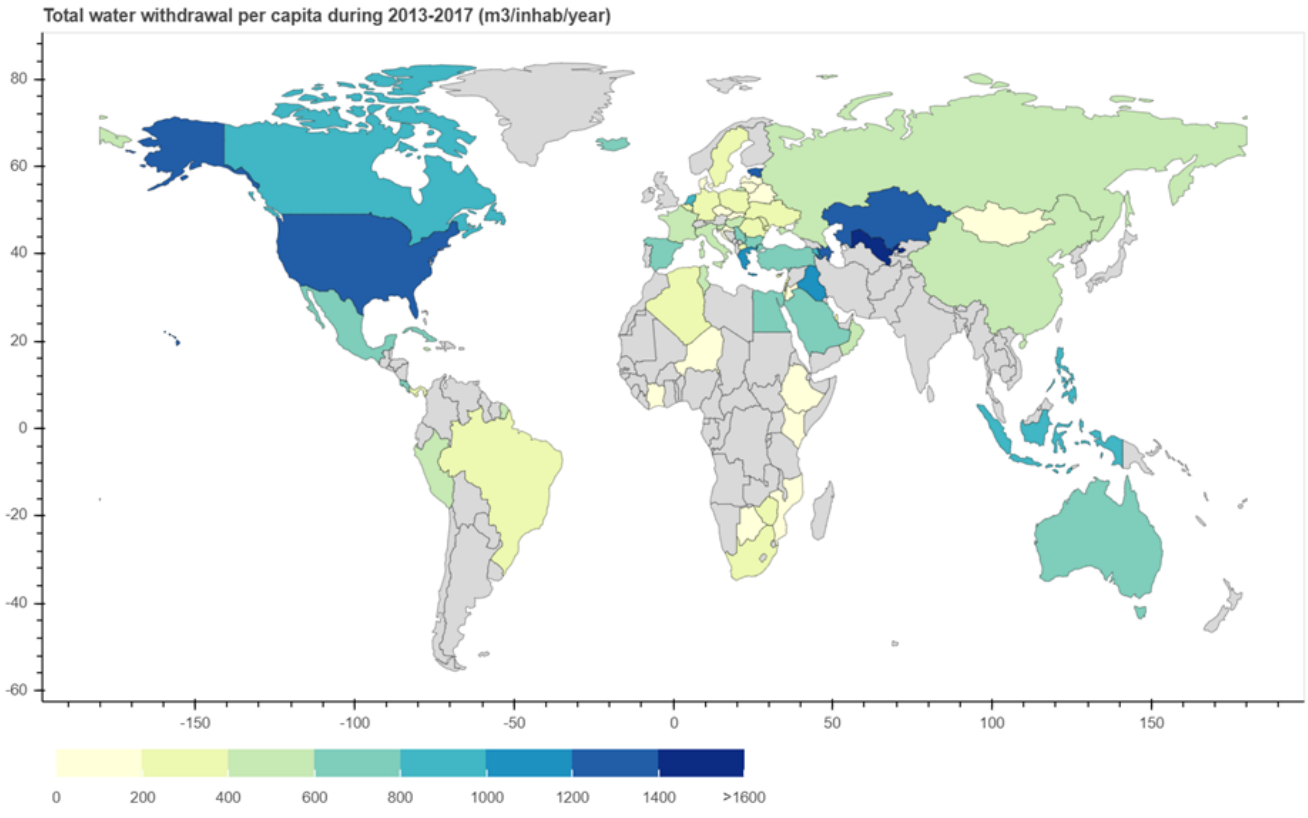

Figure 2: Total water withdrawal per capita in the world

Although the water withdrawal data during 2013-2017 of many African and west Asian countries are missing, we can find that the water withdrawal per capita varies among countries from Figure 2. For most countries, the total water withdrawal per capita is below 500 m3/inhab/year, but for Canada, the United States, Kazakhstan and Uzbekistan, the value is above 1000 m3/inhab/year.

The raw dataset includes 13 numerical independent variables. However, there are many missing values in the original dataset. It is found that “Human Development Index (HDI)” and “Total population with access to safe drinking-water” are two variables with most missing values. Meanwhile, the correlation coefficients between the target variables and these two independent variables are small. So, these two variables are dropped. After further data cleaning, finally we have a dataset with 410 samples and 11 numerical features and 2 categorical features for predicting water withdrawal per capita.

Figure 3 shows how the 11 independent variables correlate with the target variable and each other. As can be seen, “Long-term average annual precipitation in volume (10^9 m3/year)” and “Total renewable water resources (10^9 m3/year)” are highly related with the correlation coefficient of 0.97. We need to remove one of them for ML model input.

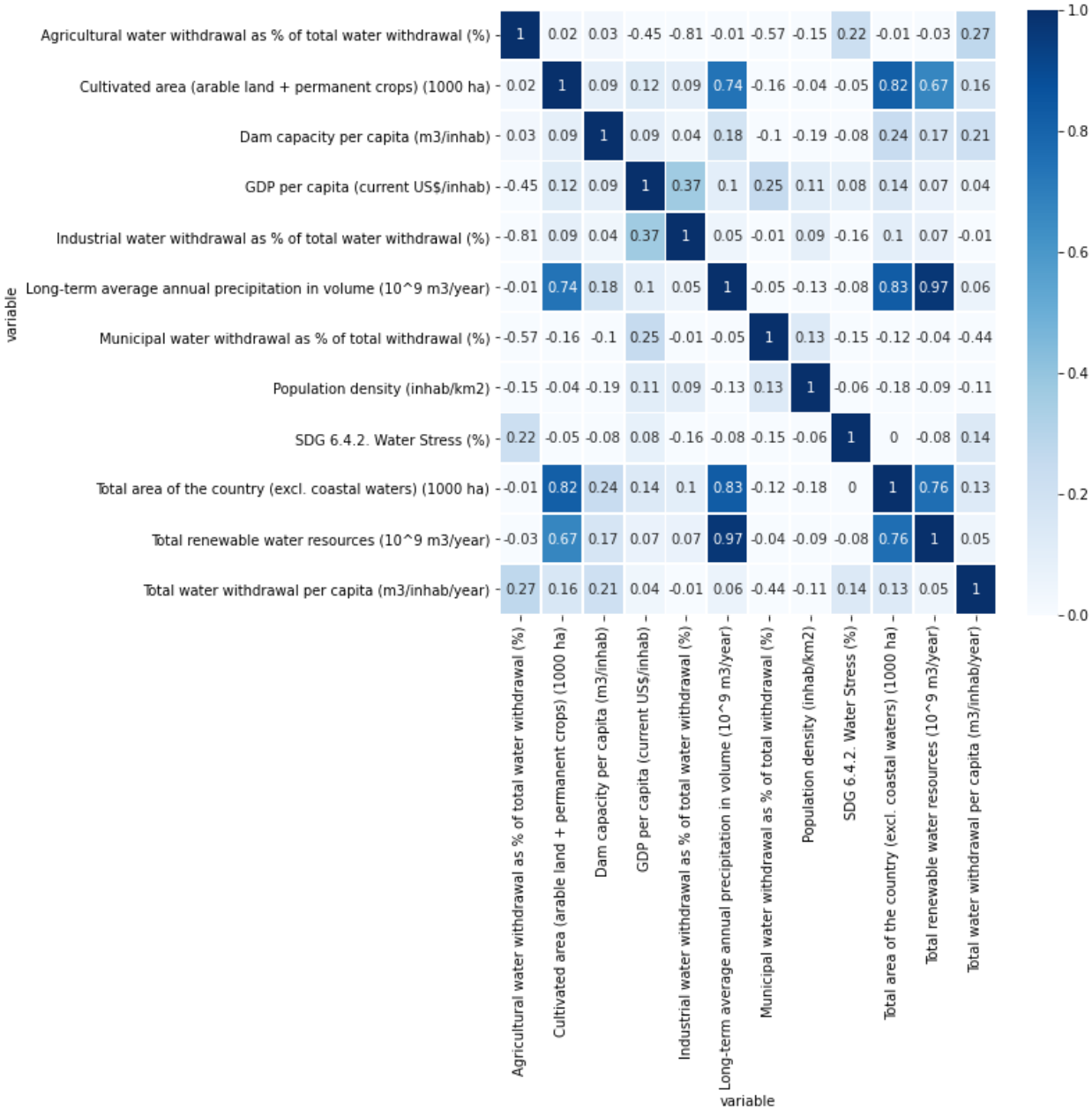

Figure 3: Correlation coefficient

3. Methods

3.1 Neural Network

Predicting water withdrawal per capita is a complex problem considering various affecting factors and non-linear relationships. We choose neural network to train our model in this project. Artificial Neural Networks have the ability to learn and model non-linear relationships, which is really important for water withdrawal prediction problem.

- Step 1: Data Preprocessing As there is no missing value in training and testing datasets, we only need to perform feature scaling for numerical variables based on the training dataset. In this step, we normalize the numerical input variables with skew higher than 3 and then standardize all the numerical variables. Considering the distribution of the target variable is skewed, we also normalize the target variable using log transformation. Since there are two categorical variables (i.e. country and year), we also do some feature engineering to prepare all the variables for use in the model.

- Step 2: Modeling We design a DNN model with one feature layer, three hidden layers, one linear single-output layer and one layer that inverses the standardization transformation. And for each hidden layer, there is 64 units. To reduce overfitting, dropout is implemented per-layer in the neural network. We choose rectified linear unit activation function (ReLU) as the activation function of hidden layers, which is not only easier to compute but also works better than a smooth function such as the sigmoid. Besides, we choose Mean Squared Error (MSE) as the loss function of our model.

- Step 3: Hyperparameter Tuning We will use Grid Search that can test the performance of different combinations of hyperparameter values and find the optimal one. The hyperparameters that will be tuned include learning rate, batch size, epochs and dropout rate. In this step, we ignore two categorical variables and only use ten numerical variables as chosen features.

- Step 4: Predicting After obtaining the best combination of hyperparameter values, we train the neural network using training set (287 examples * 80%) and check the performance of model using validation set (287 examples * 20%). Then we use it to predict the annual water withdrawal per capita in the testing set (123 examples).

3.2 Random Forests

4. Results

4.1 Results of Neural Network

We use tensorflow to build Neural Network in this project. The default values of hyperparameters are as follows: learning rate = 0.01, batch size = 40, epoch = 50, and dropout rate = 0.1. The performance of this model is shown in Figure 4 to Figure 6. The RMSE of the training data is 174.35 and the RMSE of the testing data is 259.79.

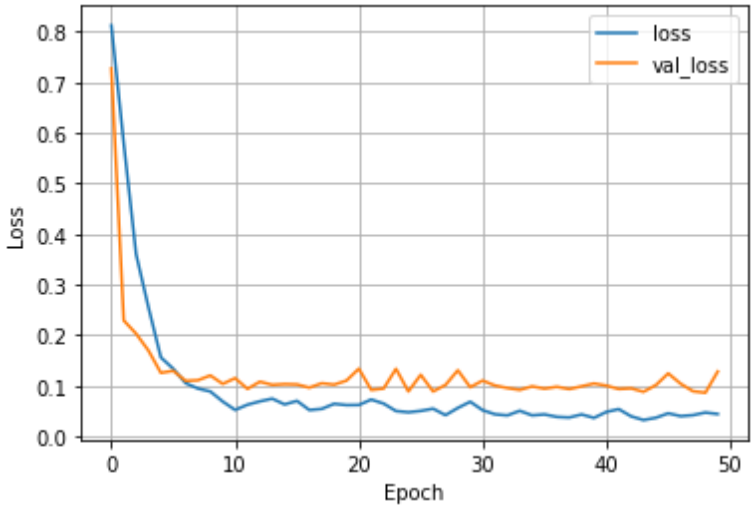

Figure 4: Model performance history

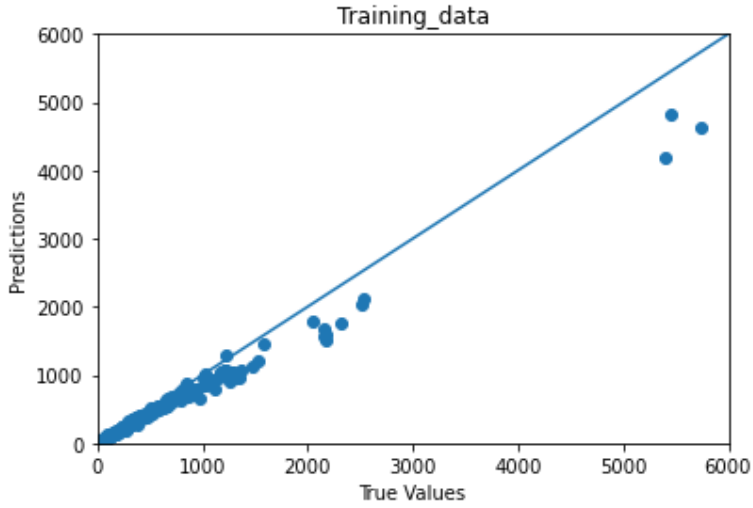

Figure 5: Predictive performance in training data

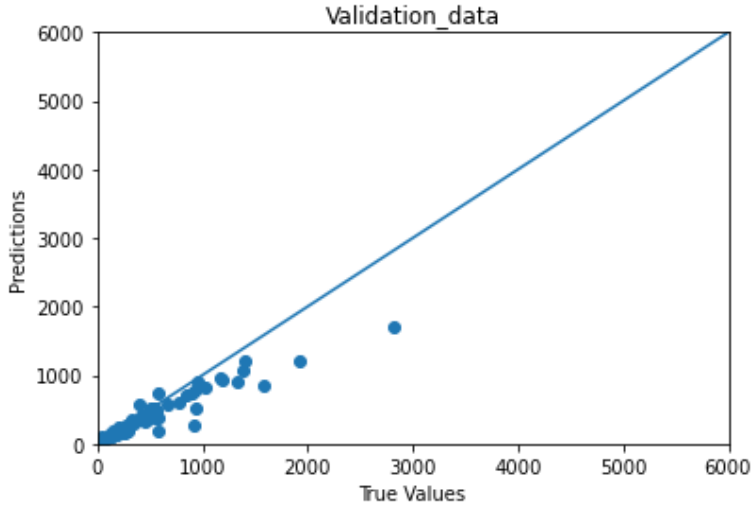

Figure 6: Predictive performance in validation data

We use Grid Search to tune the hyperparameters to improve the model performance. The results are shown in Table 1.

Table 1: The results of hyperparameter tuning

Hyperparameters	Possible values	Best value
Batch Size	2, 4, 8, 16,32, 64	64
Epochs	10, 50, 100, 200	100
Learning rate	0.001, 0.005, 0.01, 0.05, 0.1, 0.2	0.005

In addition, we test several dropout rates to find the value that can reduce overfitting. The results are shown in Table 2 and figures below.

Table 2: Model performance under different dropout rate

Dropout rate	RMSE of Traning data	RMSE of Validation data
0.1	77.03	167.24
0.2	87.31	219.42
0.3	312.99	242.12

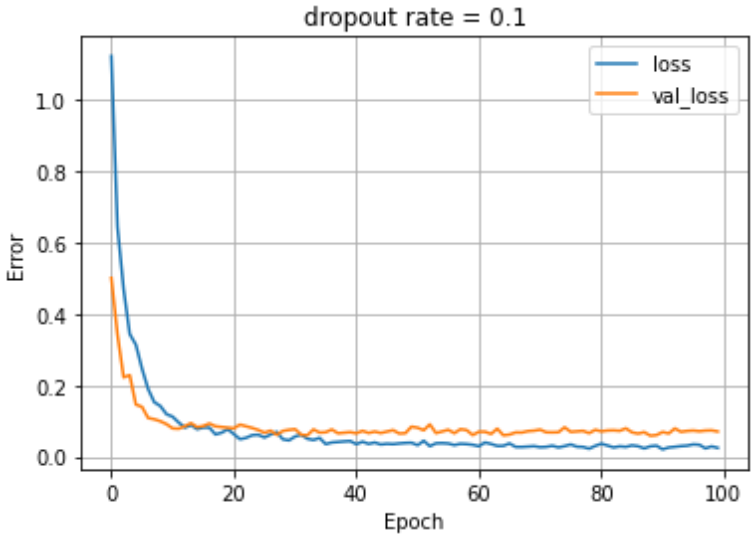

Figure 7: Model performance history (dropout rate = 0.1)

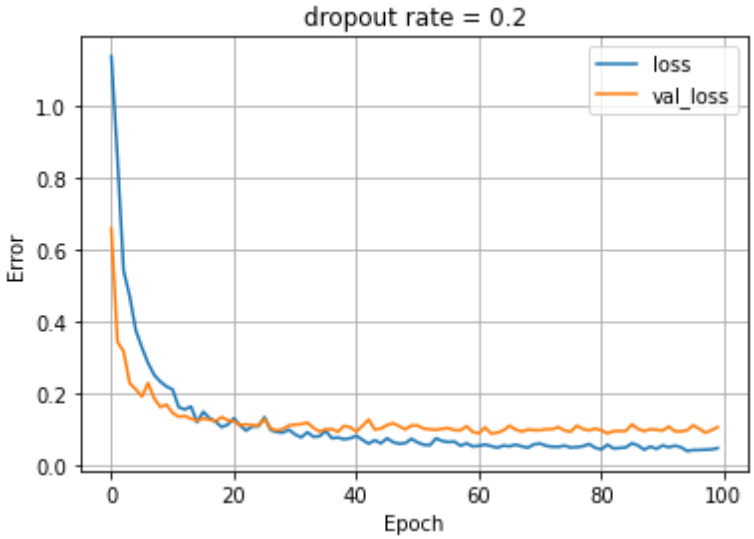

Figure 8: Model performance history (dropout rate = 0.2)

Figure 9: Model performance history (dropout rate = 0.3)

As dropout rate increases, overfitting problem can be reduced, but the RMSE of both training data and validation do not decrease. Therefore, we still set dropout rate at 0.01, which gives the best performance of the model. So, we find the best combination of hyperparameter values as follows:

- batch size = 64
- epoch = 100
- learning rate = 0.005
- dropout rate = 0.1

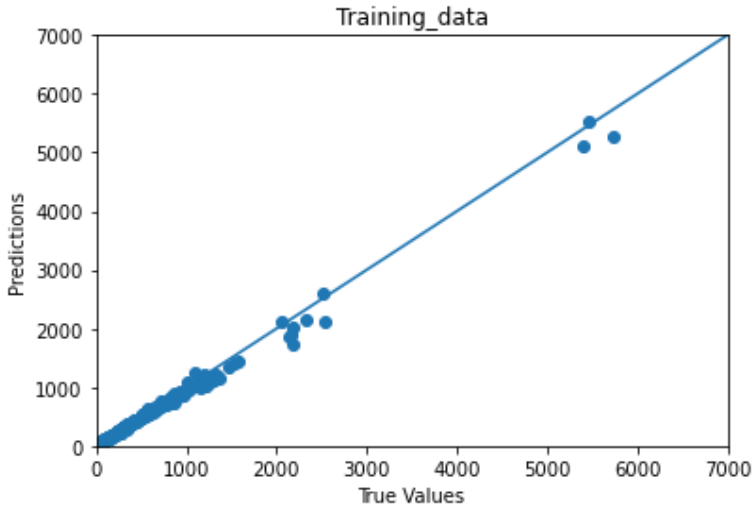

Figure 10: Predictive performance in training data after hyperparameter tuning

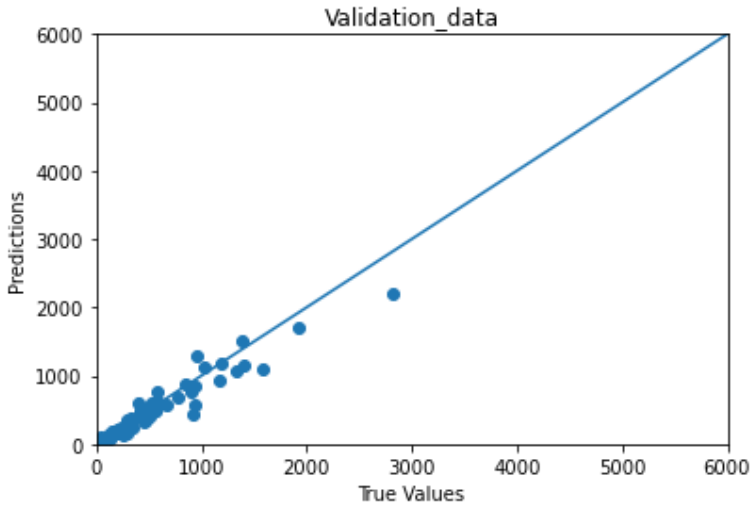

Figure 11: Predictive performance in validation data after hyperparameter tuning

Finally, we use this neural network to predict the total annual water withdrawal per capita in the testing dataset, and the Root Mean Squared Error is 144.82.

Basic formatting

Bold text

Semi-bold text

Centered text

Right-aligned text

Italic text

Combined *italics and **bold***

~~Strikethrough~~

1. Ordered list item
2. Ordered list item
    a. Sub-item
    b. Sub-item
        i. Sub-sub-item
3. Ordered list item
    a. Sub-item

- List item
- List item
- List item

subscript: H_2O is a liquid

superscript: 2^{10} is 1024.

unicode superscripts^{0123456789}

unicode subscripts$_{0123456789}$

A long paragraph of text. Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Putting each sentence on its own line has numerous benefits with regard to editing and version control.

Line break without starting a new paragraph by putting
two spaces at end of line.

Document organization

Document section headings:

Heading 1

Heading 2

Heading 3

Heading 4

Heading 5

Heading 6

A heading centered on its own printed page

Horizontal rule:

`Heading 1`'s are recommended to be reserved for the title of the manuscript.

`Heading 2`'s are recommended for broad sections such as *Abstract*, *Methods*, *Conclusion*, etc.

`Heading 3`'s and `Heading 4`'s are recommended for sub-sections.

Links

Bare URL link: https://manubot.org

Long link with lots of words and stuff and junk and bleep and blah and stuff and other stuff and more stuff yeah

Link with text

Link with hover text

Link by reference

Citations

Citation by DOI [1].

Citation by PubMed Central ID [2].

Citation by PubMed ID [3].

Citation by Wikidata ID [4].

Citation by ISBN [5].

Citation by URL [6].

Citation by alias [7].

Multiple citations can be put inside the same set of brackets [1,5,7]. Manubot plugins provide easier, more convenient visualization of and navigation between citations [2,3,7,8].

Citation tags (i.e. aliases) can be defined in their own paragraphs using Markdown's reference link syntax:

Referencing figures, tables, equations

Figure 12

Figure 13

Figure 14

Figure 15

Table 3

Equation 1

Equation 2

Quotes and code

> Quoted text

> Quoted block of text
>
> Two roads diverged in a wood, and I—
> I took the one less traveled by,
> And that has made all the difference.

Code `in the middle` of normal text, aka `inline code`.

Code block with Python syntax highlighting:

```
from manubot.cite.doi import expand_short_doi

def test_expand_short_doi():
    doi = expand_short_doi("10/c3bp")
    # a string too long to fit within page:
    assert doi == "10.25313/2524-2695-2018-3-vliyanie-enhansera-copia-i-
        insulyatora-gypsy-na-sintez-ernk-modifikatsii-hromatina-i-
        svyazyvanie-insulyatornyh-belkov-vtransfetsirovannyh-geneticheskih-
        konstruktsiyah"
```

Code block with no syntax highlighting:

```
Exporting HTML manuscript
Exporting DOCX manuscript
Exporting PDF manuscript
```

Figures

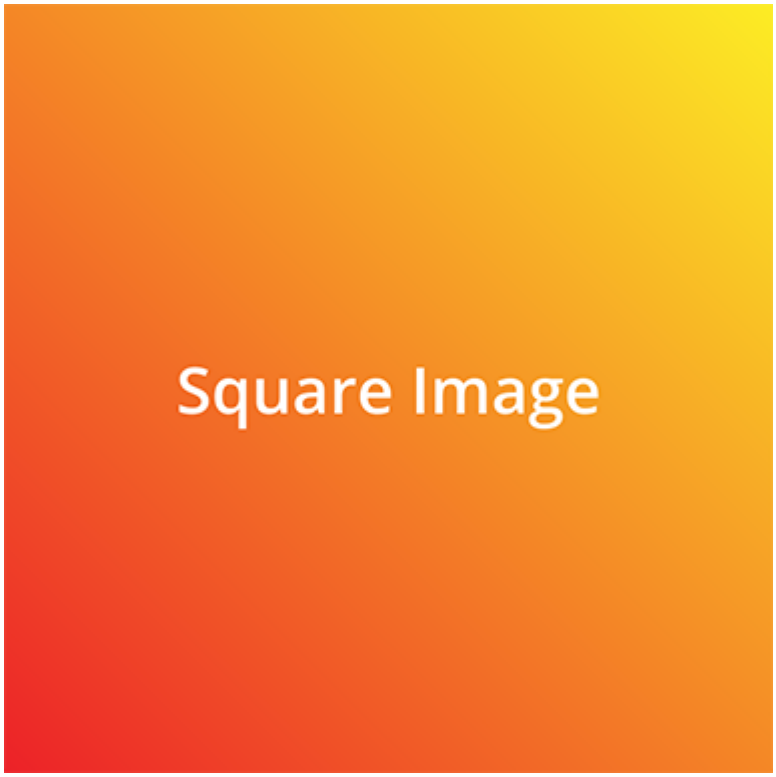

Figure 12: A square image at actual size and with a bottom caption. Loaded from the latest version of image on GitHub.

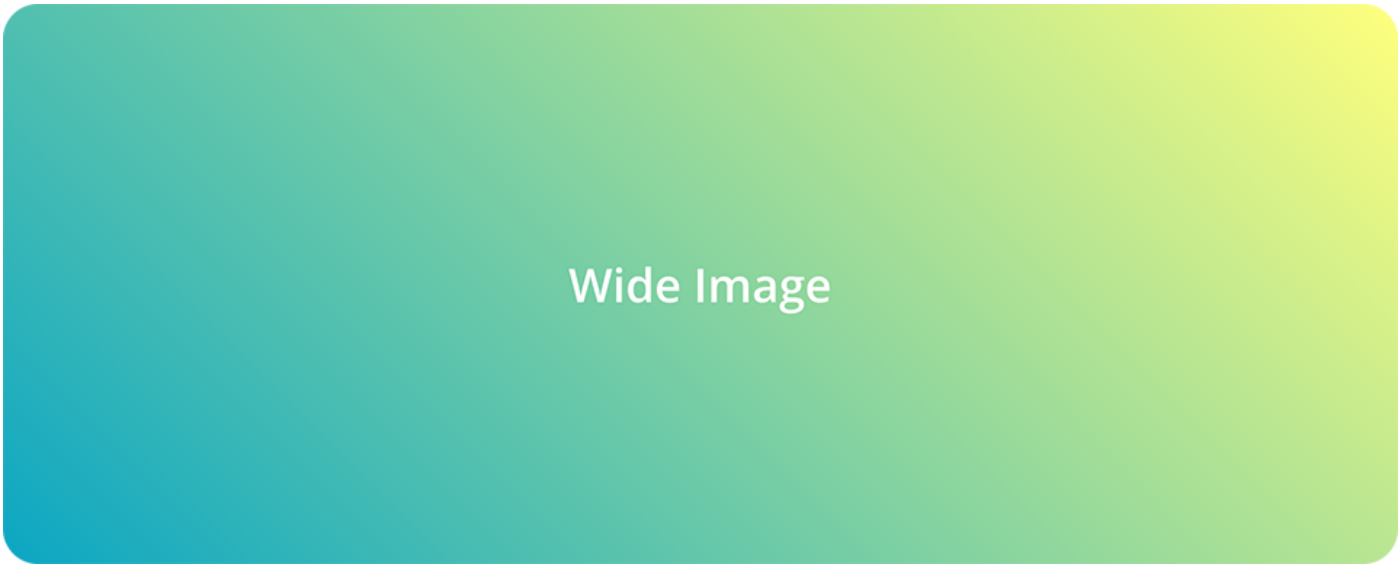

Figure 13: An image too wide to fit within page at full size. Loaded from a specific (hashed) version of the image on GitHub.

Figure 14: A tall image with a specified height. Loaded from a specific (hashed) version of the image on GitHub.

Figure 15: A vector `.svg` image loaded from GitHub. The parameter `sanitize=true` is necessary to properly load SVGs hosted via GitHub URLs. White background specified to serve as a backdrop for transparent sections of the image.

Tables

Table 3: A table with a top caption and specified relative column widths.

Bowling Scores	**Jane**	**John**	**Alice**	**Bob**
Game 1	150	187	210	105
Game 2	98	202	197	102
Game 3	123	180	238	134

Table 4: A table too wide to fit within page.

	Digits 1-33	**Digits 34-66**	**Digits 67-99**	**Ref.**
pi	3.14159265358979323 846264338327950	28841971693993751 0582097494459230	78164062862089986 2803482534211706	`piday.org`
e	2.71828182845904523 536028747135266	24977572470936999 5957496696762772	40766303535475945 7138217852516642	`nasa.gov`

Table 5: A table with merged cells using the `attributes` plugin.

	Colors	
Size	**Text Color**	**Background Color**
big	blue	orange
small	black	white

Equations

A LaTeX equation:

$$\int_0^\infty e^{-x^2} dx = \frac{\sqrt{\pi}}{2} \tag{1}$$

An equation too long to fit within page:

$$x = a + b + c + d + e + f + g + h + i + j + k + l + m + n + o + p + q + r + s + t + u + v + w + x + y + z + 1 + 2 + 3 + 4 + 5 + 6 + 7 + 8 + 9 \tag{2}$$

Special

⚠ **WARNING** *The following features are only supported and intended for* `.html` *and* `.pdf` *exports. Journals are not likely to support them, and they may not display correctly when converted to other formats such as* `.docx`*.*

LINK STYLED AS A BUTTON

Adding arbitrary HTML attributes to an element using Pandoc's attribute syntax:

> Manubot Manubot Manubot Manubot Manubot. Manubot Manubot Manubot Manubot. Manubot Manubot Manubot. Manubot Manubot. Manubot.

Adding arbitrary HTML attributes to an element with the Manubot `attributes` plugin (more flexible than Pandoc's method in terms of which elements you can add attributes to):

> Manubot Manubot Manubot Manubot Manubot. Manubot Manubot Manubot Manubot. Manubot Manubot Manubot. Manubot Manubot. Manubot.

Available background colors for text, images, code, banners, etc:

`white` `lightgrey` `grey` `darkgrey` `black` `lightred` `lightyellow` `lightgreen` `lightblue` `lightpurple` `red` `orange` `yellow` `green` `blue` `purple`

Using the Font Awesome icon set:

✔ ? ★ 🔔 ⊗ •••

Light Grey Banner
useful for *general information* - manubot.org

Blue Banner
useful for *important information* - manubot.org

Light Red Banner
useful for *warnings* - manubot.org

References

1. **Sci-Hub provides access to nearly all scholarly literature**
   Daniel S Himmelstein, Ariel Rodriguez Romero, Jacob G Levernier, Thomas Anthony Munro, Stephen Reid McLaughlin, Bastian Greshake Tzovaras, Casey S Greene
   eLife (2018-03-01) https://doi.org/ckcj
   DOI: 10.7554/elife.32822 · PMID: 29424689 · PMCID: PMC5832410

2. **Reproducibility of computational workflows is automated using continuous analysis**
   Brett K Beaulieu-Jones, Casey S Greene
   Nature biotechnology (2017-04) https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6103790/
   DOI: 10.1038/nbt.3780 · PMID: 28288103 · PMCID: PMC6103790

3. **Bitcoin for the biological literature.**
   Douglas Heaven
   Nature (2019-02) https://www.ncbi.nlm.nih.gov/pubmed/30718888
   DOI: 10.1038/d41586-019-00447-9 · PMID: 30718888

4. **Plan S: Accelerating the transition to full and immediate Open Access to scientific publications**
   cOAlition S
   (2018-09-04) https://www.wikidata.org/wiki/Q56458321

5. **Open access**
   Peter Suber
   MIT Press (2012)
   ISBN: 9780262517638

6. **Open collaborative writing with Manubot**
   Daniel S. Himmelstein, Vincent Rubinetti, David R. Slochower, Dongbo Hu, Venkat S. Malladi, Casey S. Greene, Anthony Gitter
   Manubot (2020-05-25) https://greenelab.github.io/meta-review/

7. **Opportunities and obstacles for deep learning in biology and medicine**
   Travers Ching, Daniel S. Himmelstein, Brett K. Beaulieu-Jones, Alexandr A. Kalinin, Brian T. Do, Gregory P. Way, Enrico Ferrero, Paul-Michael Agapow, Michael Zietz, Michael M. Hoffman, … Casey S. Greene
   Journal of The Royal Society Interface (2018-04-04) https://doi.org/gddkhn
   DOI: 10.1098/rsif.2017.0387 · PMID: 29618526 · PMCID: PMC5938574

8. **Open collaborative writing with Manubot**
   Daniel S. Himmelstein, Vincent Rubinetti, David R. Slochower, Dongbo Hu, Venkat S. Malladi, Casey S. Greene, Anthony Gitter
   PLOS Computational Biology (2019-06-24) https://doi.org/c7np
   DOI: 10.1371/journal.pcbi.1007128 · PMID: 31233491 · PMCID: PMC6611653