Notes

* Standardization for LassoR is must
* We propose an ensemble approach
* Outlook use RFE/forward feature selection with more computation power
* Using nested cv to fully prevent data leakage
  + Here its fine look at text, tradeoff computational power
* Hybrid cluster modeling is not fair due to inequitable treatment based on assignment of cluster possibly, we check for predictive value though
* Cluster building without outcome due to data leakage
* Explain we took a regression not classification approach
* Log transformation looks very bad – possibly log transform a few only?

Timeline:

* Create preprocessing scenarios
* Create evaluation for log/sqrt transformed outcome
* Implement rest of models including clustering
* Implement seed averaging and vip tracking that way, vip across models
* Implement better outlier strategy + write
* Check monotonicity, interpretability, model selection and robustness, single provider MAPE
* Benedict: LR, DT
* Duci:
  + Preprocessing with knn etc
    - Create other preprocessing scenarios
    - function die alle szenarien kreiert
    - optionally log transformation
    - scaling - x-scaled and x-train separately for each model whenever i need
  + Modify existing ml models
  + Cluster based approach
  + See how to outsource variable importance? Rn very printbased
  + Implement seed averaging, vip tracking / vip across models, evaluation metrics and pipeline of list of seeds and list of scenarios and then models for which, tuples
  + Outlier strategy
  + Monotonicity, interpretability, single provider MAPE check max

**Approach**: average over 10 seeds the MAPE etc. and how often which variables were selected with another column with the %s or sth or same column

* Potentially check the largest percentage deviation? To see for their model how their cap should be calculated

**Current Scenarios:**

Lasso regression

* Log transforming outcome
* Sqrt transforming outcome
* Log transforming feature variables
* Using only tot/sum variables or only disaggregates
* Splitting it into aggregating 1-4 and 5-7 and removing tot/sums
* Usage of technical blocks
  + Random grid search of technical blocks combinations to catch all 3 dimensions
* Different outlier strategy

**Ideas for aggregation scenarios:**

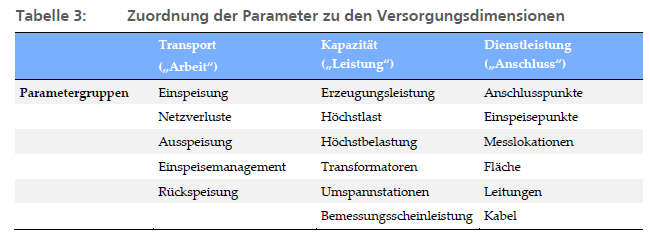
Transportation: energy delivered, network length

Capacity: peak load, installedf power

Service: metering point

The Ordinance is specific about a minimal set of cost drivers. Cost drivers suchas connections, areas, circuit length, and peak load, are obligatory -> regulation paper Ein Bild, das Text, Screenshot, Schrift, Zahl enthält.

Automatisch generierte Beschreibung Ein Bild, das Text, Screenshot, Karte Menü, Dokument enthält.

Automatisch generierte Beschreibung  Ein Bild, das Text, Screenshot, Schrift, Zahl enthält.

Automatisch generierte Beschreibung Ein Bild, das Text, Screenshot, Zahl, Schrift enthält.

Automatisch generierte Beschreibung

Capacity over transport Ein Bild, das Text, Screenshot, Software, Webseite enthält.

Automatisch generierte Beschreibung Ein Bild, das Text, Screenshot, Zahl, Software enthält.

Automatisch generierte Beschreibung Ein Bild, das Text, Screenshot, Schrift, Zahl enthält.

Automatisch generierte Beschreibung

Anhang für beschreibungen checken

12.08.2024

To-do:

Today is more exploring to understand direction:

* standardization
* Read term paper document and stuff to get more knowledge
* Try out pca for linear regression -> principal component regression with just certain variable sets such as the most important technical blocks, comes with loss of interpretability though
* Try out RF stability
  + DT, XGB; LR, possibly try RFE here, using RF for somewhere else does not mean that model perforsm with those variables the best either
    - Linear regression as base model with their variables selected
* Try out clustering and different things
  + Use the technical blocks or domain knowledge to cluster the providers into different clusters and then use different models, divide data into clusters 2-3
    - Use all of those parameters of technical blocks to separate it
* Potentially do evaluation outside of model itself, return best model and then predict and evaluate this way we can also analyze log and sqrt transformed predictions
  + Change up validation set test for evaluation bc rn we do another train test split
* Big steps:
  + Implement all of the models
  + Create all preprocessing scenarios Dienstag ideally
  + Run all models and track metrics over different seeds
  + Implement better outlier strategy Mittwoch + writing
  + Check monotonicity constraints and others
  + Focus on interpretability and model selection and robustness
* Writing and rest
* Think about outlier strategy
  + Cluster based outlier detection
  + Isolation forest
  + Dividing into different sets of DSO
  + Remove too efficient providers? We want to have a positive bias, so costs are estimated higher for the same efficiency
* Running experiments of just a select few variables to see their importance
* Look at mape but of only single providers to see how much they differ at most

Cluster-approach:

* Cluster them based on all parameters of the technical blocks
* Use the models to train on each of them separately, the ones that I have so far

Other approach:

* Use random grid search of technical block combinations to evaluate – one of the scenarios

08.08.2024

* finish monotonicety constraints etc. with lasso
* feature engineer province and lat long and east west, call log trafo etc feature engineering
* validation set always in preprocessing of scenarios
* read term paper document and stuff, get more knowledge
* PCA
* Clustering and models
* How do we make our work interpretable
  + Take variables selected through different models and try to estimate a linear regression

Lasso handles multicollinearity inherently

Homoscedasitciaty checks

Significance using many model runs and seeing how volatile the best feature subset is

For random forest use PDP

06.08.2024

To-Do:

Try out a regression? Other model, that splits it into two groups of providers and then

Use clustering to identify different number/types of clusters and run different models on each of them, assuming that the providers have different characteristics and source of costs and this differs between the providers – HAS TO ALL HAPPEN AFTER TRAIN TEST SPLIT

Ensemble method for prediction of different clusters, maybe this cluster works best with RF maybe one with regression

* Data preprocessing
  + Knn – reasoning!!
  + Outlier detection
    - Different ways of handling:
      * Log transformation
        + Create a second column with log transformation for all of them
      * Capping/clipping
        + Extreme outliers could cause large errors for regressions, more robust
  + Encode categorical variables
  + Log transformation
    - Run models with and without log transformation
  + scaling not before due to data leakage, check whcih models need scaling
* read the term paper document and stuff
  + create scenarios to run, understanding of the diff variables and create mapping or ideas
* start training and running models with scenarios, create pipelines ish
  + lasso/RF
  + PCA? With reduced number of variables
  + Maybe first analysis and seeing which types of variables important and then use that subset or take those aggregated
  + Aggregate/disaggregate
  + Types of combinations, only a few
  + Only tion, service, capacity or group them into that, mix of them, choose x of them
  + regularization
  + Hyperparameter tuning
  + Evaluation metrics#
    - Mape and justify using industry standards
    - Check variance of the MAPE
* Variable importance and interpretability
  + E.g. just additional decision tree
* Think about statistical tests and significances, constraints

capacity = peak load and installed power, transportation = energy delivered, network legnth, service = metering point, they want to change their variable selection and e.g. boundaries, negative correlation of revenue and entwork length would be bad

Scenarios: create all in preprocessing

* Let model decide (rf, lasso, feature forward selection)
* We select certain groupings
* Transform log
* Aggregate 1-4 and 5-7 or so

Lasso, RF, Linear Regression, xgboost, DT

Pca + clustering

**Adequate Number of Variables**

* **Avoid Over- or Underspecification**:
  + **Over-specification**: Including too many variables can lead to overfitting, where the model captures noise rather than the underlying relationship. This can make the model less generalizable to new data.
  + **Under-specification**: Including too few variables can lead to underfitting, where the model fails to capture important relationships and hence provides poor predictive performance.
  + **Optimal Variable Selection**: Select a sufficient number of relevant variables that contribute to the cost prediction without adding unnecessary complexity.

**No Multicollinearity**

* **Multicollinearity**: This occurs when two or more predictor variables in a regression model are highly correlated, leading to unreliable estimates of coefficients.
  + **Detection**: Use Variance Inflation Factor (VIF) or correlation matrices to detect multicollinearity.
  + **Mitigation**: Remove or combine collinear variables, or use regularization techniques like LASSO.

**Homoscedasticity**

* **Definition**: Homoscedasticity refers to the condition where the variance of the residuals (errors) is constant across all levels of the independent variables.
  + **Importance**: It ensures that the model’s predictions are equally reliable for all levels of the independent variables.
  + **Achievable Through**:
    - **Normalization**: Scale the variables to have a mean of zero and a standard deviation of one, which can help in reducing heteroscedasticity.
    - **Adequate Transformation**: Apply transformations like log or log-log (both dependent and independent variables) to stabilize variance and achieve homoscedasticity.

**Economic Effects & Significance**

* **Correct Sign (Monotonicity)**:
  + **Interpretability**: Ensure that the signs of the regression coefficients align with economic theory and intuition. For example, if an increase in a certain variable is expected to increase costs, the coefficient should be positive.
  + **Monotonicity**: The relationship between predictors and the response should be consistent with economic logic (e.g., higher input prices should generally lead to higher costs).
* **Overall Significance (F-test)**:
  + **F-test**: Used to assess whether at least one predictor variable has a non-zero coefficient.
  + **Model Validity**: Ensure the overall model is statistically significant, indicating that the predictors collectively have explanatory power.

05.08.2024

* Approach for feature/variable selection and transformation
* Feature engineering
* Writing functions/pipeline to easily run all models and get insights

Approaches

1. Data preprocessing
   1. Missing values, imputate or remove
      1. Use knn for imputation
   2. Standardization and normalization
2. EDA: DO CHECKS USING SHOW ME COLUMNS WHERE THIS APPLIES
   1. Correlation analysis
      1. Check for highest correlation with outcome variable
   2. Variable distribution
   3. Outlier detection
3. Feature selection
   1. Obvious irrelevant
   2. Lasso L1 regularization
   3. PCA
   4. Treebased like RF, for feature importance
   5. Forward subset selection

Map, interesting distributions correlations – feature engineer bundesland

PCA + X

Lasso, elastic net, ridge

RF, xgb

SVR had pretty bad results

DEA take a look at that

Aggregated/Disaggregated

Transportation, capacity, service

Step by step clean up as far as possible, with reasonable explanation

Read through regpaper/efficiency benchmarking papers

Start creating functions to automatically run, with hyperparameter tuning and cross validation (little obs so maybe think about how to do this)

Monotonicity constraints etc.

Building for different scenarios and use cases a pipeline:

* Features that are under consideration change
  + Aggregate/Disaggregate
  + Types and transformations, interactions
  + Include geographical features?
  + Different types of combinations, weighted etc.
    - Combining many of the ones that are not that useful alone, or the ones with low variance
* Also more regularization?

Pipeline that does:

* Feature selection
  + Inherent
    - Lasso l1
    - Random forest
  + other
    - PCA
    - Forward subset selection
* Model training
  + Hyperparameter tuning
  + Evaluation metrics#
    - Mape and justify using industry standards
    - Check variance of the MAPE
* Variable importance and interpretability
  + E.g. just additional decision tree

Think about statistical tests and significances

Efficiency benchmarking so think about what variables they want to use or have for deciding on the revenue cap, and which ones are here reelvant

It is also wanted to not change between years too much -> the pipeline could be run with new data

Transform all using standard or if skewed condition

Run 1 pca and combine with sth that makes sense prob not rf

Run 1 forward subset selection using xgboost

from sklearn.feature\_selection import VarianceThreshold

# Load data (example dataset)

data = pd.read\_csv('data.csv')

X = data.drop('target', axis=1)

# Apply Variance Threshold

selector = VarianceThreshold(threshold=0.01) # Adjust threshold as needed

X\_selected = selector.fit\_transform(X)

print("Original shape:", X.shape)

print("Shape after Variance Threshold:", X\_selected.shape)

1. Flow diagram for overview over all variables
   1. Na values, mehr als 50% missing z.b. weil wir nur 194 obs haben
   2. Close to 0 bei zb. Nujmber of injection oder connection points

Notieren, welche spalten rausgeschmissen werden

Interpretierbarkeit: unterscheiden von 300 verschiedenen connections

Skala von 1-7, krieg ich die anderen einsortiert

Whether traffic light just simply works also in a connection point, aggregate there

12 obervariablen, folie 19, correlation innerhalb obervariable, variable importance, welche davon korreliert am ehesten mit outcome

Mehrere ansätze fahren: rigoros, variablen selecten mit verschiedenen approaches und dann wo überlappen ist gut

Ctotexn outcome bc…

Step forward selection

3 gleise:

Remarks chatgpt

Yes, it is valid to say that the risk of data leakage is relatively small when not using nested cross-validation, especially in the context of practical machine learning applications. This is a common trade-off in machine learning between computational efficiency and the potential for slight bias.

**Why the Data Leakage Risk Might Be Small:**

1. **Same Training Data**: When you use a pipeline that combines feature selection and hyperparameter tuning, both processes are still applied only on the training data within each fold. The validation set is never used to directly influence feature selection or model training within that fold.
2. **Indirect Influence**: The potential for leakage arises because the feature selection process might indirectly affect the hyperparameter tuning, as they are interdependent within the cross-validation process. However, this influence is often minor compared to the overall variance in model performance.