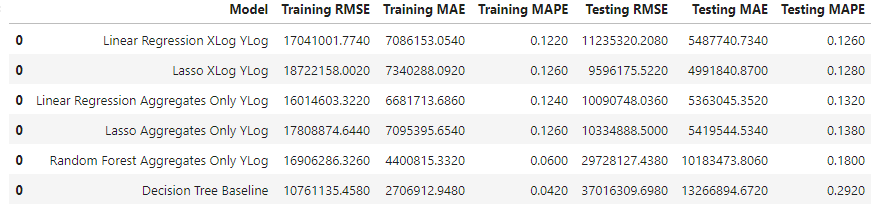
Notes

* Preprocessing
  + Knn imputation
  + Sparse features removal, columns removal
  + Apply poly or log transformation depending on …
  + Different variations of data, agg or not etc. log
    - Explain why, aggregates disaggregates, log transform x for homoskedasticity, log outcome due to skewness
* Methodology
  + General approach:
    - Explain why regression 1 sentence
    - 2 fold model training / 2 stage robust model approach
      * Train and test over 5 different seeds
        + Allows to get a better estimate of performance, especially due to the extremely high variability in such a small dataset
        + Assess stability and robustness of feature selection

We analyze how often different features were selected and identify which models rely on which features the most

* + - * Rerun best performing models on their respective dataset with robust features, set a minimum threshold for a feature to be chosen as robust feature while observing how many features are excluded to prevent oversimplification of the model
    - Evaluation
      * MAPE as the outcome variables absolute value is very high, resulting in larger RMSE and MAE
      * RMSE useful as it punishes outliers more and MAE is very intuitive and easy to interpete
      * To interprete log transformations to the outcome we exponentiate the outcome variable before calculating the metrics
    - Model-specific evaluation to identify adherence to monotonicity constraints and general variable contribution, for one random seed only
      * Lasso: coefficients, that need to be unstandardized for interpretation
      * LR: coefficients, that need to be unstandardized for interpretation
      * and p value
      * RF: partial dependency plot to analyze monotonicity
      * DT: plot tree for visualization
  + Models
    - Hyperparameter tuning generally
    - Cluster based approach
      * Intuition is that there might be several groups of network providers of different size or network type and look for the best model for each of the clusters (in this case 2)
      * First use density based spatial clustering of applications with noises (dbscan) to assign providers to clusters
        + We cluster them based on all of the variables that were assessed by expert engineers with their domain knowledge as potential cost factors (technical blocks variables)
      * Train lasso and RF to cover both linear and non linear relationships and find the best model for each cluster
      * After assigning clusters to test set we can evaluate the best models for each cluster and get a combined evaluation metric
    - Lasso regression
      * Inherent feature selection
      * Adhere to Monotonicity constraints by setting negative coefficients to 0
      * CV 5 tuning alpha
      * Standardization required
        + To interprete coefficients unstandardize
    - Lasso feature selection linear regression
      * Standardization required
      * Uses the same features from the lasso regression, but then just OLS without shrinkage from lasso
        + To interprete coefficients unstandardize
    - Rf
      * Using VIP for feature selection
    - Dt
      * Using VIP for feature selection
    - XGB?
      * With monotonicity constraints
* Results
  + 
  + These are the best performing based on mape and RMSE in the respective model category
  + You can also put the full table into appendix, which is like this table under the first summary results of evaluation metrics
  + Say that cluster based modeling does not work well because the best models for each cluster are too volatile leading to different model specifications in different seeds, since its not robust we stop exploring this model, you can also put the table in appendix if u want
  + Create a table showing the frequencies for each of those best performing models
    - Ein Bild, das Text, Screenshot, Zahl, Schrift enthält.

      Automatisch generierte Beschreibung
    - These are the frequencies, I did not know how to present them in a nice format maybe you could just aufzählen all variables tbh
  + Rerun of the best models with robust features:
    - Very last table:
    - 
    - Linear regression lower MAPE but **lasso** actually even lower RMSE and test MAE while very similar MAPE so that is probably best model
      * Also adheres to monotonicity constraints and uses rather stable features that appear with a threshold of 80%+
    - Generally log transforming outcome seems to be working well
    - Decision tree modeling leads to tendencies to overfit and does not work well here
    - Generally tree based models do not work as well as the linear models, possibly because they are worse at catching the mostly linear relationships, and also have worse interpretability so our recommendation
      * if we look at regulatory org that wants to have interpretable and monotonous effects with certain cost factors such ensemble models are not suitable in their common form
    - Describe hyperparameters possibly that were selected
    - Singular network provider Percentage deviation in lasso models mostly unter 15% with one exception around 40% you could put it into appendix or leave it doesn’t matter
  + Describe p values of linear regression,
  + Interprete coefficients:
  + Coefficients to be interpreted as log y log x so like 1% increase in factor is coefficient% increase in y, so e.g. coefficient 0.1 that means 1% increase in x means 0.1% increase in y or something
    - Need to unstandardized them before
  + RF partial dependency plot shows that it almost always adheres to monotonicity ocnstraints, which one can see with the flat and increasing parts of the line
  + DT decision tree, idk feels kinda useless make something up
* Conclusion

Limitations:

* We are aware of data leakage in the form of feature selection on the same dataset and then hyperparameter tuning but due to computational power we don’t pursue nested cv, the bias seems to be very small from initial exploration
* Hybrid cluster modeling would not be fair due to inequitable treatment based on assignment of cluster possibly, approach was tried purely for predictive value
* FEATURE ENGINEER GEOGRAPHICS? We were thinking about it but left it out

We guarantee monotonicity constraints in Lasso by only taking features subselected by lasso with a coefficient > 0

* + but we set everything <0 to 0 and don’t refit since this can lead to again negative coefficients
  + In such cases, it's important to acknowledge that you're applying a post-hoc adjustment based on external knowledge rather than purely on data-driven optimization.
  + In this case it worked very well and is still the best model
* Use more domain knowledge to refine the variable selection (is outlook at the same time)

Outlook

* Outlook use RFE/forward feature selection with more computation power
* Using nested cv to fully prevent data leakage when doing feature selection and hyperparameter tuning
* Outlier strategy
  + Cluster based outlier detection
  + Isolation forest
  + Dividing into different sets of DSO
  + Remove too efficient providers? We would want to have a positive bias, so costs are estimated higher for the same efficiency so this is reached by removing super efficient providers
* NNLS for monotonicity constraint in linear regression, non negative least squares – this has the constraint in the algorithm, unlike post processing how we did it
* Use more domain knowledge e.g. combinations of different technical blocks and test out the models again, this requires a lot of computational power because trying all combinations results in 1.6k iterations

To-Do:

* Read overleaf
* Read chloe EDA
* Xgb
* Other scenario