VU - Machine Learning
Exercise 1: Regression

Group 19:
Alexander Leitner, 01525882
Aleksander Hadzhiyski, 01426981
Peter Holzner, 01426733

Characteristics of data sets & pre-processing

Characteristics of data sets:

- Moneyball (Baseball): Moneyball (Baseball)
  - Low number of instances and low-mid number of attributes
- Appliance Energy Consumption: Appliance Energy Consumption
  - Mid number of instances and high number of attributes
- Metro Interstate Traffic Volume: Metro Interstate Traffic Volume Data Set
  - High number of instances and low number of attributes
- Bias correction of temperature forecast: Bias correction of model temperature forecast
  - Mid number of instances and mid-high number of attributes

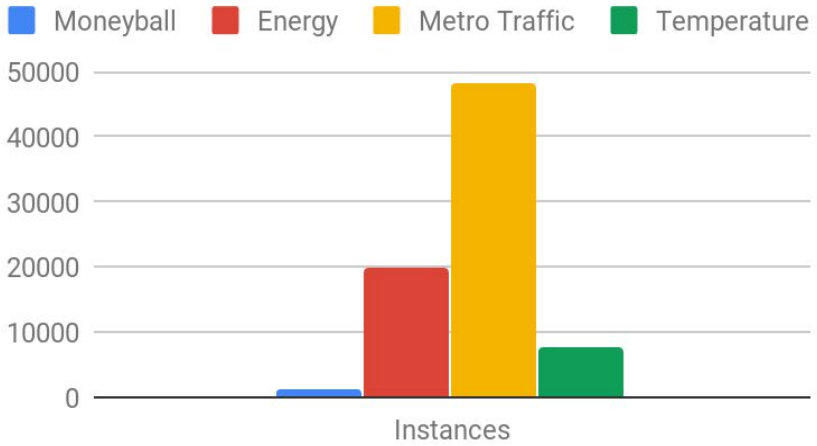
Explanation of choice for data sets & techniques

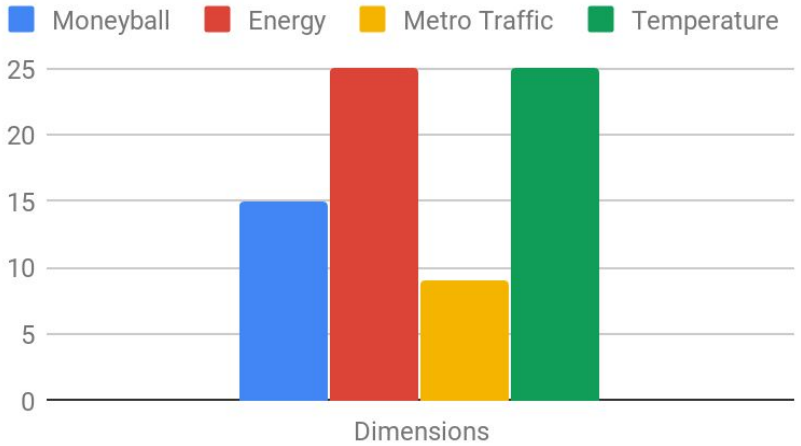
Choice for data sets:

- The datasets were chosen in order to be as diverse as possible (different number of instances and dimensions). Furthermore, our aim is to work with data sets which include real-world values and examples.

	Moneyball	Energy	Metro Traffic	Temperature
Instances	1.230	19.800	48.204	7.750
Dimensions	15	25	9	25

Explanation of choice for data sets & techniques





Explanation of choice for data sets & techniques

Choice for techniques:

- Linear Regression
- Lasso Regression
- k-Nearest Neighbors
- Random Forest

- The main idea behind the chosen techniques is to cover most of the different ways of creating regression models with different underlying methods and different algorithms, apart from the similarity between Linear and Lasso Reg.

Description of regression techniques

- **Linear Regression:** need a linear dependence between the input datas. The model tries to find the best fit linear function such as:

$$J = \frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2$$

this expression minimizes and finds the best possible values for a_0 and a_1

and forms the linear function :

$$y = a_0 + a_1 x$$

Description of regression techniques

- **Lasso Regression:** Technique description

$$J_L = \sum_{i=1}^{n} (\hat{y}_i - y_i)^2 + \alpha \sum_{j=1}^{m} |\omega_j|$$

One difference between Lasso and Linear regression is the additional term with a parameter alpha where it could be any value between 0 to infinity and omega is the slope of the linear function. It will be determined using Cross Validation.

Description of regression techniques

Pros and Cons for Lasso and Linear Regression

Pros Linear Regression	Cons Linear Regression	Pros Lasso Regression	Cons Lasso Regression
least complex if there is a linear relationship between the variables	In real world not often available	fast in terms of inference and fitting	Model choosing by lasso is not stable
		It can do feature selection	Should have a linear relationship between the variables
		Applied large numbers of features	

Description of regression techniques

- **Random Forest(RF)*:** An ensemble learning technique based on decision trees, that produces a result by aggregating the results of all trees (hence “forest”)... "wisdom of the crowd". The reduction technique used depends on the task: classification → majority vote, regression → average, are common choices. The RF tries to cultivate very different trees by assigning each tree only a randomized subset of all features + uses bootstrapping → “random” forest.

Advantages	*Disadvantages*
handles missing data well, only simple imputation needed	(sometimes) not as great for regression because it doesn't actually give continuous output
maintains data accuracy + usually doesn’t overfit	little control over what model does (black box)
parallelizes well → performance gain	computation time (when unoptimized)
large amounts of data + high dimensionality	
data analysis: feature importance	

*: from lecture notes + supplementary source Géron, A. (2019). *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow, ISBN: 9781492032618.*

Description of regression techniques

A simple Random Forest in Pseudocode*:

Training:

1. Assume number of cases in the training set is N. Then, a sample of these N cases is taken at random but with replacement (bootstrapping).
2. If there are M input variables (or features), a number m < M is specified (subset of features) such that at each node, m variables are selected at random out of the M. The best split on these m is used to split the node. The value of m is held constant while the forest is grown.
3. Each tree is grown to the largest extent possible and there is no pruning.

Prediction:

1. Let each tree produce its prediction output.
2. Aggregate the individual prediction into the final result of the Random Forest (i.e. majority vote for classification, average for regression)

*: adapted from Géron, A. (2019). *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow, ISBN: 9781492032618.*

Description of regression techniques

- **k-Nearest Neighbors:** A classification algorithm operating on the principle of a datapoint being classified based on its proximity to "neighbouring" points. Training Algorithm includes storing all the data and the prediction algorithm mainly consists of 3 steps:
  1. Calculate the distance from x to all points in the data.
  2. Sort the point in the data by increasing distance from x.
  3. Predict the majority label of the k-nearest neighbours (closest points).

  Generally, choosing a different k will affect what class new points will be assigned to. A greater k will often lead to a more "smooth" and "biased" model --> a cleaner separation of the model. It must also be considered, though, that this reduction in the noise is at the cost of more potential errors.

Description of regression techniques

k-Nearest Neighbors Pros & Cons	
Pros	**Cons**
Simple	Costly - algorithm gets worse for larger data sets
Trivial Training	Not optimal for categorical features
Works with great number of classes	Not optimal for high dimensional data sets
Adding data is easy	
Few algorithm parameters: k and distance metric	

Moneyball

Moneyball (Baseball) Data Set

Data Set description: The basic idea behind the data set is that a well analysed batch of data could prove to be groundbreaking in some cases. The presumption in baseball is that the scouts are the ones looking for new players to perfectly match them to a team in order to improve the team based on their knowledge and “feeling” for the game. With the help of the information in this data set, the “feeling” of the scouts was disputed and it was shown that some factors are far more important than previously considered.

Instances and dimensions: The Data Set has a mid number of instances: 1230 and a mid number of dimensions: 15.

Missing values: There are several rows with missing values.

Target: The target variable chosen for the task is: W...Wins of the team.

Characteristics of data set & pre-processing

Firstly, some columns were dropped, such as:

- “Team”, “League”, “Year”, “RankSeason”, “RankPlayoffs”, “Playoffs” - none of those are essential to prediction of the output variable and the regression models

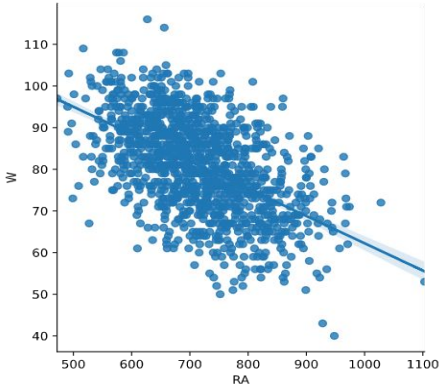
Secondly, the missing values (within the “OOBP” and “OSLG” columns) are filled with a constant value of 0 since most of the missing values are from the very early years and it’s hard to represent how the game has developed and a “mean”-imputation wouldn’t be fully correct.

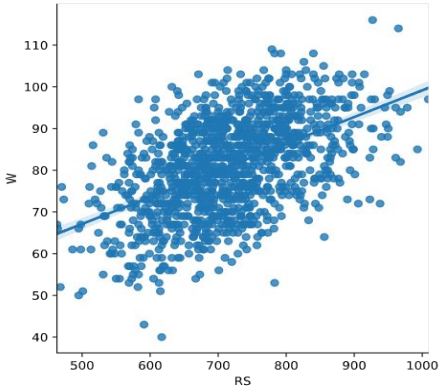
Data set experiments

- **Linear Regression experiment:**
  - Find whether there's a strong correlation between some of the parameters and the target variable.
  - Therefore, after preprocessing the data set, we plotted some of parameters alongside the Wins-variable and got the following results.
    - There's a very strong correlation between the Runs Scored (RS) and Runs Allowed (RA) variables and the Wins, which came of no surprise at all.
    - We also saw that the other "important" variable isn't only the Batting Average (BA) but also the On-Base-Percentage (OBP) alongside the Slugging-Percentage (SLG) - with the last two variables being largely ignored before the findings of the crew which inspired the movie "Moneyball".

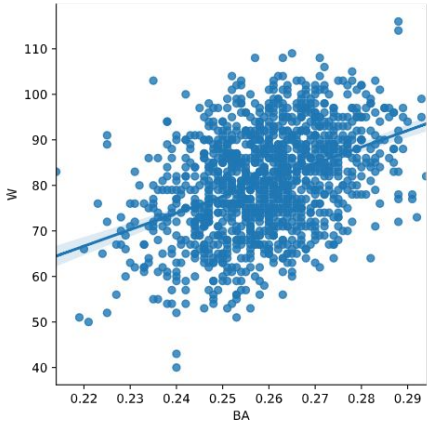
  On the next slide are plots to describe our findings.

Data set experiments - Linear Regression

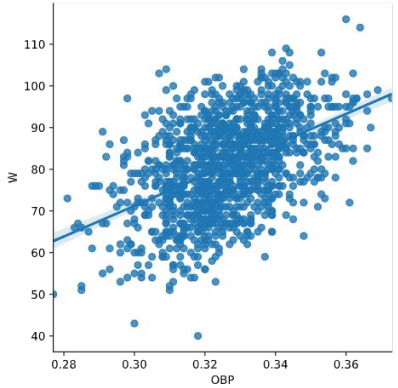


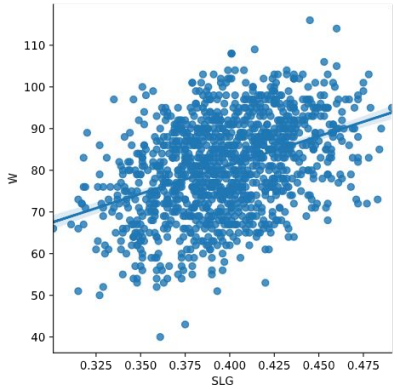


Logical dependencies



Previously considered the most crucial parameter





Found out to be as important as BA

Data set experiments - Linear Regression

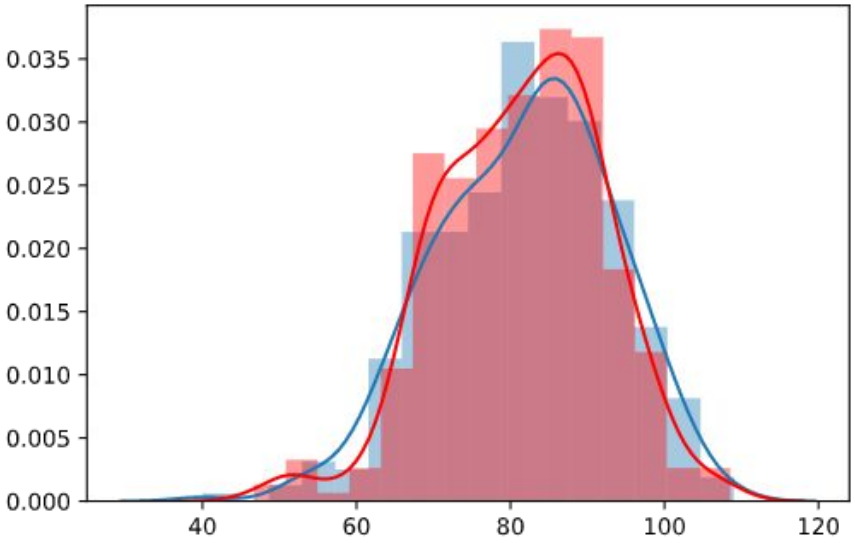
- Continuing with the Linear Regression experiment, we used separated the data in a train/test split with the test size being 30% of the whole data set and achieved the following results using the scikitlearn-Linear Regression Model:

    - Mean Absolute Error: 3.0310632639832904
      Mean Squared Error: 14.671183571500599
      Root Mean Squared Error: 3.8302981047825244

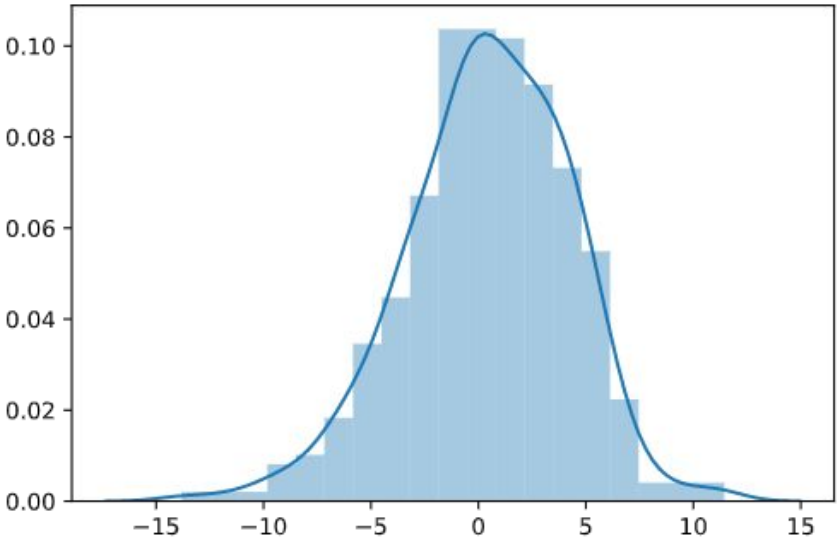
    - Model R^2-Score: 0.8877439871498782

- Those results were interpreted as very good. To visualize the model performance, we plotted the test data set with alongside the results predicted by the model and the difference between the two.

Data set experiments - Linear Regression



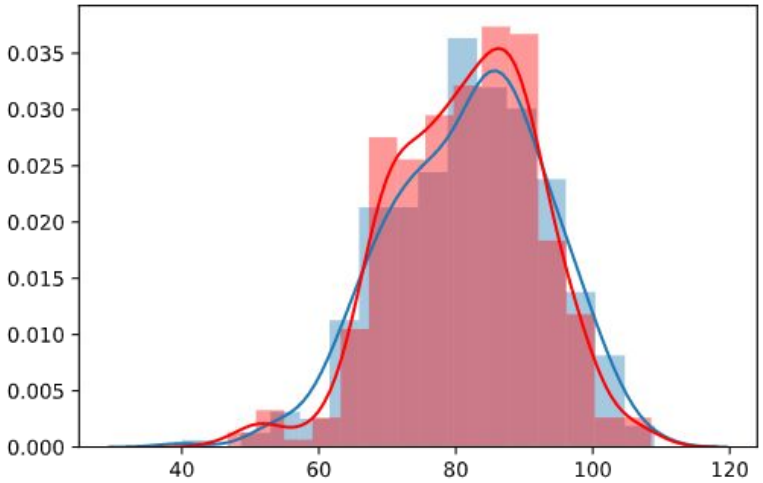
Test Data Set vs. Model Prediction



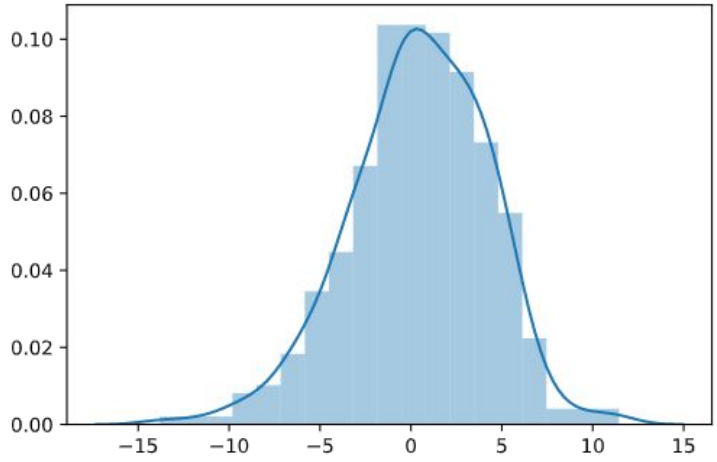
Difference between the model and the data set

Data set experiments - Lasso Regression

Based on the good performance of the linear regression model, we were quite sure that the lasso regression model would deliver good if not the same results and after trying alpha paramerest in the range from 1e-15 to 100, we found out that the best alpha-value is: 1e-8 with the following results:



Test Data Set vs. Model Prediction



Difference between the model and the data set

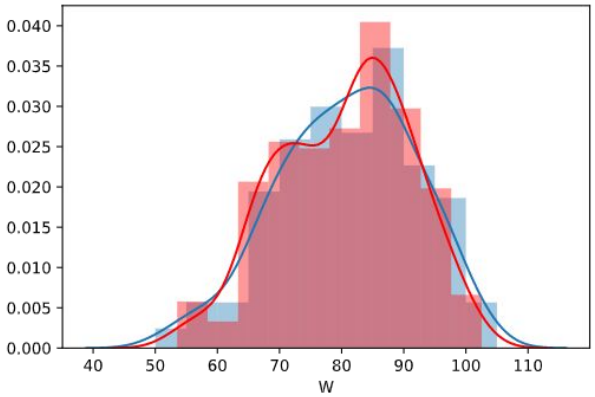
Data set experiments - Random Forest

The Random Forest regressor finds the same features to be important for its prediction, that were important for the linear regression model.

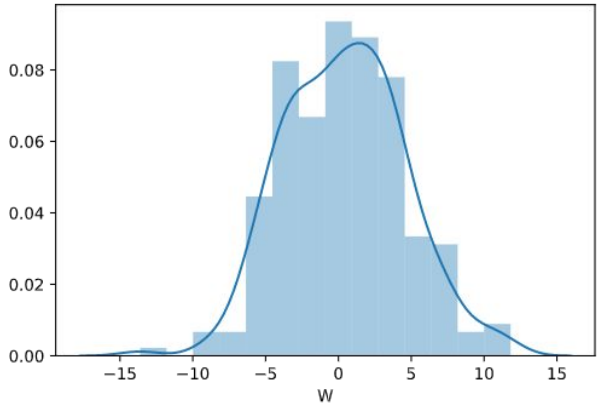
```
RS: 0.44484             ... Runs Scored
RA: 0.44004             ... Runs Allowed
OBP: 0.04944            ... On-Base Percentage
SLG: 0.02558            ... Slugging Percentage
BA: 0.01966             ... Batting Average
G: 0.00463              ... Games Played
OOBP: 0.00910           ... Opponent On-Base Percentage
OSLG: 0.00670           ... Opponent Slugging Percentage
```

Feature	Score	Comment
RS	0.4448	Runs Scored
RA	0.44	Runs Allowed
OBP	0.0494	On-Base Percentage
SLG	0.0256	Slugging Percentage
BA	0.0197	Batting Average
G	0.0046	Games Played
OOBP	0.0091	Opponent On-Base Percentage
OSLG	0.0067	Opponent Slugging Percentage

Data set experiments - Random Forest

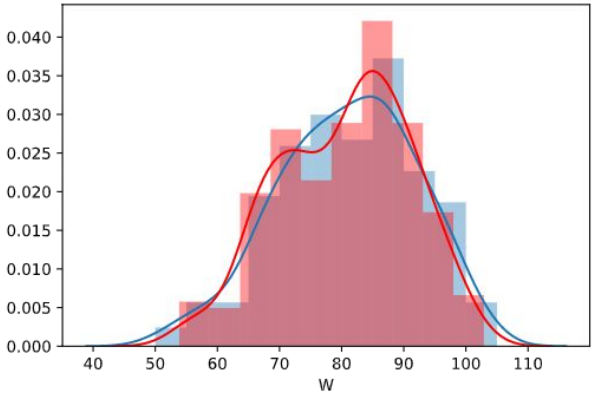


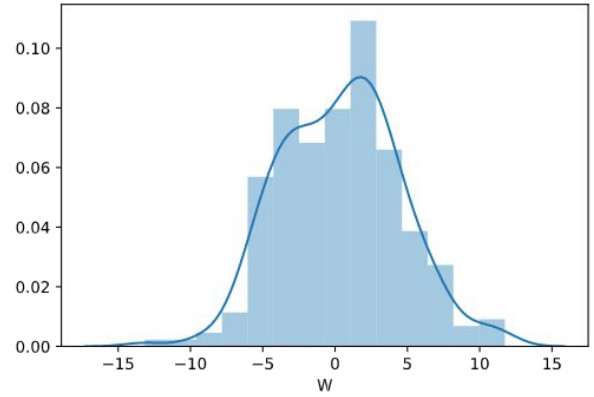
Test Data Set vs. Model Prediction



Difference between the model and the data set

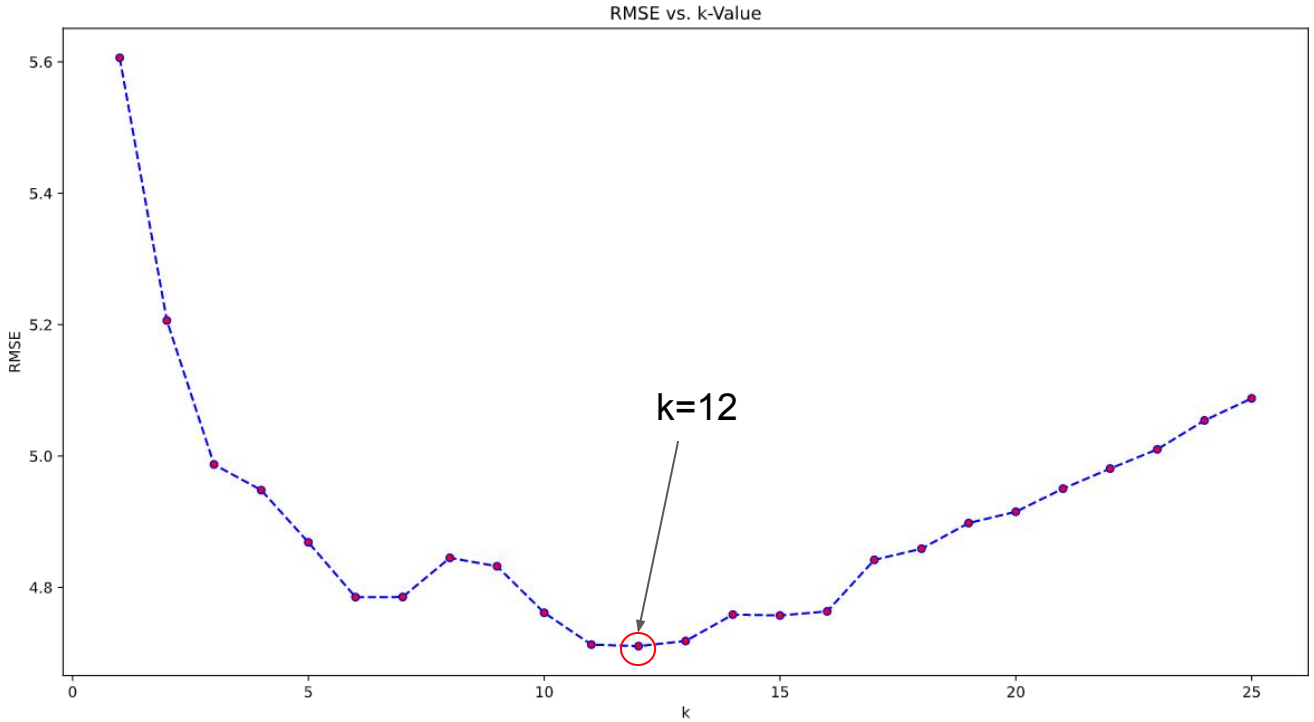
Number of trees = 100
RMSE: 4.16410
R^2-score: 0.8597059





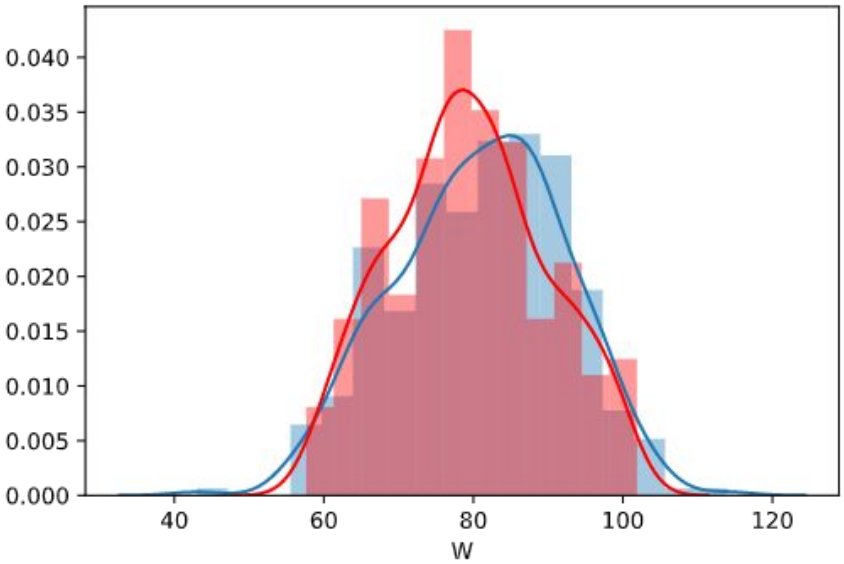
Number of trees = 1000
RMSE: 4.14277
R^2-score: 0.8611396

Data set experiments - k-Nearest Neighbors

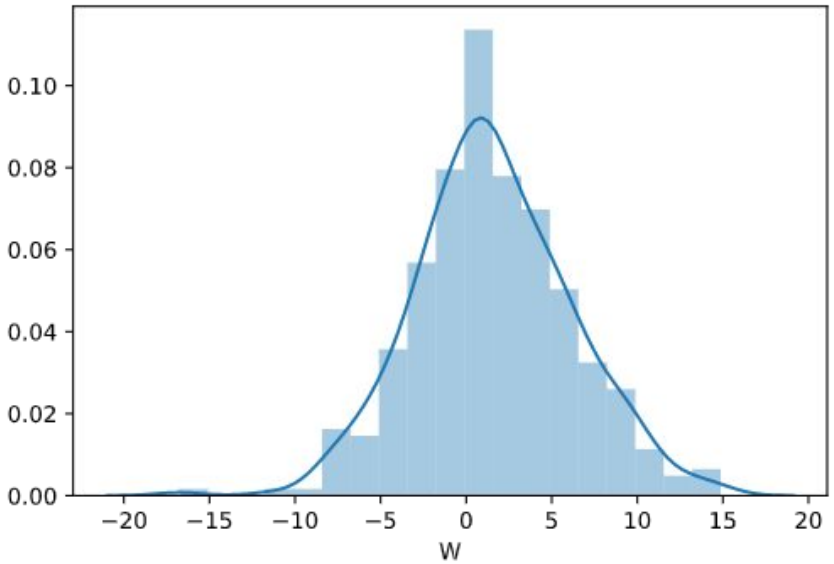


Initially, using the elbow method and trying k-values, we found out the optimal one for the data set.

Data set experiments - k-Nearest Neighbors



Test Data Set vs. Model Prediction



Difference between the model and the data set

k-Value = 12:

RMSE: 4.6441541929794035
R^2-score: 0.8256440631335493

Data set experiments

The Gridsearch performed on the random forest model returned k=13 as the best k-Value for this data set. We interpreted that as a similar situation as with the Random Forest where the initial model was already as close to best as possible and an increase in the trees by tenfold didn't bring much improvement.
In conclusion, the different models yielded the following overall results:

Linear Regressor:

- RMS(test): 3.8302981047825244,
- R2(test): 0.8877439871498782

Almost identical regression models

Lasso Regressor:

- RMS(test): 3.8302981016004654,
- R2(test): 0.8877345961507863

Best regression models for this data set

kNN Regressor:

- RMS(test): 4.6441541929794035,
- R2(test): 0.8256440631335493

Random Forest Regressor:

- RMS(test): 4.142770563194546,
- R2(test): 0.8611396757772145

Appliances Energy Prediction

Appliance energy consumption dataset

Data Set description: The dataset contains data from measurements taken within a low energy house, specifically temperature and humidity measurements by room (1-9), energy consumption of lights, and outside weather data. Everything is presented as averages across ten minute intervals of time and has previously been cleaned up by the datasets hosts - no missing or anomalous values, such as impossible or unreasonable temperatures, are present. The datasets purpose is to be able to predict the energy consumption of appliances in/of the house (recorded in the column "Appliances").

Instances: 19735, representing ~137 days (144 samples per day) - high number

Dimensions/Features: 29 (including the target and 2 random variables) - also quite high

Missing values: None

Target: The energy consumption of appliances in the house (column “Appliances”)

Features

- 'date': in format (time year-month-day hour:minute:second)
- 'Appliances': energy use in Wh, ***TARGET***
- 'lights': energy use in Wh,
- Room data: ‘T_i’ (temperature,°C) and ‘RH_i’ (humidity,%) measurements for each room. Rooms are [kitchen, living, laundry, office, bathroom, outside*, ironing, teenager, parents] with indices 1...9,
- Weather data: from nearby weather station, 'T_out'*, 'Press_mm_hg', 'RH_out', 'Windspeed', 'Visibility', 'Tdewpoint',
- Random variables: rv1', 'rv2'

*Note: house measurements for outside also included → correlates strongly with ‘T_out’

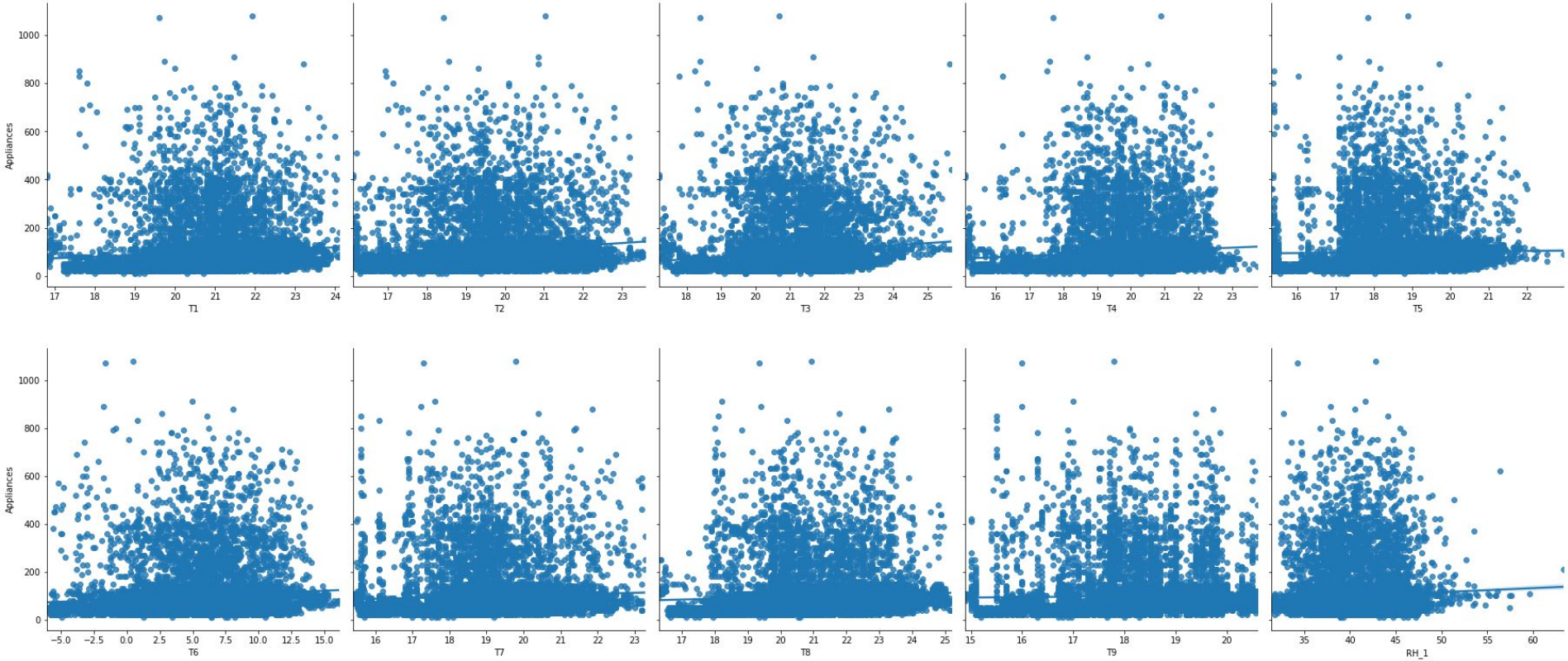
Derived features

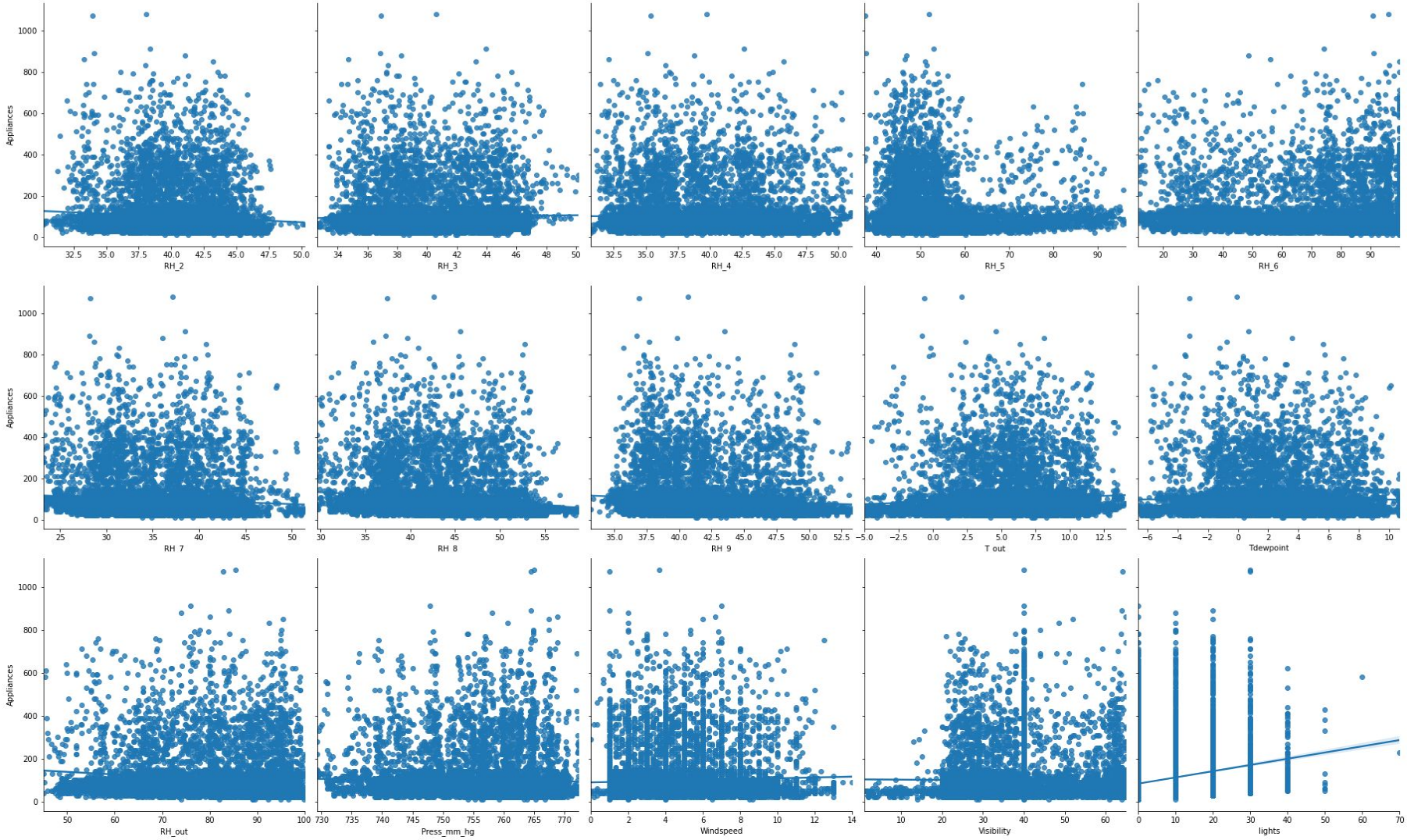
- Threshh: binning of the energy consumption (“Appliances”) into the 4 quartiles. Only used for visualizations (not in slides), for obvious reasons!
- Daytime/Daytime_str: split each day into 4 times of day.
  The idea was to gain information about human patterns encoded into the combination of time of day and energy consumption. The split into the 4 times of day is somewhat arbitrary.

Time	daytime_str	daytime
06:00 - 11:59	morning	0
12:00 - 14:59	midday	1
15:00 - 19:59	afternoon	2
20:00 - 05:59	night	3

Linear regression plots:

(only ~½ of the instances for better readability)

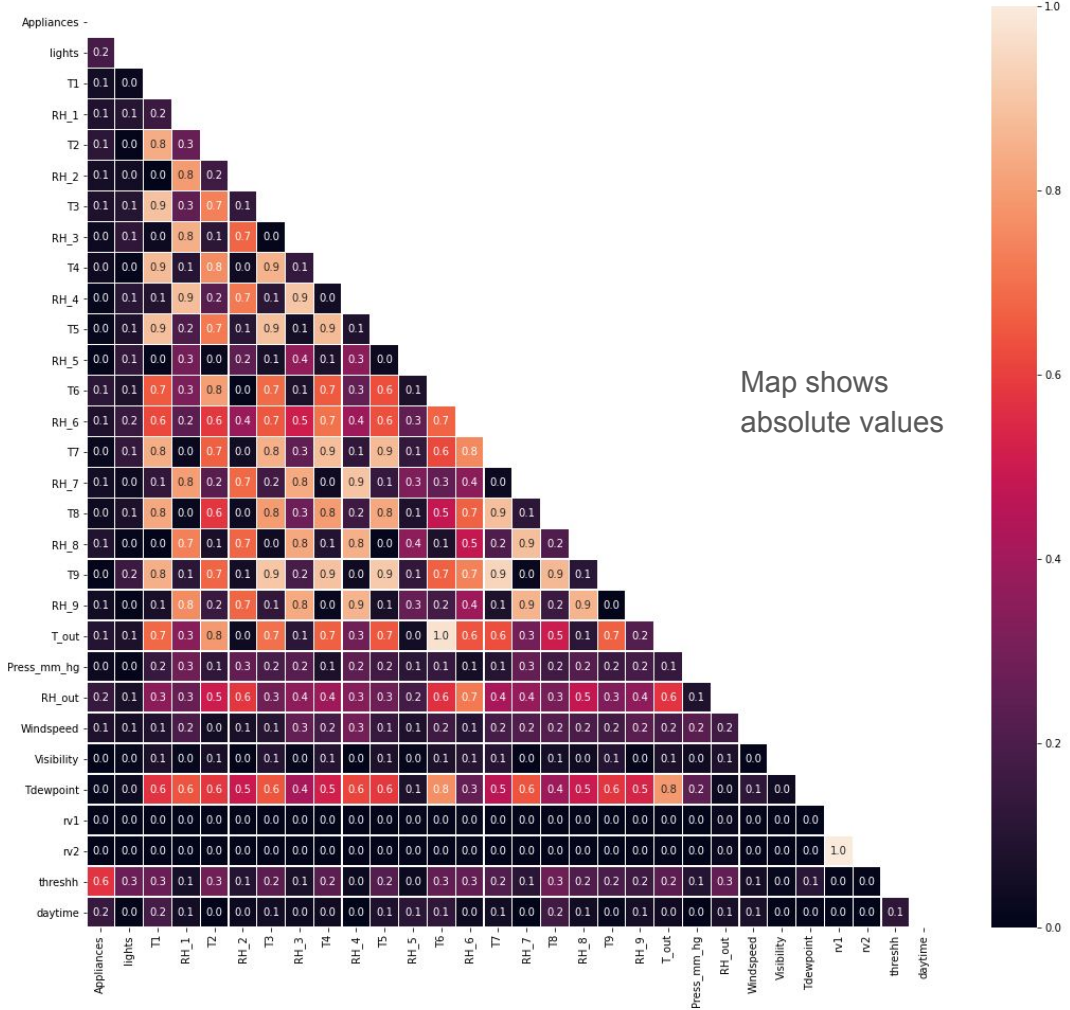



Appliances
RH_2
RH_3
RH_4
RH_5
RH_6
RH_7
RH_8
RH_9
T_out
Tdewpoint
RH_out
Press_mm_hg
Windspeed
Visibility
lights

Correlation heat map

- Standout corrs: RH_out, T_out, lights, T6, T2 (>0.1) and the derived daytime and threshh
- Temps and humidity of the same room correlate as expected
- T9 correlates highly with the other Ti
- T6, T9 correlate highly with the other Ti; T6 also with T_out (0.97)
- Visibility and Tdewpoint are not worth it (low corr with target + high corr with other features)
- The random variables do not correlate, as expected
- Daytime seems to correlate well

We therefore decided to always drop: ['date', 'rv1', 'rv2', 'Visibility', 'Tdewpoint'] (Set1) and created a second set (Set2) where we also drop: ['T6', 'T9'] to see if that has an impact on the scores.



Map shows absolute values

Conclusion of data analysis

- No apparent strong linear relationships between any of the features and the target: Linear and Lasso not expected to work particularly well.
- There are correlations between some features and the target
- kNN and Random Forest will probably work better, as they can fit non-linear/more complex relationships better
- There are some features that can be dropped because they do not actually contain any meaningful information about the target (‘Visibility’ and ‘Tdewpoint’].
- Some features correlate with each other → feature set can be reduced without any or much loss of information → “leaner” models possible

Pre-processing

Pre-processing of data sets:

- Appliance Energy Consumption data set:
  - Added derived feature “daytime” instead “date”: 4 bins=[0,3]=[“morning”, “midday”, “afternoon”, “night”] → input for models
  - Derived features “daytime_str” and “threshh” (energy consumption into bins by quartil) only used for visualization
  - Set1: Dropped ['date', 'rv1', 'rv2', 'Visibility', 'Tdewpoint'] due to low correlation with target=”Appliances” (energy consumption)
  - Set2: Dropped ['T6', 'T9'] aswell as the above due to good correlation with other features (=no information gain)
  - All data was either numeric or datetime to begin with.
  - A standard scaler was used on all columns

Best model for this set?

Linear Regressor:

- RMS(train): 0.9091848811402113,
- RMS(test): 0.8791985912456376,
- R2(train): 0.18314455200158364,
- R2(test): 0.1880996369544411

kNN Regressor:

- RMS(train): 0.4042780730865147,
- RMS(test): 0.6941401902798184,
- R2(train): 0.8384893467936548,
- R2(test): 0.49391536797945856

Lasso Regressor:

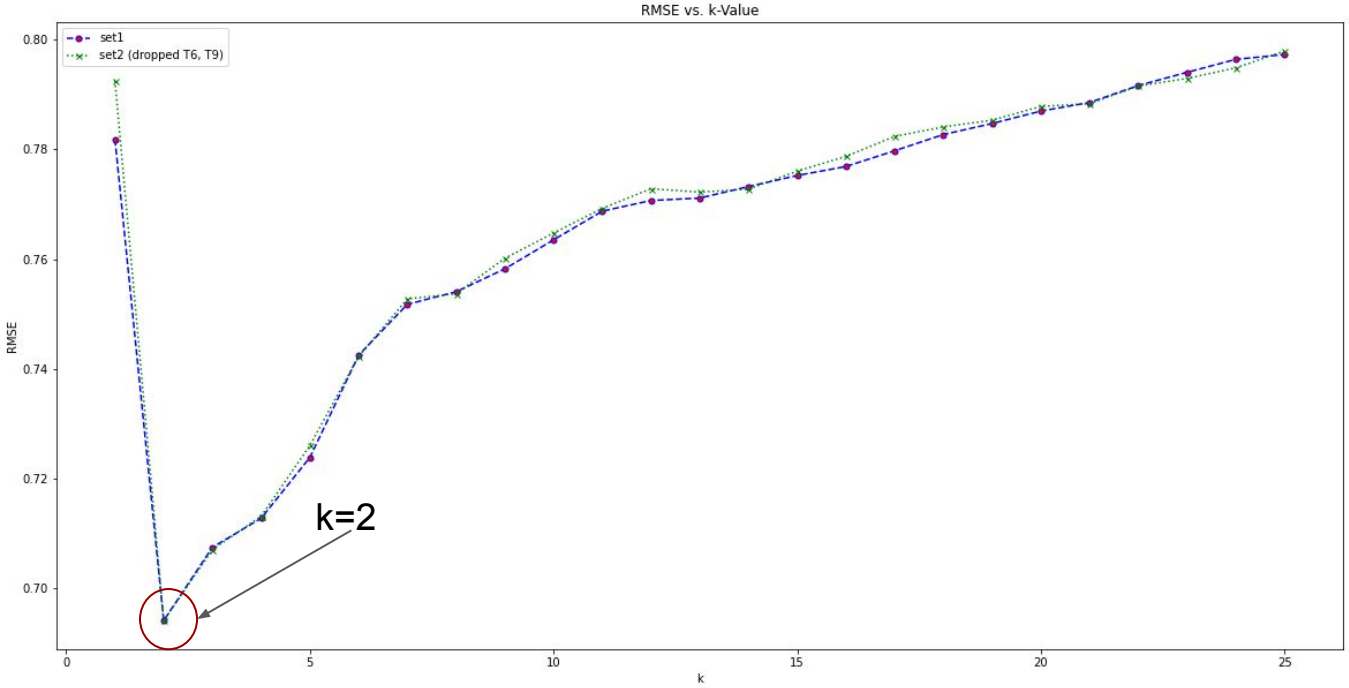
- RMS(train): 0.9091848814383682,
- RMS(test): 0.8791979841629626,
- R2(train): 0.18314455146582664,
- R2(test): 0.1881007581811559,

Random Forest Regressor:

- RMS(train): 0.2428459669193612,
- RMS(test): 0.6334570066832834,
- R2(train): 0.9417222749625794,
- R2(test): 0.578533503576818

The random forest model seems to perform the best for this task (lowest RMS error, highest R2-score on the test sets). As expected, the linear and Lasso regressors do not perform well - the data did not “look” to be very linear.

kNN - number of neighbors



k=2 seems to be the best number of neighbors here.

Feature importance (Random Forest)

Feature	Score	Comment
lights	0.0395	energy use of light fixtures in the house in Wh
T1	0.0298	Temperature in kitchen area: in Celsius
RH_1	0.0434	Humidity in kitchen area: in %
T2	0.035	Temperature in living room area: in Celsius
RH_2	0.0428	Humidity in living room area: in %
T3	0.0652	Temperature in laundry room area
RH_3	0.0475	Humidity in laundry room area: in %
T4	0.0316	Temperature in office room: in Celsius
RH_4	0.0364	Humidity in office room: in %
T5	0.0326	Temperature in bathroom in Celsius
RH_5	0.0505	Humidity in bathroom in %
T6	0.0367	Temperature outside the building (north side: in Celsius
RH_6	0.0362	Humidity outside the building (north side): in %
T7	0.0349	Temperature in ironing room: in Celsius
RH_7	0.0348	Humidity in ironing room in %
T8	0.0391	Temperature in teenager room 2 in Celsius
RH_8	0.0417	Humidity in teenager room 2 in %
T9	0.0373	Temperature in parents room in Celsius
RH_9	0.0423	Humidity in parents room: in %
T_out	0.0338	Temperature outside (from Chievres weather station): in Celsius
Press_mm_hg	0.0451	from Chievres weather station) in mm Hg
RH_out	0.0426	Humidity outside (from Chievres weather station): in %
Windspeed	0.0333	(from Chievres weather station) in m/s
daytime	0.0878	time of the day (extracted from date), as integer: (0...morning, 1...midday, 2...afternoon, 3...night)

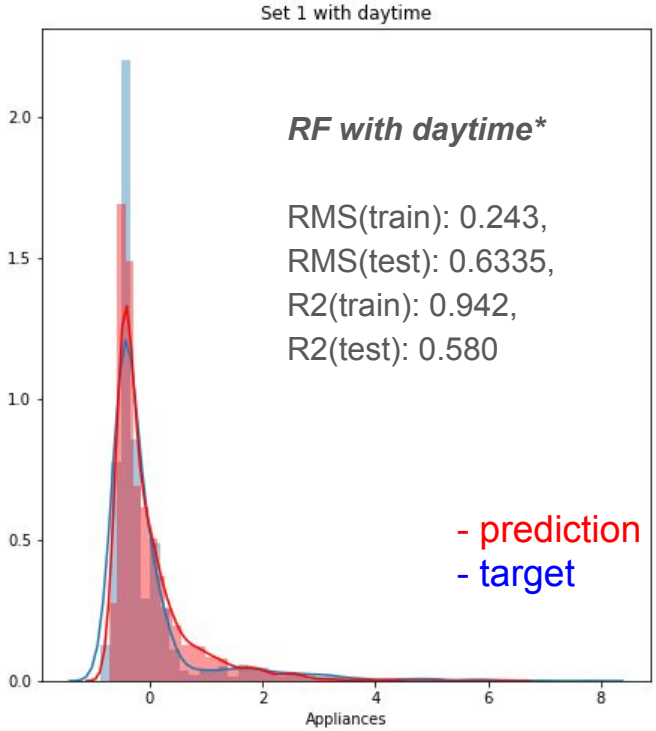
Feature importance (Random Forest)

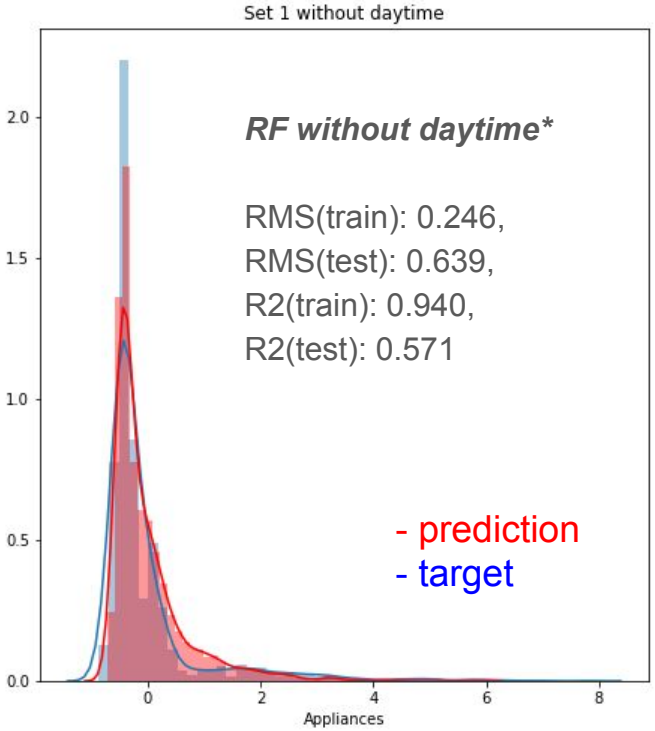
The derived feature daytime seems to very valuable to the random forest (highest importance score) - but it only very slightly improves the final prediction (next slide).

Surprisingly, the highly correlated values (heat map) do not have particularly high importance scores (T6, T_out, lights)

T3, that did not stand out in the correlation heat map, has a high importance score.

Random Forest: w/ vs w/o daytime

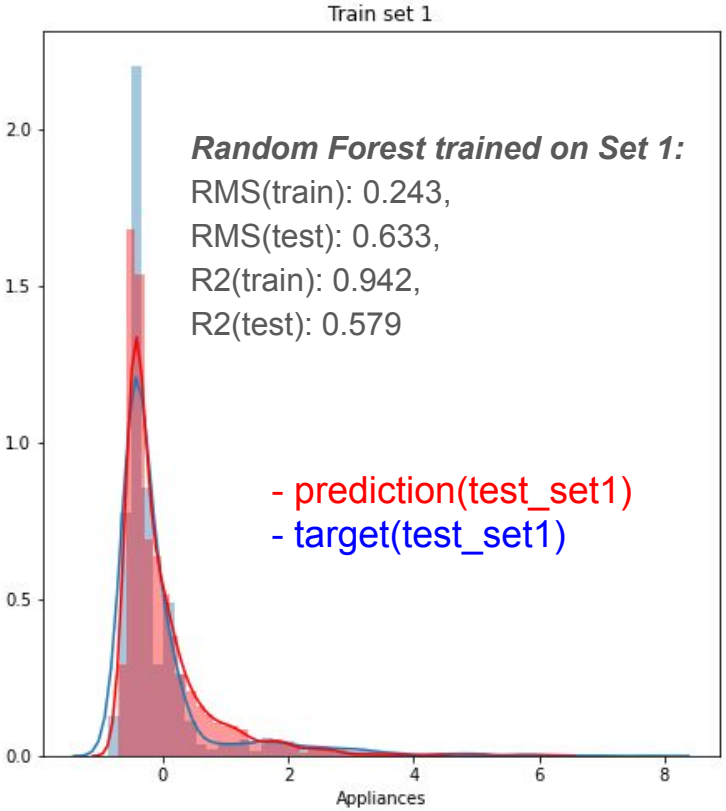




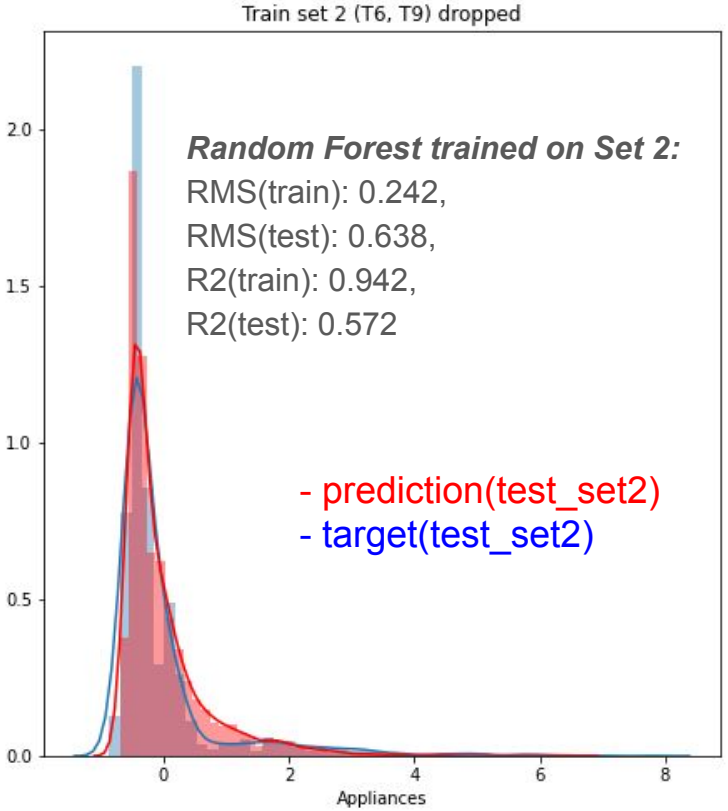
The inclusion of the derived daytime feature seems to have a very slight positive effect (if at all).

**Both RF-Regressors were trained with the same hyperparameters: n_estimators=500, max_features=0.5, min_samples_leaf=1,*

Set 1 vs Set 2:

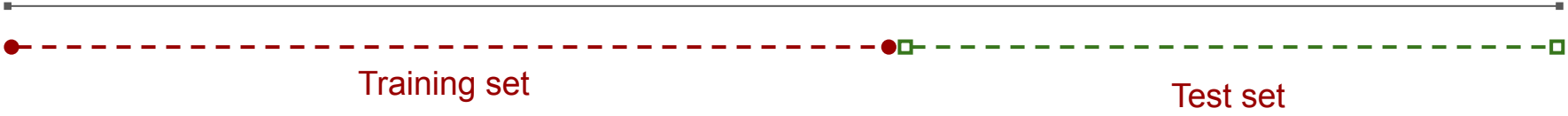


Set2 = Set1 - {T6, T9}



Train/test split by time

Idea: The dataset was gathered over a series of ~137 days and is given in temporal order. What happens, when the dataset is split into train/test sets based on the date(-time), instead of the usual randomized spl?



Conclusion: (Results on next page)

- The linear and lasso regression are almost unaffected by the different train/test-split.
- However, the kNN and random forest models seem to overfit the training data, or do generalize well, when given the time split. This can be seen in the huge drops in the RMS and R2 scores when going from the training to the test set.

Train/test split by time: Scores

Linear Regression

- RMS(train): 0.922,
- RMS(test): 0.837,
- R2(train): 0.193,
- R2(test): 0.111

Roughly same as before

Lasso Regression

- RMS(train): 0.922,
- RMS(test): 0.837,
- R2(train): 0.193,
- R2(test) 0.111

kNN:

- RMS(train): 0.401,
- RMS(test): 1.386,
- R2(train): 0.847,
- R2(test): -1.436

horrible!

Random Forest

- RMS(train): 0.228,
- RMS(test): 1.728,
- R2(train): 0.951,
- R2(test): -2.788

Metro Interstate

Metro Interstate Traffic Volume Data Set

Dataset description: It contains weather data (hourly) and the count of traffic (hourly) at a certain point in the USA.

Dimensions: high number of Instances 48204 and low number of attributes 9

Missing values: none but some values are wrong for example temperature (0 Kelvin)

Target: traffic_volume(hourly)

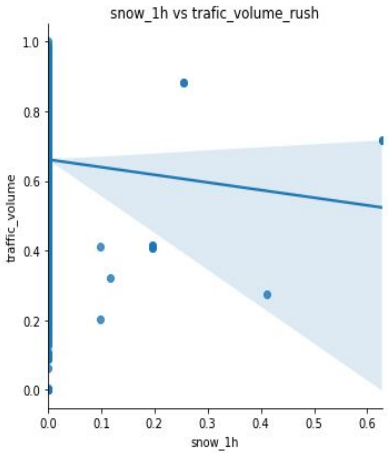
Characteristics of data sets & pre-processing

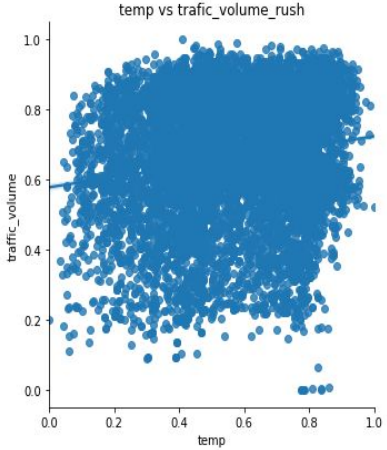
Pre-processing of data sets:

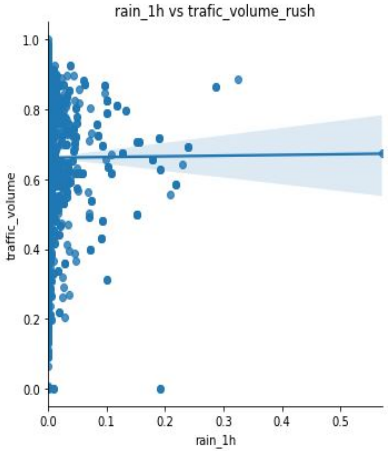
- Metro Interstate Traffic Volume:
  - Converted the date_time column into 4 different columns (year, month, day and hour)
  - Categorized the two columns for the weather description “weather_main” and “weather_description”
  - Seperated the raw_dataset into a new dataset which contains only the rows (8,9,10,16,17 hour)

Data set experiments

- **Linear Regression experiment:**
  - Find the variables which have a good linear relationship
  - Scale the dataset is not important for the linear regression but it is then easier to compare the different methods.



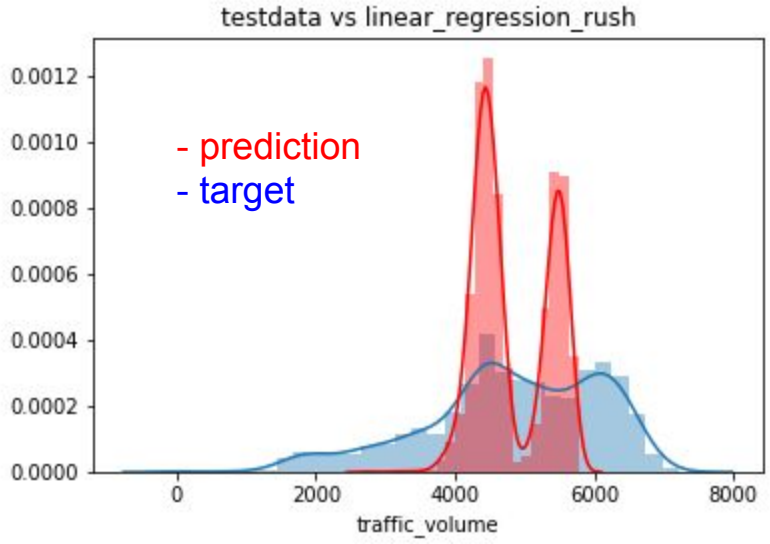




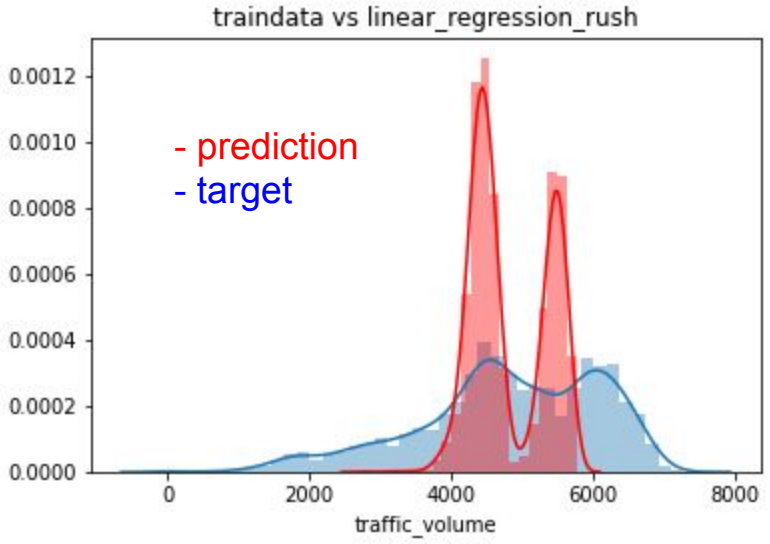
Some input variables for the training for the linear regression model. They have a bad linear relationship. Expect a bad performance of the model.

Data set experiments - Linear Regression

RMS(test): 1165.2671
R2(test): 0.15466606



RMS(train): 1167.82
R2(train): 0.170502



Data set experiments - Lasso Regression

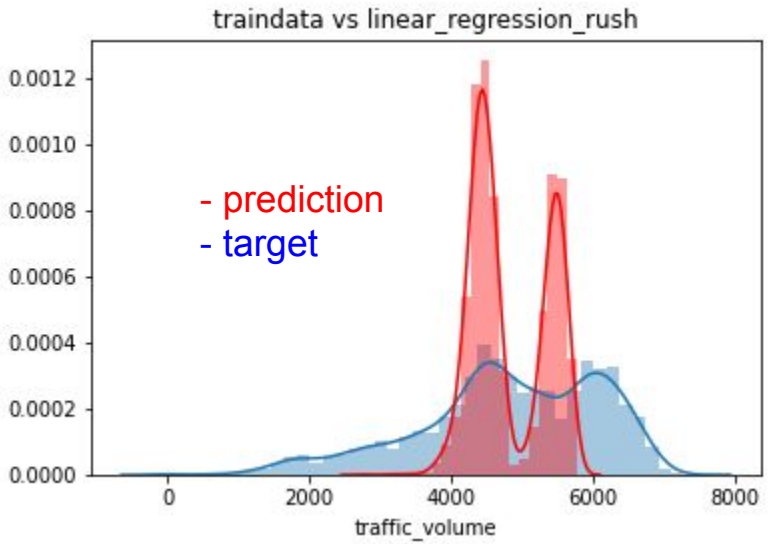
The algorithmus choose as alpha 0.01 and the score for

RMS = 1165.1331

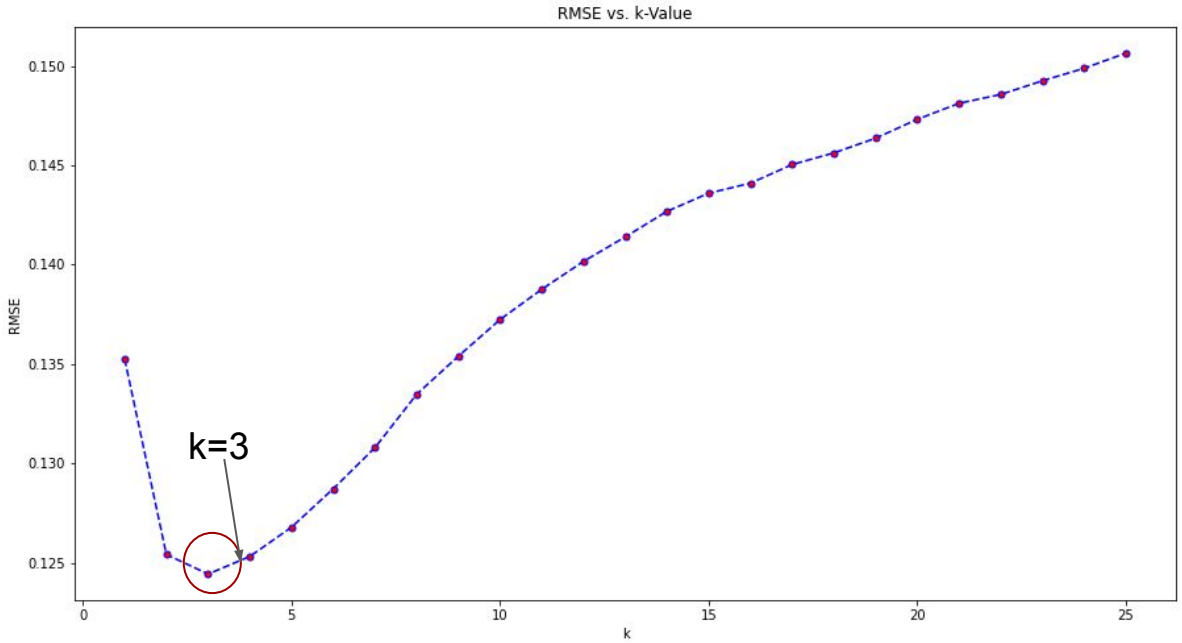
In the end both methods perform

similarly because of the

nonlinear behavior form the variables.



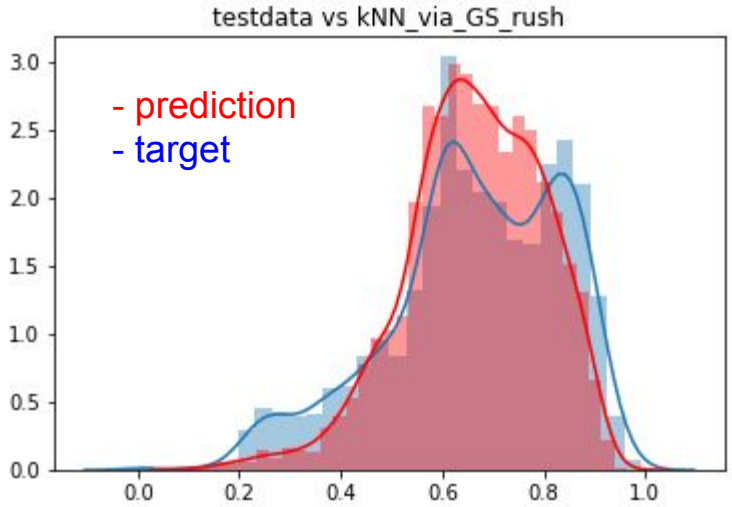
Data set experiments - kNN



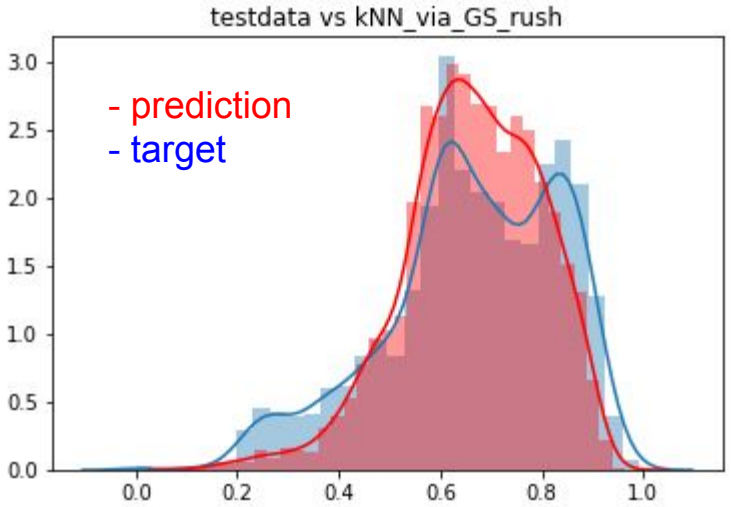
For k = 3 the best amount of neighbours

Data set experiments - kNN with Gridsearch

RMS(train): 0.080931444539265 03
R2(train): 0.78886743804805



RMS(test): 0.12443281439874476
R2(test): 0.48913056725980353



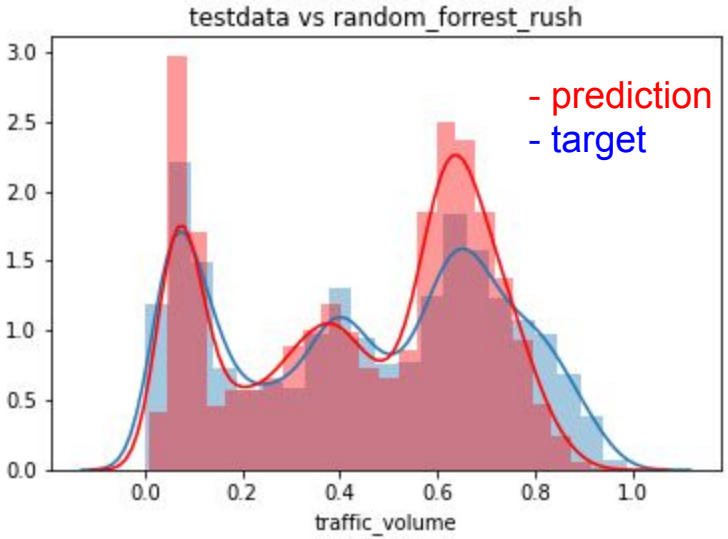
Feature importance (Random Forest)

Feature	Score	Comment
holiday	0.0	Name of the holiday (object)
temp	0.0765	temperature[Kelvin]
rain_1h	0.0062	Amount of rain in 1 hour[mm]
snow_1h	0.0	Amount of snow in 1 hour[mm]
clouds_all	0.0158	percentage of clounds
weather_main	0.0085	discribe the current wheather situation
weather_description	0.0143	more precise description of the weather situation
hour	0.781	hour
day	0.0516	day
month	0.0241	month
year	0.022	year

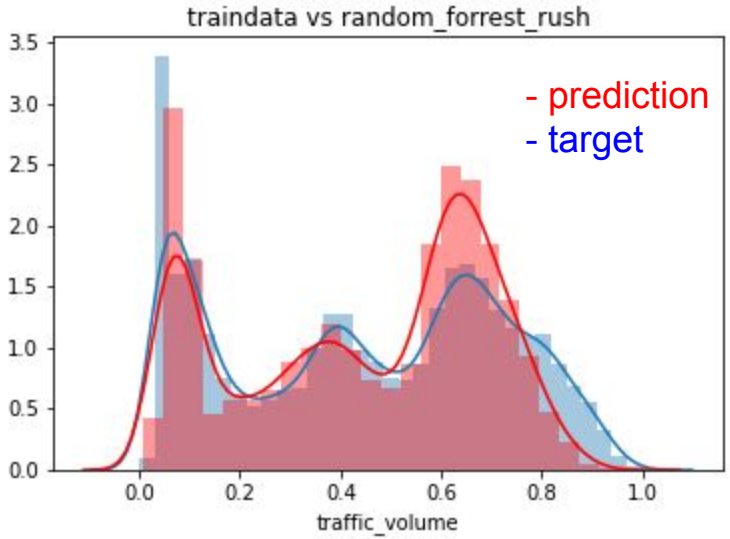
Feature importance (Random Forest)

n_estemators = 100

RMS(train): 0.0407606
R2(train): 0.97768529



RMS(test): 0.108789863
R2(test): 0.8413440405



Feature importance (Random Forest) summary

- Default mode n_estemators = 100
  - RMS(test): 0.10849827390563874
  - R2(test): 0.8421933922768938

- Optimize Hyperparameters via GridSearch n_estemators = 100
  - RMS(test): 0.101910706041439 59
  - RMS(Test): 0.101910706041439 59

Data set experiments

The Gridsearch performed on the random forest model returned k=3 as the best k-Value for this data set. We interpreted that as a similar situation as with the Random Forest where the initial model was already as close to best as possible.
In conclusion, the different models yielded the following overall results:

Linear Regressor:

- RMS(test): 0.1600955848899509
- R2(test): 0.15433417518307213

Lasso Regressor:

- RMS(Test): 0.16009537812514907
- R2(Test): 0.025630530097034462

kNN Regressor:

- RMS(test): 0.12443281439874476
- R2(test):0.48913056725980353

Random Forest Regressor:

- RMS(test): 0.10849827390563874
- R2(test): 0.8421933922768938

Best regression model for this data set.

Temperature Forecast

Bias correction of numerical prediction model temperature forecast Data Set

Data Set description: This data is for the purpose of bias correction of next-day maximum and minimum air temperatures forecast of the LDAPS model operated by the Korea Meteorological Administration over Seoul, South Korea. This data consists of summer data from 2013 to 2017. There are two outputs (i.e. next-day maximum and minimum air temperatures) in this data.

Instances and dimensions: The Data Set has a mid number of instances: 7750 and a relatively high number of dimensions: 25 (including the 2 output columns).

Missing values: There are several rows with missing values, approximately 220.

Target: The target variable chosen for the task is: Next_Tmin.

Characteristics of data sets & pre-processing

Pre-processing of data sets:

- Bias correction of temperature forecast data set: Firstly, several columns were dropped at the beginning. Those were:
  - “station”, due to the fact that it contains simply IDs which didn’t bring much information since those stations are already presented through the latitude and longitude columns;
  - “Date”, because we considered that it wouldn’t bring any valuable information for the regression models;
  - “Next_Tmax” since we decided to use the other output variable

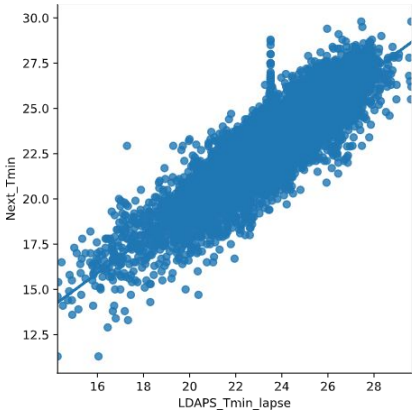
  Secondly, the missing values (mostly within the “Present_Tmax”, “Present_Tmin” columns) were imputed with the mean-values of those columns

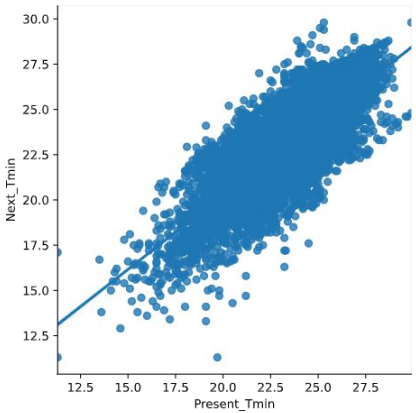
Data set experiments

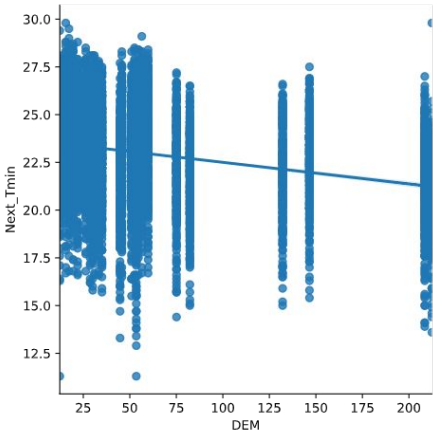
- **Linear Regression experiment:**
  - Find whether there's a strong correlation between some of the parameters and the target variable.
  - Therefore, after preprocessing the data set, we plotted some of parameters alongside the Wins-variable and got the following results.
    - There's a very strong correlation between the LDAPS Tmin and Present day Tmin which is natural.
    - We were also considering that the Relative Humidity might have a somewhat important role but that proved to not be the case. The rest of the variables didn't have a clear influence on the next day minimum temperature as well .

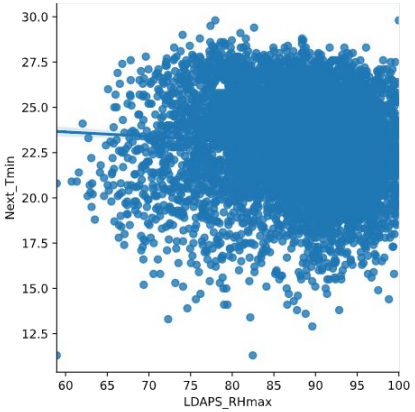
    On the next slide are plots to describe our findings.

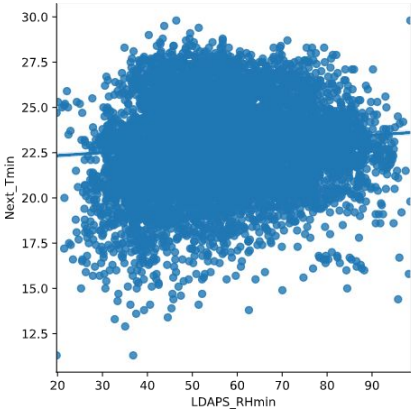
Data set experiments







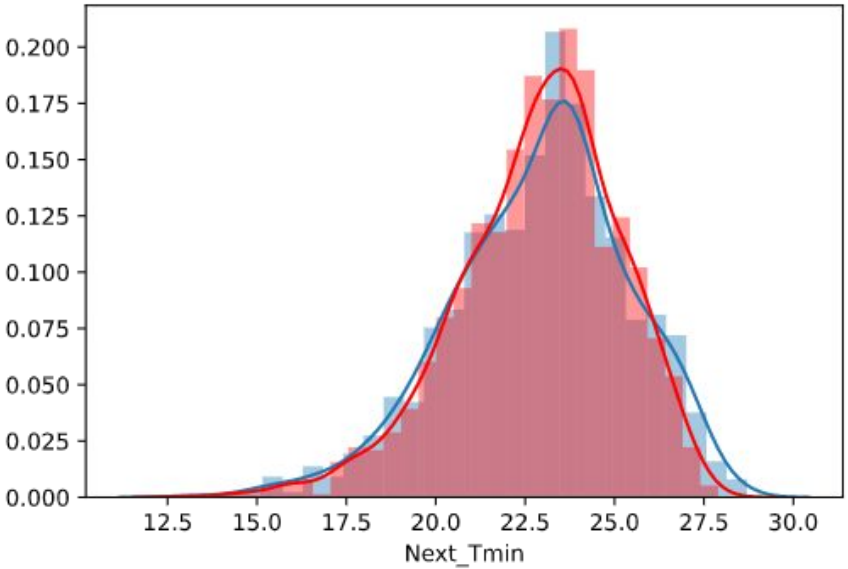




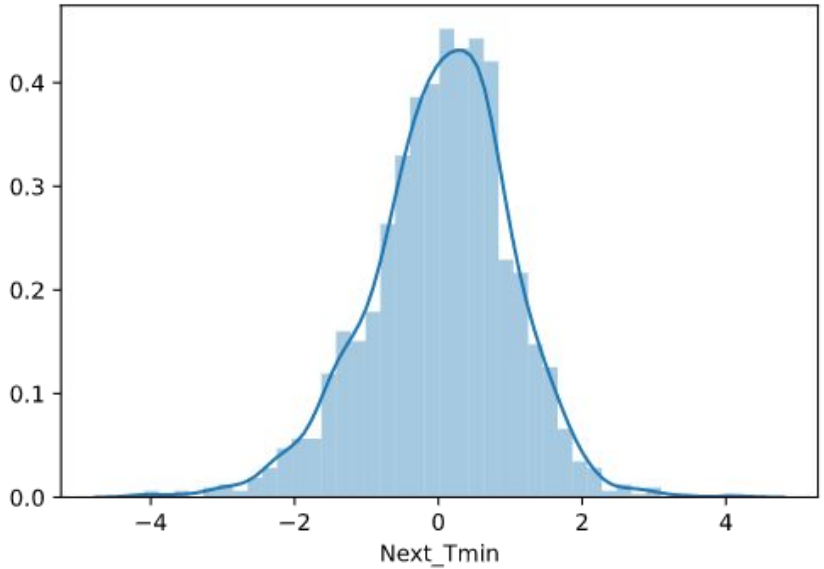
Data set experiments - Linear Regression

- Continuing with the Linear Regression experiment, we used separated the data in a train/test split with the test size being 30% of the whole data set and achieved the following results using the scikitlearn-Linear Regression Model:
    - Root Mean Squared Error: 0.9818378084004215
    - Model R^2-Score: 0.8453046823462249
- Those results were interpreted as very good. To visualize the model performance, we plotted the test data set with alongside the results predicted by the model and the difference between the two.

Data set experiments - Linear Regression



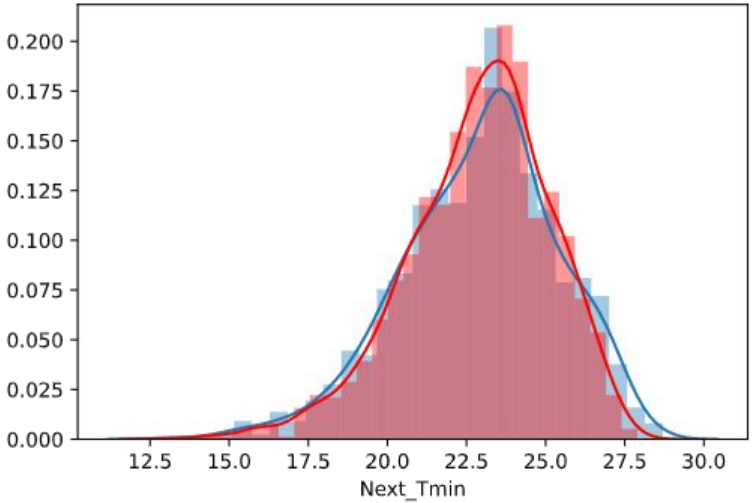
Test Data Set vs. Model Prediction



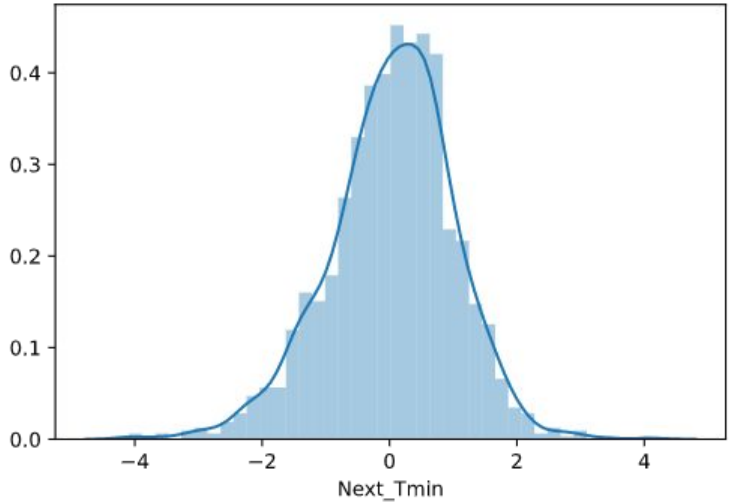
Difference between the model and the data set

Data set experiments - Lasso Regression

Based on the good performance of the linear regression model, we were quite sure that the lasso regression model would deliver good if not the same results and after trying alpha paramerest in the range from 1e-15 to 100, we found out that the best alpha-value is: 1e-8 with the following results:



Test Data Set vs. Model Prediction



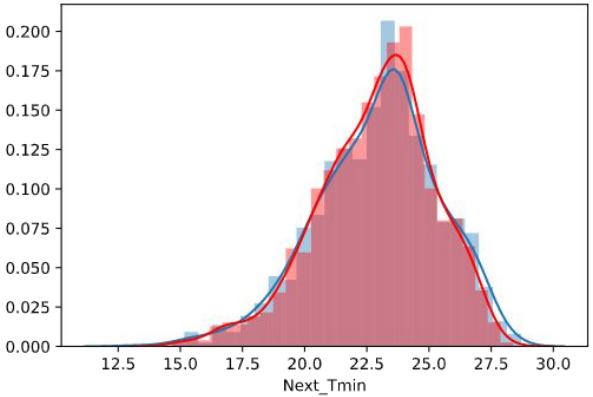
Difference between the model and the data set

Data set experiments - Random Forest

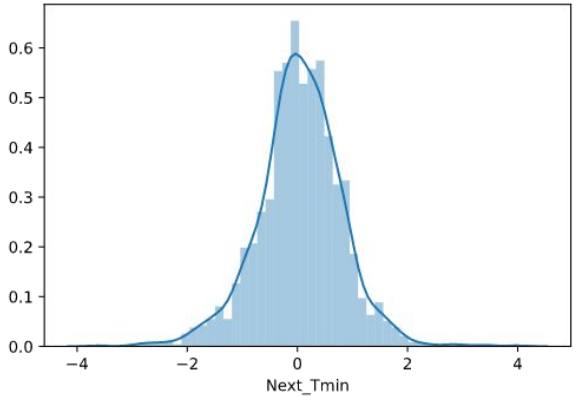
Es ist sichtbar, dass die “wichtigsten” Variablen aus der linearen Regression auch hier hohes Feature Importance haben.

Feature	Score	Comment
Present_Tmax	0.0164	2
Present_Tmin	0.0621	3
LDAPS_RHmin	0.0091	4
LDAPS_RHmax	0.0083	5
LDAPS_Tmax_lapse	0.0079	6
LDAPS_Tmin_lapse	0.7683	7
LDAPS_WS	0.0137	8
LDAPS_LH	0.007	9
LDAPS_CC1	0.012	10
LDAPS_CC2	0.0095	11
LDAPS_CC3	0.0104	12
LDAPS_CC4	0.013	13
LDAPS_PPT1	0.0053	14
LDAPS_PPT2	0.0045	15
LDAPS_PPT3	0.0029	16
LDAPS_PPT4	0.0042	17
lat	0.0072	18
lon	0.0081	19
DEM	0.0089	20
Slope	0.0076	21
Solar radiation	0.0136	22

Data set experiments - Random Forest

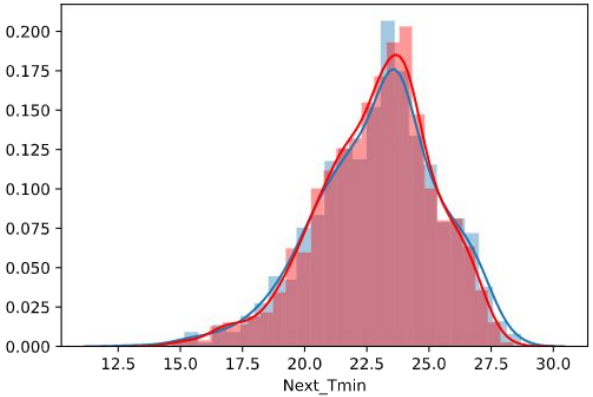


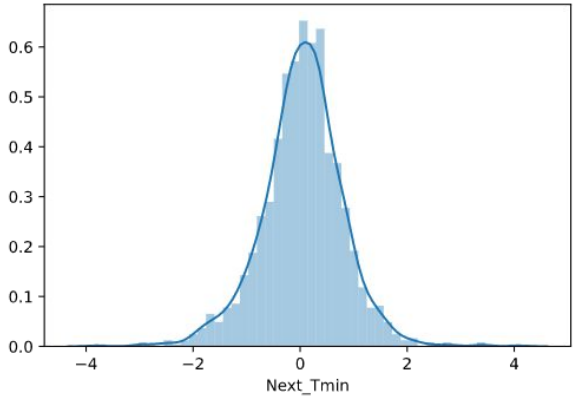
Test Data Set vs. Model Prediction



Difference between the model and the data set

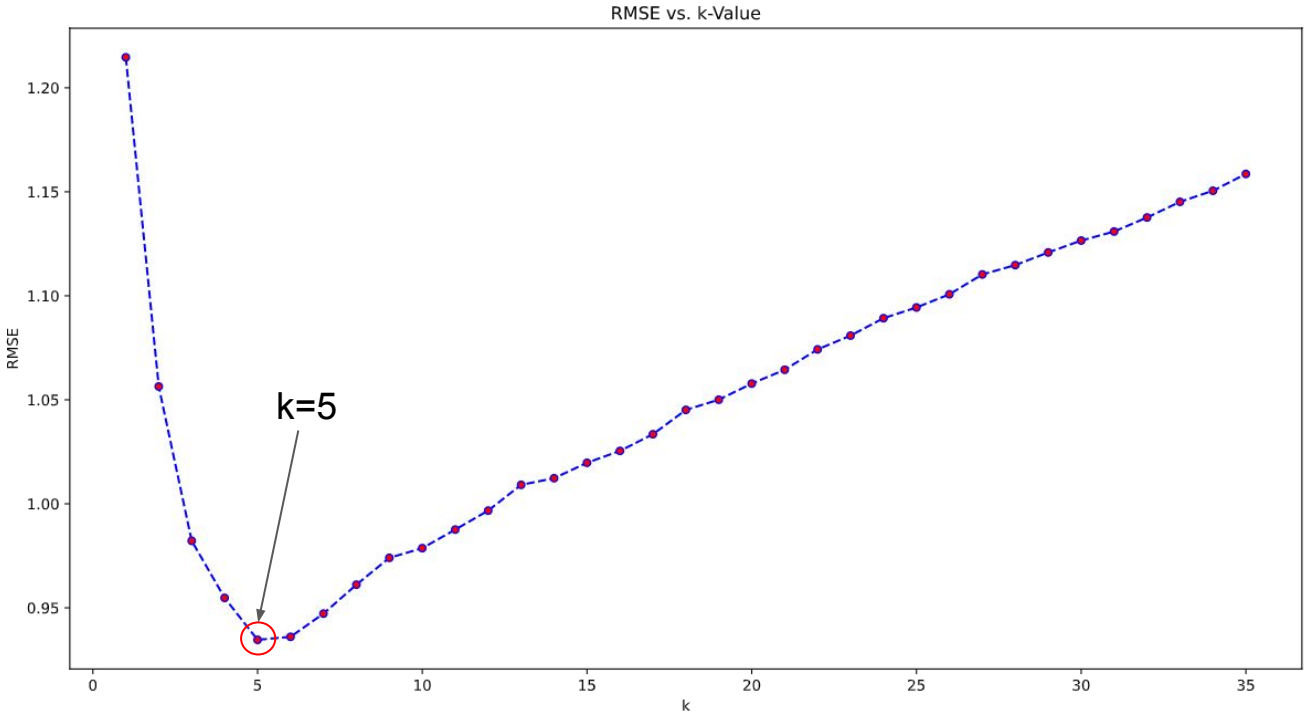
Number of trees = 100
RMSE: 0.7595007
R^2-score: 0.9074334





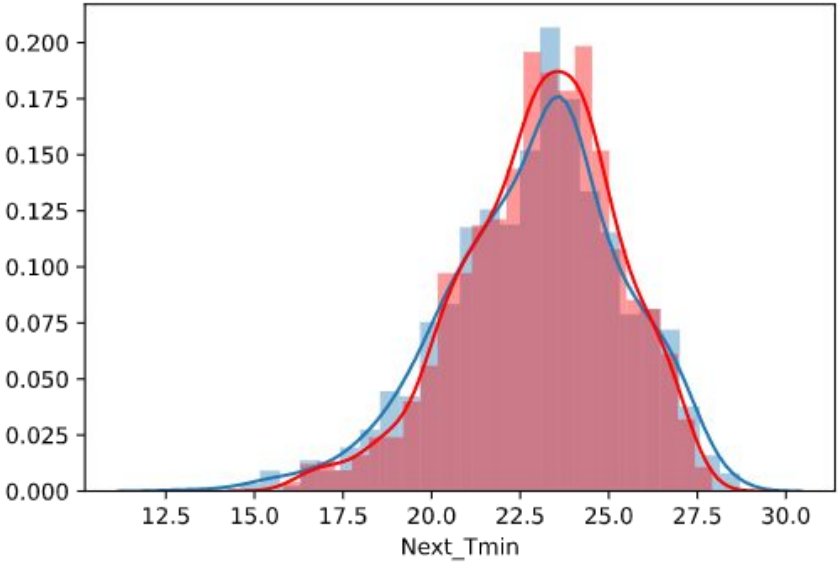
Number of trees = 500
RMSE: 0.7595007
R^2-score: 0.9074334

Data set experiments - k-Nearest Neighbors

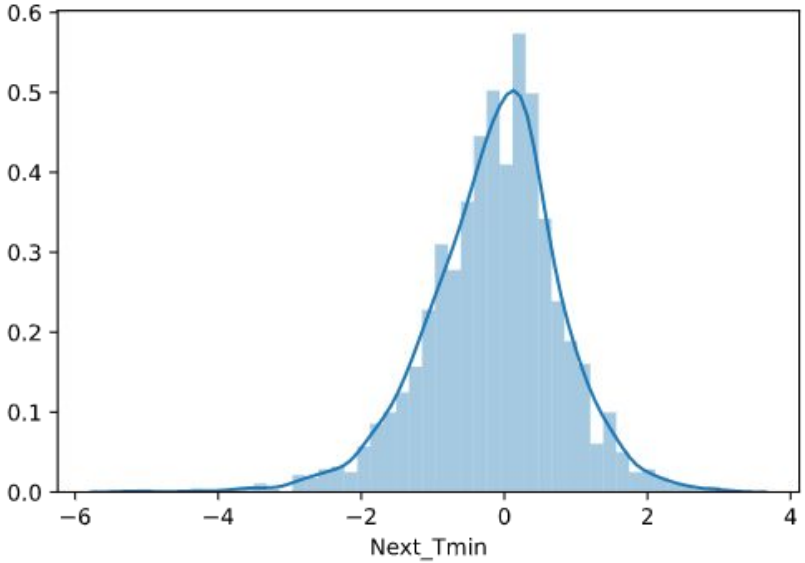


Initially, using the elbow method and trying k-values, we found out the optimal one for the data set.

Data set experiments - k-Nearest Neighbors



Test Data Set vs. Model Prediction



Difference between the model and the data set

k-Value = 5:

RMSE: 0.9345947103920778
R^2-score: 0.859833476332077

Data set experiments

The Gridsearch performed on the random forest model returned k=6 as the best k-Value for this data set. We interpreted that as a similar situation as with the Random Forest where the initial model was already as close to best as possible.
In conclusion, the different models yielded the following overall results:

Linear Regressor:

- RMS(test): 0.9818378084004215,
- R^2(test): 0.8453046823462249

Lasso Regressor:

- RMS(test): 0.9818375978966962,
- R2(test): 0.84621570835671161,

kNN Regressor:

- RMS(test): 0.9345947103920778,
- R2(test): 0.859833476332077

Random Forest Regressor:

- RMS(test): 0.7595007001970385,
- R2(test): 0.9074334618330312

Best regression model for this data set.