



Milestone II Project Report

A Thirsty Valley

Aim: Predict San Joaquin Valley (CA) groundwater depth in feet

California's Sustainable Groundwater Management Act (SGMA)^[1] was passed in 2014 with the intention to address over pumping, halt chronic water-level declines and bring long-depleted aquifers into balance. "Despite SGMA, a frenzy of well drilling has continued on large farms across the **San Joaquin Valley, the state's largest and most lucrative agricultural zone**. As a result, shallower wells supplying nearly a thousand family homes have gone dry in recent years."^[2]

Frequently, perniciously drought-inflicted California, depends on groundwater for a major portion of its annual water supply, particularly for agricultural and domestic usage. This project seeks to aid policy makers and natural resource management agencies preemptively identify areas prone to overdraft and bring groundwater basins into **balanced levels of pumping and recharge**.

Focused on the San Joaquin Valley, the objectives are:

- **Supervised Learning:** Predict the groundwater depth in feet below ground surface (GSE_GWE). This value portends shortage in a TownshipRange. Increase or decrease in GSE_GWE should then indicate if there will be more requests for well construction. This in turn should provide a quantitative metric for whether SGMA is functioning and areas to focus on for recharge.
- **Unsupervised Learning:** cluster areas into sustainable and unsustainable areas. Detect groundwater depth anomalies within Township-Ranges in the river basin.

The Data

Data Sources

We have collected 10 geospatial datasets from federal and state (CA) government agencies for the 2014-2021 period, on the factors we think are impacting groundwater depth: San Joaquin Valley Public Land Survey System data, current groundwater levels and consumption through well completion reports, agricultural crops, population density, regional vegetation, reservoir capacity, recharge through precipitation and soils survey (see Appendix 2).

Although we retrieved the data for dry wells and water shortage reports, this dataset was not included as this is a voluntarily reported information which is not verified, is not complete and has potential errors.

Dataset	Retrieval Location	Retrieval Mode	Format	Records	Time Period
Crops	California Natural Resources Agency	Download	Geospatial datasets	1159979	2014, 2016, 2018
Groundwater	CNRA Groundwater	API	CSV	5064676	2014-2022
Population	US Census Bureau	API	JSON	5465	2014-2020
Precipitation	CDEC Precipitation	Web scraping	HTML	1418	2013-2022
Reservoir	CDEC Water	Web scraping	HTML	224	2018- 2022
Water Shortage	CNRA Well Water Shortage Reports	Download	CSV	4792	2015-2022
Soils	USDA Soils	Download + manual extraction of the table	Microsoft Access Table	2139	2016
Vegetation	USDA Forest Service	Download	Zip File	54809	2018, 2019
Well Completion	CNRA Well Completion Reports	API	JSON	2139	2014-2022
Census Bureau Shapefile	L.A. Times GitHub page	Download	GeoJSON	478	Constant

Table 1: Data Sources (click on the dataset links for explanation on features and extraction).

Data Manipulation and Aggregation

All the datasets were aggregated at the Township-Range granularity (see Appendix 1.2) on the spatial dimension and at the year level on the time dimension. For each dataset we computed each feature value per Township-Range and year. To do so we performed two main types of data transformation.

The first one was to overlay the Township-Range boundaries over geospatial data like crops (Fig. 1), soils, vegetation, population density, to compute the land surface used by each feature in each Township Range and year (e.g., the land surface used by each crop).

Some datasets, like precipitation, provide point measurements instead of spatial ones. To transform such data into spatial data, we used the Voronoï Diagram method. It computes the Voronoï polygons by using the measurement points as the Voronoï polygons center. We then overlaid the Township-Ranges

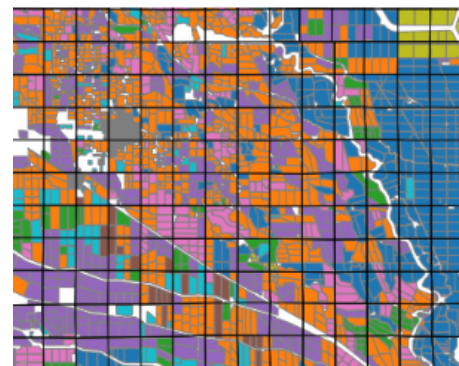



Figure 1: Overlaying Township-Range borders on the Crops dataset



boundaries on top of the computed Voronoï diagrams and for each Township-Range took the mean value of all the intersecting polygons.

The result of the transformations and aggregations applied was a dataset of 478 Township-Ranges, each containing a multivariate (80 features) time series (data between 2014 to 2021).

Imputation Pipeline

As seen in the Appendix 3 Missing Data, there are missing data that need to be addressed. We used a pipeline to both impute and normalize data after the dataset is split into train and test sets. Creating this pipeline and fitting it to the training set makes the application of normalization uniform across the datasets when we then transform the test set. It also guards against data leakage. While crops, soil and vegetation were imputed applying custom forward-fills, for reservoir capacity and ground surface elevation we created custom group transformers, taking the minimum and median of each Township-Range group respectively.

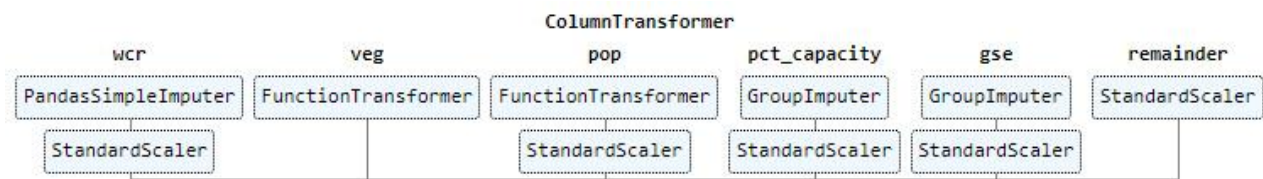



Figure2: The Imputation Pipeline

What is lookahead: The pipelines we created posed a special challenge in splitting the data since several of the imputations required past information from the same township range which were in the training set alone. We needed to guard against lookahead to avoid leaking knowledge of the future when designing, training, or evaluating a model.

Walk-forward validation: In walk-forward validation, the dataset is first split into train and test sets by selecting a cut point. For the machine learning approach, we chose this as the year 2019 e.g. data for years 2014-2019 are used for training, 2020 used for testing and 2021 for prediction.

While tree based algorithms are not sensitive to scale of data and do not require scaling and normalization, algorithms such as Support Vector Regression (SVR) that aim at maximizing the margin between the hyperplane and the closest supporting data vectors are sensitive to the scale of individual features. Both of these are discriminative models, predicting Y given X . We evaluated MinMaxScaler (range 0-1) and StandardScaler (subtracting mean and unit variance) and found that evaluation metrics and PCA components generated varied based on the scaling used. StandardScaling resulted in a marginally better R-squared and a more intuitive set of top features in PCA components. As stated in scikit-learn documentation^[21], “many elements used in the objective function of a learning algorithm (such as the RBF kernel of Support Vector Machines) assume that all features are centered around 0 and have variance in the same order.”



However, in order to have the values between 0 and 1 for the deep learning algorithm, a MinMax scaling method was used.

Supervised Machine Learning


Motivation

In predicting groundwater depth in the Township-Ranges, the primary goal was to predict the areas where the water resource management and affiliated agencies should focus their effort to address water shortage and wells over drilling. The second objective was to identify the most important subset of factors contributing towards the predictions.

Train-Test and Target Split

For the machine learning approach converting the multivariate, multi-indexed (Township-Range + year) dataset to a supervised learning prediction task required (Appendix 4):

1. Including the current groundwater depth as a feature : (GSE_GWE renamed as CURRENT_DEPTH)
2. Shifting the groundwater depth of the next year as the prediction target for the current year.

The shift resulted in the loss of one feature data in 2021. Which implied that while each train, test and prediction set contain all Township-Ranges, the train set years are 2014-2019, the test set year is 2020 and the prediction set included year 2021. While transformation of the target is typically not necessary, in the context of deep learning, “A target variable with a large spread of values, in turn, may result in large error gradient values causing weight values to change dramatically, making the learning process unstable.”^[14] Sklearn’s TransformedTargetRegressor was used to wrap the regressors used so that the transformation can occur in the pipeline without manually converting the target. Our target depth varied from 0.5 to 727 feet and we observed improvement in metrics upon transformation. 

For the deep learning approach, to fit our dataset and objective, as well as Long Short-Term Memory (LSTM) neural network architectures, we split the train-test sets by group (Township-Ranges) and the inputs and target by time (refer to Appendix 7.1 for the diagram). We used 15% of the Township-Ranges for the test set. This means that models were trained on 406 Township-Ranges and tested on 72 of them. During training 10% of the training data was used for cross-validation.

The Machine Learning Approach

Feature Subset from Feature Correlation and Dimensionality Reduction

The dataset contains 80 features, some of which are intuitively highly correlated (>0.50) such as well yield and its static water level and dimensions. Average precipitation is correlated to reservoir

capacity by a Pearson correlation value of 0.31. Crops such as Onion and Garlic are correlated to Tomatoes. And Blue oak-gray pine which forms one of the largest ancient forest types in California, is naturally correlated to Chaparral with which it mingles at lower elevation. To select features using

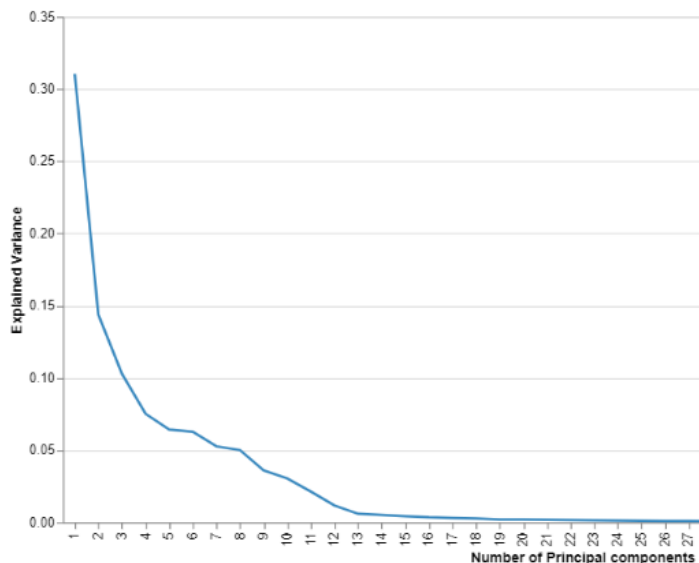


Figure 2: Scree Plot - Explained Variance based on the number of components

quantitative measures, feature correlations to each other and to target were studied along with top features constituting the principal components of a PCA analysis.

Unsurprisingly, predicted depth correlates strongly with current depth. Correlation with other features is significantly weaker (<0.3). Well depth, precipitation, and crops such as Onion and Garlic along with pasture land crops are more correlated with the target than others. (Appendix 6). Some other correlations that were noticed to be higher than others in the same category are those between Onions and Garlic crops that are tolerant to arid soils, mixed pastures to precipitation and

current depth to ground surface elevation.

After observing the feature correlation in the heatmap, we decided to check for latent features in the dataset that will effectively combine the correlated features, and create a new set of features that are a weighted linear combination of original features, using PCA. The number of components selected explain 75% (a threshold we selected based on the Scree plot in Fig 2), of the original variance of the dataset. The PCA biplot (Fig 3) emphasizes the collinearity we saw in the heatmap in features related to a well such as completed depth, top and bottom of perforations (for filter). Vectors with a smaller angle of separation are more correlated.


- The first component contains well features in an area such as its depth and static water level.
- The second component includes well counts and ground surface elevation.
- The third component is largely about precipitation and reservoir capacity, and well counts.
- The fourth component includes precipitation, ground surface elevation and population density and well counts.

We thus derived the sense that well counts play a significant role in the variance of the data along with ground surface elevation, well features, precipitation and reservoir capacity.



Setting a baseline through dummy regressor and linear regression

6



Since this is a regression problem with continuous variables, the evaluation scores find the difference (error) between the predicted value and the observed value. From among the choices of R-squared, Mean Square Error, Root Mean Square Error, we selected the mean magnitude of the errors in a set of predictions (Mean Absolute Error) as the error to be communicated since it uses the same units as the target and is relatively easy to understand in the context. The Mean Absolute Error (MAE) is calculated by taking the summation of the absolute difference between the actual and predicted values of each observation over the entire set and then dividing the sum obtained by the number of observations in the dataset. These are negatively oriented scores which means lower scores are better.

R-squared and MAE

For the initial comparison of algorithms, before hypertuning and for generating a list of models for PyCaret to compare, R-squared gave us a quick indication of the relative performance. R-squared provides the proportion of the variance for the target that's explained by selected features in the model. An R-squared value of 0.9, for instance, would indicate that 90% of the variance of the dependent variable being studied is explained by the variance of the independent variable. It is independent of the scale of the features and ranges from 0 to 1. A negative value implies worse than the mean model.

When evaluating an algorithm, it is prudent to look into multiple regression scores and not just R-squared since the acceptable threshold of the error will depend on the distribution of the target value itself. To keep the groundwater depth prediction difference from the mean in check, we additionally look into MAE, MSE and RMSE and communicate the percentage mean absolute error relative to the mean of the true target value with the stakeholder.

Reasoning behind the models selected

The baseline R-squared, established with DummyRegressor is 0.054, a very low value. PyCaret's initial model comparison results hinted at tree based models being among the top five including: GradientBoostedTrees (creates sequential learners with learning rates that works very well with heterogeneous tabular data with categories mixed in), RandomForestRegressor (ensemble bagging model) and CatBoost Regressor (which is surprising since the dataset has continuous variables) and XGBoostRegressor. PyCaret also did not pull up Support Vector Machines in the first five top models although testing this algorithm with a radial based kernel showed early promise with the many featured dataset. Other advantages of SVM considered:

- Effective in high dimensional spaces and where number of dimensions > number of samples.
- Uses a subset of training points in the decision function (called support vectors), so it is also memory efficient.
- Versatile: different [Kernel functions](#) can be specified for the decision function. Common kernels are provided, but it is also possible to specify custom kernels^[21].

Train			Test		
Regressor	MAE	R^2	Regressor	MAE	R^2
Random Forest	22.9536	0.9006	SVR	29.1754	0.8779
XGB	22.9395	0.8951	RandomForest	30.9034	0.8468
GradientBoost	23.171	0.8903	GradientBoost	32.5584	0.8547

Table 2: Machine Learning Model Error Metrics

Manually setting up the parameter grid for RandomForestRegressor and SVR after this initial analysis through PyCaret, provided better results in optimizing the parameters through RandomSearchCV. R-squared shows how well the data fit the regression model.

After evaluation on both train and test set results and a combination of evaluation metrics, R-squared and MAE, RandomForestRegressor

was picked as the top model as it has a superior train score and acceptable generalizability test scores of the various models. Advantages of Random Forest algorithm over Decision Trees and SVM :

- It can be understood easily
- It can handle large datasets efficiently
- Provides more regularization over decision trees.

The other factors considered were the number of samples and absence of categorical features and sensitivity to outliers. Results of the top three algorithms are shown in Table 2. Cross-validation for hypertuning of parameters of each of the models, was carried out using RandomizedSearchCV^[20] which unlike GridSearchCV, does not try out all possible parameter values, but performs a fixed number of parameter settings, randomly sampled from the specified distributions.

The MAE of the chosen regressor is 31 feet which when put into perspective in relation to the mean of the target (167 feet), is a large percentage error of 18.5%.

The Deep Learning Approach

The dataset can be seen as a 3 dimensional dataset of 478 Township-Ranges x 80 features x 8 years, with the objective to predict the 2022 groundwater depth. LSTMs are used for time series and NLP because they are both sequential data and depend on previous states and thus perfectly fit such a prediction scenario.

We trained 3 different LSTM model architectures. A simple one with a unique LSTM layer. A second one with an LSTM layer followed by a dense and dropout layer. The third one was an encoder-decoder model^[9] made of 2 LSTM layers followed by a dense and dropout layer (refer to Appendix 7.2 for the model architectures' diagram). The hypertuning of the model parameters was performed using a Bayesian search of the hyperparameter space.

The best model on the test set was the simplest model and the one we used for the rest of the analysis. Once trained on the 2014-2020 data to predict 2021, the model was used to predict the

	MAE	MSE	RMSE
model 1	23.79	1208.83	34.77
model 2	30.34	1814.70	42.60
model 3	30.21	1610.08	40.13

Table 3: Deep Learning Model Error Metrics



2022 groundwater depth based on the 2015-2021 data. The Township-Ranges with high or low groundwater depth or with high or low year-to-year variation of groundwater depth were retained by the model. But considering the groundwater depth varies in the dataset from 0.5 to 727 feet and has a mean of 167 feet, a root mean square error (RMSE) of 34.7 feet and mean average error (MAE) of 24.8 feet feels too high to achieve the objective of predicting areas where water management requires attention.

Results Evaluation and Analysis

Sensitivity Analysis

We performed two sensitivity analyses of the models. The first one on the amount of data the model was trained on, the second one on the hyperparameters. Due to the higher number of data the deep learning was trained on (7 years of data compared to 1 for the machine learning models) and the higher amount of hyperparameters used by the deep learning model, we limited this analysis to the deep learning model.

Training Data Size Tradeoffs and Sensitivity

To evaluate the impact of the amount of historical data used to train the model and perform predictions, we recursively trained the model with 1 to 7 years of data. The first model was trained only on the training set of 406 Township-Ranges 2020 data, to predict the 2021 groundwater depth and tested on the 72 Township-Ranges in the test set. Then the model was trained with 2019-2020 data, then 2018-2020 and so on. We see in Appendix 9.1 that although, at the beginning, the RMSE reduces as we add yearly data to train the model, the improvement in prediction is minor. The RMSE only reduces from ~42.5 feet to ~34 feet by increasing the number of historical data from 1 year to 4 years. This indicates that just adding a little bit of yearly data to very little yearly data to train the neural network model is not enough to significantly improve its performance. As neural networks tend to require a lot of data, the hypothesis that having much more historical data would improve the model performance is still a valid hypothesis. But the analysis also suggests the hypothesis that the model performance issue might also be related to the quality of the data or the features used.

Hyperparameters Sensitivity

To perform a sensitivity analysis of the deep learning model to the hyperparameters, we trained 33,345 models in parallel on 3 cloud servers during ~12 hours across an exhaustive combination of all hyperparameters within the defined ranges. The parameters included: the optimizer used (Adam RMSprop, Adagrad), the training/validation dataset split, the learning rate, the batch size, the



number of training epochs and the number of lstm units. The full results are shown in the visualization in Appendix 9.2.

Displaying for each hyperparameter, the concentration of models per RMSE score and the average RMSE mean (using the color) depending on the hyperparameter values (the lower and the more blue the peak is, the better the hyperparameter value is), we found that the choice of the optimizer seems to have the largest impact on the model performance as seen in Fig. 4 where an Adagrad optimizer generates mainly models with an RMSE above 100 feet. The best training-validation percentage split seems to be at 10% (Appendix 9.2), but the difference with a split of 5% or 15% looks small, suggesting this hyperparameter has little impact. The bigger the

learning rate the better, while on the other hand, the smaller the batch size the better. With a learning rate of 0.1 most models have an RMSE under 60 feet, while a learning rate of 0.001 shows a binomial distribution of the models with an RMSE under 60 feet or above 130 feet. Although there is less of a difference if we compare close values, the bigger the number of training epochs, the better. With 50 epochs, the distribution of models' RMSE peaks at 140 feet, while with 290 it peaks at 40 feet. Finally, the number of LSTM units (impacting the number of neurons in the model) seems to have less impact on the performance of the model. A small number of LSTM units seems to be better as the distribution shows mainly models with an RMSE of 30 feet while, with a high number of LSTM units, the distribution of the models is much more broad. Following these results we retrained a model using the combination of the best hyperparameters: an Adam optimizer, a validation split of 10%, a learning rate of 0.01, a batch size of 32, 290 epochs and just 10 LSTM units. The model had a MAE Of 29.38 feet and a RMSE of 39.82 feet, much worse than our best model. This shows that the best hyperparameters' combination which produces the best performing model is not the result of the simple combination of the individual best hyperparameters.

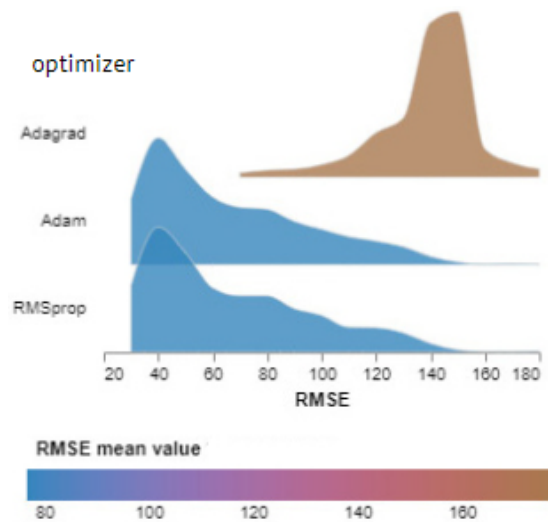


Figure 4: Concentration of models per RMSE score for different optimizers (the lower and more blue the peak is, the better the hyperparameter value)

Failure analysis

Since the MAE and RMSE are quite high, a minimum of 29 feet among the top models(while the range of groundwater depth is from 0.5 to 727 feet), an in-depth analysis was conducted on failure. We wanted to detect the cause and common areas of failure. There were certain Township-Ranges for which quite curiously, there was a spike in depth predicted by all models. As seen in Fig. 5 where

y-axis is the absolute error, graphed against true target, these spikes are between target value of 0-50 feet and then again between 150-200 feet.

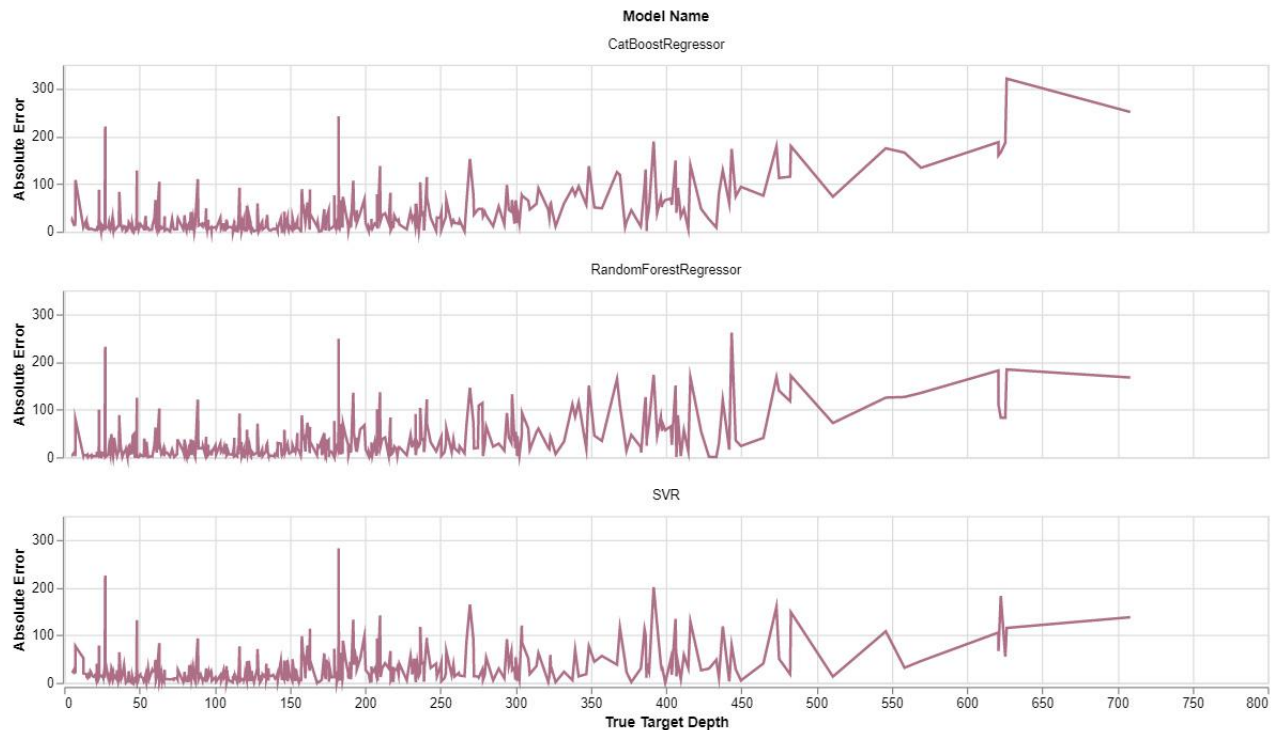


Figure 5: Model absolute errors plotted against target groundwater depth

Isolating these values, shows us that these two Township-Ranges have a very different groundwater depth in the test set (year=2020) from the mean for all years past. Since current depth is the biggest predictor of the target, this is a data issue that will need investigation in further iterations of this study. The Township-Ranges isolated for the spikes (T10S R21E and T15S R10E) have a mean groundwater depth value of 15.18 and 248.85 feet respectively prior to 2020. The depth unaccountably changes to 267.65 and 483.50 feet in 2020, indicating potential data issues.

Explainability through SHAP

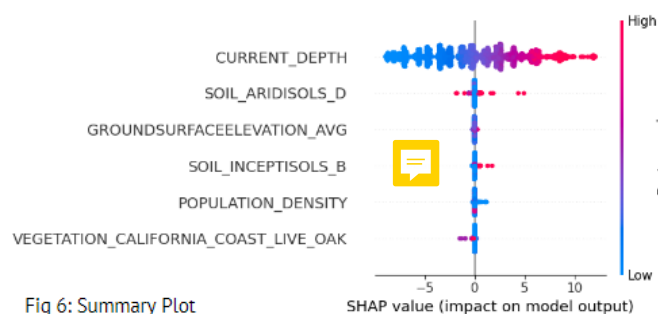


Fig 6: Summary Plot

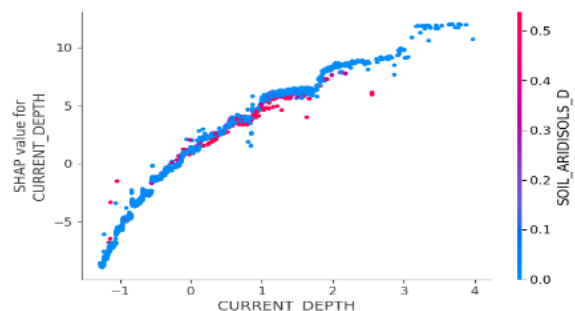


Fig 7: Dependency plot

We used **SHapley Additive ExPlanations** to explain the predictions of the RandomForestRegressor by generating summary plot, dependency plot and force plots. The SHAP model is trained on the transformed imputed train set which is the background distribution. The tests are conducted on the test set on which we predict values. We have the choice to predict on singular instances or multiple instances. The summary plot (Fig. 6) shows current depth as the most dominant predictor followed by arid soils. The dependency plot (Fig. 7) further emphasizes the relation between increasing current depth and arid soils.

For the force plot (Fig. 8) we pick an instance of over and under prediction by sorting the prediction by the difference of prediction and observed value in the test set. The third lowest under-prediction and the fifth highest over-prediction instance are shown in figure. The base value of 11.9 is the prediction when no feature is taken into account. The plot will display features upon hover. Here we observe that lower values of Groundsurface elevation, followed by arid soil, reduce the value of the prediction in the under-prediction. Whereas the over-prediction in this instance is fueled by Current Depth and arid soils and reservoir capacity.

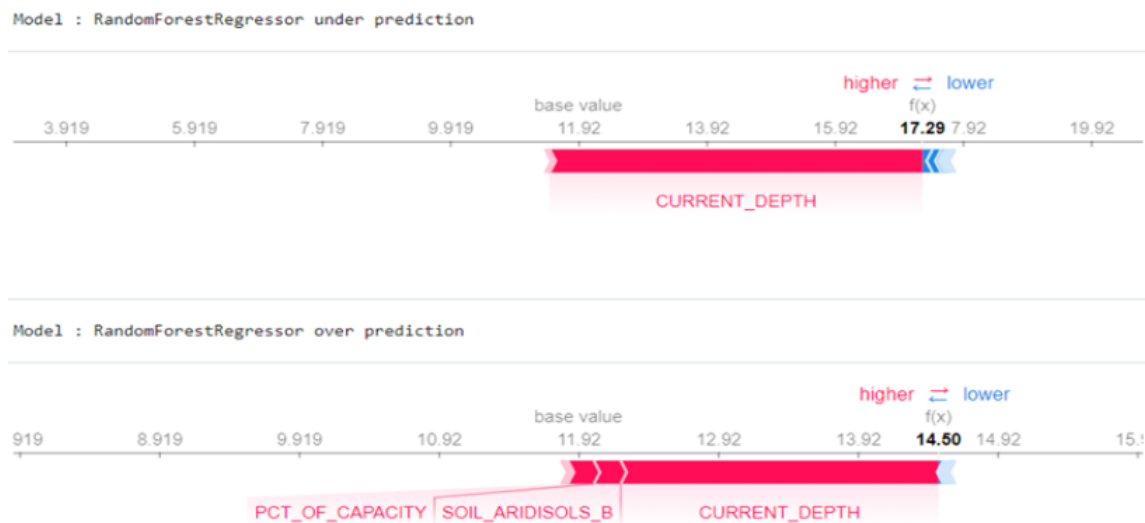


Fig 8: Force Plot for an instance of over and under prediction

Prediction Uncertainty

The predictions made by the RandomForestRegressor are point estimates for each of the Township-Ranges. We create a prediction interval to demonstrate the uncertainty of prediction. “A prediction interval for a single future observation is an interval that will, with a specified degree of confidence, contain a future randomly selected observation from a distribution”^[17]. Linear regression estimates the conditional mean of the response variable given certain values of the predictor

variables, while quantile regression aims at estimating the conditional quantiles of the response variable. By combining the predictions of two quantile regressors, it is possible to build an interval. Each model estimates one of the limits of the interval. For example, the models obtained for $Q=0.1$ and $Q=0.9$ produce an 80% prediction interval ($90\% - 10\% = 80\%$). This is the interval in which a median point estimate will lie 80% of the time.

In the quantile loss equation below, with q having a value between 0 and 1, when $y_i^p > y_i$ (over-prediction) the first term will influence the loss function and the second term will similarly influence it for under predictions. So as q is set closer to 1, over-predictions will be penalized more than under predictions. Regression based on quantile loss provides sensible prediction intervals even for residuals with non-constant variance and non-normal distributions.

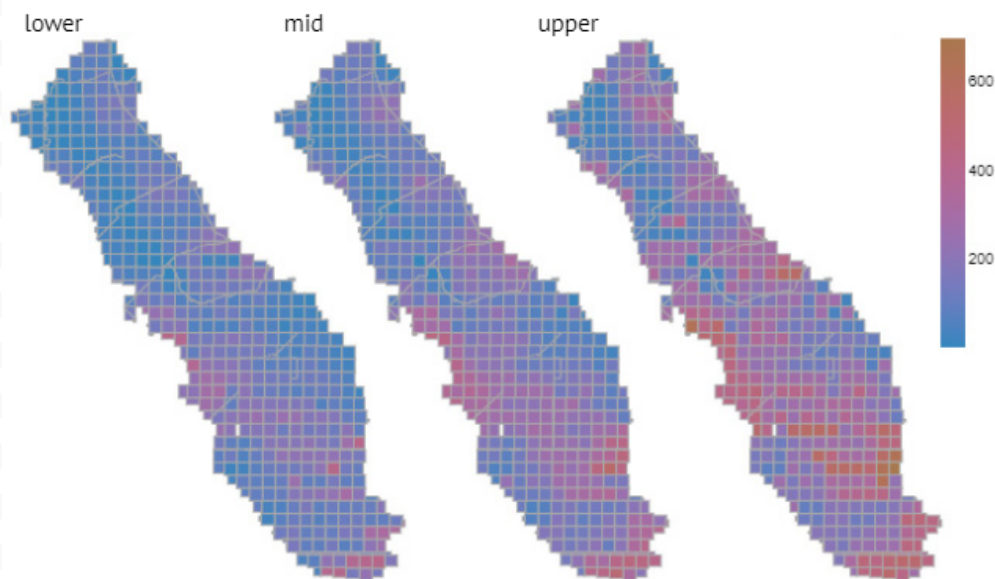


Figure 9: Quantile Prediction Point Estimates

$$L(y_i^p, y_i) = \max[q(y_i^p - y_i), (q-1)(y_i^p - y_i)]$$

Prediction values calculated for RandomForestRegressor are seen in Fig. 9. For instance, in the interactive map in the notebook, T16S 13E in Fresno County indicated a lower quantile prediction of 415.75, median quantile prediction of 462.11 and upper quantile prediction of 538.34. The prediction quantile of 90% interval implies that there is a 90% likelihood that the true outcome is in the 415.75 to 538.34 range.

While there is a range of values for every Township-Range, we do see areas of sustainable depths in Stanislaus and San Joaquin counties in the north of the San Joaquin valley, whereas lower Kern county towards the southern tip of the basin is clearly at risk of much higher groundwater depths.

Unsupervised Machine Learning

Motivation

The objective of this unsupervised learning analysis is, through clustering, to try to identify groups of Township-Ranges and their characteristics. Ideally this analysis would help reveal the characteristics of the Township-Ranges with sustainable water situation and those with systemic water problems.

Methodology

The dataset contains 8 years of data for each Township-Ranges. We decided not to average the yearly data to have one clustering for each Township-Ranges but perform the clustering year by year. This allows us to see if there are a lot of year-to-year variations of Township-Ranges clustering.

We used 3 different techniques to perform clustering: K-Means ,Density-Based Spatial Clustering of Applications with Noise (DBSCAN) and Hierarchical Clustering. For all of these techniques, the models were trained with a grid-search approach to find the hyperparameters giving the best clustering results.

To choose the best k value for the K-Means algorithm, we ran the K-Means algorithm for different values of k between 2 and 16 and evaluate the clustering results based on the following metrics: the *Calinski-Harabasz* score, the *Davies-Bouldin* score, the *Silhouette* score, the *Inertia*. As can be seen in Appendix 8.1, the primary challenge faced is the absence of a clear indication of good k clustering value on any of the 4 evaluation metrics. Having a too high number of clusters doesn't seem appropriate for this analysis. Since the Davis-Boudin (Fig. 10) score seems to indicate a better clustering score

for $k=2$ amongst the lower number of clusters and it matches with the motivations, we decided to use that value for the rest of the unsupervised analysis. With that value of k defined, we then applied the 3 chosen clustering techniques on the dataset and analyzed the results.

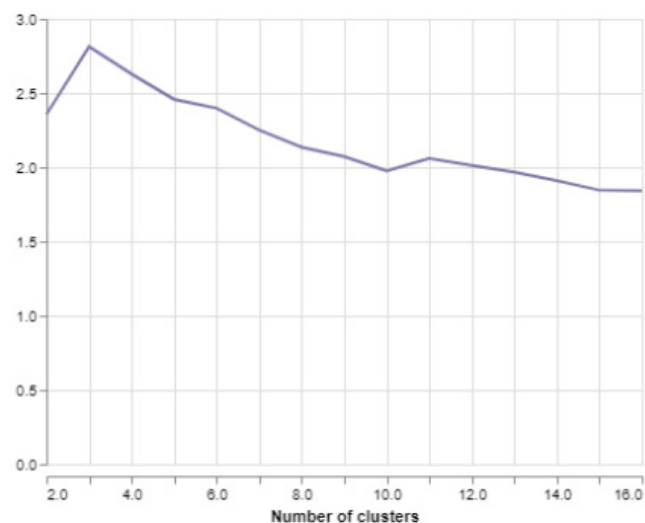
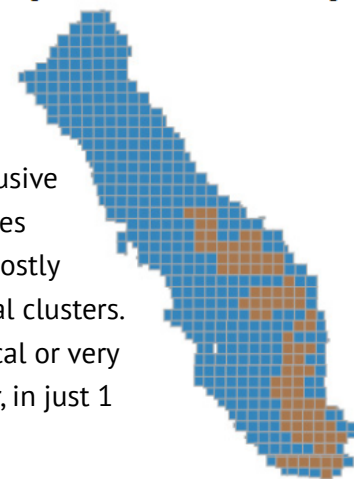


Figure 10: Davies-Bouldin score (the lower the better)

Clustering Results

The K-Means algorithm showed an almost consistent year-to-year classification of the Township-Ranges (see Appendix 8.2 for the full results) with just 10 Townships showing some variations in their clustering for some years. The DBSCAN algorithm results were not conclusive at all regardless of the hyperparameters used. Almost all Township-Ranges were assigned to one cluster while just a handful of Township-Ranges mostly located in the South-East border of the valley were assigned to individual clusters. Finally the Hierarchical Clustering algorithm where either almost identical or very similar to the K-Means results, almost identical to the DBSCAN results or, in just 1 case, proposing a very different clustering.

Figure 11: K-Means 2021 Clustering



Analyzing the Clusters

Although we analyzed the results of that single different clustering computed by one of the Hierarchical Clustering outputs, the analysis was not very conclusive and we focused on the analysis of the results of the K-Means clustering.

As can be clearly seen on the K-Means clustering map for 2021 (Fig. 11), one of the cluster (the brown cluster) is mainly located on the South-East of the San Joaquin Valley, on a axis following the road 99 between the towns of Madera, Fresno, Tulare, Delan and Bakersfield in the Madera, Fresno, Tulare and Kern counties.

As we analyzed the details of the most important features for the two clusters (Fig. 12, see Appendix 8.3 for a more complete visualization), we found out that in addition to the South-East location along the road 99, the Township-Ranges in the brown cluster have poorer sandy soils, require to dig deeper to reach groundwater, have less precipitation, have water reservoirs with less capacity.

To try evaluate if this clustering result could help water resource management agencies to identify areas they should focus on, we used the well depth, the amount of wells drilled and the amount of well shortage reports as potential proxies for measuring water resource sustainability. We compared the Township-Ranges clusters with the top 30 Township-Ranges with the deepest wells (biggest GSE_GWE value) drilled, the highest number of wells drilled, the highest number of reported well shortages, averaged over the 2014-2021 period

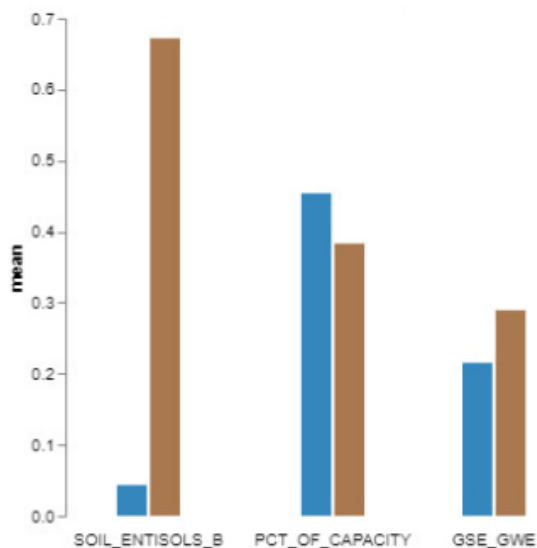



Figure 12: Difference in mean of the top 3 most important clustering features (see Appendix X.X)



(see Appendix 8.4). In all three cases, the clustering does not properly capture these Township-Ranges.

In summary, although partially meaningful, the clustering fails at achieving the objective of identifying Township-Ranges where water management requires attention. Our hypothesis is that the features in the dataset are not appropriate to perform such a clustering.

Limitations

As the results and the failure analysis show, this project suffers various limitations. The range of prediction uncertainty is too wide for us to achieve the project's objectives and advise our stakeholders with enough confidence. For all approaches used, the metrics were too low to promote the project to production. More features and certainly more historical data would have made the results more robust. Gathering insights from environmental field experts would probably help us to select more appropriate datasets or features to achieve our objectives.

Conclusion

Despite the results not being sufficient to achieve the initial objectives, this project was however a great learning opportunity. First of all, we learned to manipulate and extract insights from geo-spatial data. The multivariate time series format of the dataset also forced us to go beyond classic Machine Learning techniques learned during the MADS courses and learn and implement creative machine learning and deep learning approaches. We created custom class libraries and extended classes to genericize standard ML operations for consistency and reproducibility.

Comparing the supervised learning and deep learning approaches we observed that the differences in the prediction output period, the input data structures, and the feature scaling all played significant roles in the results produced. Also, while the deep learning LSTM approach fits the multivariate multi time series structure of the data well, the low amount of data makes supervised learning more appropriate resulting in slightly lower MAE for the Support Vector Regressor. Finally, each approach used in the project emphasized different features in regards to our objective, it did bring some features like soil aridity and some crops (such as pasture crops) to our attention and would be useful findings in a second iteration of this project.

What we will do next:

This project could benefit with a new iteration starting with a qualitative analysis by including environmental experts in the team, reaching out for more data and performing more feature engineering.



Ethical Considerations

We are keenly sensitive to the fact that the subject and set of stakeholders we have chosen to address has real world implications that could potentially impact the residents and farmers in the San Joaquin valley. By no means is this analysis complete or production ready and is a work that has many possibilities to be extended. It is meant to be thought-provoking in the features correlations to the target that it reveals. We have taken some precautions to communicate results effectively and forestall unintended usage of the prediction in this manner:

1. We present results in terms of Township-Ranges in a county that lie within the San Joaquin valley as point estimates but clearly indicate the error of predictions in each model and also derive prediction quantiles for the best performing model.
2. We highlight that more historical data needs to be collected for the Deep Learning module to learn from the data meaningfully.
3. The distribution of resources using Voronoi diagrams needs to incorporate geological factors as well.
4. We include the possibility of extending the search of features to include weather characteristics such as temperature, drought conditions and GDP of the area to be sensitive to the fairness in water distribution.
5. We are considering replacing the groundwater depth in feet below ground surface (GSE_GWE) as a target feature by another feature which could better fit our objective of identifying areas with water resource issues.
6. Ellul(1964) stated that technique and technical processes strive for the “mechanization of everything it encounters”. We have to remind ourselves at every turn in this analysis, that water forms the backbone of a human settlement and lives can be just as adversely affected by decisions made as a result of this analysis, as they can be improved. Our analysis is an attempt to employ learned as well as current techniques. As stated by Nielsen M.^[23], “Such improvements to the way discoveries are made are more important than any single discovery”.

The Team and Contribution

Team: Simi Talkar and Matthieu Lienart

- Both: Reports and visualizations, data cleaning and manipulation, pipelines
- Simi Talkar: Supervised learning, failure analysis, prediction uncertainty and explainability, Streamlit app.
- Matthieu Lienart : Voronoi diagram, Deep Learning techniques and unsupervised learning, sensitivity analysis.



Credits :

1. Sustainable Groundwater Management Act (SGMA). *California Department of Water Resources*.
<https://water.ca.gov/programs/groundwater-management/sgma-groundwater-management>.
Last Accessed 15th October 2022.
2. M. L. La Ganga, G. L. LeMee and I. James. "A frenzy of well drilling by California farmers leaves taps running dry". *Los Angeles Times*.
<https://www.latimes.com/projects/california-farms-water-wells-drought/>. 16th December 2021.
3. I. James. "Despite California groundwater law, aquifers keep droppin in a 'race to the bottom for agricultural wells'". *Los Angeles Times*.
<https://www.latimes.com/environment/story/2021-12-16/its-a-race-to-the-bottom-for-agricultural-wells>. 16th December 2021.
4. R Pauloo. "An Exploratory Data Analysis of California's Well Completion Reports".
https://richpauloo.github.io/oswcr_1.html. 30th April 2018.
5. A. Fulton, T. Dudley, and K. Staton. "Groundwater Level Monitoring: What is it? How is it done? Why do it?".
<https://www.countyofcolusa.org/DocumentCenter/View/4260/Series1Article4-GroundwaterLevelMonitoring>. Last Accessed 15th October 2022.
6. J. Miles, "Getting the Most out of scikit-learn Pipelines". Towards Data Science.
<https://towardsdatascience.com/getting-the-most-out-of-scikit-learn-pipelines-c2afc4410f1a>.
29th July 2021
7. "Typical water well construction and terms". Ground Water Information Center Online.
<https://mbmggwic.mtech.edu/sqlserver/v11/help/welldesign.asp>. Last Accessed 15th October 2022.
8. "Groundwater". United States Geological Survey.
<https://www.usgs.gov/special-topics/water-science-school/science/groundwater>. Last Accessed 15th October 2022.
9. CNN-LSTM-Based Models for Multiple Parallel Input and Multi-Step Forecast.
<https://towardsdatascience.com/cnn-lstm-based-models-for-multiple-parallel-input-and-multi-step-forecast-6fe2172f7668>. Last accessed 17th November 2021.
10. A. Nielsen. "Practical Time Series Analysis". O'Reilly Media. ISBN: 9781492041658. October 2019.
11. A. L. D'Agostino. "Bounding Boxes for All US States".
<https://pathindependence.wordpress.com/2018/11/23/bounding-boxes-for-all-us-states/>. 23rd November 2018. Last Accessed 15th October 2022.
12. S. M. Lundberg S. Lee. "A unified approach to Interpreting Model Predictions". *NIPS'17: Proceedings of the 31st International Conference on Neural Information Processing Systems*.



<https://proceedings.neurips.cc/paper/2017/file/8a20a8621978632d76c43dfd28b67767-Paper.pdf>. 25th May 2017.

13. J. Brownlee. "Prediction Intervals for Machine Learning". *Machine Learning Mastery*. <https://machinelearningmastery.com/prediction-intervals-for-machine-learning/>. 17th February 2021. Last Accessed 15th October 2022.
14. J. Brownlee. "Machine Learning Mastery How to improve Neural Network Stability". *Machine Learning Mastery*. <https://machinelearningmastery.com/how-to-improve-neural-network-stability-and-modeling-performance-with-data-scaling/>. 4th February 2019. Last Accessed 15th October 2022.
15. A. M. MacEachren, A. Robinson, S. Hopper, S. Gardner, R. Murray, M. Gahegan & E. Hetzler. "Visualizing Geospatial Information Uncertainty: What We Know and What We Need to Know". *Cartography and Geographic Information Science*, 32:3, 139-160, DOI: 10.1559/1523040054738936. 2005
16. S. Kandi. "Prediction Intervals in Forecasting: Quantile Loss Function". *Medium*. <https://medium.com/analytics-vidhya/prediction-intervals-in-forecasting-quantile-loss-function-18f72501586f>. 5th September 2019. Last Accessed 15th October 2022.
17. W. Q. Meeker, G. J. Hahn, L. A. Escobar "Statistical Intervals: A Guide for Practitioners and Researchers". *Wiley* ISBN: 978-0-471-68717-7. April 2017.
18. B. F. Froeschke, L. B. Jones, B. Garman, "Spatio-temporal Models of Juvenile and Adult Sheepshead (*Archosargus probatocephalus*) in Tampa Bay, Florida from 1996 to 2016". *Gulf and Caribbean Research*, 31(1): 8 – 17. [<https://doi.org/10.18785/gcr.3101.04>]. 2020.
19. "sklearn.preprocessing.StandardScaler". *Scikit Learn*. <https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html>. Last Accessed 15th October 2022.
20. "sklearn.model_selection.RandomizedSearchCV". *Scikit Learn*. https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.RandomizedSearchCV.html. Last Accessed 15th October 2022.
21. "Support Vector Machines". *Scikit Learn*. <https://scikit-learn.org/stable/modules/svm.html>. Last Accessed 15th October 2022.
22. "The Public Land Survey System (PLSS)". *Sidwell*. <https://www.sidwellco.com/company/resources/public-land-survey-system/>. Last Accessed 15th October 2022.
23. Nielsen, M. (2013). *Reinventing Discovery: The New Era of Networked Science* (Reprint edition). Princeton University Press.

Appendix

1. Definitions

1.1 Sustainable Groundwater Management

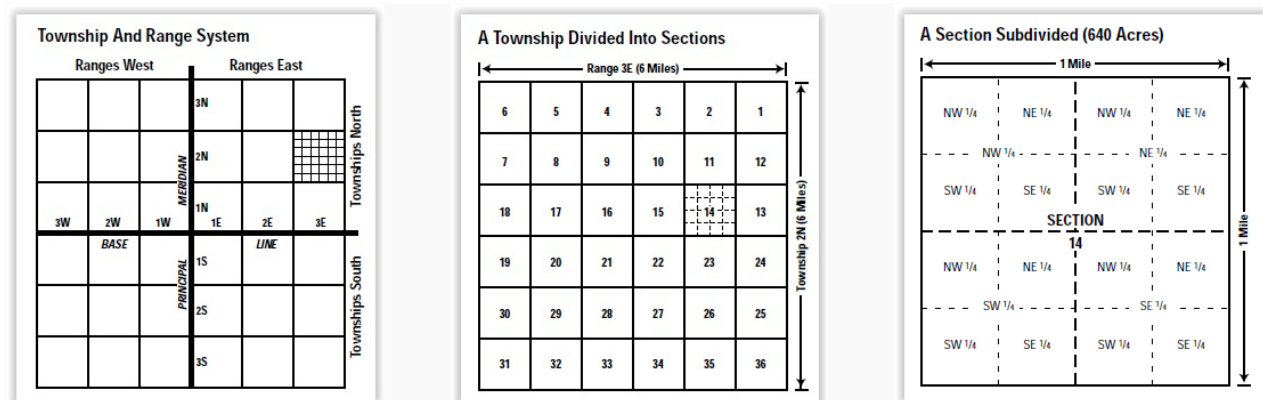
Sustainable groundwater management is defined as managing water supplies in a way that can be maintained without “causing undesirable results,” such as **chronic declines in groundwater levels** or “significant and unreasonable” depletion, adverse effects on surface water, **degraded water quality** or land subsidence.

1.2. PLSS Township-Range

According to Sidewell “The Public Land Survey System”^[22], the “Public Land Survey System (PLSS) is the surveying method developed and used in the United States to plat, or divide, land for sale and settling”^[22]. “Under this system the lands are divided into ‘townships,’ 6 miles square, which are related to base lines established by the federal government. The township numbers east or west of the Principal Meridians are designated as ranges; whereas, the numbers north and south of the base line are tiers”^[22].

“Thus, the description of a township as ‘Township 16 North, Range 7 West’ would mean that the township is situated 16 tiers north of the base line for the Principal Meridian and 7 ranges west of that meridian. Guide Meridians, at intervals of 24 miles east and (or) west of the Principal Meridian, are extended north and (or) south from the base line; Standard Parallels, at 24-mile intervals north and (or) south of the base line, are extended east and (or) west from the Principal Meridian.

The Township is 6 miles square. It is divided into 36 square-mile “sections” of 640 acres, each which may be divided and subdivided as desired. The diagram below shows the system of numbering the sections and the usual method of subdividing them”^[22].

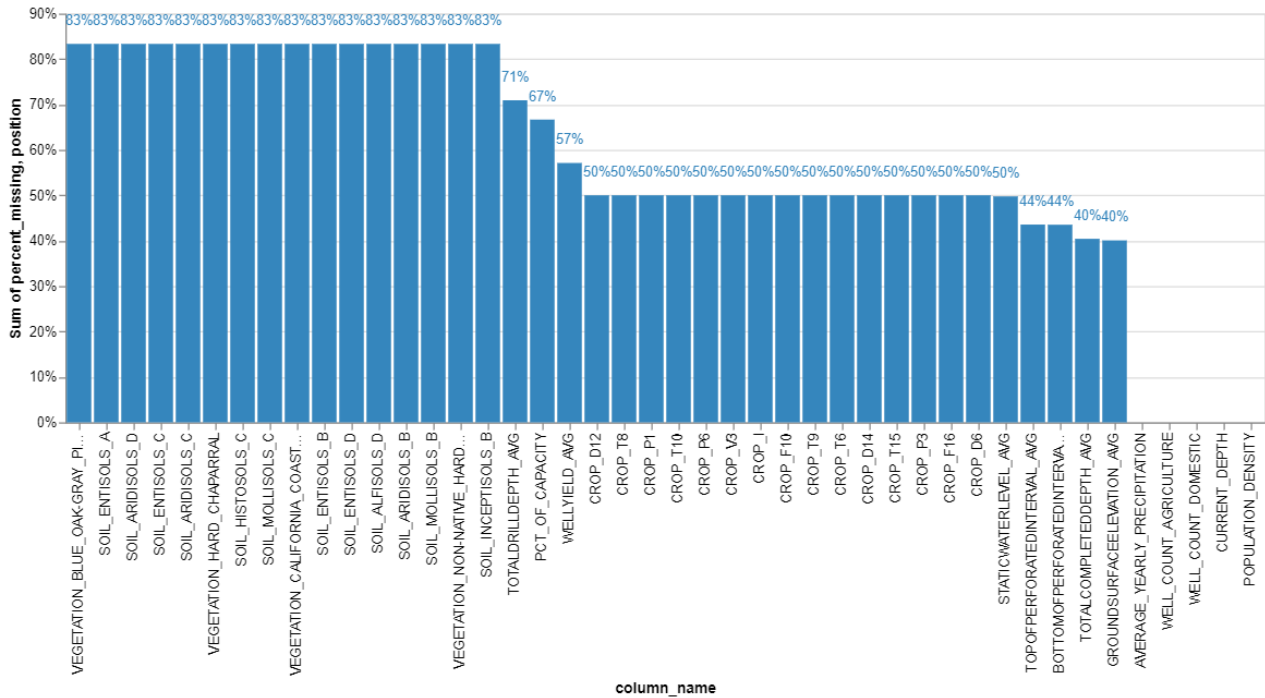


Figures from Sidewell “The Public Land Survey System”^[21]

2. Datasets

<p>well_completion_report</p> <p>TOWNSHIP_RANGE YEAR</p> <p>BOTTOMOFPERFORATEDINTERVAL_AVG GROUNDSURFACEELEVATION_AVG STATICWATERLEVEL_AVG TOPOFPERFORATEDINTERVAL_AVG TOTALDRILLDEPTH_AVG TOTALCOMPLETEDDEPTH_AVG WELLYIELD_AVG WELL_COUNT WELL_COUNT_AGRICULTURE WELL_COUNT_DOMESTIC WELL_COUNT_INDUSTRIAL WELL_COUNT_PUBLIC BOTTOMOFPERFORATEDINTERVAL GROUNDSURFACEELEVATION STATICWATERLEVEL TOPOFPERFORATEDINTERVAL TOTALDRILLDEPTH TOTALCOMPLETEDDEPTH WELLYIELD USE</p>	<p>Soils</p> <p>TOWNSHIP_RANGE YEAR</p> <p>SOIL_ALFISOLS_B SOIL_ALFISOLS_C SOIL_ALFISOLS_D SOIL_ARIDISOLS_B SOIL_ARIDISOLS_C SOIL_ARIDISOLS_D SOIL_ENTISOLS_A SOIL_ENTISOLS_B SOIL_ENTISOLS_C SOIL_ENTISOLS_S SOIL_HISTOSOLS_C SOIL_INCEPTISOLS_B SOIL_INCEPTISOLS_D SOIL_MOLLISOLS_B SOIL_MOLLISOLS_C SOIL_MOLLISOLS_D SOIL_ROCK_OUTCROP_D SOIL_VERTISOLS_D SOIL_WATER_</p>	<p>crops</p> <p>TOWNSHIP_RANGE YEAR</p> <p>CROP_C CROP_C6 CROP_D10 CROP_D12 CROP_D13 CROP_D14 CROP_D15 CROP_D16 CROP_F1 CROP_F10 CROP_F16 CROP_F2 CROP_G CROP_G2 CROP_G6 CROP_J CROP_P1 CROP_P3 CROP_P6 CROP_R CROP_R1 CROP_T10 CROP_T15 CROP_T18 CROP_T19 CROP_T21 CROP_T26 CROP_T30 CROP_T31 CROP_T6 CROP_T8 CROP_T9 CROP_V CROP_V3 CROP_YP</p>	<p>vegetation</p> <p>TOWNSHIP_RANGE YEAR</p> <p>VEGETATION_BLUE_OAK_GRAY_PINE VEGETATION_CALIFORNIA_COAST_LIVE_OAK VEGETATION_CANYON_LIVE_OAK VEGETATION_HARD_CHAPARRAL VEGETATION_KNOBCONE_PINE VEGETATION_NON-NATIVE_HARDWOOD_FOREST VEGETATION_PINYON-JUNIPER</p> <p>precipitation</p> <p>TOWNSHIP_RANGE YEAR</p> <p>AVERAGE_YEARLY_PRECIPITATION</p> <p>reservoir</p> <p>TOWNSHIP_RANGE YEAR</p> <p>PCT_OF_CAPACITY</p> <p>spring_groundwater_levels</p> <p>TOWNSHIP_RANGE YEAR</p> <p>GSE_GWE</p>
---	--	---	--

3. Missing Data



4. Target Shifting

The method used to capture part of the intrinsic time series nature of the data, the target variable is preserved as an input and the target becomes the target variable itself shifted by one time stamp. Instead of predicting $Y(t)$ based on on the features $X_1(t) - X_4(t)$,



The output $Y(t)$ is then predicted based on the previous value of the features ($X_1(t-1) - X_4(t-1)$) but also based on its own previous value $Y(t-1)$.



This results in the following dataset transformation.



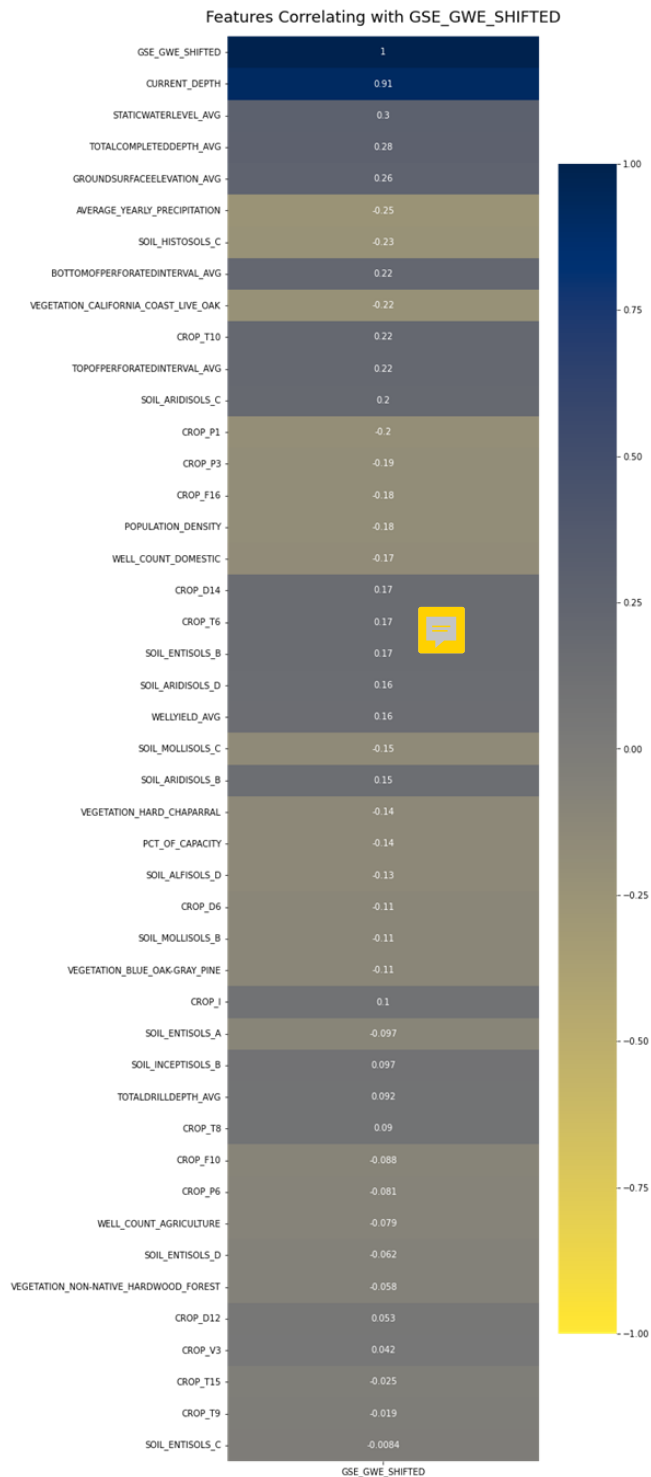
Original Dataset							Dataset with Shifted Target as Input									
		Inputs (X)				Outputs (Y)			Inputs (X)					Outputs (Y)		
		Time	X ₁	X ₂	X ₃	X ₄	Y			Time	X ₁	X ₂	X ₃	X ₄	Y	Shifted
Township Range 1	t	X ₁ (t)	X ₂ (t)	X ₃ (t)	X ₄ (t)	Y(t)	Township Range 1	t	X ₁ (t)	X ₂ (t)	X ₃ (t)	X ₄ (t)	Y(t)	Y(t+1)		
	t+1	X ₁ (t+1)	X ₂ (t+1)	X ₃ (t+1)	X ₄ (t+1)	Y(t+1)		t+1	X ₁ (t+1)	X ₂ (t+1)	X ₃ (t+1)	X ₄ (t+1)	Y(t+1)	Y(t+2)		
	t+2	X ₁ (t+2)	X ₂ (t+2)	X ₃ (t+2)	X ₄ (t+2)	Y(t+2)		t+2	X ₁ (t+2)	X ₂ (t+2)	X ₃ (t+2)	X ₄ (t+2)		
	Y(t+n-2)	Y(t+n-1)		
	t+n-1	X ₁ (t+n-1)	X ₂ (t+n-1)	X ₃ (t+n-1)	X ₄ (t+n-1)	Y(t+n-1)		t+n-1	X ₁ (t+n-1)	X ₂ (t+n-1)	X ₃ (t+n-1)	X ₄ (t+n-1)	Y(t+n-1)	Y(t+n)		
	t+n	X ₁ (t+n)	X ₂ (t+n)	X ₃ (t+n)	X ₄ (t+n)	Y(t+n)										
Township Range 2	t	X ₁ (t)	X ₂ (t)	X ₃ (t)	X ₄ (t)	Y(t)	Township Range 2	t	X ₁ (t)	X ₂ (t)	X ₃ (t)	X ₄ (t)	Y(t)	Y(t+1)		
	t+1	X ₁ (t+1)	X ₂ (t+1)	X ₃ (t+1)	X ₄ (t+1)	Y(t+1)		t+1	X ₁ (t+1)	X ₂ (t+1)	X ₃ (t+1)	X ₄ (t+1)	Y(t+1)	Y(t+2)		
	t+2	X ₁ (t+2)	X ₂ (t+2)	X ₃ (t+2)	X ₄ (t+2)	Y(t+2)		t+2	X ₁ (t+2)	X ₂ (t+2)	X ₃ (t+2)	X ₄ (t+2)		
	Y(t+n-2)	Y(t+n-1)		
	t+n-1	X ₁ (t+n-1)	X ₂ (t+n-1)	X ₃ (t+n-1)	X ₄ (t+n-1)	Y(t+n-1)		t+n-1	X ₁ (t+n-1)	X ₂ (t+n-1)	X ₃ (t+n-1)	X ₄ (t+n-1)	Y(t+n-1)	Y(t+n)		
	t+n	X ₁ (t+n)	X ₂ (t+n)	X ₃ (t+n)	X ₄ (t+n)	Y(t+n)										

The current year's features of well details, precipitation, reservoir level, vegetation, crops and groundwater depth GSE_GWE are used to predict next year's groundwater depth GSE_GWE_SHIFTED.

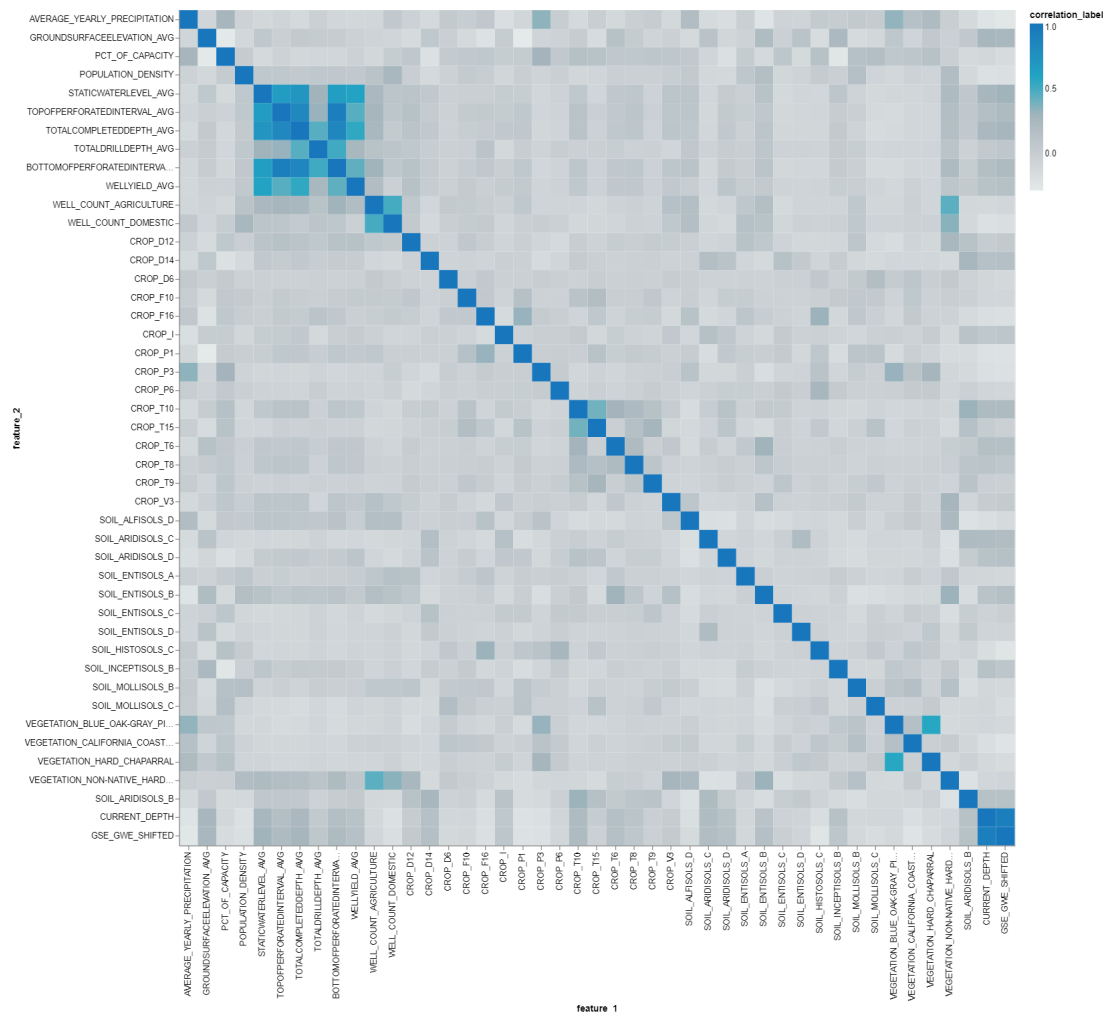


5. Feature Correlation

The below chart has been sorted by absolute value of correlation of features with the target.



6. Feature Correlation



7. Deep Learning

7.1 Train-Test Split

Below is a conceptual visualization of the Train-Test split for the deep learning approach using LSTM neural network, where $t=2014$ and $t+n=2021$ and Y is the groundwater depth (GSE_GWE) feature of the dataset.

The split of training and test sets inputs and targets was performed as follow:

- Training and Test sets were split by Township-Ranges. I.e., some Township-Ranges had all their 2014-2021 data points in the training set, some others were in the test set.

- The inputs and targets were split by time. The model was trained based on all the 2014-2020 data with the target being the 2021 value of the target feature.

Training Set

Test Set

		Time	X_1	X_2	X_3	X_4	Y
Inputs (X)	Township Range 1	t	$X_1(t)$	$X_2(t)$	$X_3(t)$	$X_4(t)$	$Y(t)$
		t+1	$X_1(t+1)$	$X_2(t+1)$	$X_3(t+1)$	$X_4(t+1)$	$Y(t+1)$
		t+2	$X_1(t+2)$	$X_2(t+2)$	$X_3(t+2)$	$X_4(t+2)$	$Y(t+2)$
	
		t+n-2	$X_1(t+n-2)$	$X_2(t+n-2)$	$X_3(t+n-2)$	$X_4(t+n-2)$	$Y(t+n-2)$
		t+n-1	$X_1(t+n-1)$	$X_2(t+n-1)$	$X_3(t+n-1)$	$X_4(t+n-1)$	$Y(t+n-1)$
	Township Range 2	t	$X_1(t)$	$X_2(t)$	$X_3(t)$	$X_4(t)$	$Y(t)$
		t+1	$X_1(t+1)$	$X_2(t+1)$	$X_3(t+1)$	$X_4(t+1)$	$Y(t+1)$
		t+2	$X_1(t+2)$	$X_2(t+2)$	$X_3(t+2)$	$X_4(t+2)$	$Y(t+2)$
	
		t+n-2	$X_1(t+n-2)$	$X_2(t+n-2)$	$X_3(t+n-2)$	$X_4(t+n-2)$	$Y(t+n-2)$
		t+n-1	$X_1(t+n-1)$	$X_2(t+n-1)$	$X_3(t+n-1)$	$X_4(t+n-1)$	$Y(t+n-1)$
	Township Range 3	t	$X_1(t)$	$X_2(t)$	$X_3(t)$	$X_4(t)$	$Y(t)$
		t+1	$X_1(t+1)$	$X_2(t+1)$	$X_3(t+1)$	$X_4(t+1)$	$Y(t+1)$
		t+2	$X_1(t+2)$	$X_2(t+2)$	$X_3(t+2)$	$X_4(t+2)$	$Y(t+2)$
	
		t+n-2	$X_1(t+n-2)$	$X_2(t+n-2)$	$X_3(t+n-2)$	$X_4(t+n-2)$	$Y(t+n-2)$
		t+n-1	$X_1(t+n-1)$	$X_2(t+n-1)$	$X_3(t+n-1)$	$X_4(t+n-1)$	$Y(t+n-1)$

		Time	X_1	X_2	X_3	X_4	Y
Inputs (X)	Township Range 4	t+1	$X_1(t)$	$X_2(t+1)$	$X_3(t+1)$	$X_4(t+1)$	$Y(t+1)$
		t+2	$X_1(t+1)$	$X_2(t+1)$	$X_3(t+1)$	$X_4(t+1)$	$Y(t+1)$
		t+3	$X_1(t+2)$	$X_2(t+2)$	$X_3(t+2)$	$X_4(t+2)$	$Y(t+2)$
	
		t+n-2	$X_1(t+n-2)$	$X_2(t+n-2)$	$X_3(t+n-2)$	$X_4(t+n-2)$	$Y(t+n-2)$
		t+n-1	$X_1(t+n-1)$	$X_2(t+n-1)$	$X_3(t+n-1)$	$X_4(t+n-1)$	$Y(t+n-1)$

Outputs (Y)	Township Range 1	t+n	$Y(t+n)$
	Township Range 2	t+n	$Y(t+n)$
	Township Range 3	t+n	$Y(t+n)$

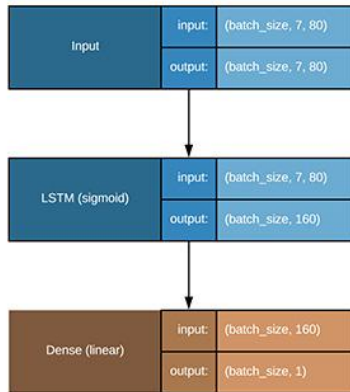
Outputs (Y)	Township Range 4	t+n	$Y(t+n)$
----------------	---------------------	-----	----------

7.2 LSTM Model Architectures

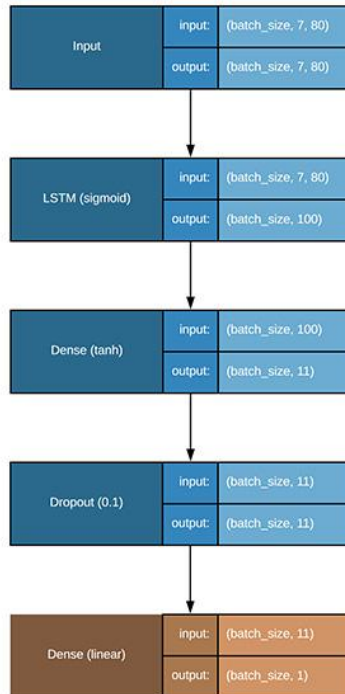
We tried 3 different LSTM models:

1. A simple model made of a single LSTM layer and an output Dense layer
2. A model made of a LSTM layer followed by a Dense and Dropout layers before the output layer
3. An Encoder-Decoder model made of 2 LSTM layers followed by a Dense and Dropout layers

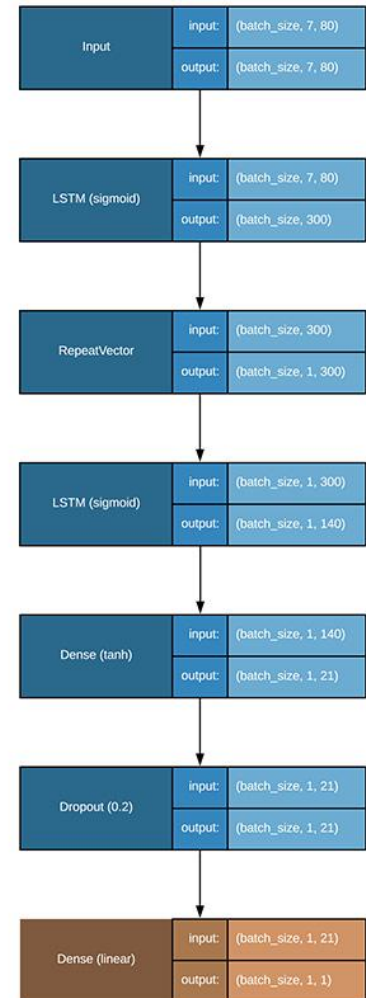
Model 1 : Simple LSTM Model



Model 2 : LSTM with Dense Layer



Model 3 : Encoder-Decoder LSTM

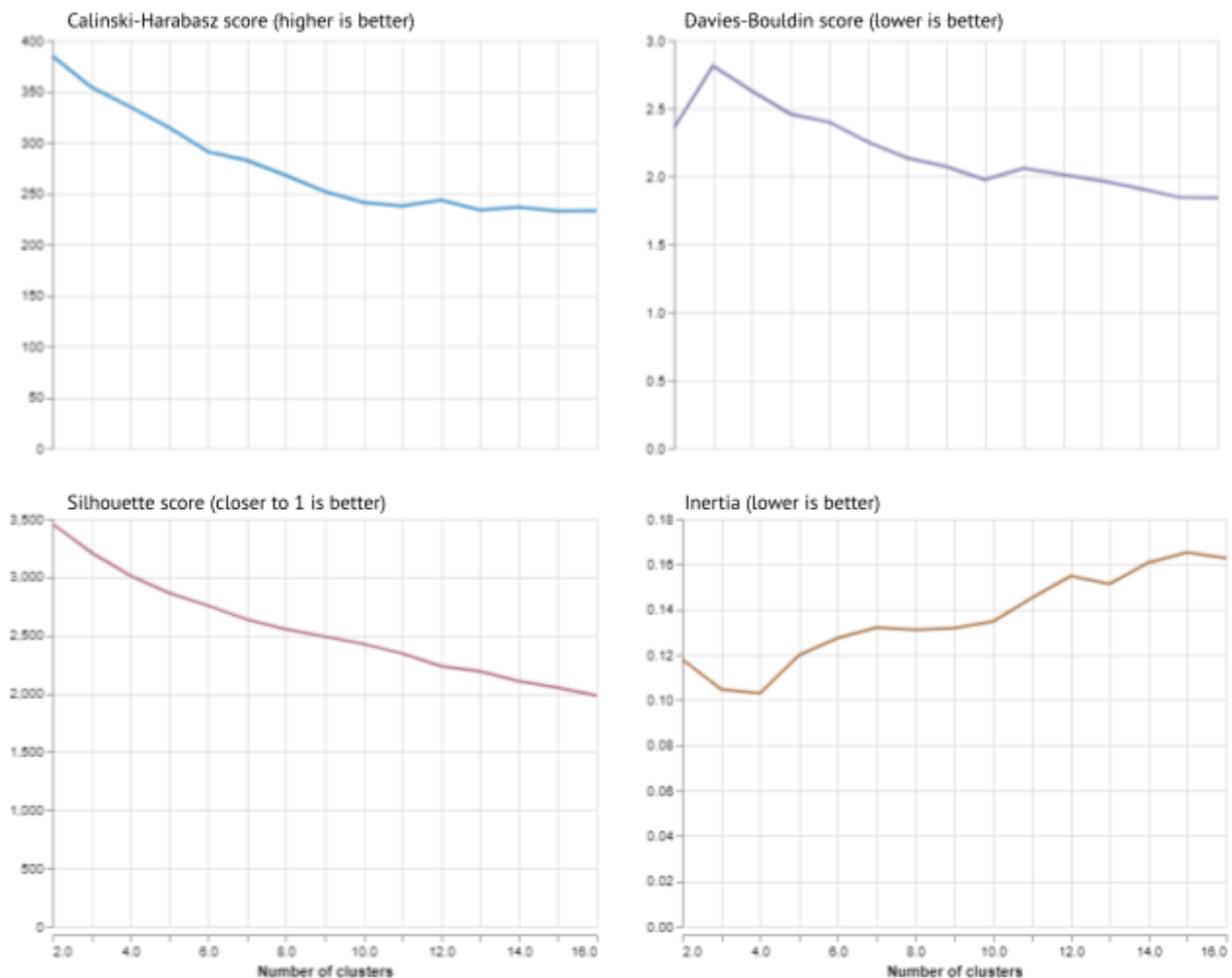


Encoder-decoder architectures are more common for sequence-to-sequence learning e.g., when forecasting the next 3 days (output sequence of length 3) based on the past year data (input sequence of length 365). In our case we only predict data for 1 time step in the feature. The output sequence being of length 1 this architecture might seem superfluous but has been tested anyway. This architecture was inspired by the Encoder-Decoder architecture in this article: CNN-LSTM-Based Models for Multiple Parallel Input and Multi-Step Forecast^[9].

8. Unsupervised Learning

8.1 Clustering - Estimating the Best “k”

K-Means Calinski-Harabasz, Davies-Bouldin, Silhouette scores and Sum of Squared Distances for different K values



Based on the above plots of the various clustering metric scores we get

- the Calinski-Harabasz score (indicating better cluster compactness) is best when **k=2**
- the Davies-Bouldin score (indicating better separation between clusters) is best when **k=14** but also good when **k=2**
- although the Silhouette scores increases with **k**, it remains similar (between 0.1 and 0.18), very far from the optimum value of 1
- the Inertia doesn't show any "elbow" in the plot to help deciding the best value of **k**

We thus decided to use **k=2** as our best value for the K-Means algorithm.

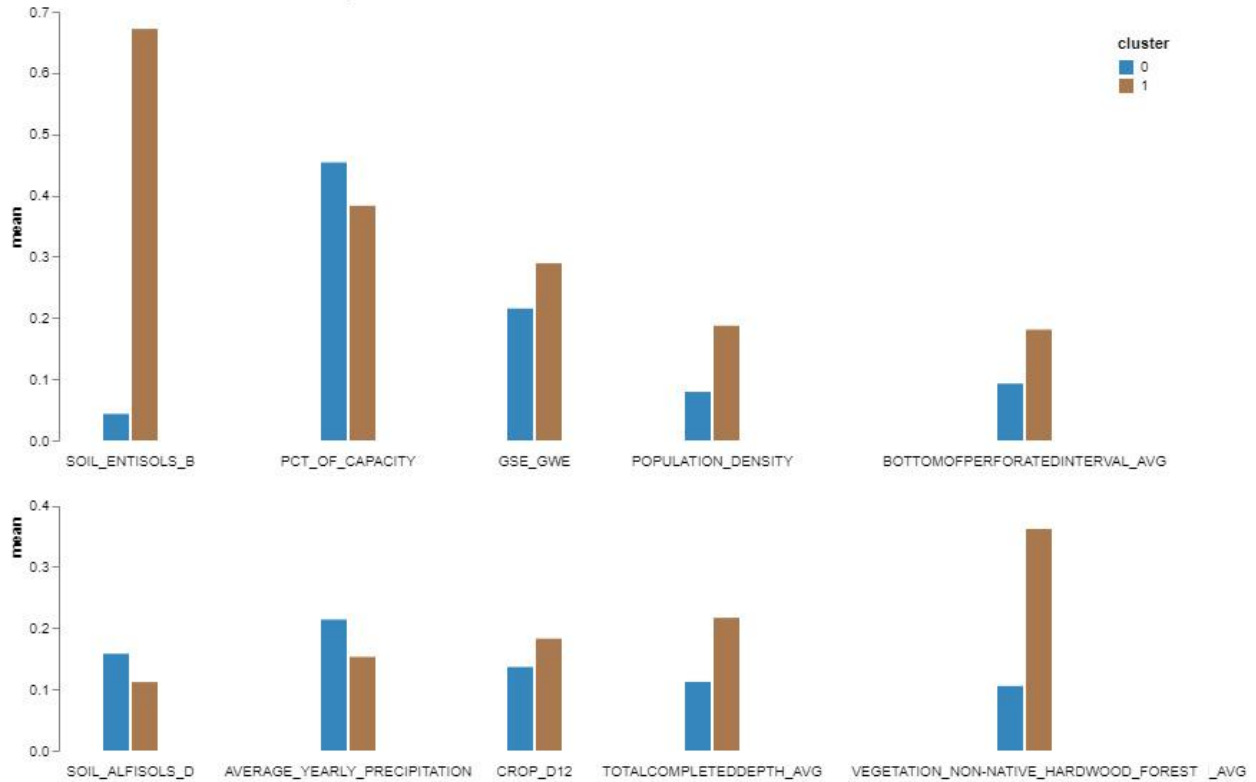
8.2 K-Means Clustering Year-to-Year Results

The figure below shows the results of K-Means clustering for 2 clusters where each Township-Range yearly data point has been clustered independently. There are just about 10 Township-Ranges showing some variations in their clustering over the 8 years period.



8.3 Most Important Features

Difference in mean values of the most predominant features for each cluster

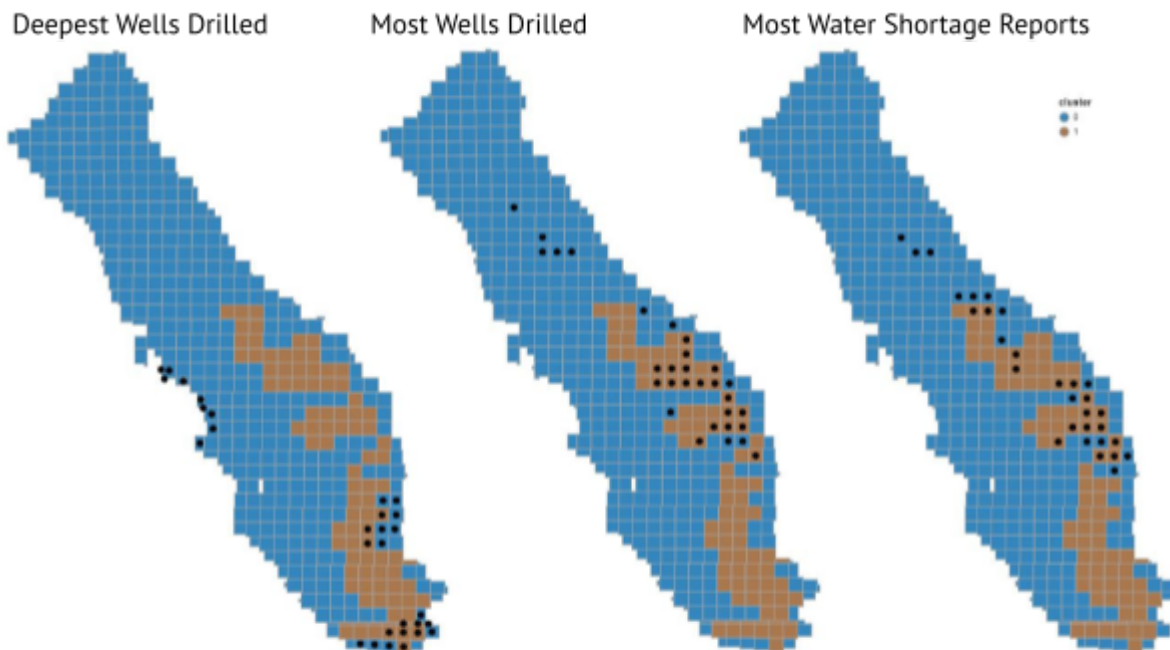


8.4 Comparing Clustered Township-Ranges

In the below visualization we compare the Township-Range clusters with the top 30 Township-Ranges with:

1. the average deepest wells (biggest GSE_GWE value) drilled,
2. the highest number of wells drilled,
3. the highest number of reported well shortages,

during the 2014-2021 period, we see that our classification fails to identify such Township-Ranges. It falls short of identifying 2/3 of the top 30 Township-Ranges with the highest number of wells drilled and the highest number of reported well shortages, and barely includes any of the top 30 Township-Ranges where the deepest wells are drilled.



Township-Ranges Clusters vs. Township-Ranges with deepest wells drilled, most wells drilled, most water shortage reports (based on 2014-2021 averages)

9. Sensitivity Analysis

9.1 Training Data Learning Curve

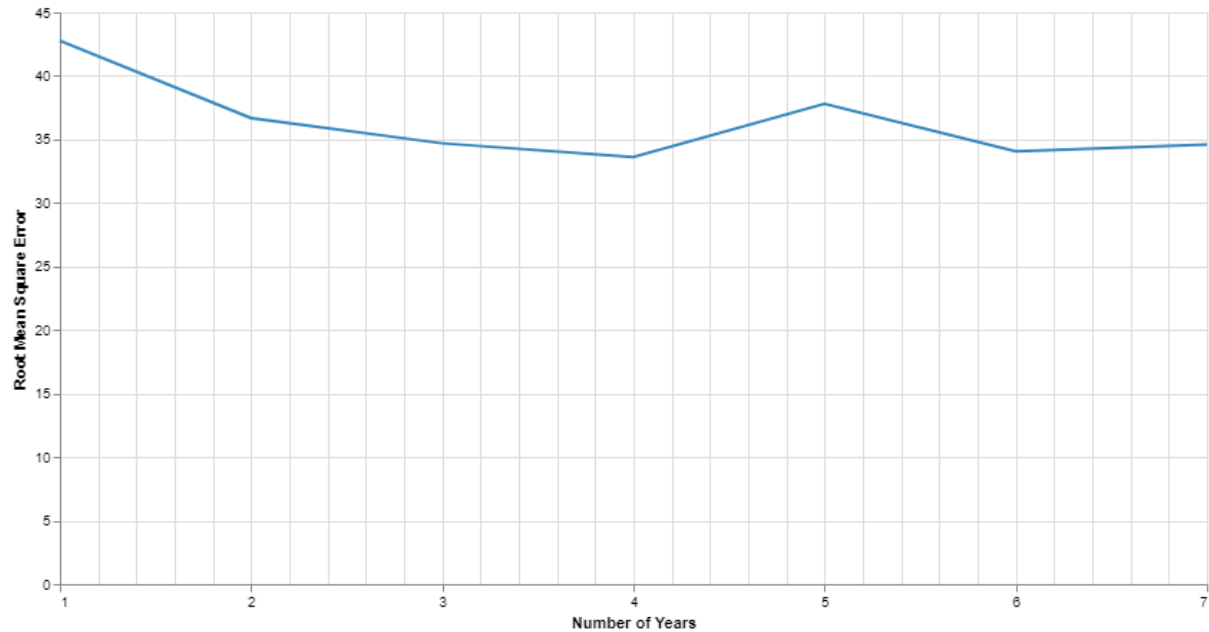
Taking the best deep learning model, we analyzed how much the amount of historical data impacts the LSTM performance. To do so we recursively trained the LSTM model based on more and more historical data. E.g. the model was first trained only based on 2020 data, then 2019-2020 data, etc. Although at the beginning the RMSE reduces as we add yearly data to train the model, the improvement in prediction is minor. The RMSE only reduces from ~42.5 feet to ~34 feet by increasing the number of historical data from 1 year to 4 years.





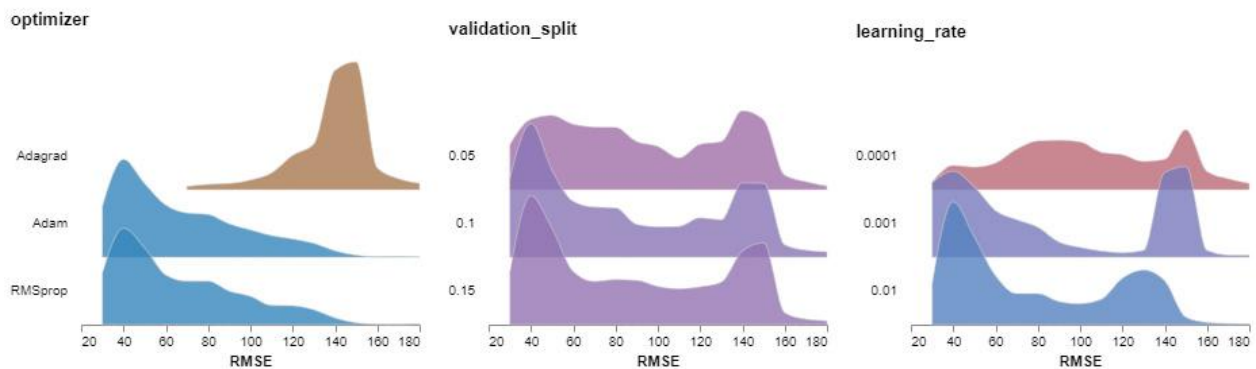
LSTM Model RMSE

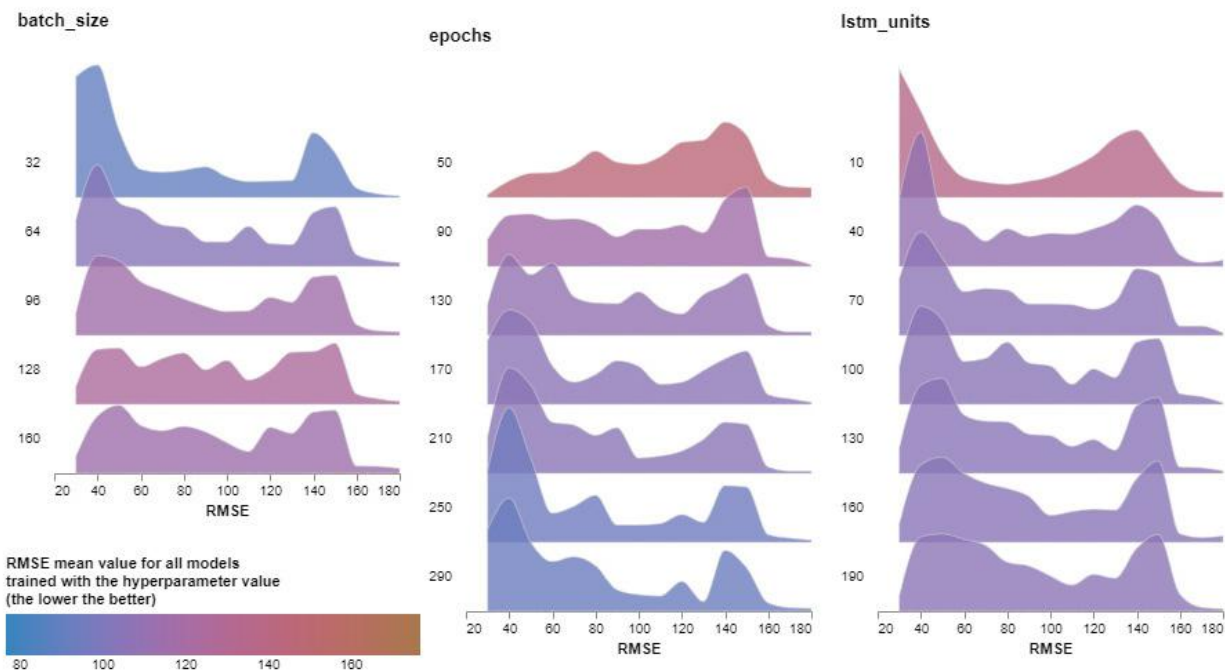
Based on the number of years the model was trained on.



9.2 Hyperparameters Sensitivity

We are displaying here for each hyperparameter value, the concentration of models per RMSE score and the average RMSE mean (using the color), for all 33,345 trained LSTM models depending on the hyperparameter values (The lower and the more blue the peak is, the better the hyperparameter value is).





Looking at this visualization, we can see - with some surprise - that the hyperparameters which seem to have the biggest impact on the model performance have little to do with the model architecture itself (the number of LSTM units) but with how the model is trained.

- The choice of the optimizer seems to have the largest impact on the model performance, with both the mean and distribution of the RMSE for all models trained with an Adagrad optimizer being really bad.
- The training-validation percentage split seems to have little impact. The best performance is obtained with assigning 10% of the training data to the validation set, but with 15% of the training data to the validation set the results are close.
- The bigger the learning rate, the better the model performs in terms of RMSE. The distribution of all models RMSE shows that with a learning_rate of 0.01, most models have low RMSE around 40. With a learning_rate of 0.001 we have a bimodal distribution of the RMSE with models performing either around 40 or very poorly around 150. With a learning rate of 0.0001, the distribution is more even with most models having an RMSE above 60.
- On the other hand, the smaller the batch size, the more models have a low RMSE.
- Although there is less of a difference if we compare similar values (e.g. 50 and 70 epochs or 270 and 290 epochs), we still see clearly that the bigger the number of training epochs the more there are trained models with a low RMSE.
- The number of LSTM units, impacting the number of neurons in the LSTM model, seems to have less impact on the performance of the RMSE. The distribution of all models RMSE does show differences between 10 and 190 LSTM units but not as much as other hyperparameters.

