EDA (Exploratory data analysis):

Solar Dataset

Insights:

This dataset has 456 columns and 6909 rows. The first 99 columns indicate the real values of solar production of each of the stations per day. The remaining variables are PCAs (Principal Components Analysis) over the original dataset, given by weather predictors. From row 5113 (01/01/2008), there are NA or missing values, so we focus on predicting the ACME, GOOD, and WYNO stations after 2008.

Data Preprocessing/Cleaning:

- Import the solar_dataset CSV file and infer data types from the schema tab. The station columns are “bigint” type, and the PCA’s are “double” type
- Parse the date column to a standard format
- We joined the prepared dataset of solar information with the prepared dataset of additional variables (which also has the date column parsed)
- We joined these two datasets on the date columns using a left join
- From solar dataset, we selected the stations of interest- ACME, GOOD, WYNO and the PCA components
- For the additional variables dataset, we included all of the descriptor variables
- We then split this joined dataset into a predictor set which contains the dates from 2008 onwards and a train_val_test, which contains the rows from the start (1994) to the end of 2007 (reference figure 1)
- Perform “analyse” on the train_val_test set

Figure 1

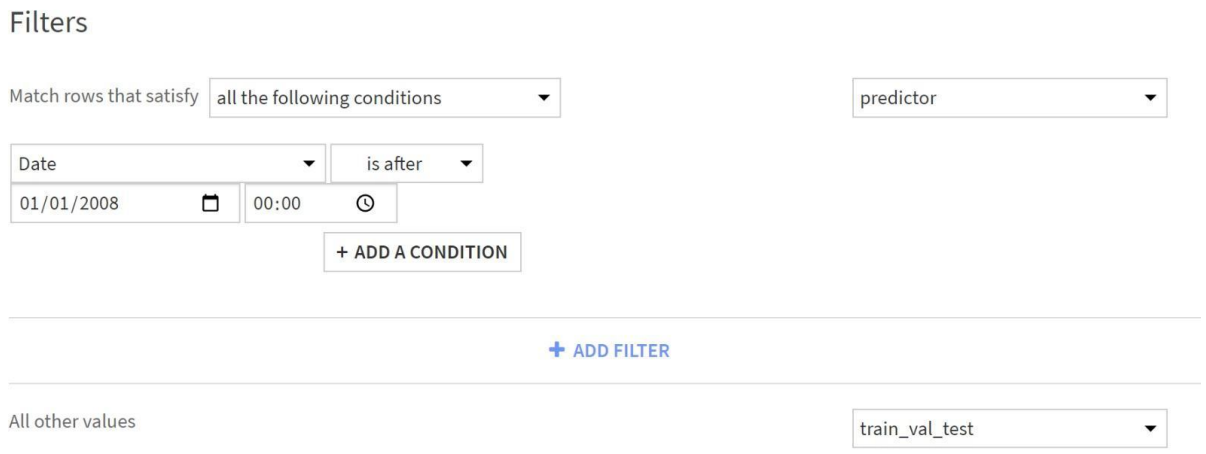


Figure 2

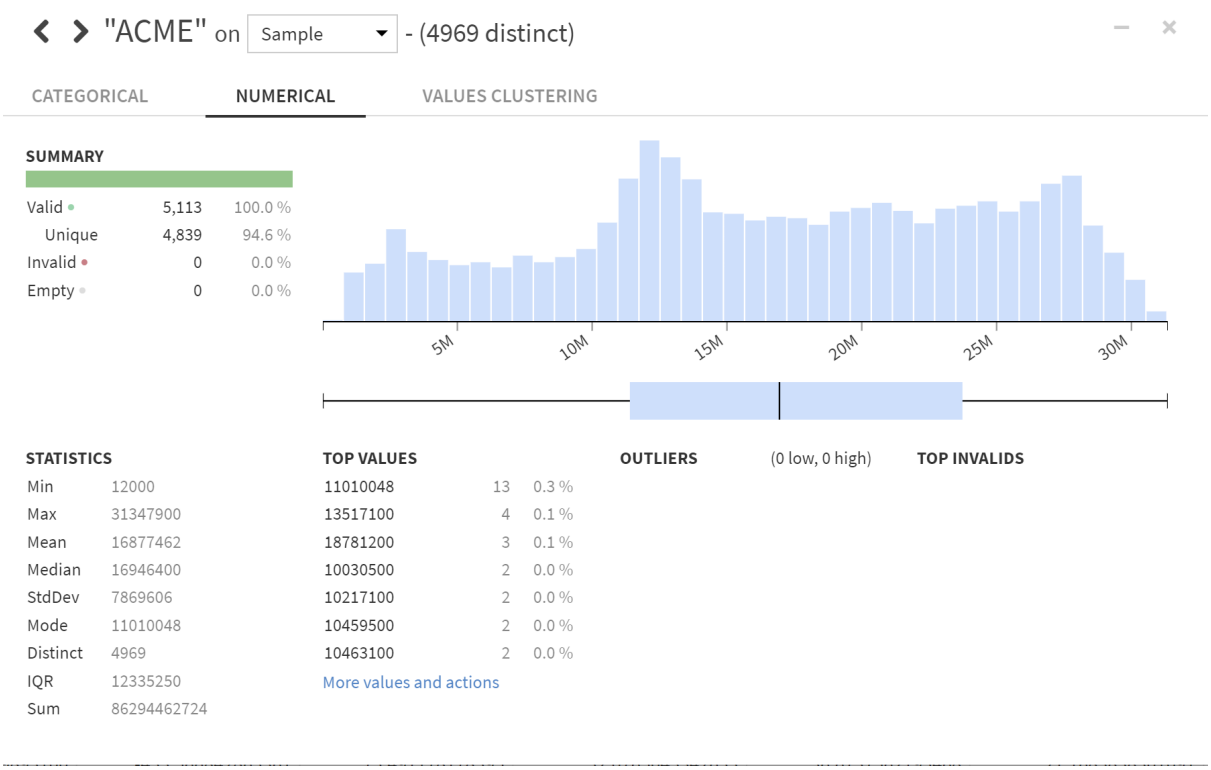


Figure 3

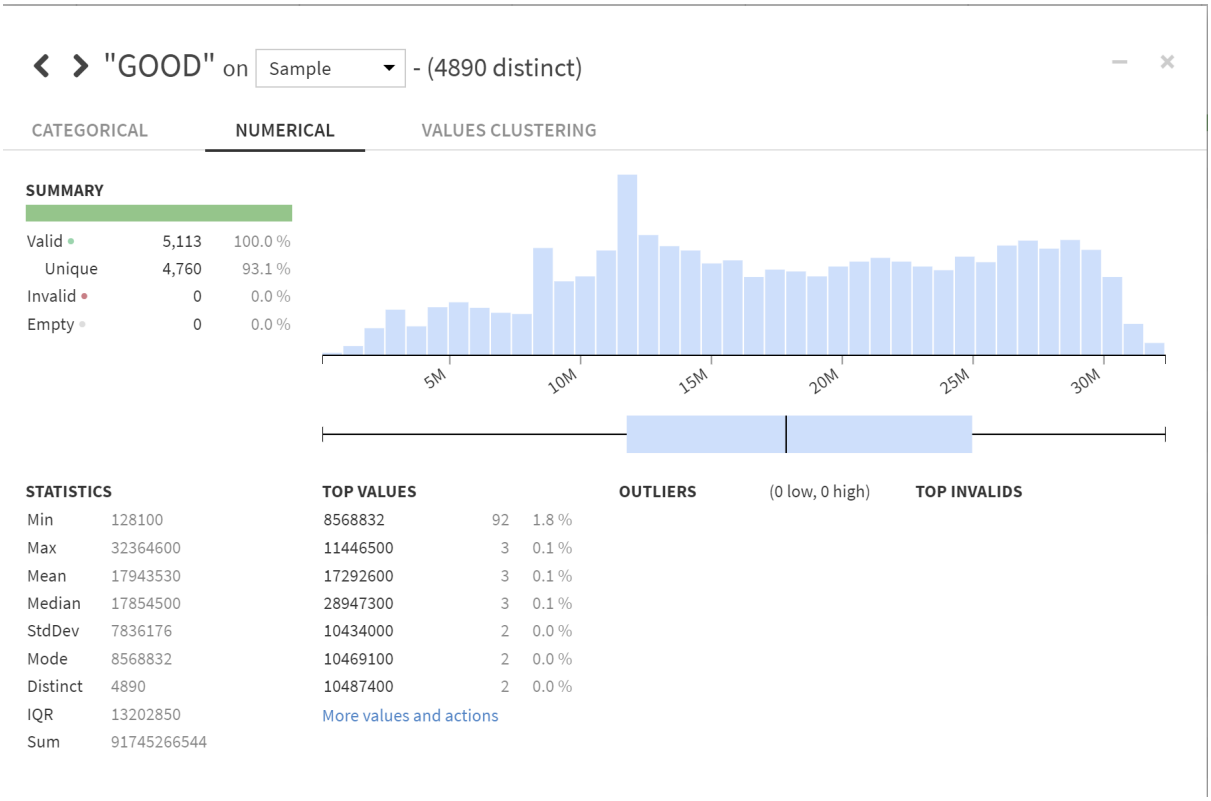
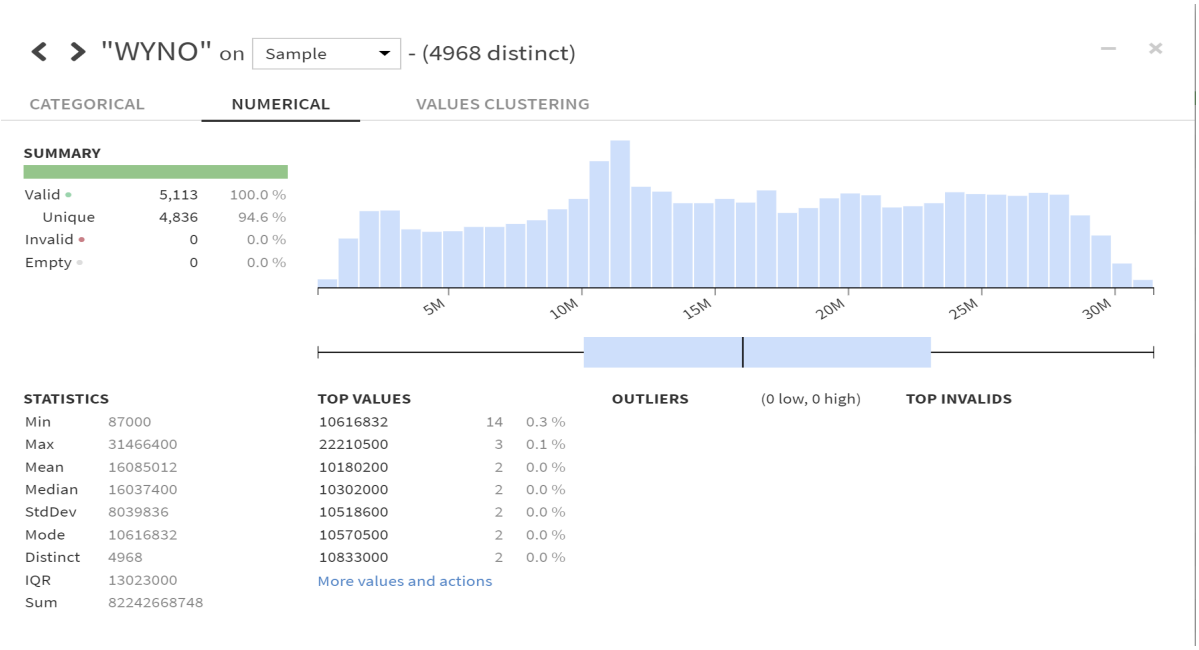
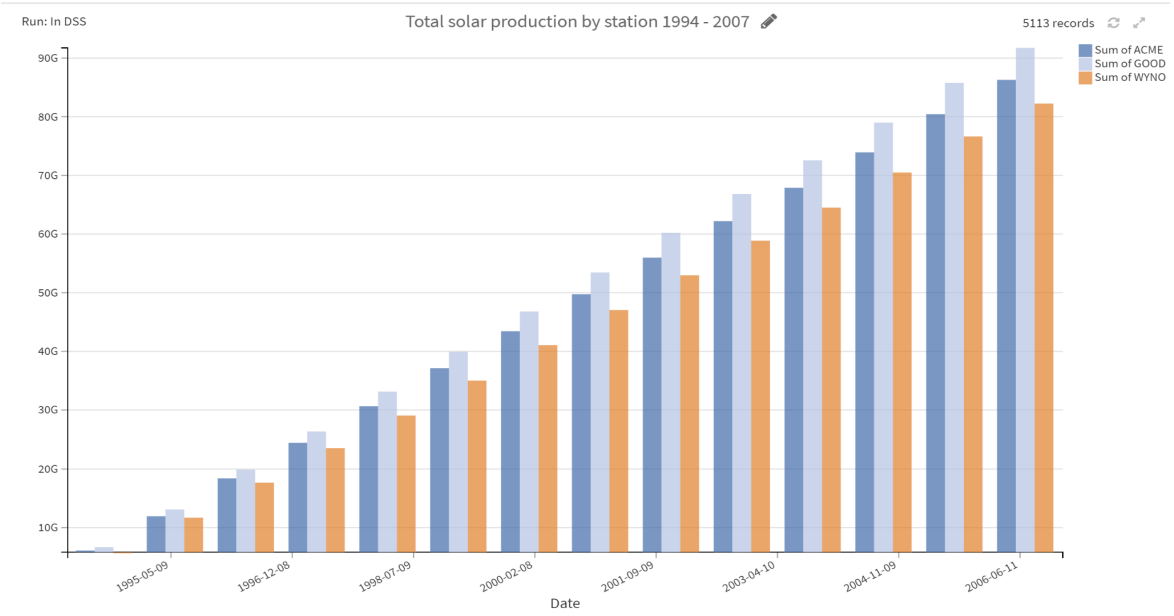


Figure 4



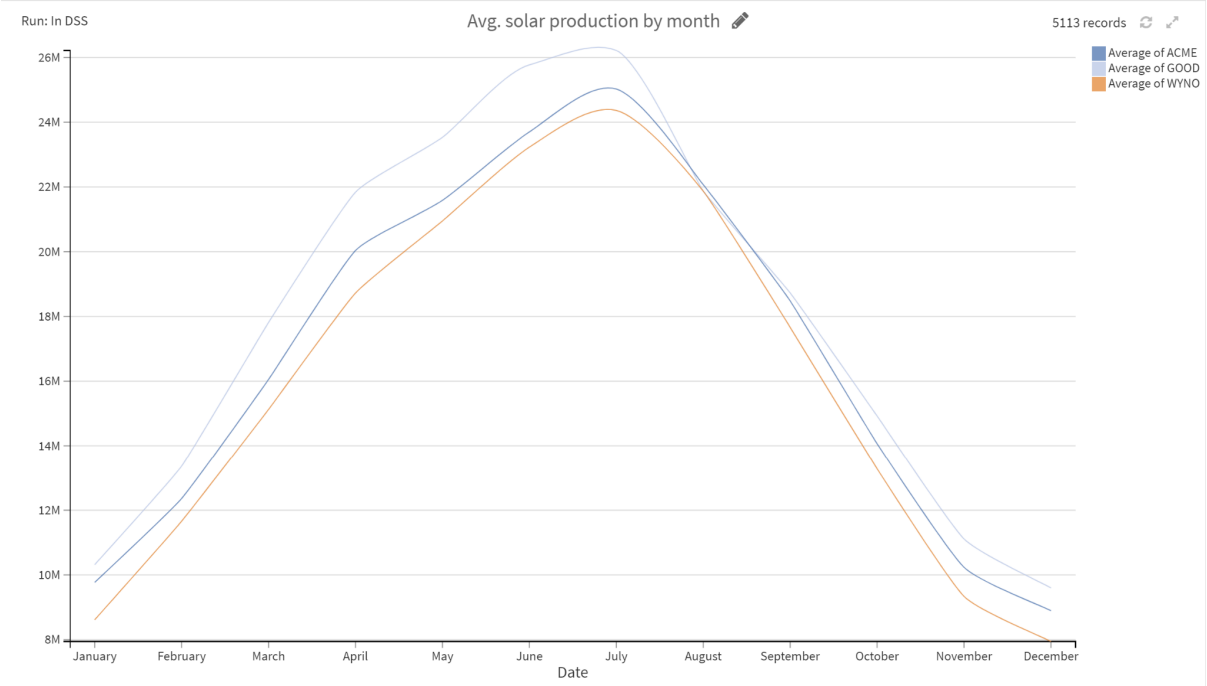
- We can see with this analysis that we have no empty values up until 2008, of course
- We do not detect any outliers and use the boxplot to represent this visually

Figure 5



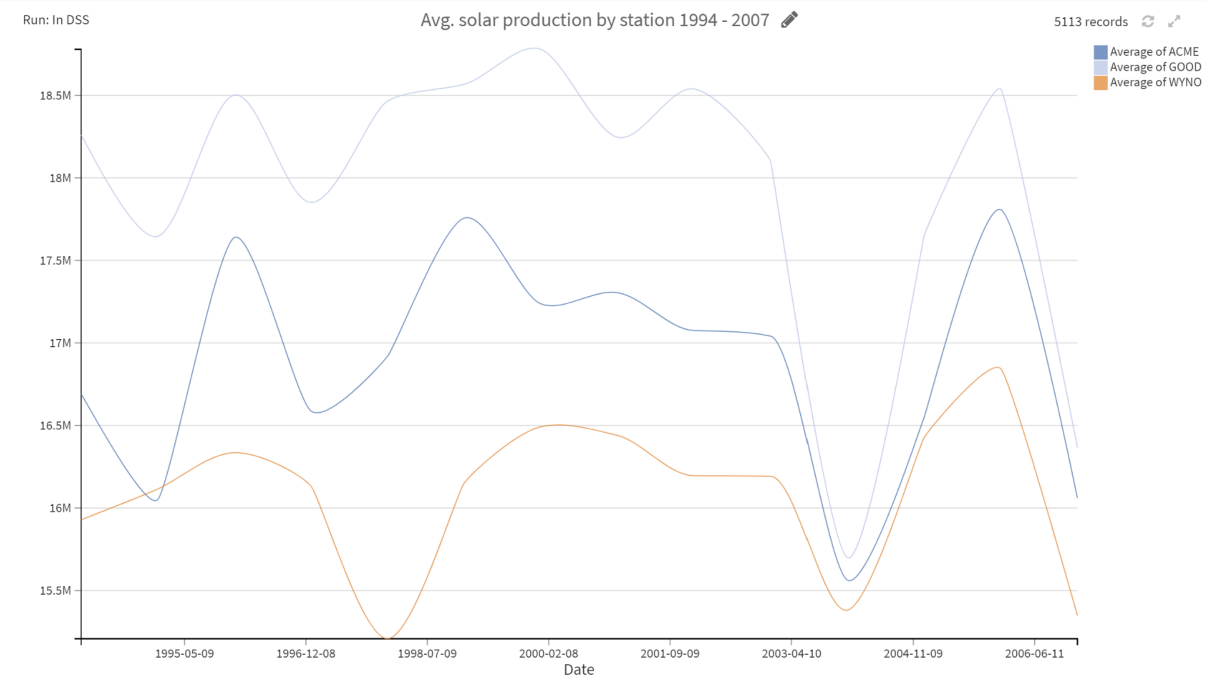
- We can use this cumulative sum of the total production to get a better understanding of the top solar power producer
- GOOD: 91.75 GJ/m2
- ACME: 86.29 GJ/m2
- WYNO: 82.24 GJ/m2

Figure 6



- Here we see the average solar production by month
- We see a seasonal relationship between the production of solar energy and the different seasons- seasons with less sun are not able to produce as much solar energy

Figure 7



- This is a visualisation of the average production rate over all of the years we are studying. We notice a large decrease at the beginning of 2004 potentially due to “chaotic” weather such as a snow storm

Additional variables

Insights:

This dataset consists of variables that correspond to real Numerical Weather Prediction, NWP, values. In particular, these are the top 100 most important variables to predict ACME stations. As in solar_dataset.csv, each row corresponds to a particular day. As there are 100 variables, we have randomly selected 8 variables from them (reference figures 8 and 9).

Data Preprocessing/Cleaning:

- Import the additional variables dataset and infer data types using the schema tab. There are missing values which cause some columns to become a string type after inferring. We change these columns to “double” type.
- We do not fill missing values here because we are not analysing the columns at this point
- The missing values are filled when we create our models
- Parse the date column to standard format
- We joined the prepared dataset of solar information with the prepared dataset of additional variables (which also has the date column parsed)
- We joined these two datasets on the date columns using a left join. For the solar dataset, we selected the three stations of interests- ACME, GOOD, WYNO and the PCA components. For the additional variables dataset, we included all of the descriptor variables

Figure 8

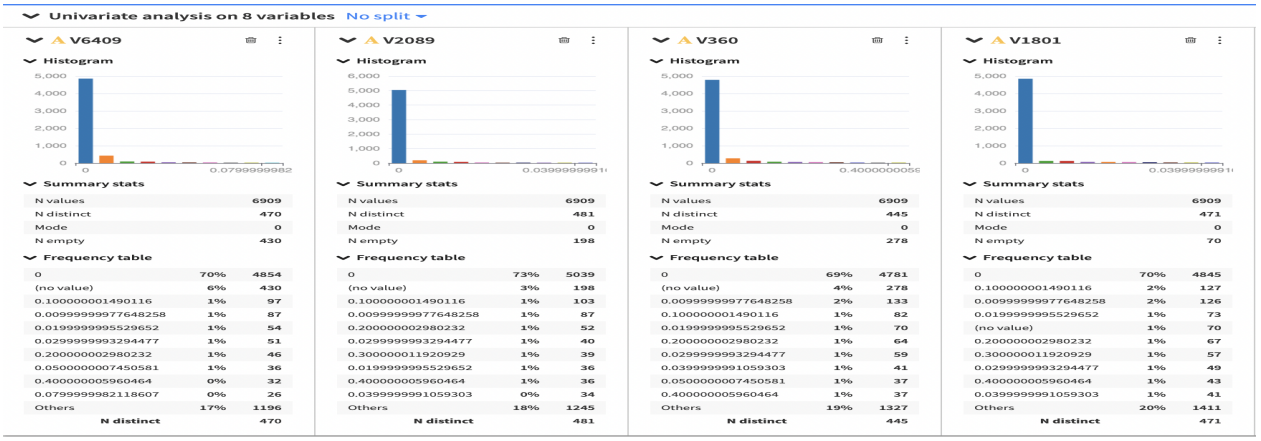
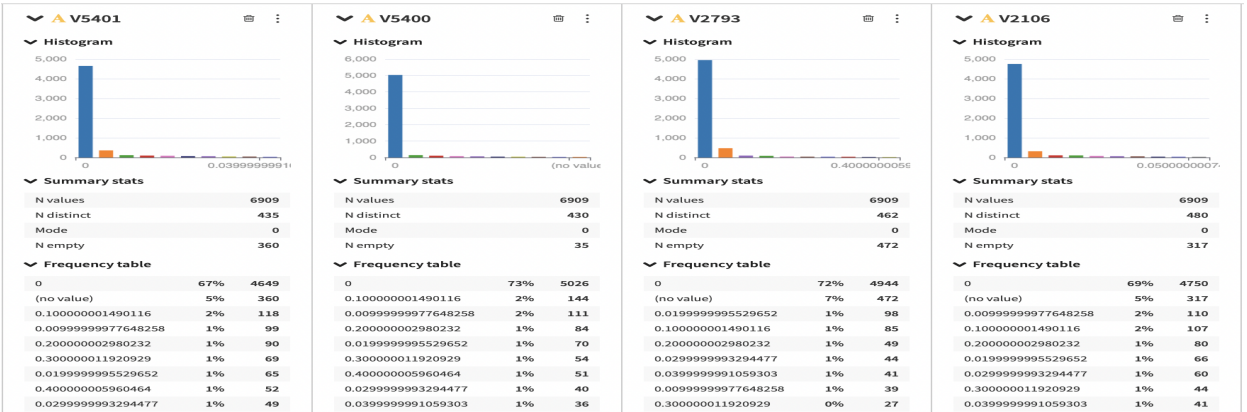


Figure 9



Station Info

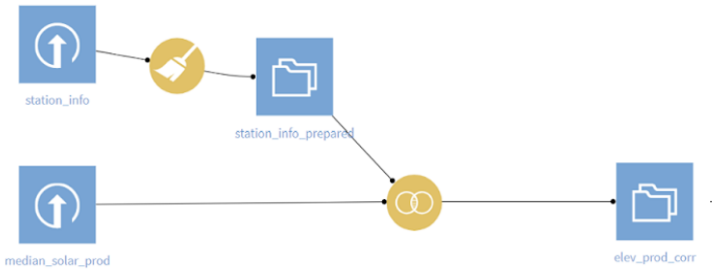
Insights:

File with name, latitude, longitude, and elevation of each of the 98 stations.95% of the values in this dataset are between 0 and 0.4. Distinct Values are NA, which is why the column data type is categorical instead of numerical. In all cases, the mode is 0. Around 70% of the values are 0 value.

Data Preprocessing/Cleaning:

- We import the additional variables CSV and infer data types using the schema tab.
- Create GeoPoint from columns “nlat” and “elon”, output column: “coordinates”. Results are shown on a scatter map for Oklahoma.
- We took the solar dataset and took the median production for each station ID and then created another table with station ID and corresponding median. Using this we can join the station info dataset and the new dataset on the station_ID and get the median production of each station and its elevation, latitude and longitude
- Create via MS Excel a CSV dataset with median solar production per station (1994-2007); file is named: median_solar_prod
- Left join Station_info_prepared with Median_solar_prod on Station_id

Figure 10



- We analyse the correlation between median solar production and geographical attributes, i.e. latitude/longitude/elevation. Results are shown in three scatter plots and the correlation matrix. These are included below.

FIgure 11

- As follows from the correlation matrix, the variable most correlated with the median production is longitude (0.931). Next is elevation (0.818). Latitude is correlated the least (0.034).

Figure 12

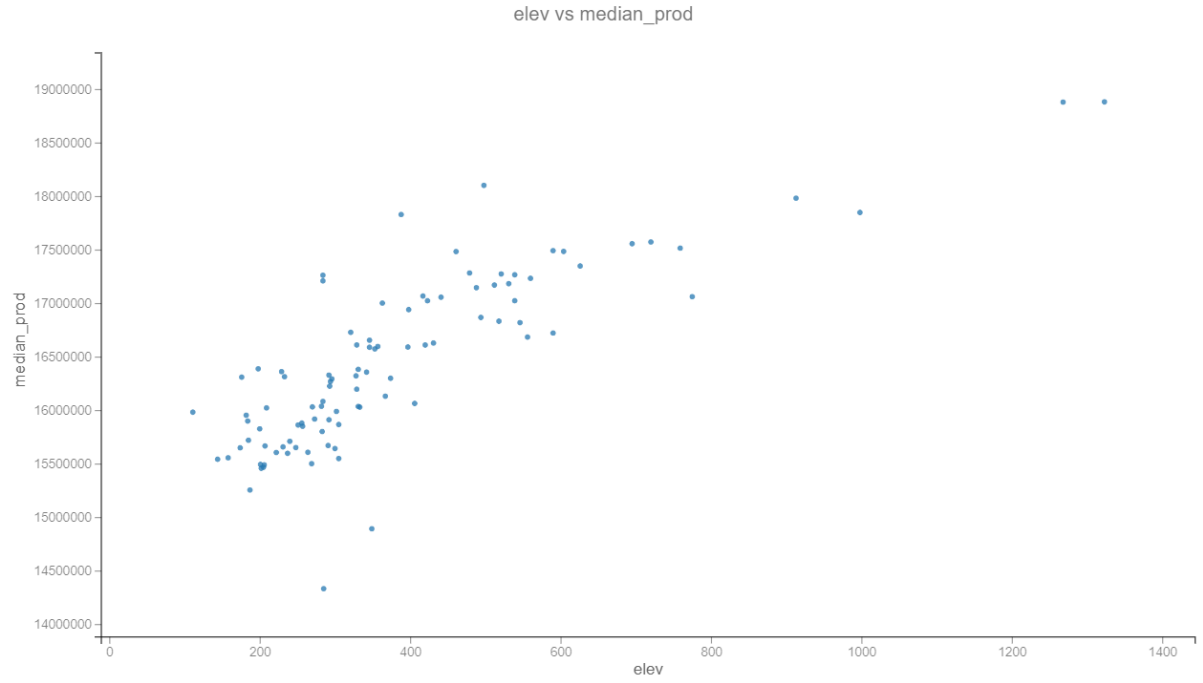


Figure 13

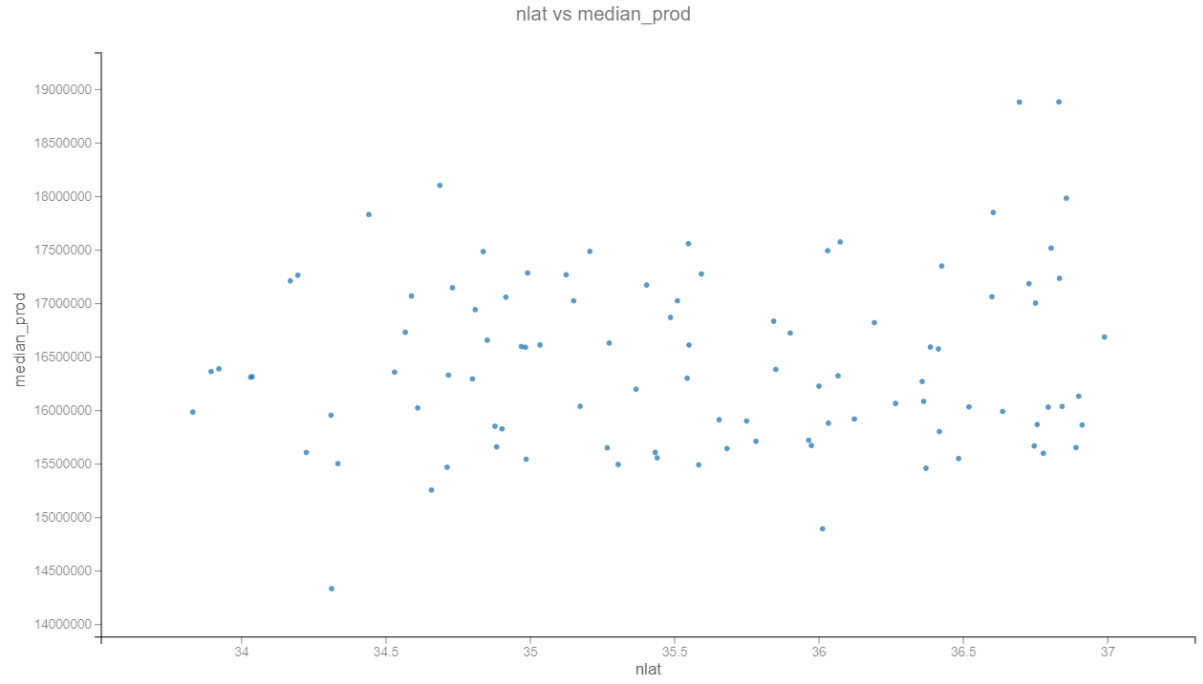
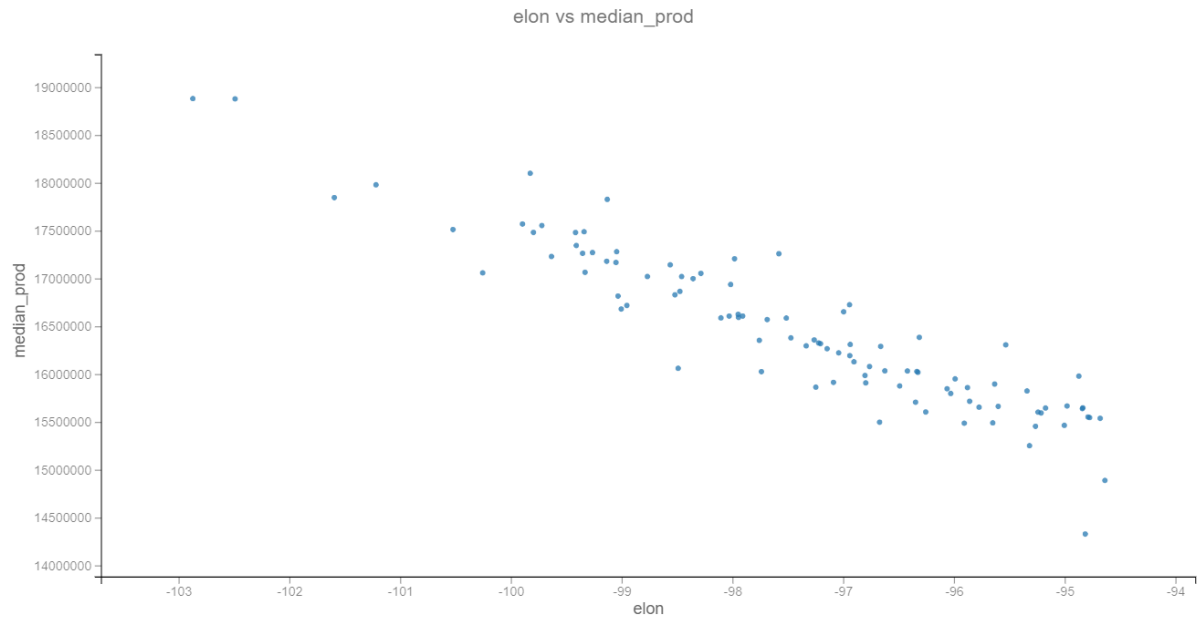
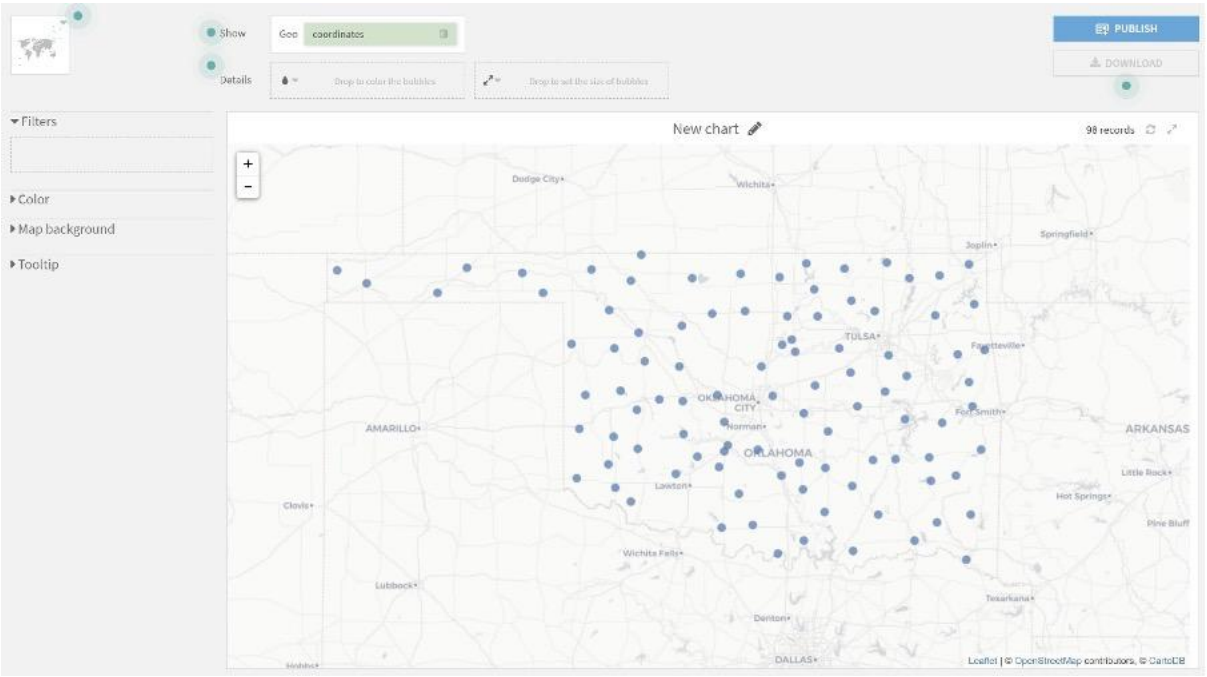


Figure 14



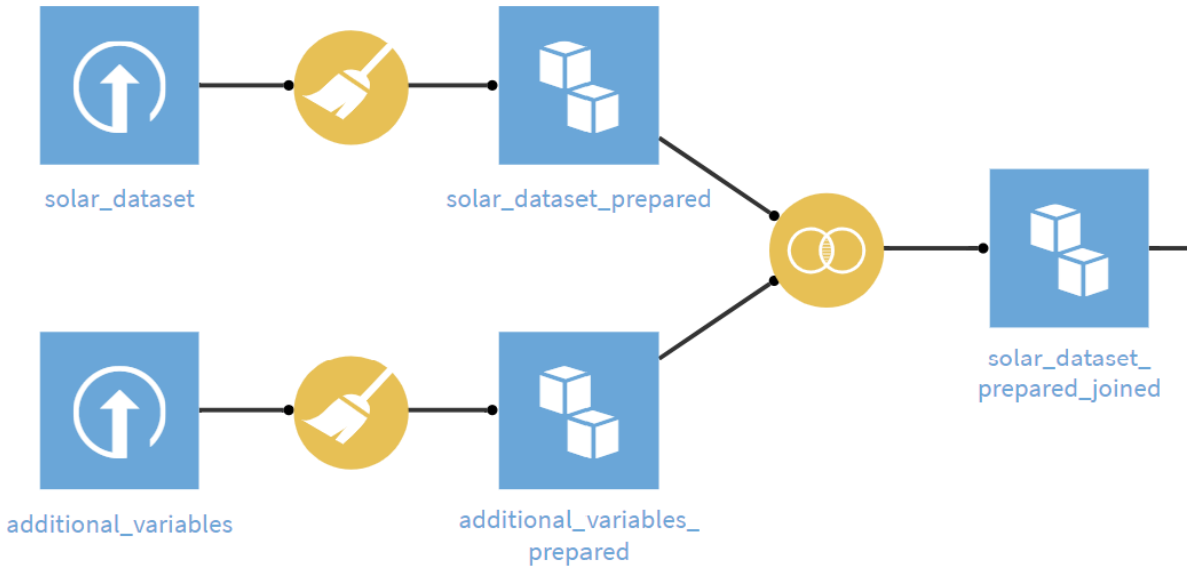
- Latitude is not correlated with production. Longitude and elevation are both highly correlated with solar production.
- Stations on the left of the map produce the highest solar energy compared to the rest (due to the longitude having a higher correlation).
- After noting this, we could suggest that, from a business standpoint, increasing the number of stations in the West would be beneficial
- These results match with what we have obtained from the correlation matrix.

Figure 15



- Visual representation of the 98 stations that we have in the dataset

Final dataset
- This dataset is the final result of our EDA and preprocessing
- the screenshot below demonstrates that we have chosen to label this dataset as "solar_dataset_prepared_joined"
- We will use this dataset to perform the modelling below



Modelling

The steps below describe the actions we took to train, validate and test our model on ACME, GOOD and WYNO.

- Select solar_dataset_prepared_joined, split it into **model** and **predictor**.

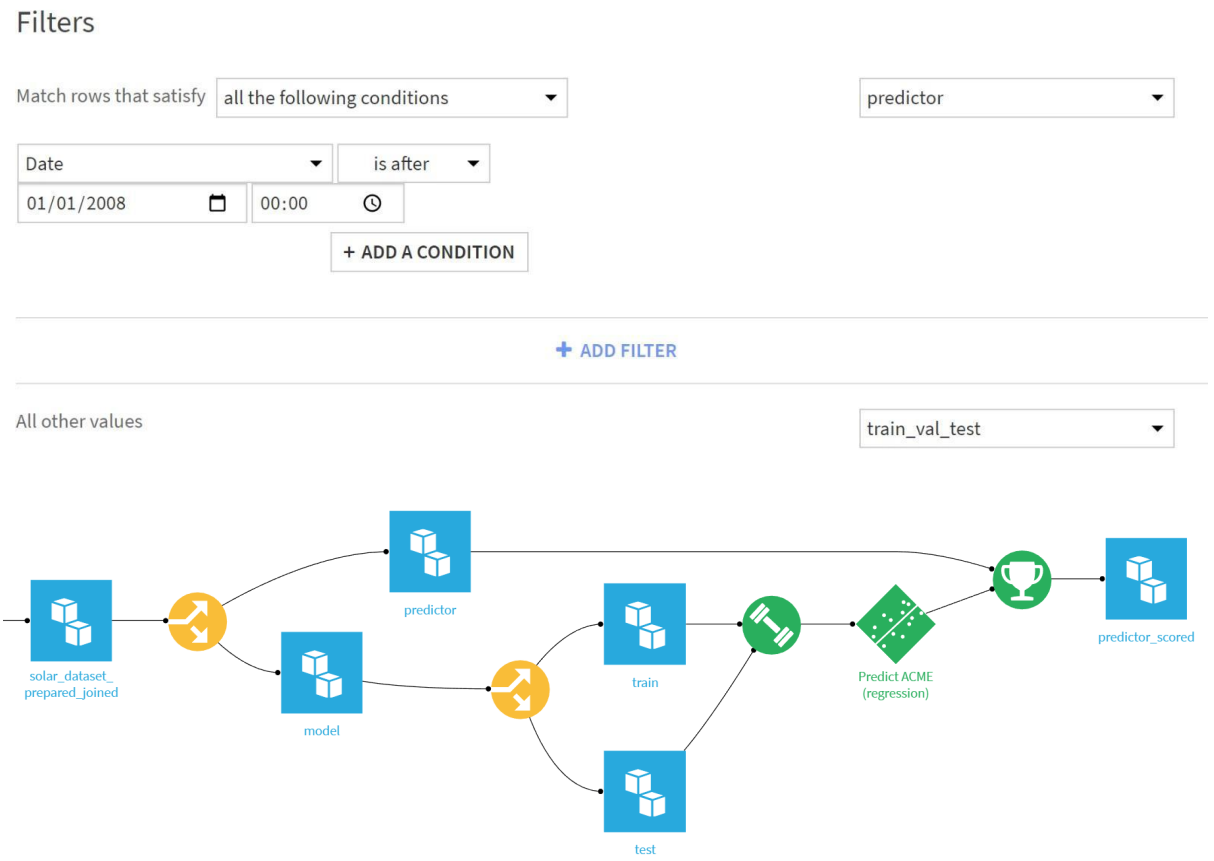
- Model contains all the data from 01/01/1994 to 31/12/2007. We split this dataset further into train and test:

- Train will be part of the dataset to do both training and validation, including data from 01/01/1994 to 31/12/2004.

- Test will include data from 01/01/2005 to 31/12/2007.

- Note that initially, we went with a different splitting approach: 80% train and validation, 20% test. Afterwards, we chose to split the dataset by year 2005 (we believed the data should go on a yearly basis) as it improved the model performance. The split used in the final model is still quite close to 80/20.

- Predictor contains all the data from 01/01/2008 onwards, where we will deploy the best model to predict values from 01/01/2008 to 30/11/2012.



For each station, we explored different algorithms and hyperparameters to achieve the optimal prediction model, based on the MAE score.

We noted a few overarching observations across roughly 80 models we tried as a team:

- Random Forest in most cases produced slightly worse results vs. XGBoost (~0.01 MAE score difference).

- With feature reduction, PCA was less effective than tree-based or lasso. There was no clear “winner” within the latter two options, but the tree-based showed better results with XG Boost and lasso performed better when used with SGD.

- Feature generation produced too many features, which slowed Dataiku down considerably.

- Grid search resulted in better performance vs. random search. Bayesian search failed due to technical issues with Dataiku.

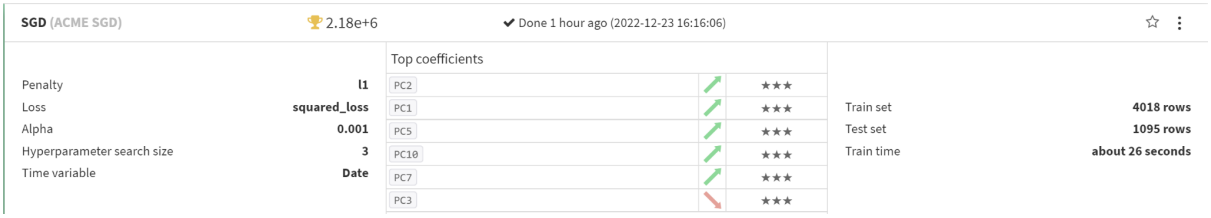
Please refer to the sections below where we explain in detail the models we finally chose for ACME, GOOD and WYNO.

ACME MODEL 2.18e+6 SGD

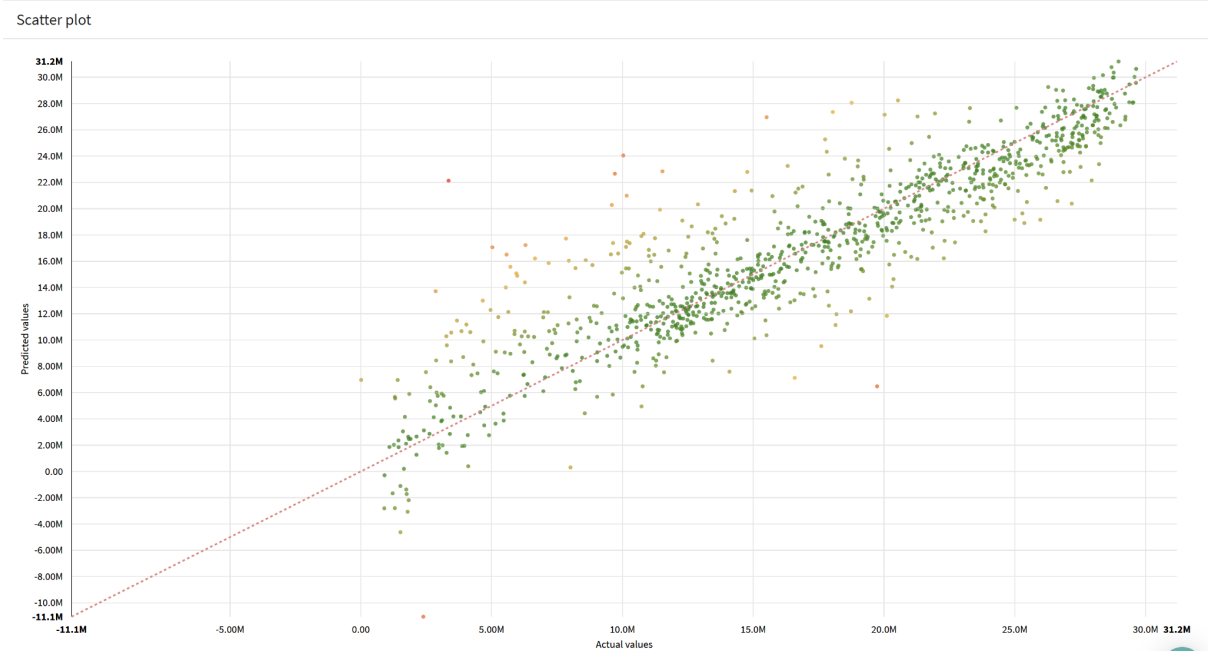
Score with main parameters:

1. We use additional variables (made to help predict ACME) and PC components
2. We start by applying all available models in dataiku.
3. Then out of all models, we choose the best performing ones: xgboost, random forest, sgd, and lasso.
4. After tweaking with the dimensionality reduction, we find that dimensionality reduction does not really help with improving our scores.
5. So we decided to do it without dimensionality reduction.
6. We filled the missing values with the mean.
7. Then we experiment with the rescaling, and we find that min max with sgd on acme.
8. For sgd, we have experimented with all available options and find that the L1 regularization works best.

SGD Model with MAE score

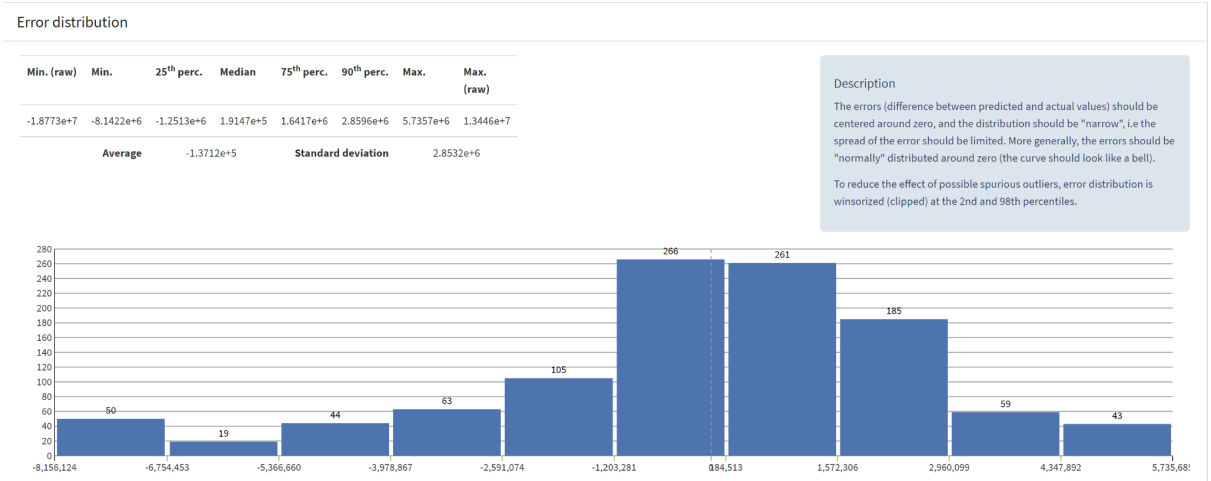


Scatter plot



We can see the line of best fit works better from 10M onwards, with lower values, the error increases.

Error distribution



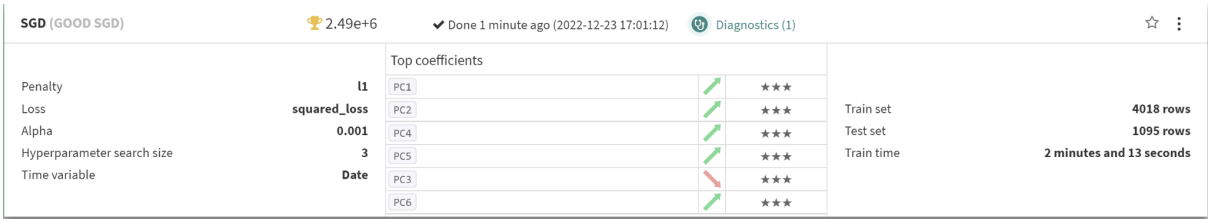
Most errors are centred around 0 and somewhat follow a normal distribution.

The best model is to the right.
We don’t see a change in the model score when we change the hyperparameter alpha. We cannot conclude if there is either overfitting or underfitting.

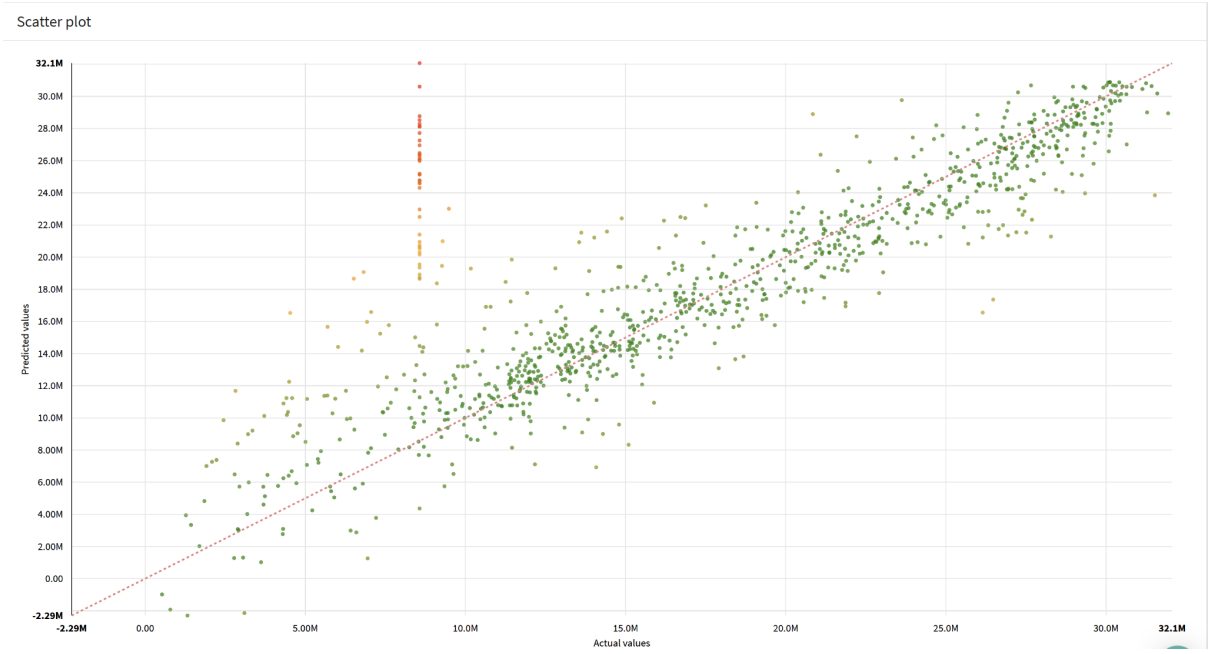
GOOD MODEL 2.49+e6 SGD

1. We use additional variables and PC components.
2. Although additional variables are made for ACME, we decided to include them for GOOD and WYNO because they improved our score.
3. We start by applying all available models in Dataiku, then out of all models, we choose the best performing ones: xgboost, random forest, sgd, and lasso.
4. After tweaking with the dimensionality reduction, we find that lasso reduction with regularisation ranging .1 to 100 seems to work best with our models.
5. We found that alpha with 0.001 seems to be the optimal balance between model complexity and predictive performance.
6. We filled the missing values with the mean.
7. Then we experiment with the rescaling, and we find that min max works best with sgd on GOOD.
8. Finally we deploy the model with sgd with l1 regularisation as penalty.

SGD Model with MAE score

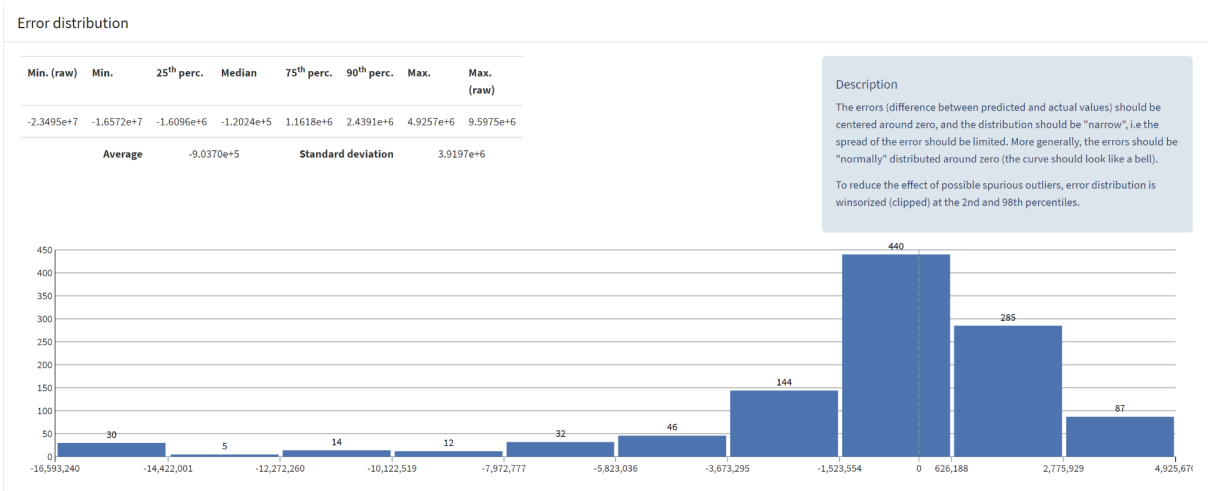


Scatter plot

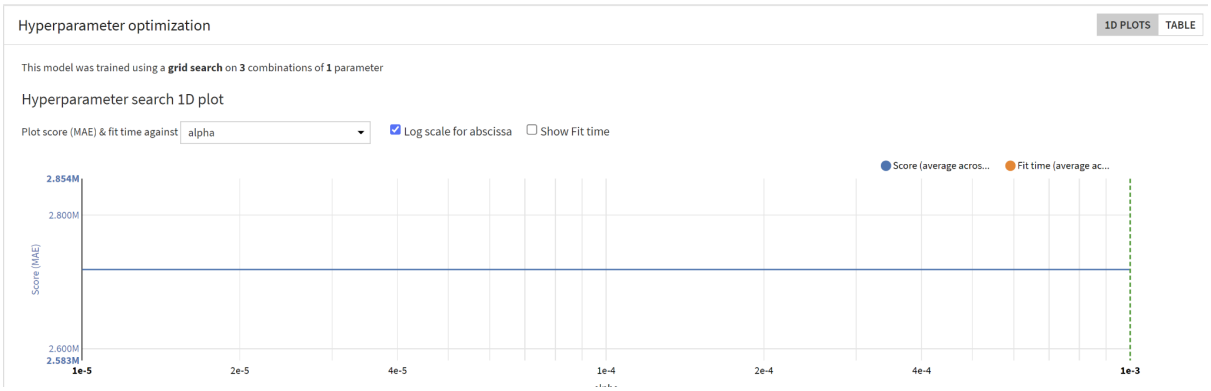


We can see the line of best fit works better from 10M onwards, with lower values, the error increases. In this case we can also see a cluster of errors at around 8M (Actual values).

Error distribution



Most errors are centred around 0 and somewhat follow a normal distribution.

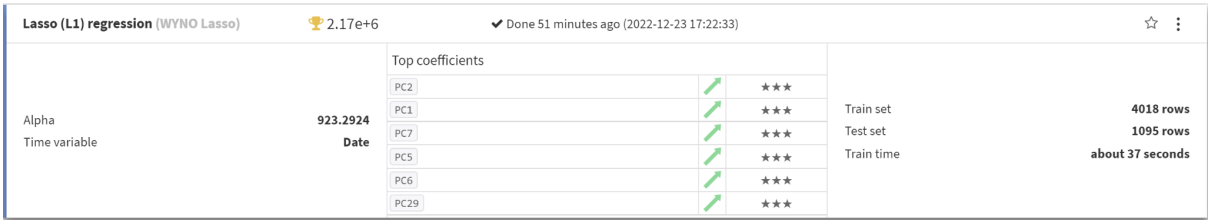


The best model is to the right.
We don't see a change in the model score when we change the hyperparameter alpha. We cannot conclude if there is either overfitting or underfitting.

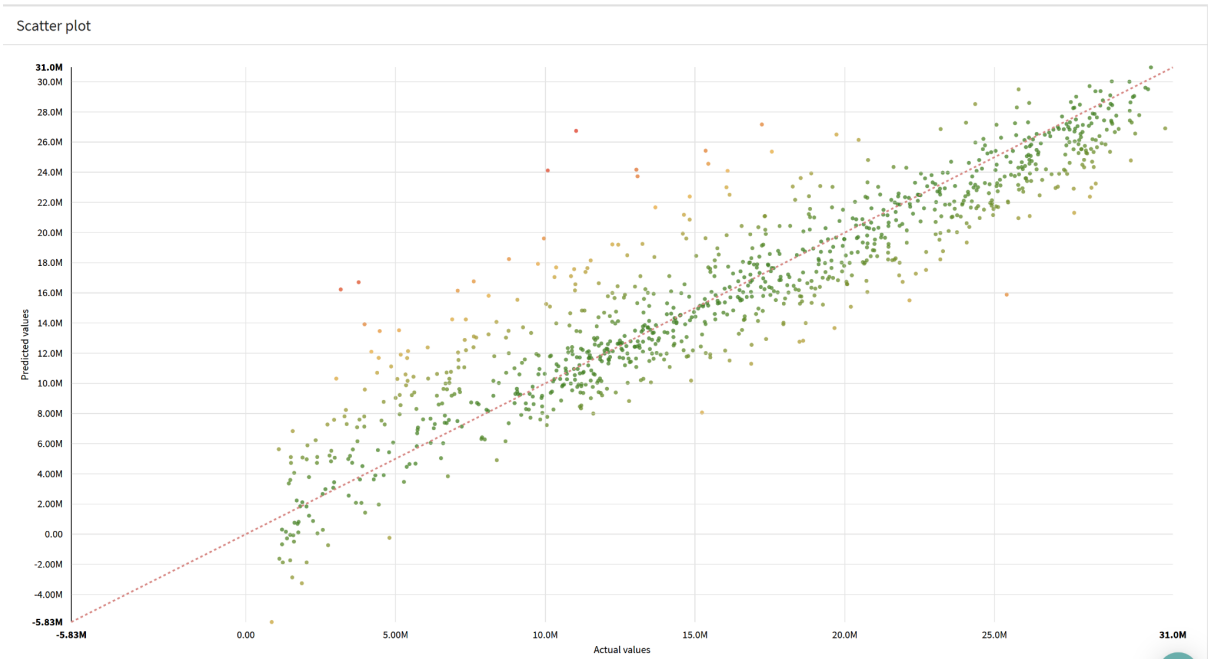
WYNO Model 2.17+e6 Lasso

1. We use additional variables and PC components
2. Although additional variables are made for ACME, we decided to include them for GOOD and WYNO because they improved our score.
3. We start by applying all available models in Dataiku, then out of all models, we choose the best performing ones: xgboost, random forest, sgd, and lasso
4. After tweaking with the dimensionality reduction, we find that none of them really improved our score. So we decided with no reduction.
5. We filled the missing values with the mean.
6. Then we experiment with the rescaling, and we find that min max rescaling works best.
7. We are using lasso regularisation with automatic optimization for alpha and the resulting alpha is 923.2924.

Lasso Model with MAE score

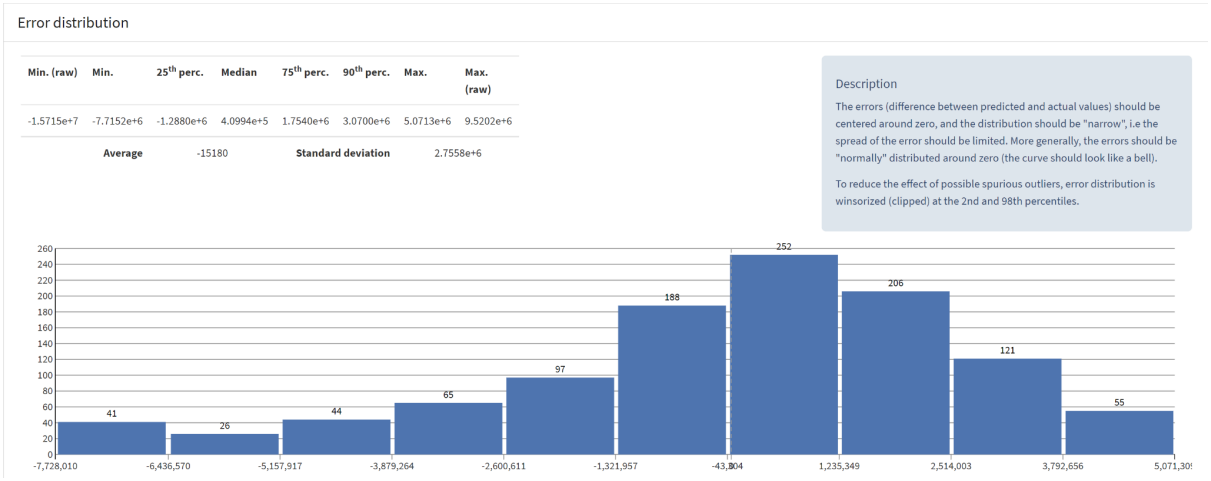


Scatter plot



The error is evenly distributed starting from 2.5M to 31M.

Error distribution



Most errors are centred around 0 and somewhat follow a normal distribution. More normally distributed than the SGD models.

Conclusion

So, why is Lasso the better performing algorithm in comparison to SGD for WYNO?

- We are using squared loss function and lasso for feature selection. SGD uses all the combinations of all features available rather than just a subset of features selected by Lasso, which might lead to worse performance if the model is overfitting to the training data.
- Lasso regularisation is often used for feature selection because it tends to set the coefficients of less important features to zero, effectively eliminating them from the model. This can be beneficial if you have a large number of features and want to reduce the complexity of the model, but it can also cause the model to lose valuable information if some of the eliminated features are actually important for making accurate predictions. It could be suggested that lasso is achieving better results with WYNO because lasso dropped the additional variables which are less important to WYNO.

Why is the MAE score worse for GOOD in comparison to the other 2 stations?

- GOOD has higher deviations in the dataset. We can confirm this by figure 7 plotted in the EDA section where we can see that GOOD has higher data deviations than ACME and WYNO.