CEE 498 Project 10: Water Withdrawal Prediction

This manuscript (permalink) was automatically generated from gooood-night/Project10_Water_Withdrawal_Prediction@ce1d476 on December 6, 2020.

Authors

- **Kathryn Grace Gunderson**
  · kggunde2
  Civil and Environmental Engineering, University of Illinois at Urbana-Champaign

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

1. Introduction

1.1 Background

Water is a crucial and finite resource. 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 by many factors, such as water availability, climate, demographics, economics, etc. Machine learning (ML) methods have the ability to capture the correlations between independent variables and the non-linear relationships between features and 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 chose 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 the RF model performs better than the CART model.

In another study done by Sahour et al. (2020), scientists used machine learning methods to predict groundwater salinity, which is an affect of excessive groundwater withdrawal. Their study is relevant because it gives a layout of how to setup different machine learning algorthims for hydrological data with many indpendent variables. They tested three methods extreme gradient boosting, deep neural network, and multiple linear regression. They had a small dataset of 162 observations therefore extreme gradient boosting gave the more accurate model. The scope of their study was concentrated in one 10,000 km squared location, while we plan to look at global data for our project, therefore their study may not transfer well to our data.

1.2 Objective

The goal of the project was to use machine learning algorithms to predict water withdrawal per capita in the world. Compared with previous studies, our project used worldwide data at national scale rather than data from one city or one country, and we involved more independent variables such as water stress, renewable water resources, and GDP per capita in order to improve the prediction results. Meanwhile, we planned to test two machine learning algorithms: Neural Networks and Random Forest to predict the target variable and compare the model performances. The goal was to find a ML method that produced accurate predictions of water withdrawal, which would allow for 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 dating back to the 1960s. The variables 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, economy, climate, water resource utilities, etc. After variable selection, 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

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

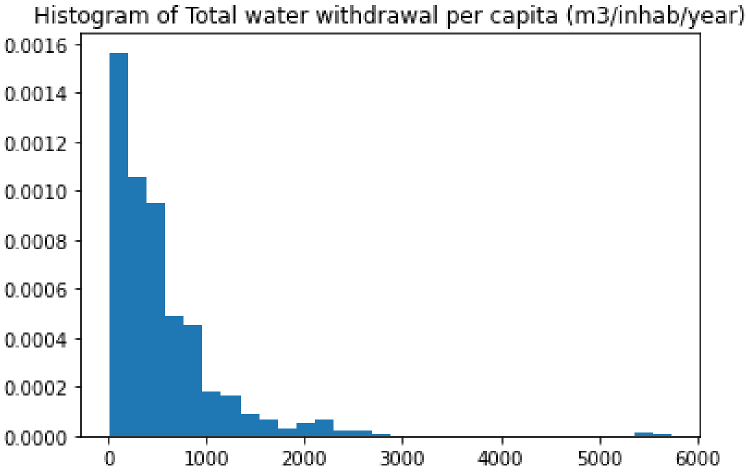


Figure 1: Histogram of the target variable

As shown in Figure 1, the distribution of total water withdrawal is not symmetric. For 75% of the 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.

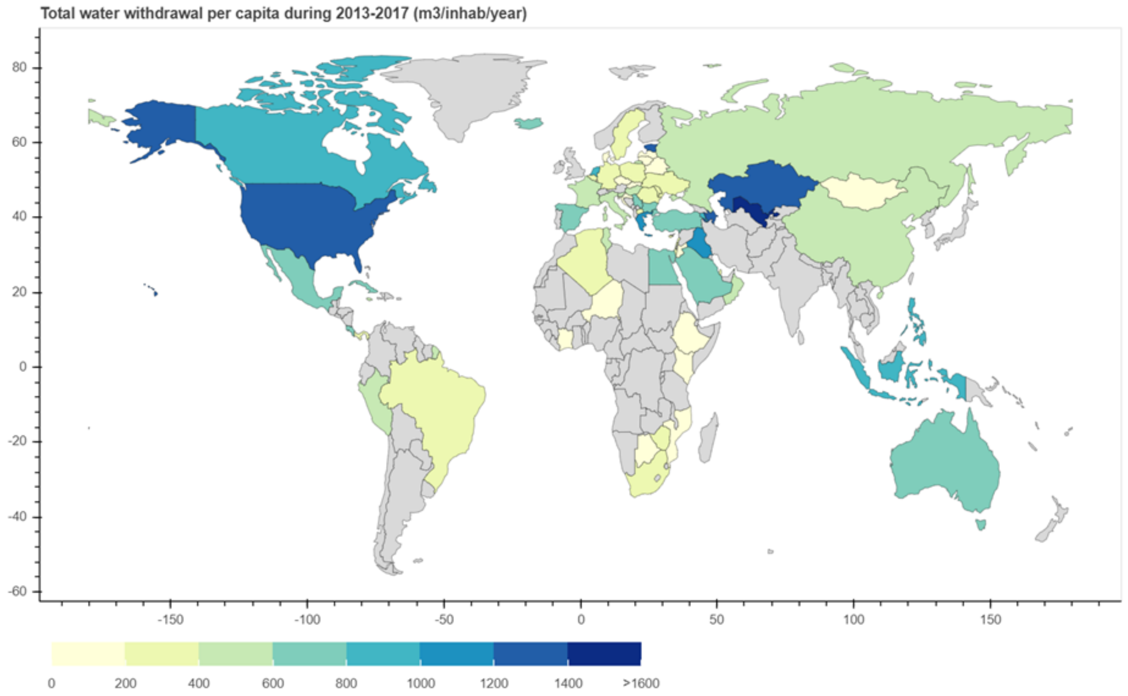


Figure 2: Total water withdrawal per capita in the world

Although the water withdrawal data during 2013-2017 of many African and west Asian countries is 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.

We can visualize the missing data in the features by using an msno matrix.

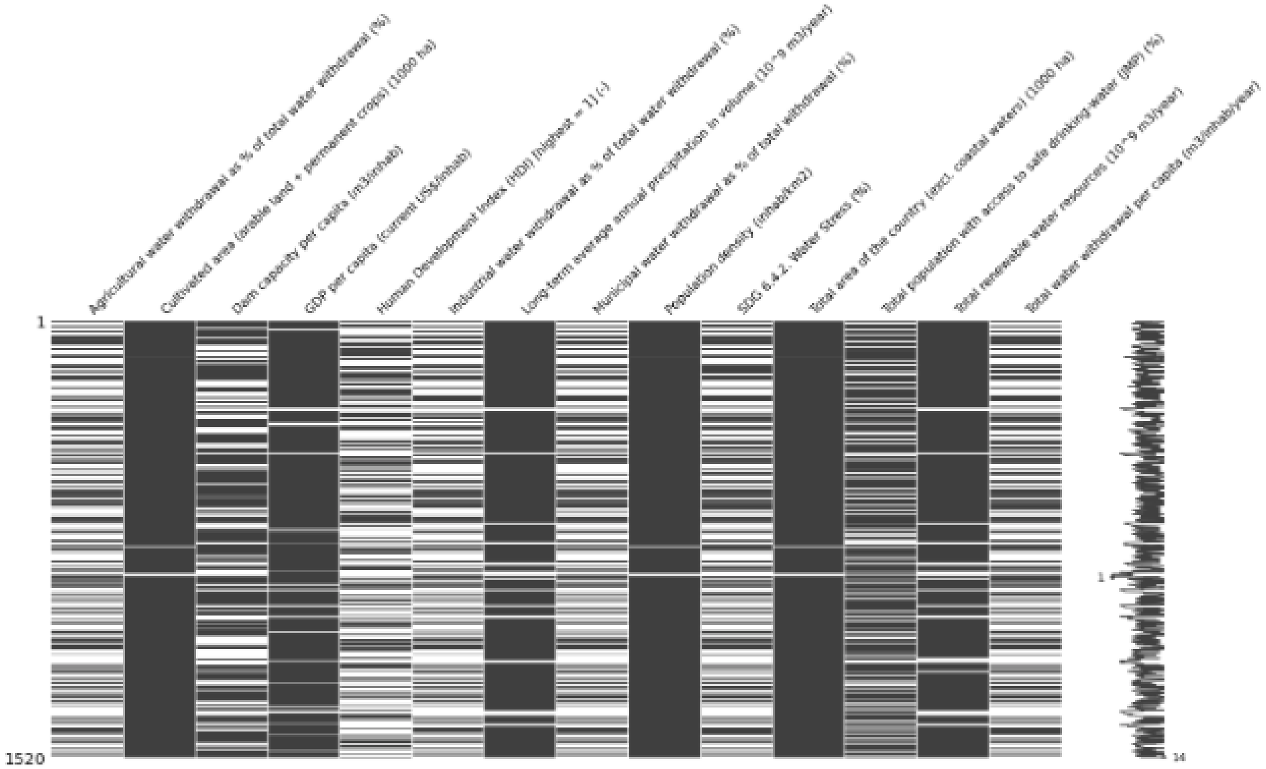


Figure 3: Missing Data Matrix

As we can see above from the msno matrix, the features with the least amount of missing data are cultivated area, GDP per capita, long-term average annual precipitation, population density, total area of the country, and total renewable water resources. This information helps us select which features to include in the model.

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

The figure below 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 in data preprocessing.

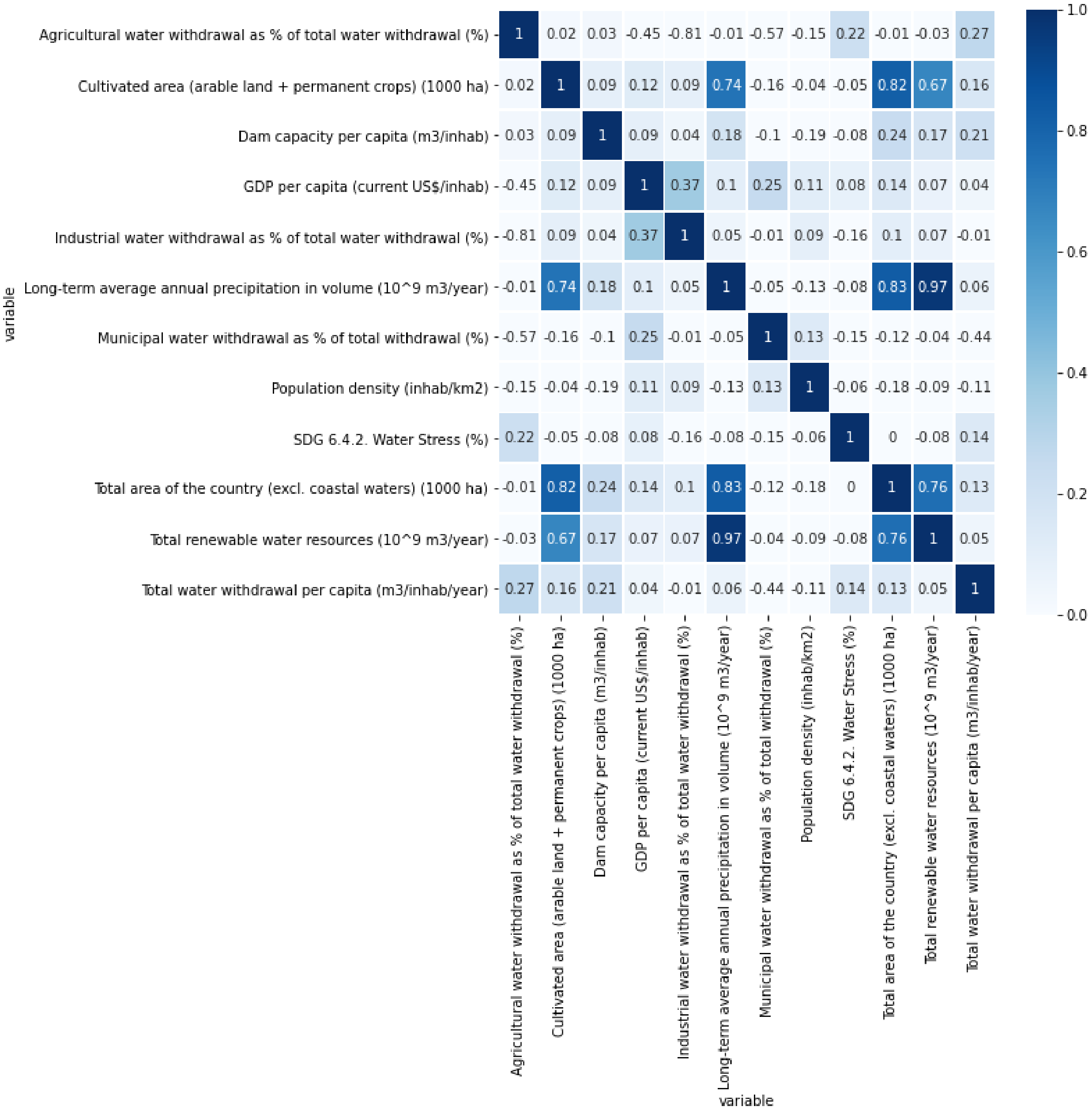
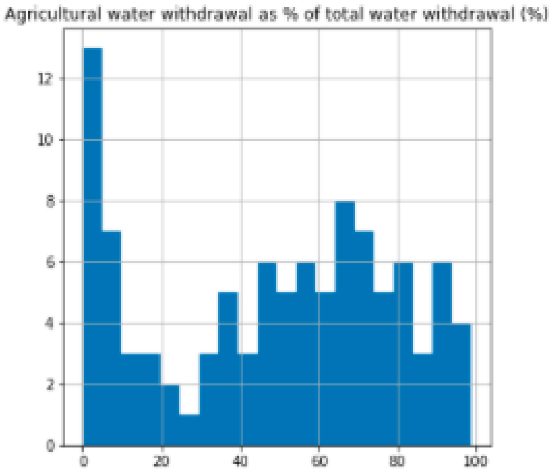
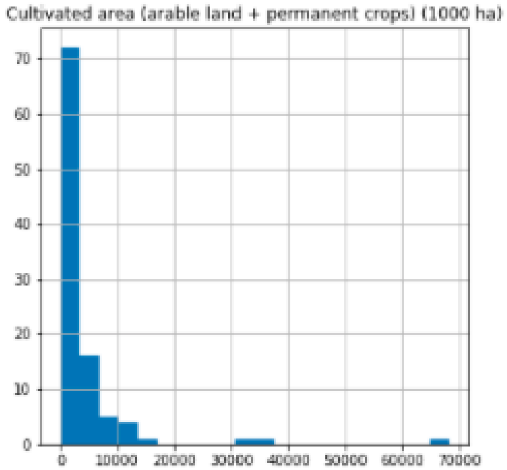


Figure 4: Correlation coefficient

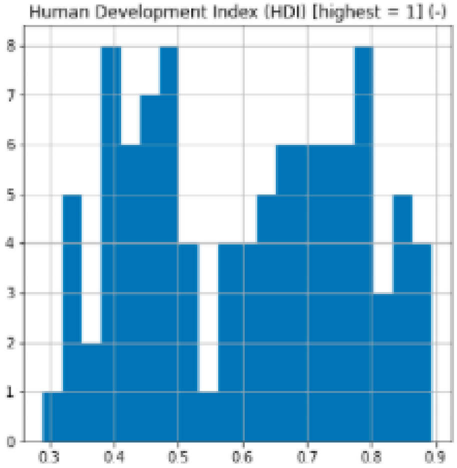
As mentioned previously most of the values for total water withdrawal per capita around below 500 m^3/inhab/yr. To further explore if any features correlate with high and low values of total water withdrawal the average value for each feature was found for each country over the different time periods. Then the data was broken up into countries that have values below 400 m^3/inhab/yr and above that value. Then we looked at the distribution of the features above and below to see if any conclusions could be made about how the features relate to the total water withdrawal. Below are histograms of the features that correspond to countries with an average water withdrawal per capita less than 400.



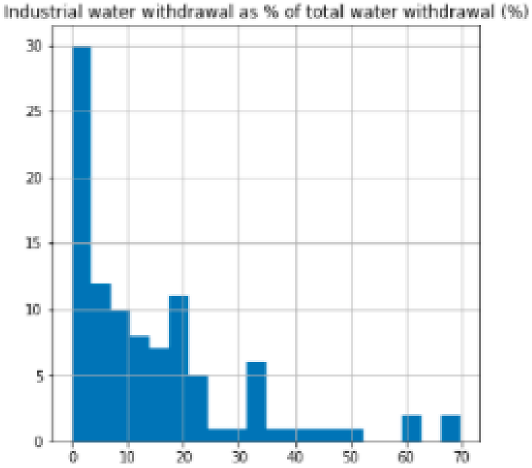
Agricultural water withdrawal as % of total water withdrawal (%)



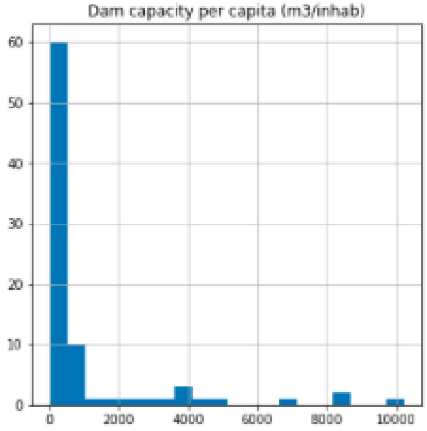
Cultivated area (arable land + permanent crops) (1000 ha)



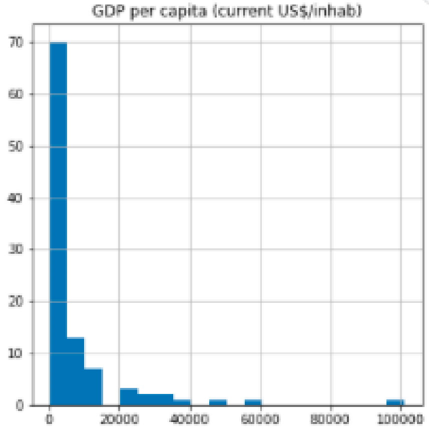
Human Development Index (HDI) [highest = 1] (-)



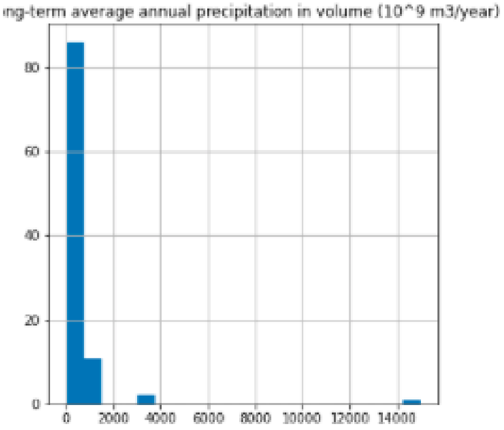
Industrial water withdrawal as % of total water withdrawal (%)



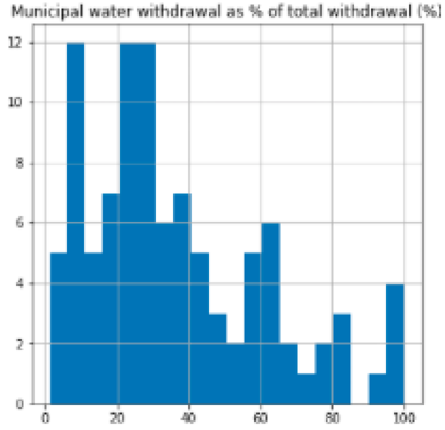
Dam capacity per capita (m3/inhab)



GDP per capita (current US$/inhab)



ng-term average annual precipitation in volume (10^9 m3/year)



Municipal water withdrawal as % of total withdrawal (%)

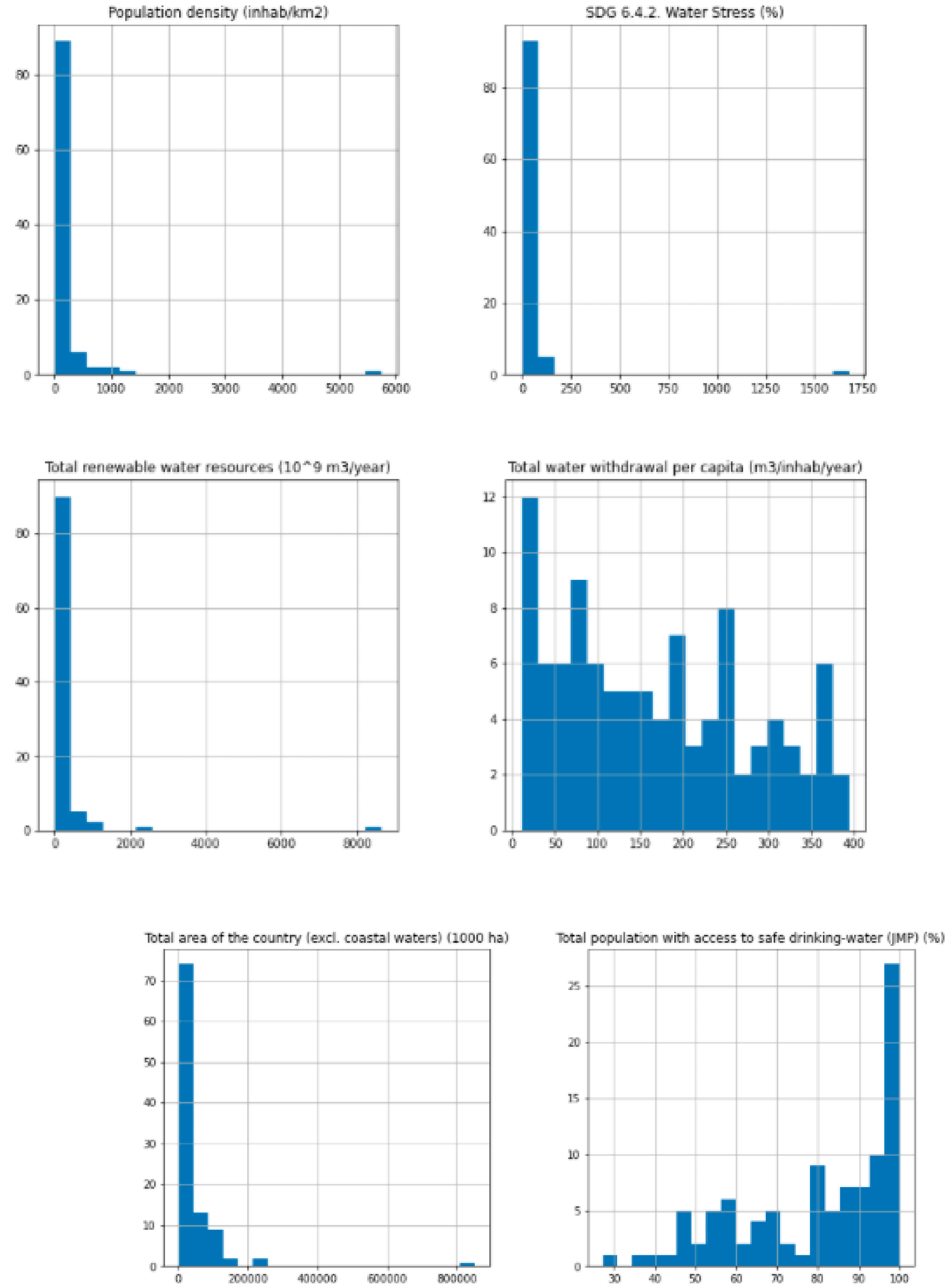
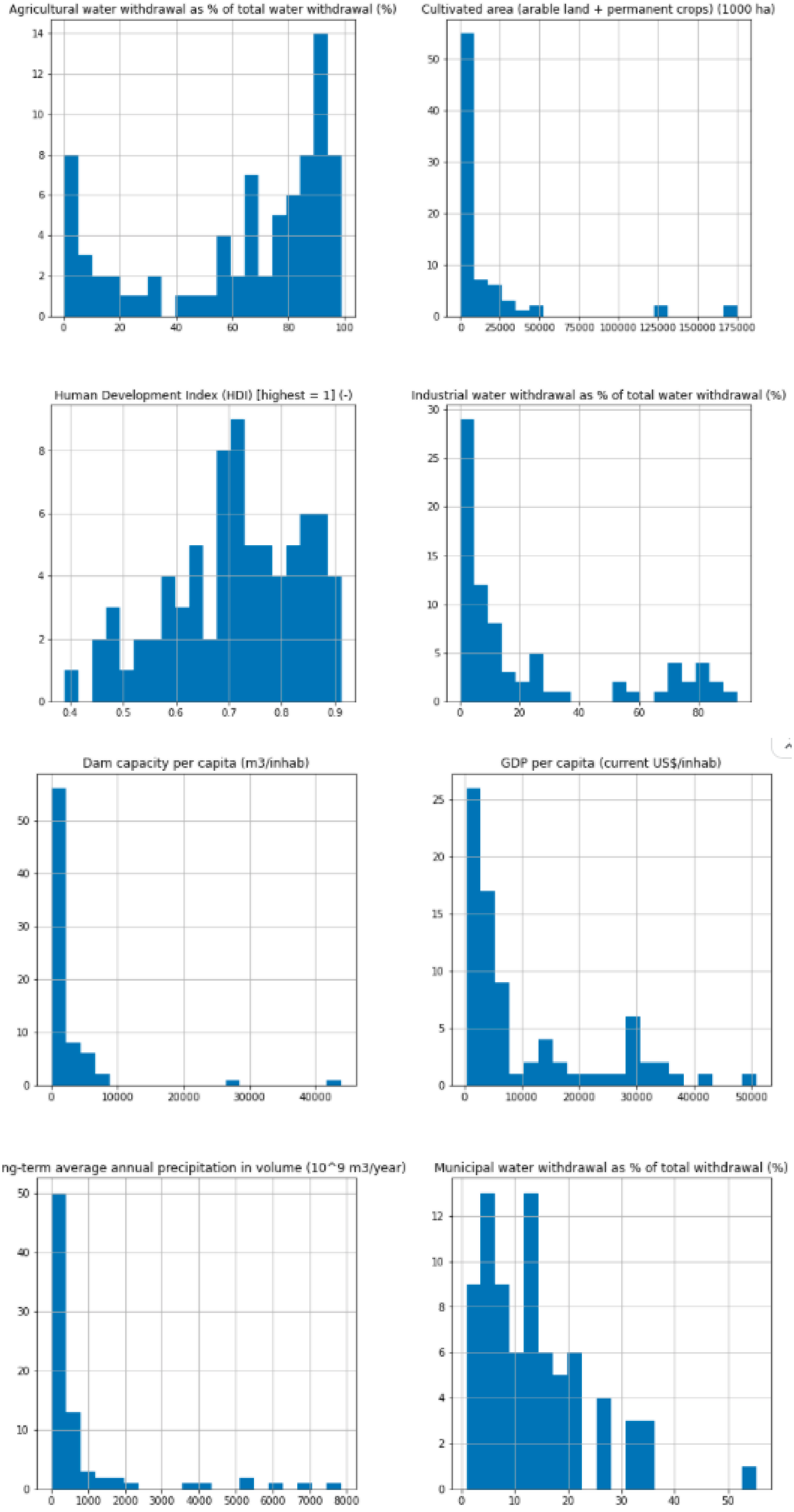


Figure 5: Distribution of Features Corresponding to Countries with Average Water Withdrawal per Capita less than 400 m^3/inhab/yr

Below are histograms of the features that correspond to countries with an average water withdrawal per capita greater than 400.


Agricultural water withdrawal as % of total water withdrawal (%)
0 2 4 6 8 10 12 14
0 20 40 60 80 100
Cultivated area (arable land + permanent crops) (1000 ha)
0 10 20 30 40 50
0 25000 50000 75000 100000 125000 150000 175000
Human Development Index (HDI) [highest = 1] (-)
0 2 4 6 8
0.4 0.5 0.6 0.7 0.8 0.9
Industrial water withdrawal as % of total water withdrawal (%)
0 5 10 15 20 25 30
0 20 40 60 80
Dam capacity per capita (m3/inhab)
0 10 20 30 40 50
0 10000 20000 30000 40000
GDP per capita (current US$/inhab)
0 5 10 15 20 25
0 10000 20000 30000 40000 50000
ng-term average annual precipitation in volume (10^9 m3/year)
0 10 20 30 40 50
0 1000 2000 3000 4000 5000 6000 7000 8000
Municipal water withdrawal as % of total withdrawal (%)
0 2 4 6 8 10 12
0 10 20 30 40 50

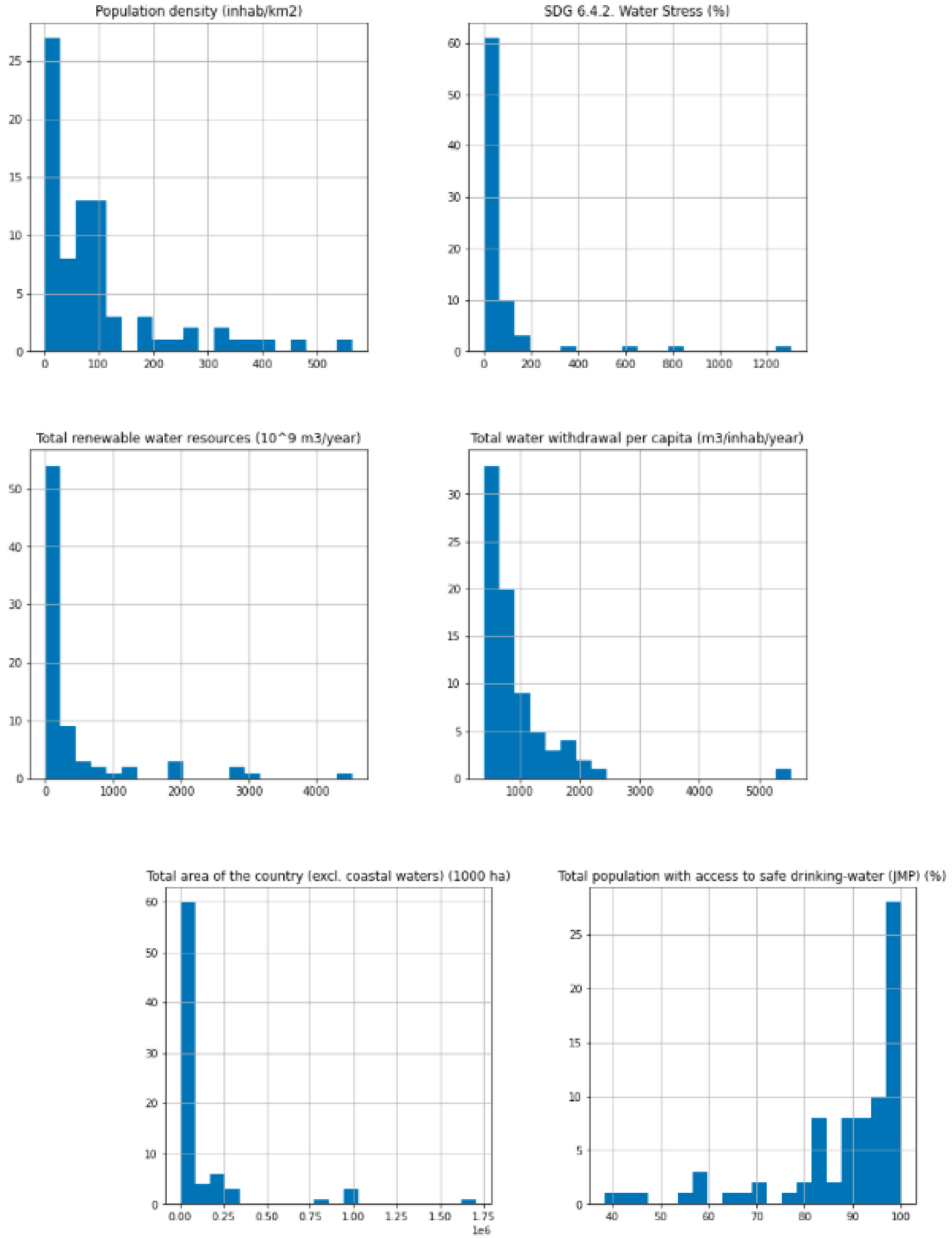


Figure 6: Distribution of Features Corresponding to Countries with Average Water Withdrawal per Capita more than 400 m^3/inhab/yr

Comparing the set of histograms for the countries that have an average water withdrawal lower than 400 m^3/inhab/year and those that have higher than 400 there are some differences that could help predict the average water withdrawal. The agricultural water withdrawal, population density, cultivated area, and GDP tends to be higher for countries with a higher average water withdrawal than 500. Total area of the country, municipal water withdrawal tends to be higher for countries with a lower average water withdrawal than 500.

3. Methods

3.1 Neural Network

Predicting water withdrawal per capita is a complex problem considering various affecting factors and non-linear relationships. In this project, one of the models we chose was neural network. 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 performed feature scaling for numerical variables based on the training dataset. In this step, we normalized the numerical input variables with skew higher than 3 and then standardized all the numerical variables. Considering the distribution of the target variable was skewed, we also normalized the target variable using log transformation. Since there are two categorical variables (i.e. country and year), we also did some feature engineering using one hot encoding to prepare the categorical variables for use in the model.

- Step 2: Modeling

We designed a DNN model with one feature layer, three hidden layers, one linear single-output layer and one layer that inversed the standardization transformation. And for each hidden layer, there was 64 units. To reduce overfitting, dropout was implemented per-layer in the neural network. We chose 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 chose Mean Squared Error (MSE) as the loss function of our model.

- Step 3: Hyperparameter Tuning

We used Grid Search to tune hyperparameter, which can test the performance of different combinations of hyperparameter values and find the optimal one. The hyperparameters that were tuned included learning rate, batch size, epochs and dropout rate. In this step, we ignored two categorical variables and only used ten numerical variables as the chosen features.

- Step 4: Predicting

After obtaining the best combination of hyperparameter values, we trained the neural network using training set (287 examples * 80%) and checked the performance of model using validation set (287 examples * 20%). Then we used it to predict the annual water withdrawal per capita in the testing set (123 examples).

3.2 Random Forest Regression

The other model tested to predict our target variable is a Random Forest Regression (RFR). RFR was selected because we need a non linear model. In this case we want a regression model rather than a classification model because we have numerical values rather than categorical values to predict. One of the shortcomings that could inhibit the accuracy of the model is that a Random Forest Regression cannot predict values that are higher than the values in the training data.

- Step 1: Data Preprocessing

There are not missing values from our training and testing models. First we looked at a description of the data to make sure there were no constant values that needed to be removed. After that we checked the data types and changed all object variables to integers using one hot encoding. After this we looked at the distribution of the target variable to check for skewness. The target variable had right skewness, therefore we normalized the target variable using a log transformation to improve the model predictions. We split the training data into training data and validation data to prevent overfitting the model to the training data.

- Step 2: Modeling

The first random forest regression model we made based on adjusting the n-estimaters parameter until we got the highest percent accuracy on the validation data. The accuracy was calculated using the mean absolute error of the predictions compared to the actual values of the validation data. A random state of 42 was added to the model to create consistency.

- Step 3: Hyperparameter Tuning

To improve the accuracy of the model we used both randomized search cross validation and grid search cross validation. The hyperparemeters tested were n_estimators, min_samples_split, min_samples_leaf, max_features, max_depth, and bootstrap. The randomized search CV was used to narrow down the range of the parameters. Then grid search CV compares all combinations of inputs. We performed the grid search CV and stopped when the accuracy stopped improving. Next we checked the feature importances and checked the accuracy of the model using the four most important features. The accuracy did not change when only the four most important features were used therefore we changed the model to only use those four features in predicting the target variable.

- Step 4: Predicting

We used the best combination of hyperparemeters from the grid search CV and the four most important features to predict the target variable.

4. Results

4.1 Results of Neural Network

We used 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 figures. The RMSE of the training data was 174.35 and the RMSE of the testing data was 259.79.

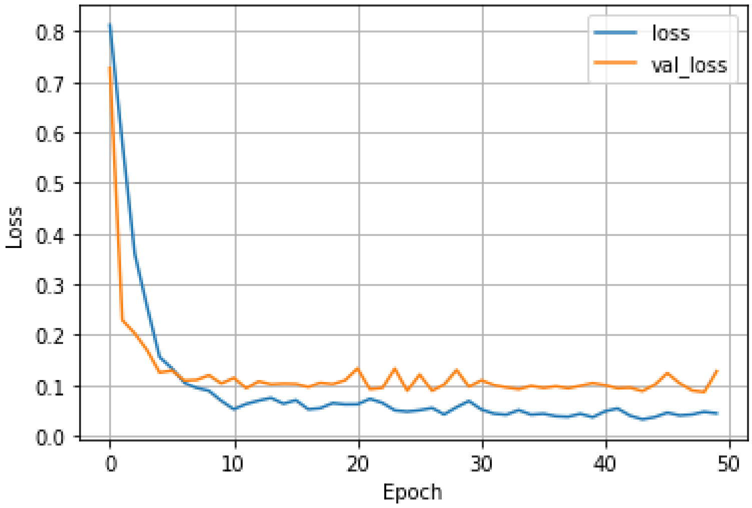


Figure 7: Model performance history

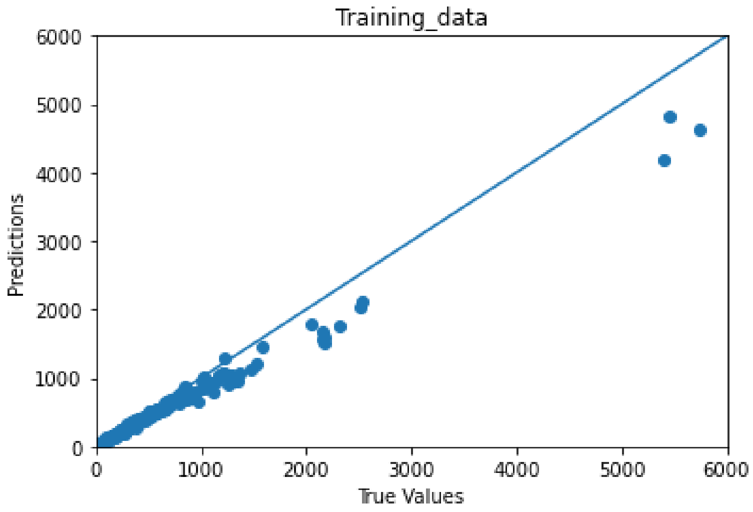


Figure 8: Predictive performance in training data

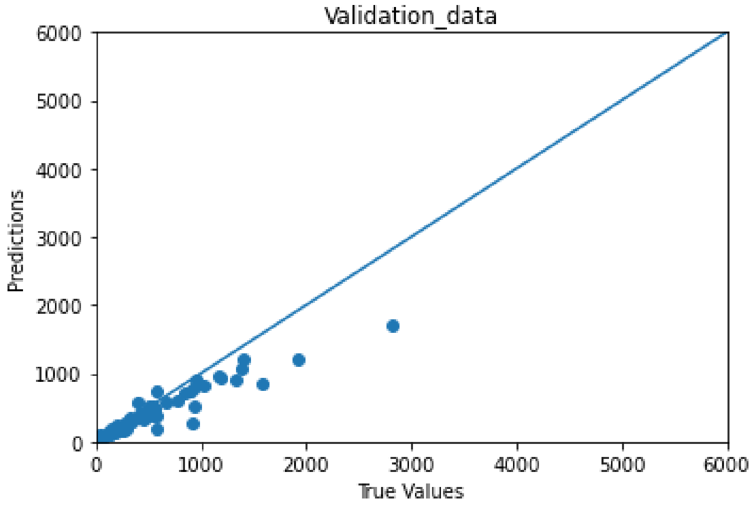


Figure 9: Predictive performance in validation data

We used 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 tested 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

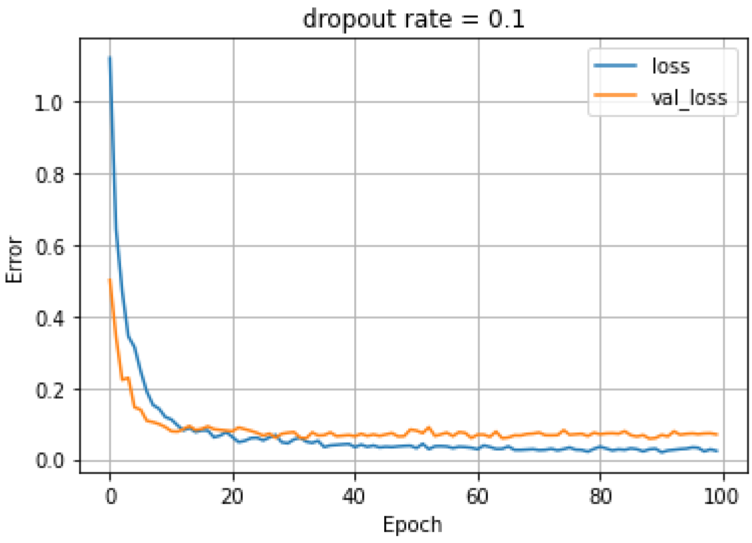


Figure 10: Model performance history (dropout rate = 0.1)

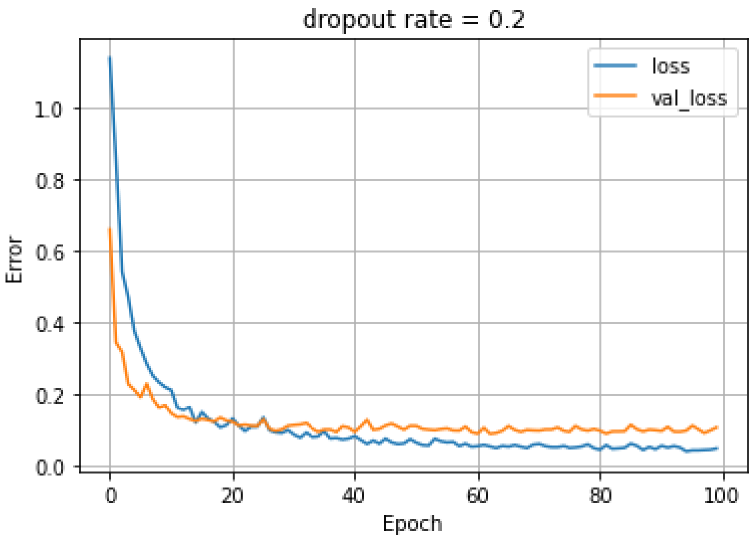


Figure 11: Model performance history (dropout rate = 0.2)

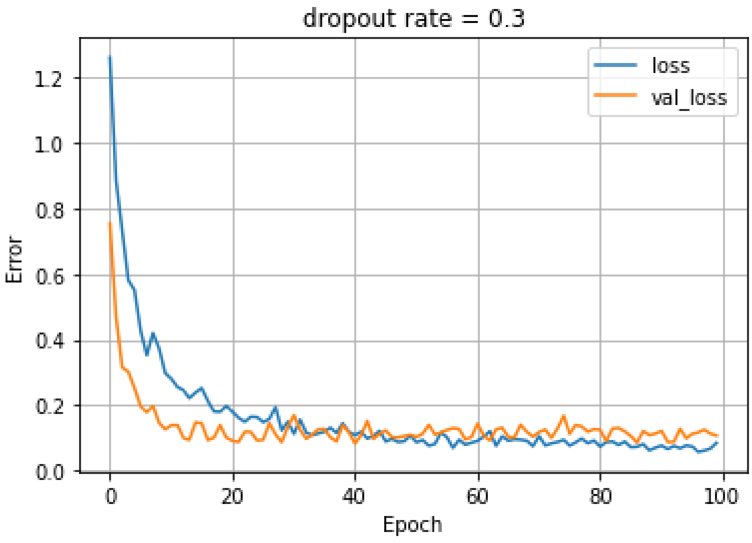


Figure 12: Model performance history (dropout rate = 0.3)

As dropout rate increased, the overfitting problem can be reduced, but the RMSE of both the training data and the validation did not decrease. Therefore, we still set the dropout rate at 0.01, which gave 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

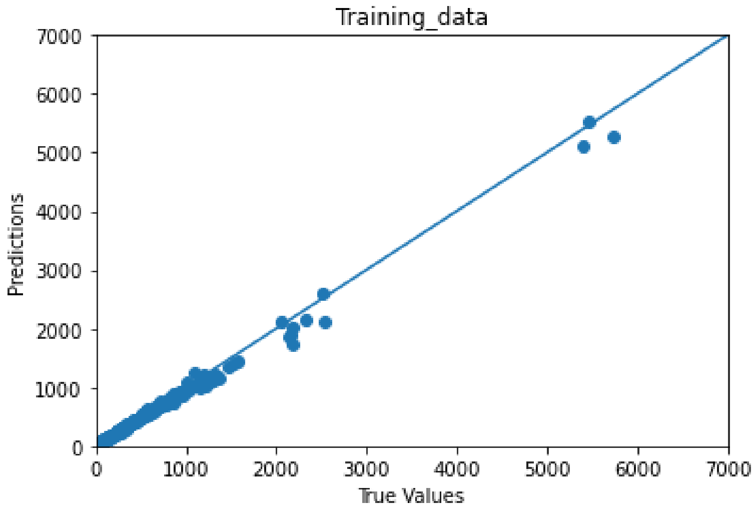


Figure 13: Predictive performance in training data after hyperparameter tuning

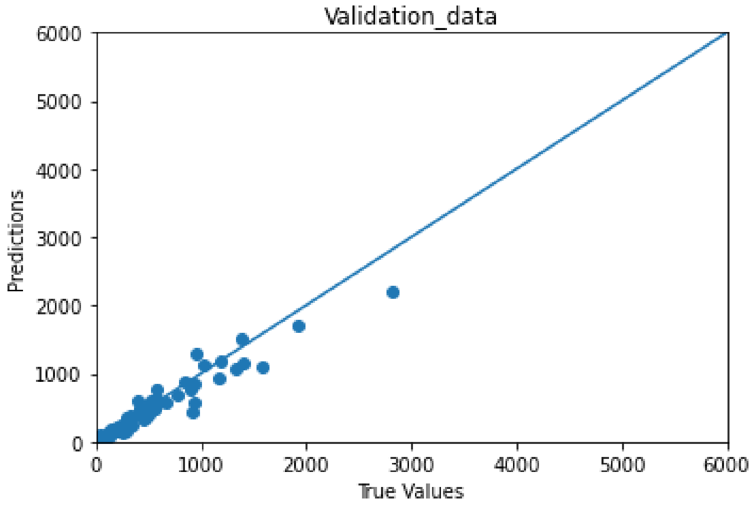


Figure 14: Predictive performance in validation data after hyperparameter tuning

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

4.2 Results of Random Forest Regression

The model was built using random forest regressor with the initial hyperparameters set to n-estimators = 100 and random state = 42. The mean absolute error was 0.41 degrees and the percent accuracy was 92.98%.

Next we used randomized search CV and did two grid search CV to tune the hyperparemeters and find the best combination. Below are the mean absolute error and percent accuracy for each of the attempts.

Table 3: Model performance

Tuning Attempt	MAE of Validation data	% Accuracy of Validation data
Randomized Search CV	0.4	93.2%
Grid Search CV 1	0.4	93.2%
Grid Search CV 2	0.4	93.2%

As can be seen in the table, the accuracy of the model did not improve much after the first cross validation. The final hyperparamters selected for the model were:

- n estimators: 1000
- min samples split: 2
- min samples leaf: 1
- max features: 3
- max depth: 60
- bootstrap: true
- random state: 42

After selecting the hyperparameters I next looked at the figure importance and made a plot of the top 12 features to better visualize the weight of each feature on the model.

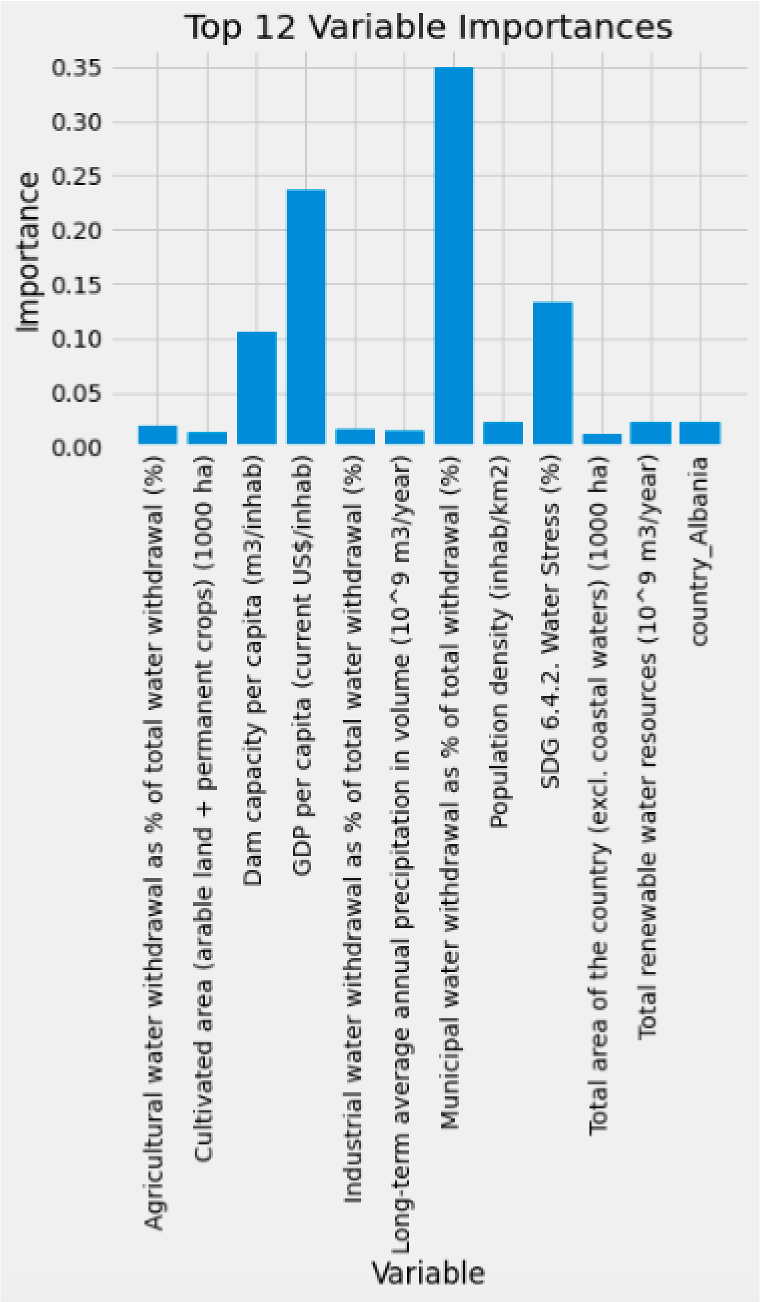


Figure 15: Top 12 variable importances

The top four most important features were % Municipal Water Withdrawal, GDP per Capita, Water Stress, and Dam Capacity per Capita. We wanted to test how the accuracy of the model changed if we only used the top four most important features. We made a new dataframe that only included those four features and used the same hyperparamters as above and tested for the mean absolute error and percent accuracy in the validation data. The mae was 0.41 degrees and the percent accuracy was 93.09%. Since the accuracy changed by 0.11%, We decided to go with the simpler model and only include those four features.

After applying the model to the testing dataframe the final root mean squared error was 264.13 m^3/inhab/yr.

5. Conclusion

Comparing the root mean squared error of the two methods used, the Neural Network algorithm had the lower value of root mean squared error. Therefore this model was more accurate at predicting the target variable: annual total water withdrawal per capita. This model had a root mean squared error of 122.08 m^3/inhab/yr, which in the context of the problem is acceptable due to the spread of the data for the target variable. The model could give countries an idea of where to expect their total water withdrawal to be when changing different parameters that were included in the model. To expand on this model, different features could be tested and having more data could improve the accuracy of the model.

6. References

AQUASTAT Database , Food and Agriculture Organization of the United Nations, www.fao.org/nr/water/aquastat/data/query/index.html?lang=en.

Sahour, Hossein, et al. “A Comparative Analysis of Statistical and Machine Learning Techniques for Mapping the Spatial Distribution of Groundwater Salinity in a Coastal Aquifer.” Journal of Hydrology, Elsevier, 19 July 2020, www.sciencedirect.com/science/article/pii/S0022169420307812.

Villarin, M. C., & Rodriguez-Galiano, V. F. (2019). Machine learning for modeling water demand. Journal of Water Resources Planning and Management, 145(5), 04019017.