1. Description

The problem we tacked in this project is that of predicting California’s urban water demand and the available fresh water supply to address it.

The limited natural water resources and increasing population have led to an increasing demand in enhanced forecasting and subsequently managing water consumption. To address this, we started out by looking at the various sources that California’s water comes from. We identified that there are three major sources: surface, ground, aqueducts.

We collected the openly available amount data for all three, in addition to getting snow and precipitation sensor recordings. The sensors recorded at various frequency intervals and we utilized Hadoop to map reduce the data to monthly intervals. This data mostly consisted of averages of various types.

Initially, we identified our primary features as snow and precipitation because we thought they are the major contributors of water to reservoirs. However, when we normalized our data to consider the statewide trends in snow and precipitation with those of reservoirs, we did not find a strong correlation. A major contributing factor to this was the fact that our reservoir data does not take into consideration the aqueducts that feed water into the reservoirs. Only the North California region can be expected to have strong correlation with rain and reservoir levels. We talked to a subject matter expert who works for the Fresno County State department who provided this information.

One key interesting point between snow and precipitation data was that it had a correlation of 75%, which provided us a confidence metric to the validity of the data and the consistency of our mappers and reducers for different data types.

Next, we looked for features that would allow us to capture how much water there was actually being supplied from the reservoirs to the state. We found in our reservoir datasets a set of rows that captures the outflow for a given sensor. Since our problem was for entire california, we were able to normalize the outflow to an amount per month using Hadoop and begin building our model.

To build our model, we initially started looking at linear models. Very soon we realized that the approach was incorrect and not going to work. Next, we started using multiple features to predict the outflow such as storage, precipitation and snow. That yielded poor results. Subsequently, we tried to apply lasso and ridge regression models to scale the features mentioned in order to see improvements. We got slight improvements, but our accuracy was still below 50% which was unsatisfactory.

Lastly, what worked for us was a successive approximation model. We initially used optimization functions in scipy by trying to find a fixed point and extending it. The improvements were there, but it was taking extensive time and effort to understand and apply the mathematics behind.

Lastly, what worked for us is the successive approximation library implemented in the fb-prophet package. It’s built for time series analysis for tracking human activity. We were able to successfully apply it and achieve 85% accuracy on our time series validation.

The second half of the problem was the demand side. We had data from 2005-2015 for various municipalities that supplied water to different regions in California. This key caveat here was that they included *most* of the water supplied but not all. Since the data was from CDEC, these municipalities were required to report their data to the statewide department (CDEC), but not all of them did. This was a skewing factor in our study that had to be adjusted for later.

Overall, we were able to find a high correlation between the changes in population of california and the growing demand for urban water. We chose to do only urban water because it seemed to be the most tracked metric by the government.

A linear regression model fit nicely here and we were able to achieve accuracy north of 85%.

1. Requirements

Our main requirement for this project was to provide water availability forecast to a state department stakeholder. This top level requirement was further decomposed into a forecasting problem that had two components:

* Fresh water availability
  + Understand how reservoir water is used for urban consumption
  + Obtain data regarding how much water reservoirs are providing
  + Model the water provided based on a correlating feature (so that the problem statement can be extended)
* Urban water demand
  + Needed data regarding water consumption trends
  + Find a set of features to model water consumption
* Model the demand vs. availability
  + Deduce a common metric that can be used to compare demand vs supply

1. KDD

## A. Data Selection

Based on our problem statement, we first set out to determine what types of water sources exist. Currently, there are three primary sources of water for various regions in California. These include: freshwater, groundwater, aqueducts.

Based on our problem, the best means to obtain water usage data seemed to be through a government sponsored organization. Private water data collection methods are localized to small geographic regions, while we are interested in water availability in California as a whole. We used California Data Exchange Center (CDEC) [1] to collect precipitation, snow and reservoir level sampled data. Additionally we used the United States Geological Survey (USGS) [4] to collect data on water levels in various rivers that aqueducts deliver water from. Another source of data was from the California State Water Resources Control Board (California Water Boards) [11], we use them to collect data on amounts of water supplied. The files were in csv formats with various units for different types of data collected by each sensor, or in different formats.

## B. Data Pre-Processing

From the various sources that we pulled data from, we had to start cleaning it and determine what we can use to best solve our problem.

We started with the precipitation, snow and reservoir sensor data that we found from the California Data Exchange center. Looking over the sensor data that we had collected for California precipitation, we were able to determine that we were going to use the available hourly rainfall data. Similarly looking at the snowpack data, we would be using the hourly snowpack data that was available. And for the data collected for each of the reservoirs, we would take the total outflows for each hour. We had collected hourly data from January 2005 to October of 2019.

We then looked over the ground water and recycled water data that we had pulled, and we were to use the data that showed how much groundwater is available at the current period of time. Our data was for the years 2005, 2010, and 2015. We pre-processed our data points to only include information about the state of California.

Finally, we had sector data. From this we would look at the counties of each sector and how much water they had used in the years 2014 -2017. We pre-processed our data to only include data points for the state of California.

## C. Data Transformation

Once we had pre-processed our data to look for our relevant fields. We started to transform our data to reduce the dimensionality and to potentially have our model run faster.

Once again we started without data from the California data exchange center. Our twelve gigabytes of data needed to be consolidated to monthly totals. We setup a Hadoop environment where we stored our data. We then designed a python mapper to look at each sensor, determine the type of data that we were given. We would then map this data to a common metric: “inches” From their hourly data would be added up and we would be given the total amount of precipitation for each hour in our defined metric.

Once we had those results, our reducer would aggregate our hourly data into daily data, and then our daily data into monthly data.

This process would be repeated for our Snowpack in which we looked for “SNOW WC and SNOW ADJ”, these values would provide us the water content within the snow.

For Reservoir data, we would look at the amount of outflows for each hour, this would then be consolidated into an average for each day, and finally, we would consolidate these values into an average for the month. These values would be provided to us in “CFS” units, which would then have to be converted into gallons.

We now had data values available for each sensor for every month between January 2005 and October 2019.

Similar to our transformation of our California data exchange center data, we reduced the amount of groundwater data to include data only for the state of California. This data additionally included the amount of pop

As for our sector data, we once again selected data for the state of California. Otherwise our data has been transformed to best fit our needs.

## D. Data Mining

We used regression to find a best fit line on X (snowfall received, precipitation received), Y (Deltas in reservoir levels). This gave us the correlation of reservoir levels with precipitation and snow.

Next, we took the reservoir level delta and ran regression with X (reservoir level delta) and Y (groundwater pulled). This gave us the relationship between how the fluctuations in reservoir levels relate to the amount of groundwater that needs to be pulled.

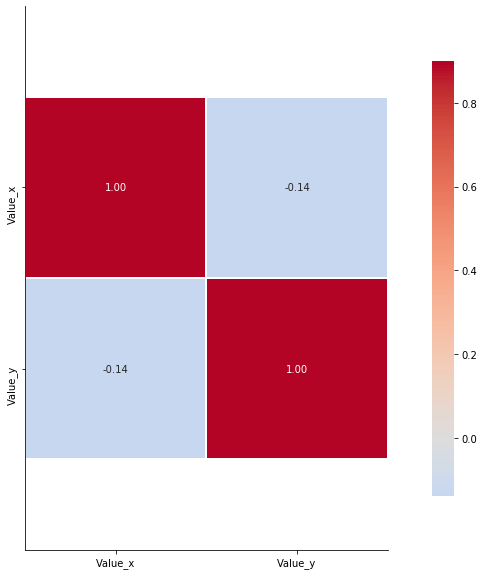
Another factor in this equation is the aqueduct water which is generally the go-to choice for water before groundwater. We built a second regression model with the amount of water available.

We found that for the entire state, having a large correlation between two metrics, snow water content and precipitation would not be feasible, we would see a large correlation between one of the metrics and deltas in reservoir levels based on the region.

We then looked at the water supplier to look at the water usage among the population. We found a linear relationship between the population and water usage.

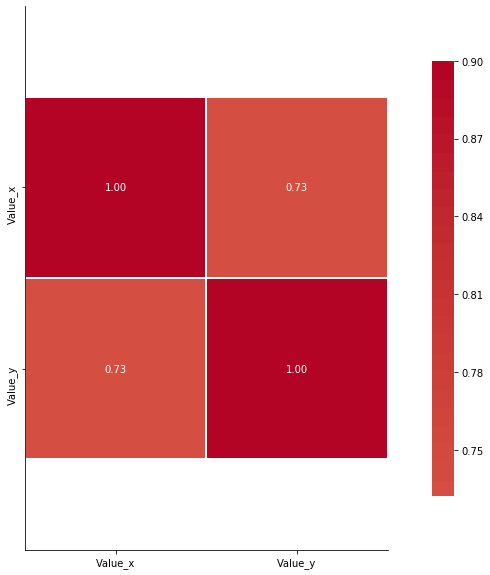
1. Feature Engineering

Feature engineering was a challenging aspect due to the fact that California has probably the most complicated water system in the world. Initially, we thought that it would be possible to treat water that is built up in a given reservoir as a flow networks problem with only inputs into the reservoir being rain and snow melts. Once we started exploring the data, our assumptions turned out to be incorrect.



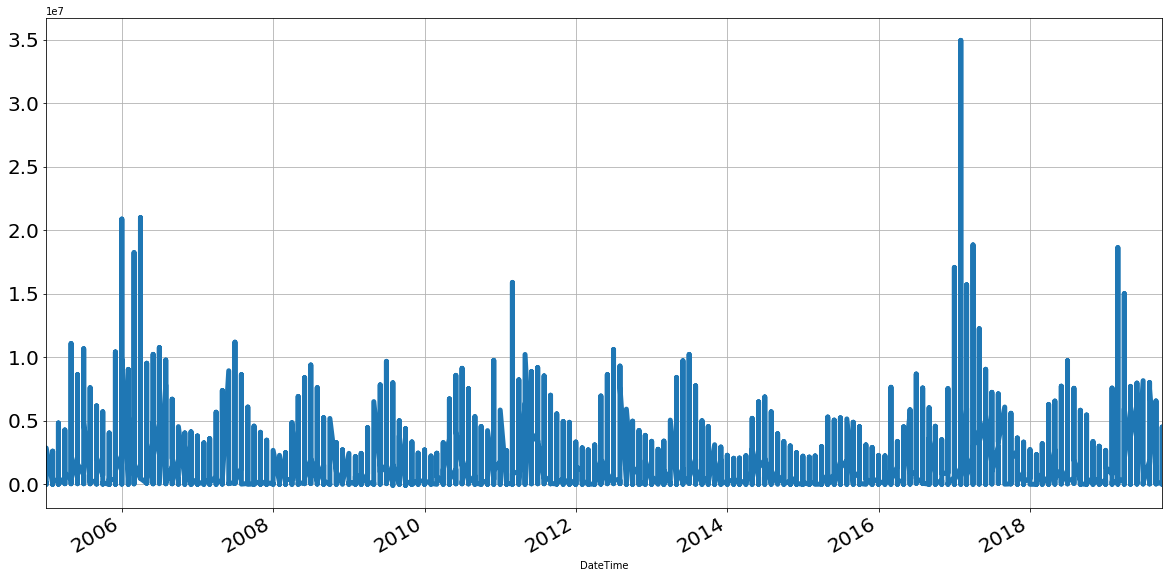
**Fig 1**: Rain vs. Reservoir level correlation. Value\_x are the reservoir storage amounts Value\_y are the precipitation amount

This figure shows our correlation between precipitation amounts and reservoir levels. The low correlation can be attributed to the fact that there is an extensive network of watersheds in California that transport water. Some of the watersheds directly dump water into the outflow of a reservoir or put it into a lake. This means that rainfall in one location does not necessarily correlate with the amount of water that ends up in a reservoir. The same is the case for snow



**Fig 2**: Rain vs. Snow level correlation. Value\_x are the snow cap in inches Value\_y are the precipitation amount

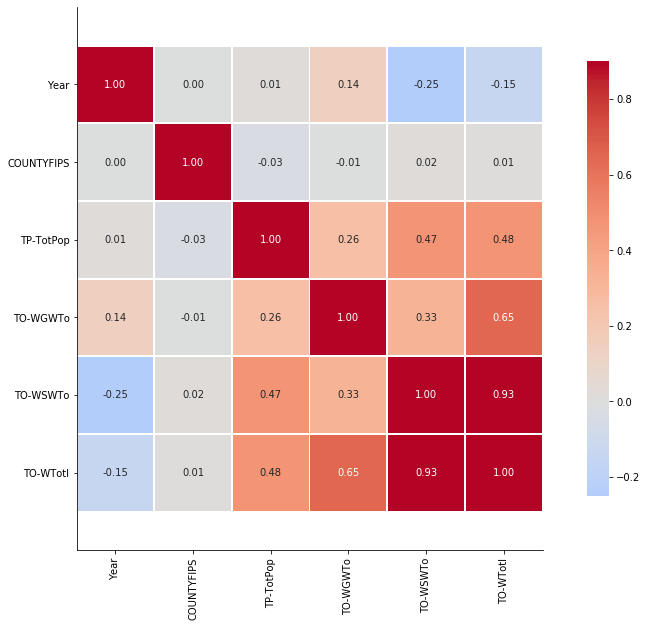
This was performed as a sanity check, because multiple sources state that rain and snow are correlated. It can also be ascertained from the fact that snow is frozen water vapor. The above chart validated our data.



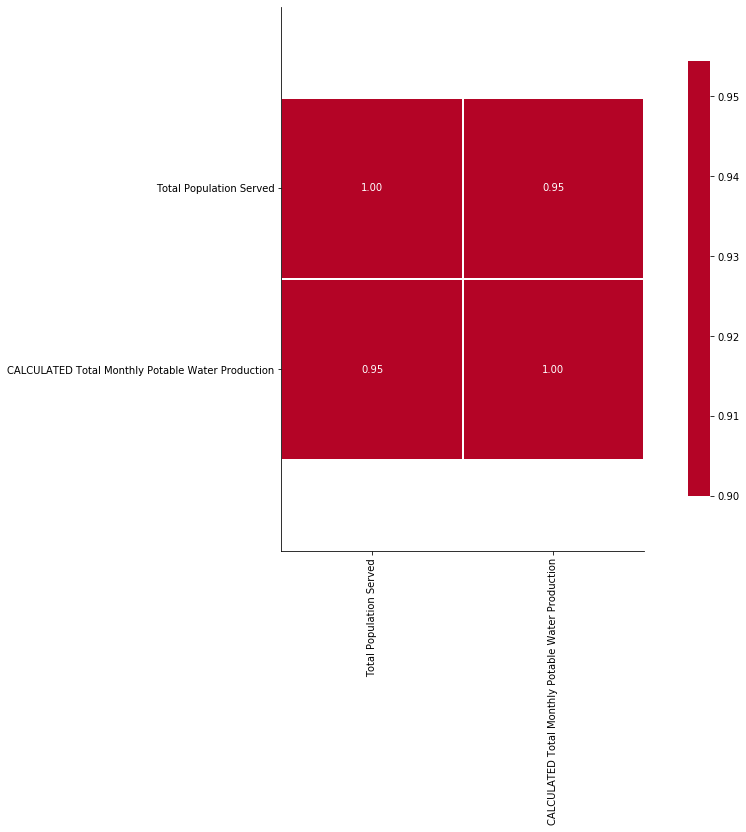
**Fig 3:** Normalized total reservoir outflow value time series

The next phase was looking for a metric that actually would lead to conclusive results with respect to available fresh water supply. From the reading about watersheds, we realized that the outflow metric of a reservoir is tracked. Then, we proceed to look for those values in our data. The unit used for the outflow values is CFS which is cubic feet/sec. This allowed us to convert the outflow values into the amount of water. As a rough estimate, we could use 1 CFS = 7.3 gallons of water. Gallons could then be mapped to maf (million acre feet).

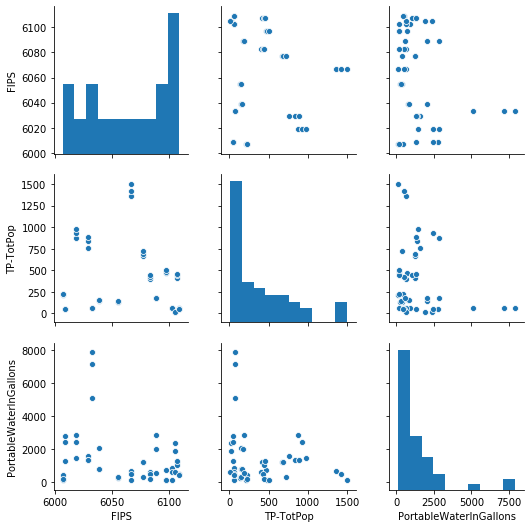
Next part of feature engineering was modeling the demand side of the problem. We drew a correlation plot to see which features were most intimately related with growing total water demand. This was a relatively easy problem to answer because what we saw in the matrix below was that population was one of the most directly related factors, as can be seen in figure 4.



**Fig 4:** Correlation plot showing relationships between total population, total ground water, surface water, and total water



**Fig 5:** Correlation plot showing relationships between total population and total potable water



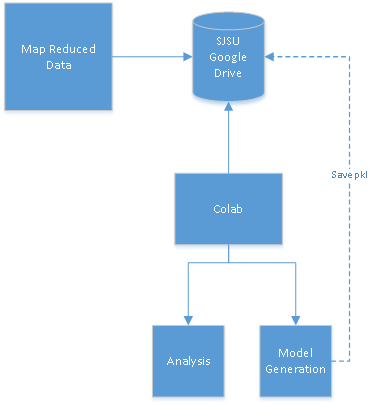
**Fig 6:** Scatter plot representation for total population vs. potable water supplied

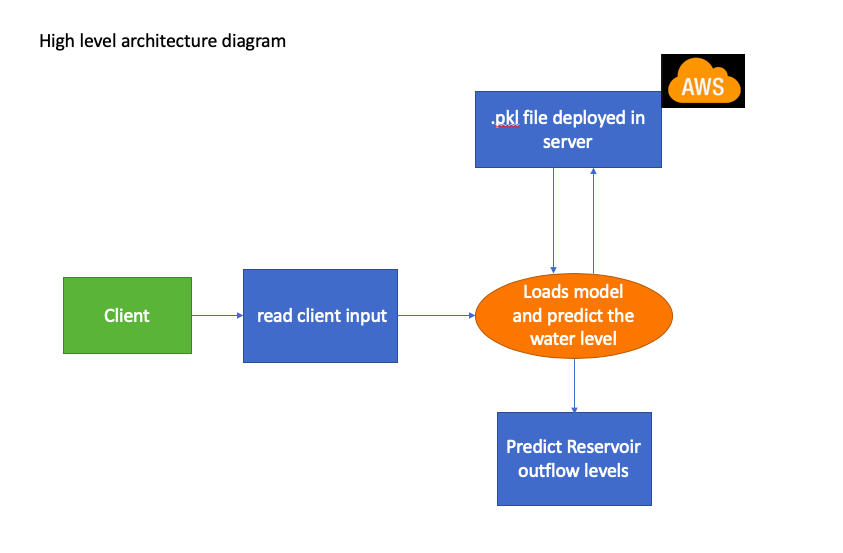
Last and probably the most important aspect of feature engineering we did is that after realizing there is no correlation between outflow from a sensor in a given county and the water used by that county, we decided to do the two models separately and combine them with a common metric which is the volume of water used vs. outflow from reservoirs.

1. High Level Architecture Design

In the high level architecture below, we will deploy our pickle file in aws, and based on the user input we will load our models.

These models will predict the reservoir water level as well as predict the water usage of the population. Then a final model will predict the available water supply for the population.





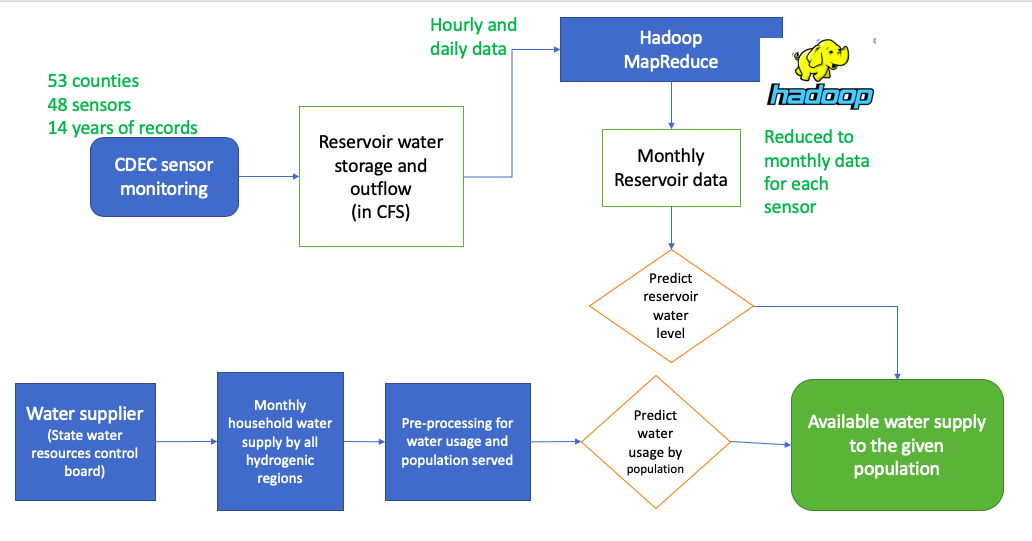
1. Data Flow Diagram and Component Level Design

We started with our sensor data from the California Data Exchange center. From this we gathered the storage and outflows (CFS). This data was converted from hourly and daily data to monthly data. To reduce the large scale data quickly and efficiently we used Hadoop mapreduce. Using the reduced reservoir data, we used that to predict reservoir water levels.

On a second level, we used our Water supplier data to determine the monthly water supply to households within a hydrogenic region. The water usage was then used to determine usage based on the population.

At the endpoint, we combined both our predictions to determine if we would have enough water for the population based on the usage and the amount of water available in California’s reservoirs.

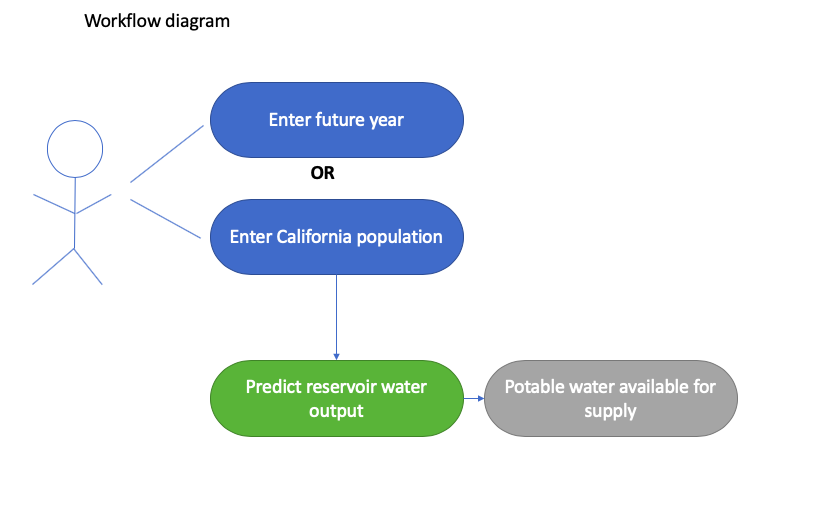
In the diagram below you can see the flow of the data from collection to building a model.



1. Sequence/Workflow

The user will navigate to <http://ec2-13-52-178-250.us-west-1.compute.amazonaws.com:8080/>. Here they will have the option to either enter the future year or enter the california population.

This will then output the reservoir water level and portable water supply for California.



1. Data Science Algorithms and features used

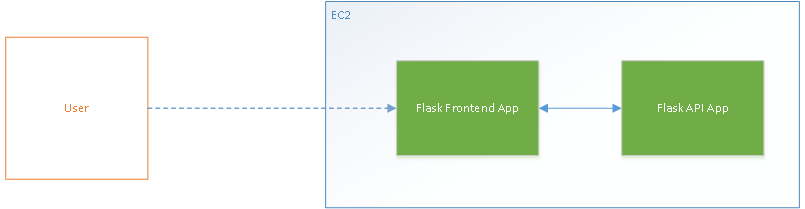
We used variations of two algorithms mainly.

* Linear Regression
  + Ridge
  + Lasso
* Successive approximation model
* Performance metrics
  + R-squared test
  + Time series

We also used Random Forest Clustering for predicting the weight of each of the variables when forecasting water demand for population, but it yielded poor results.

1. Server Side Design

We use flask, scipy, fb-prophet on the server side to do the computations and serve the results via a REST API. Another flask application hosts the front-end and calls the REST Apis to take input and serve the results to the front-end.



Interfaces:

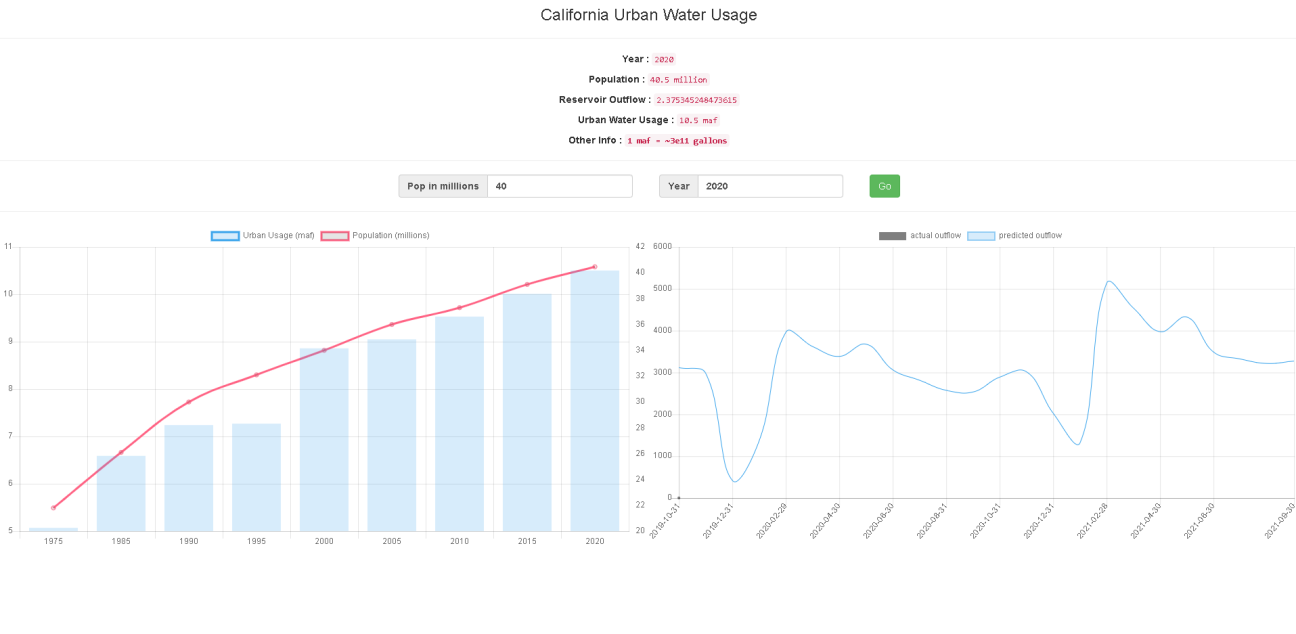
GET Homepage → Returns 2019 Population and outflow data

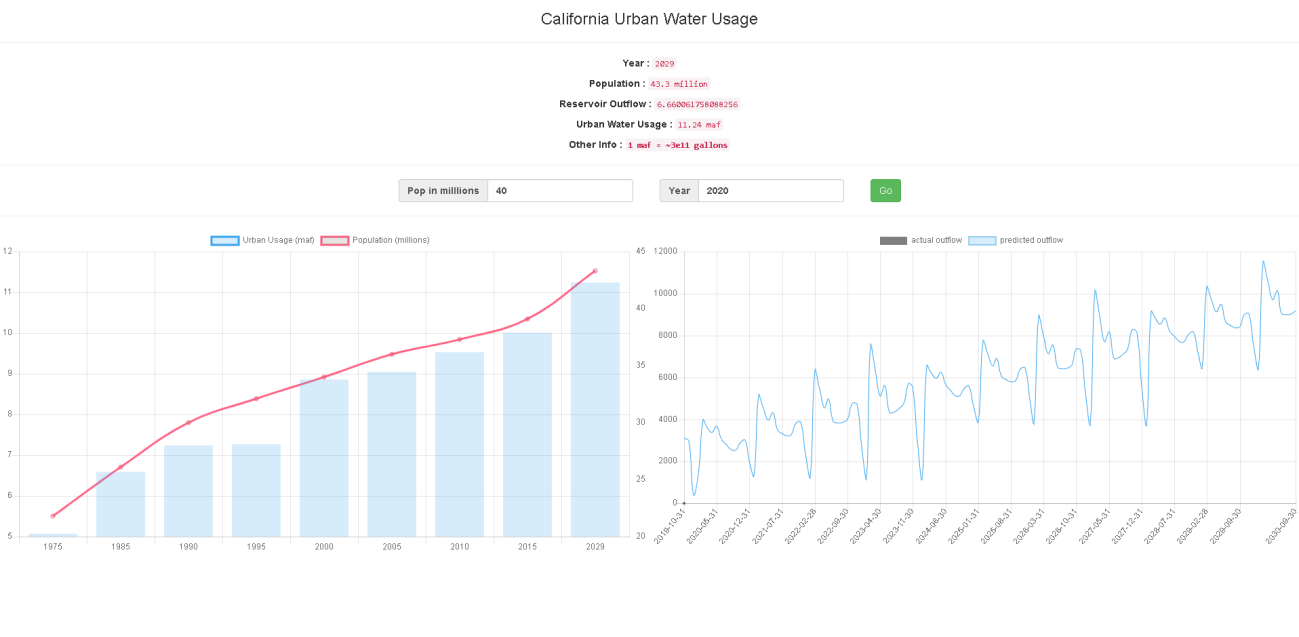
GET Args: pop=X&year=Y → Returns query based on forecasted population for given year. If no pop is provided, fetches forecasted population for the year itself.

1. Client Side Design

Client side was designed using Client.js and jquery.js. The top of the page shows information about the data displayed in the charts below. The year is for the year being considered, population (fetched or entered) value for the year, forecasted outflow from all reservoirs and lastly, expected urban usage based on population.

More info section is included for handy tips to convert between various units, etc.



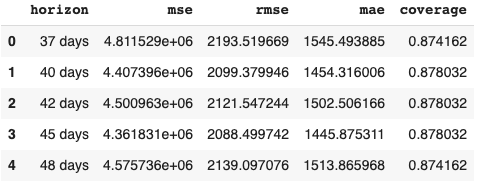


1. Testing – Data and Model Validation

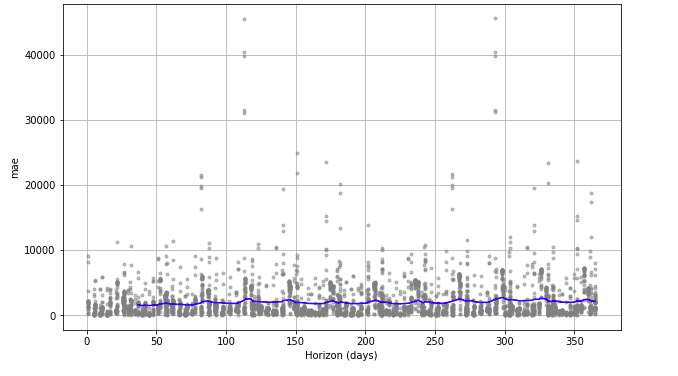
We have developed two models for predicting the water usage statistics. The linear model for predicting the water usage by population number is validated with accuracy score. The accuracy of the model is 91.22%.

Another multiple time-variant regression model to predict the reservoir water level for the future year. We have applied cross validation to evaluate time series data. The period was splitted into 5 years. The model was training in historical data and forecasted on future data.

The validation results are shown in the table below.



The validation plot for performance metrics



The blue line indicates the ‘mean’ value, which is the mean of the rolling window. As the figure indicates, this forecast has an error of 5% and the error increases to 10% for a few years. Overall coverage of the model ranges around 87%.

1. Model Deployment

Our model was deployed on Amazon Web Services. We built a web app that was deployed on an AWS medium sized ec2 instance. The instance had 4gb of ram and 8gb internal storage. This was capable enough of hosting our web app and running our model.

Our model was loaded into a pickle file, which was then deployed to our AWS instance with flask and python.

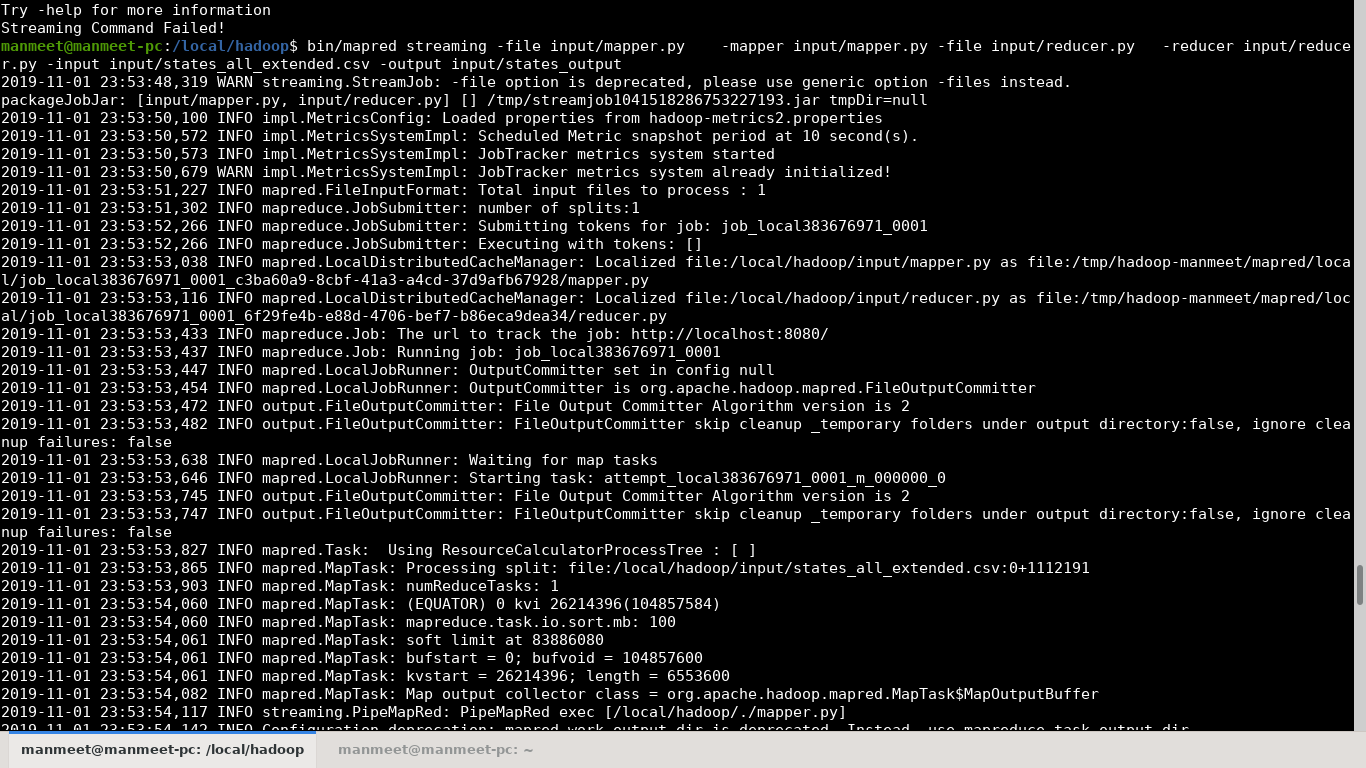
1. HPC

We used a hadoop cluster to reduce our data from the California Data exchange center. We initially started with 12gb of data, which was provided to us in both hourly and daily format. Each sensor had its own csv file that contained different levels of information. For example, precipitation data was primarily sourced from rain sensors. While snow sensors had data about snow depth, snow water content, and adjusted water content. Reservoir data consisted of storage water, outflows, inflows, and other items.

We preprocessed the data to determine which types of data we would require to keep. Once we had determined that information we wrote scripts in python to help find those data points and save them to a new csv file. While running this script we noticed that it took quite a long time and to be able to do this for the entire dataset would not be feasible.

Our solution to this problem was to setup an hadoop cluster on local hardware that could run our mapper and reducer. The mapper searched for the points that we wanted and output them to the standard in. We then took this standard in and used our reducer to consolidate the relevant information into monthly data.

Below is a screenshot of us running the hadoop cluster to help reduce the amount of data.



Using hadoop, we were able to reduce the time it took for us to consolidate the data to about 15 minutes. The amount of data as well reduced from 12gb to approximately 100 mb.

1. Documentation

To get started on the project we take a look at the github project. <https://github.com/manmeet3/Cmpe256_H2O/tree/webapp>. This project's readme tells how to deploy our webapp.

We start with creating a virtual environment with python 3.6.9, then install the requirements.txt file with command: pip install -r requirements.txt. Finally we run the application with python app.py.

This will launch the application at <http://locahost:5000>

To go through the jupyter notebooks, we can launch jupyter notebook. Navigate to the directory that you had cloned above. Then go ahead and open the notebook to view the metrics, evaluation, visualization, and other items relevant to this topic.

1. Design Patterns Used

For our webapp, we used the classic MVC pattern. The models and their deployment is kept in a file, while the controller is the REST API. Our view is the front-end flask application.

The flow of our application from data to pre-processing to model generation is mentioned in sections V and VI.

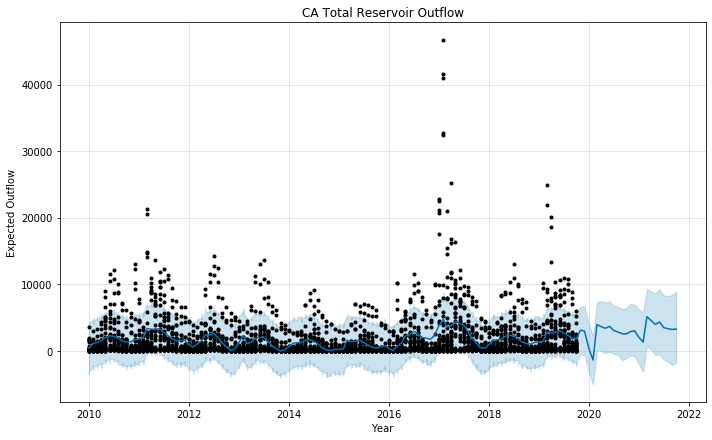
1. Data Engineering

California water boards maintain water supply data for each city separated by hydrogenic stations. Hydrogenic stations are not categorized by county. State water resources control board maintains water usage data across California in terms of counties and population as per 10000 people in county.

We have first converted all the water units into Gallons and aggregated multiple hydrogenic station data into California state data. CDEC sensors have reported water levels in various units and different time patterns.

Using Hadoop mapreduce we have aggregated hourly, daily data into monthly pattern as per the overflow and storage information. The reservoir outflow water value is measured in CFS, which is a new metric and we have converted all data to Gallons for further analysis.

1. Model Interpretation



The above model shows time series of outflow from reservoirs. The black dots are the data points provided to the fb-prophet. As can be seen above, the model tries to find a best fit line using all the data points and extend it into the future. It’s fairly resilient to the outliers, which can be tuned with the hyper parameter called changepoint\_prior\_scale which changes the level of fluctuations it takes into account while modeling.

One limitation that can be seen from the above model is that it can at best be expected to predict one decade into the future, and probably less. From historical events we know that there was a drought in California that was followed up by higher amounts of water in reservoirs and higher outflows.

This bias in the model trains it to expect rising reservoir outflow levels into the future.

1. References

[1] California State Water Resources Control Board. “Water Conservation and Production Reports.” *SWRCB.gov*, www.waterboards.ca.gov/water\_issues/programs/conservation\_portal/conservation\_reporting.html#monthly\_archive.

[2] *Science Explorer*, www.usgs.gov/science-explorer-results?es=National%2BWater-Use%2BData&classification=pub

[3] *Total Water Use*, www.usgs.gov/mission-areas/water-resources/science/total-water-use?qt-science\_center\_objects=0#qt-science\_center\_objects.

[4] Water-Use Data Available from USGS.” *USGS Water Resources of the United States*, water.usgs.gov/watuse/data/index.html.

[5] *USGSWaterDatafortheNation*, waterdata.usgs.gov/nwis/water\_use?format=rdb&rdb\_compression=value&wu\_year=2005&wu\_category=PS&wu\_category\_nms=Public%2Bsupply%252C%2Bin%2BBgal%252Fd.

[6] Bawsca.org, bawsca.org/uploads/userfiles/files/BAWSCA\_Annual\_Survey\_FY17-18\_Final.pdf.

[7] “California Water 101.” *Water Education Foundation*, www.watereducation.org/photo-gallery/california-water-101.

[8] *California Water*, 2018, [water.ca.gov/-/media/DWR-Website/Web-Pages/Programs/California-Water-Plan/Docs/Update2018/Final/Accessible-California-Water-Plan-Update-2018.pdf](http://water.ca.gov/-/media/DWR-Website/Web-Pages/Programs/California-Water-Plan/Docs/Update2018/Final/Accessible-California-Water-Plan-Update-2018.pdf).

[9] “As California's Groundwater Free-for-All Ends, Gauging What's Left.” *For*, 9 Nov. 2018, west.stanford.edu/news/blogs/and-the-west-blog/2018/california-s-groundwater-free-all-ends-grappling-how-measure-what-s-left.

[10] Area Wise High Resolution Water Availability Estimation Using Heterogeneous Remote Sensing and Ensemble Machine Learning - IEEE Conference Publication, ieeexplore.ieee.org/document/6985424.

[11] California State Water Resources Control Board. “California State Water Resources Control Board.” *SWRCB.gov*, www.waterboards.ca.gov/.