Doing More with Less - 1

Masters Thesis

Applied Data Science, Utrecht University

Varoon Sushil Agrawal

Utrecht, July 3, 2022

Information Page Graduation Report

Utrecht University
Heidelberglaan 8, 3584 CS Utrecht, Netherlands

Master Thesis

Name of student:	Varoon Sushil Agrawal
Student number:	2306557
Course:	Masters in Applied Data Science, Utrecht University
Period:	April 2022 - July 2022
External company name:	Intergas Verwarming B.V.
Address:	Europark Allee 2
Postcode, City:	7742 NA Coevorden
Country:	Netherlands
External supervisor:	Erwin Bisschop
Email:	erwin@inversable.com
University supervisor:	Arno Siebes
Email:	a.p.j.m.siebes@uu.nl
First Examiner:	Arno Siebes
Second Examiner:	Ad Feelders
Non-disclosure agreement:	Yes
Number of words:	5026

NDA and Working Arrangements

for Students working with the Intergas Data

1. You have your own clean spark environment (no other users are using it). After your research is done, we will destroy your access to the environment (but results will be kept on the server).
2. Do not copy data from the server to your local device but do all processing on server. Datasets are intellectual property of Intergas.
3. Outcomes (not raw data) may be downloaded to your own pc for the report.
4. There are no access limitations to folders on the server / filesystem, but try not to delete files which you will need yourself. If something is gone, it is gone.
5. Make sure you have discussed with Intergas about which findings you may / may not publish. By default you cannot publish anything you discover in the data without consent from Intergas. So make sure Intergas has enough time to give you feedback on your (semi-)final version of your thesis.
6. Do not use the server for things that are not a part of your data assignment.
7. Do not provide access to the server to others outside of your group.
8. Never distribute, copy, sell or share the data of Intergas and make the necessary precautions to prevent this from (accidentally) happening. For example, store your access token in a secure way.

I, Varoon Sushil Agrawal (full name),

have read the agreement and will stick to it.

Date: 28-04-2022

Signature: 

Statement of Authenticity

I, the undersigned, hereby certify that I have compiled and written the attached document / piece of work and the underlying work without assistance from anyone except the specifically assigned academic supervisors and examiners. This work is solely my own, and I am solely responsible for the content, organization, and making of this document / piece of work.

I hereby acknowledge that I have read the instructions for preparation and submission of documents / pieces of work provided by my course / my academic institution, and I understand that this document / piece of work will not be accepted for evaluation or for the award of academic credits if it is determined that it has not been prepared in compliance with those instructions and this statement of authenticity.

I further certify that I did not commit plagiarism, did neither take over nor paraphrase (digital or printed, translated or original) material (e.g. ideas, data, pieces of text, figures, diagrams, tables, recordings, videos, code, ...) produced by others without correct and complete citation and correct and complete reference of the source(s). I understand that this document / piece of work and the underlying work will not be accepted for evaluation or for the award of academic credits if it is determined that it embodies plagiarism.

Name:	Varoon Sushil Agrawal
Student Number:	2306557
Place/Date:	Rotterdam, July 1, 2022

Signature:

Abstract

Adding insulation to the households offers unique opportunity to save energy by reducing gas consumption by space heaters. In this research, insulation change at households is being detected from the gas use data available for heaters from 2015 to 2020. The research question is to identify how soon can we detect this insulation change with a certain amount of certainty. The feature of interest is the temperature difference between inside and outside the house. The data is preprocessed to obtain the final dataset for analysis. A correlation value of 0.632 is obtained between gas use and temperature difference. This paper discusses the Linear Regression approach to establish a relationship between gas use and temperature difference per heater for 12675 households. The model runs independently for each of the five heating seasons per heater in the dataset. The slopes of the regression lines are compared per heater to identify the change in slope and calculate the difference in percentage. Filters are applied on this percentage difference to detect households where insulation might have been added. The results reveal the heaters where potential insulation change has taken place. Based on the results, it can be concluded that data for two heating periods is required to detect this insulation change. Data for a third period is needed to add certainty to the change detection. This research can be utilised as a building block for future research in energy savings by detecting insulation change of households.

Keywords: Insulation; Heaters; Gas use; Linear Regression

Preface

The study was a part of the research project for the Intergas Verwaming BV and is divided into three main tasks.

The first task is a collaboration between the three MSc. Applied Data Science students from Utrecht University, in which they prepare the data provided by Intergas. The aim of this task is to create a data set that is as meaningful as possible and suitable for data analysis.

In the second task of this thesis, each of the three students work individually on the modelling process with different approaches and evaluating the results. Varoon Sushil Agrawal uses the Linear Regression approach, Moritz Muenten uses the Support Vector Regression approach, and Maria Fakou focuses on the Random Forest approach.

The third task of this study is to draw conclusion from the findings and comparison of the different approaches in order to make assumptions about which model is most suitable in the context of the research question. Here the students answer the research question and make a recommendation for further research.

Contents

List of Figures **VIII**

List of Tables **X**

1 Introduction **1**

2 Data **3**

2.1 Data Preprocessing 4

2.2 Exploratory Data Analysis 7

2.3 Ethical and Legal Consideration of Data 9

3 Methods **10**

3.1 Translation of the Research Question to a Data Science Question 10

3.2 Motivated Selection of Method for Analysis 10

3.3 Motivated Settings for Selected Method(s) 11

4 Results **12**

4.1 Density Plots 14

4.2 Detection of Insulation Change 16

4.3 Some Heater Examples 16

5 Discussion **21**

5.1 Answering the Research Question 21

5.2 Ethical Implications 21

5.3 Limitations and Future Research 21

5.4 Comparison of Models 22

5.5 Conclusion 22

Reference **23**

Appendices **24**

A Scripts of analyses **25**

A.1 Data Preprocessing 25

A.2 Data Exploration 29

A.3 Linear Regression - Monthly 30

A.4 Linear Regression - Period-wise 32

A.5 Evaluating Results 35

B Full Data Exploration Results **43**

C Full Analysis Results **44**

List of Figures

1.1 Linear regression of a household gas consumption vs temperature difference. 2

2.1 Scatter plot of gas use vs temperature difference 7
2.2 Boxplot of temperature difference and gas use per month 8
2.3 Line plot of average daily gas use per period 9

3.1 Scatter plot of temperature difference vs gas use for heater ID 24171 10

4.1 Density plot of percentage difference between period 1 and 2 14
4.2 Density plot of percentage difference between period 2 and 3 15
4.3 Density plot of percentage difference between period 3 and 4 15
4.4 Density plot of percentage difference between period 4 and 5 16
4.5 Slopes comparison for heater id 25007 17
4.6 Percentage Difference comparison for heater id 25007 17
4.7 Slopes comparison for heater id 30549 18
4.8 Percentage Difference comparison for heater id 30549 18
4.9 Slopes comparison for heater id 18680 19
4.10 Percentage Difference comparison for heater id 18680 19
4.11 Slopes comparison for heater id 27729 20
4.12 Percentage Difference comparison for heater id 27729 20

A.1 Read Data 25
A.2 Data preprocessing - Stage 1 26
A.3 Data preprocessing - Stage 2 26
A.4 Data preprocessing - Stage 3 27
A.5 Data preprocessing - Aggregate Data 28
A.6 Data Exploration 29
A.7 Monthly LR - Part 1 30
A.8 Monthly LR - Part 2 31
A.9 Monthly LR - Part 3 32

A.10 Period LR - Part 1 . . . 32
A.11 Period LR - Part 2 . . . 33
A.12 Period LR - Part 3 . . . 34
A.13 Results Summary . . . 35
A.14 Density Plot 1 . . . 35
A.15 Density Plot 2 . . . 35
A.16 Density Plot 3 . . . 36
A.17 Density Plot 4 . . . 36
A.18 Filter and Save Detected Heaters . . . 37
A.19 Evaluate Heater 25007 . . . 38
A.20 Evaluate Heater 30549 . . . 39
A.21 Evaluate Heater 18680 . . . 40
A.22 Evaluate Heater 14237 . . . 41
A.23 Evaluate Heater 27729 . . . 42

B.1 Distribution of data by heating periods. . . . 43
B.2 Boxplots of temperature difference by day of month. . . . 43

C.1 Density plot of Percentage Difference for Monthly LR. . . . 44
C.2 Slope comparison for heater ID 14237. . . . 44
C.3 Percentage Difference comparison for heater ID 14237. . . . 45

List of Tables

2.1 Ig_gasuse_hourly. . . . 3
2.2 ig-heater-info-nl-2. . . . 3
2.3 od_knmi_hourly_wijken_v2. . . . 3
2.4 House_prop. . . . 4
2.5 Join operation details. . . . 4
2.6 Datasets size before and after filtering 5
2.7 Heating periods. . . . 5
2.8 Final dataset structure. . . . 6
2.9 First rows of the final dataset. . . . 6
2.10 Descriptive Statistics. . . . 7

4.1 Selected monthly linear regression slopes and differences 12
4.2 Period-wise linear regression slopes and differences - Part 1 12
4.3 Period-wise linear regression slopes and differences - Part 2 13
4.4 Summary of slope_df - Part 1 13
4.5 Summary of slope_df - Part 2 14

C.1 Complete results of Monthly LR for heater id 8736 46

1 Introduction

The all-time heat records are being set worldwide with several global locations recording their highest ever temperature (*Insulation: the Missing Key to Energy Efficiency* 2018). Energy savings are an important aspect in the fight against climate crisis and to save us from the worsening heat waves. European Union's dependence on energy import has been the topic of discussion since the Russian aggression against Ukraine as gas imports from Russia cover 45% of EU's gas demand (BPIE 2022).

The housing sector is a huge consumer of the energy and plays a vital role in achieving energy efficiency targets in the EU (Faidra Filippidou 2018). Due to poor energy performance of buildings, they account for 38% of total energy consumption in the European Union (Delft CE 2015). Out of which, households are responsible for 24.8% of final energy consumption in the EU (*Consumption of energy* 2016). Thermal comfort in housing is established by space heating by maintaining the indoor temperature at a desired, uniform level and providing proper admission of fresh air (Haris Lulic 2013). In the Netherlands, 85% of the households are heated using natural gas (Faidra Filippidou 2018).

Various studies have identified that insulation in households reduces the energy consumption for heating. A report based on a study in eight EU countries by BPIE in 2022 revealed the savings in energy consumption by 45% and reduction in gas use by 44% with improved insulation (BPIE 2022). In all these countries, space heaters use between 15% - 21% of total energy consumption in the country. Another study in the UK concluded that if the average energy band is moved from D to C with insulation, the energy consumption would be reduced by 20% (*Insulation and gas prices* 2022). A simulation research on thermal insulation of walls by adding Styrofoam and new windows in Sarajevo, Bosnia and Herzegovina resulted in more than 30% energy savings (Haris Lulic 2013). A different study on mechanical insulation of heating water pipes revealed that heat loss can be reduced by 95% with insulation and concluded that mechanical insulation is potentially greener than planting trees (*Insulation: the Missing Key to Energy Efficiency* 2018).

Insulation can be of different types. One of them is the envelope insulation which is measured by the U-value of the roofs and walls (BPIE 2022). Window insulation is another example and every type of insulation has a different effect on energy savings (Majcen 2016).

Intergas Verwarming BV(Intergas) builds and sells heating equipment varying from gas boilers, water heaters, hybrids, control devices to heat pumps and researches future possibilities such as heating using hydrogen (*Intergas* 2022). A study on factors affecting gas use by heaters inferred that outdoor temperature has the highest influence on gas consumption (Dotzauer 2002). Figure 1.1 shows a plot provided by Intergas of linear regression of daily gas use versus temperature difference(between indoor and outdoor temperature) for a household. They observed a linear relationship with decrease in slope after insulation in 2017 which is in conjunction with other studies indicating that insulation reduces energy consumption. Through various contracts with their customers, Intergas has a large and detailed data of the energy use of their clients' households. This data does not contain insulation details of the houses, hence, with the available data, they would like to identify those houses where insulation has already been added by detecting a change in slope similar to Figure 1.1. There are two essential challenges - firstly, Intergas would like to know how quickly can this change be detected. This is about the temporal aspect as they collect data over time. Secondly, how certain is the change in slope? This is about the consistency of the new slope. This research would help Intergas to identify potential customers to install energy efficient heating systems. Additionally, it would assist them in identifying customers with high energy consumption and provide them with an insulation advise.

Thus, the main question of this research is: How soon can we say something about the new slope (gas consumption per temperature difference between outside and inside) with a certain amount of certainty?

In this research, the data is pre-processed to have a final dataset for modelling. Furthermore, three different modelling approaches have been used by different students involved in the research - Linear Regression, Support Vector Regression, and Random Forest. These models establish a relation between temperature difference and gas. This research paper focuses on the Linear Regression Model. Differences between coefficients(slopes) of temperature difference are calculated for separate timeframes to identify the change in slope. Moreover, filters are applied on these slope differences to detect the potential households where insulation has been added. Finally, the three different modelling approaches and their results are compared with suggestions for future research.

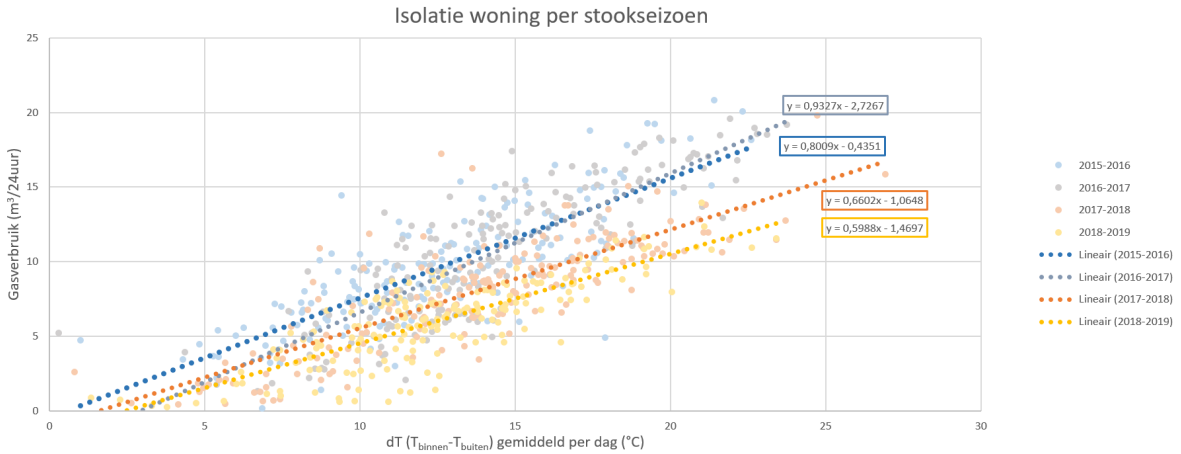

Figure 1.1: Linear regression of a household gas consumption vs temperature difference.

2 | Data

The data for this research is provided by Intergas. To gather all the needed information, four datasets were combined. This data is from the households in the Netherlands for the time period from 2015 to 2020. Tables 2.1 - 2.4 show the details of raw data obtained from Intergas. Intergas' database is built on Hadoop. To deal with large amount of data, Apache Spark is one of the engine options and is deployed by Intergas. Spark is a multi-language engine for executing data engineering, data science, and machine learning on single-node machines or clusters (*Spark* 2022). It is fast, offers less reading and writing from and to the disk, and due to the Python API PySpark, it is easy to use. It was also possible to work on smaller portions of the data with other Python packages (e.g. Pandas). Overall, the following software packages were used: PySpark, Pandas, NumPy, Matplotlib, Seaborn, and scikit-learn.

Column Name	Type	Description
heater_id	Integer	Heater identification number
gas_use	Double	Gas consumption (m^3)
surface_area	Integer	Surface area of the house (m^2)
t_set	Double	Temperature set on the thermostat (°C)
t_act	Double	Inside house temperature (°C)
TimeKey	Timestamp	Time of recording data

Table 2.1: Ig_gasuse_hourly.

Column Name	Type	Description
HEATER_ID	Integer	Heater identification number
wijk	Integer	Neighborhood ID
building_year	Integer	Building year

Table 2.2: ig-heater-info-nl-2.

Column Name	Type	Description
wijk	Integer	Neighborhood ID
rain	Double	Rainfall (cm)
sun	Double	Sun intensity (W/m^2)
temp	Double	Outside temperature*10 (°C)
wind	Double	Wind speed (km/h)
TimeKey	Timestamp	Time of recording data

Table 2.3: od_knmi_hourly_wijken_v2.

Column Name	Type	Description
HEATER_ID	Integer	Heater identification number
WONING_TYPE	String	House type

Table 2.4: House_prop.

2.1 Data Preprocessing

In the first stage of data preprocessing, the datasets were individually cleaned before joining them to improve the quality of data for analysis and decrease computation time in the later stages. Rows with missing values were removed. Outliers were removed with filters and data for summer months was removed. Following are the steps taken:

- Houses of surface area below 40 or above 400 square meters, in *Ig_gasuse_hourly*, were filtered out, as they are assumed to be too small or too big for a household and are not relevant for the research.
- Rows outside the range of [0,26] for t_set and rows outside the range of [10,30] for t_act were filtered out from *Ig_gasuse_hourly*. It was assumed to be highly unlikely for these variables to have values outside of this range and are thus considered noise.
- The data for the months of May, June, July, and August was removed, since the gas consumption during these months is negligible for heating and any high values are considered as unreliable data. This step was applied to *Ig_gasuse_hourly* and *od_knmi_hourly_wijken_v2*.
- Houses that had a missing house type in *house_prop* were discarded.
- Heaters that have missing values for building year or neighborhood were removed from *ig-heater-info-nl-2*.
- The research was limited to buildings constructed after 1950 and all rows with building year before 1950 were removed from *ig-heater-info-nl-2*.

In the second stage, the resulting four datasets were joined as per details in Table 2.5 to obtain an integrated dataframe for further analysis. Left inner joins were performed in all the join operations.

Left dataframe	Right dataframe	Key	Joined dataframe
Ig_gasuse_hourly	ig-heater-info-nl-2	heater_id	join_1
join_1	od_knmi_hourly_wijken_v2	Wijk, TimeKey	join_2
join_2	House_prop	heater_id	df_joined

Table 2.5: Join operation details.

Table 2.6 describes the datasets size before and after the filters. Note that the high percentage(> 30%) is due to removal of data for summer months.

Dataset	Before filtering	After filtering	Percentage removed
Ig_gasuse_hourly	558,960,694	354,261,532	36.6%
ig-heater-info-nl-2	39,305	39,175	0.33%
od_knmi_hourly_wijken_v2	74,894,318	51,603,006	31.09%
House_prop	39,305	39,155	0.38%
Final_df	324,849,444	222,216,880	31.6%

Table 2.6: Datasets size before and after filtering

In the third stage, duplicate rows were removed from df_joined. There were several heaters with different gas_use values for the same TimeKey. This was due to one heater_id representing multiple households. Rows for all such heater_id were removed from df_joined as they represent inconsistent data. Thus, heater_id and TimeKey together are the primary key for this dataframe. Furthermore, new columns were added in the data transformation process. Year, month, and day of the month were extracted from TimeKey. To calculate a new column t_diff, the outside temperature was subtracted from the indoor temperature (t_act - temp). Rows with a value of t_diff less than zero were removed as it was assumed to be noisy data to have temperature inside the house less than the outside temperature.

A new column "period" was created assigning it the values of heating periods. These values are defined as in table 2.7.

period	**Time Range**
1	Sept. 2015 - Apr. 2016
2	Sept. 2016 - Apr. 2017
3	Sept. 2017 - Apr. 2018
4	Sept. 2018 - Apr. 2019
5	Sept. 2019 - Apr. 2020

Table 2.7: Heating periods.

It is important to mention here that a heating period includes the data for a heating season from September to April and it is not the calendar year. Heaters with data for only a single heating period were also removed from the dataset as they contain little data for comparison which is unsuitable for analysis. There were 308 such heaters.

It is assumed that the insulation of a house directly affects the temperature difference between inside and outside and thus t_diff can be used to build a simple model. Moreover, Intergas is interested in relationship between t_diff and gas_use without including any other variables in the regression analysis. Thus, the columns selected for next steps are - heater_id, period, month, dayOfMonth, gas_use, t_diff. Moreover, zero gas use during some hours of the day implies a regression model would be better suited for daily data compared to hourly data. For this purpose, the data was grouped by heater_id, period, month, and dayOfMonth. Further, aggregate values like sum of gas_use and average of t_diff were calculated for the final dataset.

The structure of the final dataset and its first five rows are depicted in tables 2.8 and 2.9, respectively.

Column Name	Type
heater_id	Integer
period	Integer
month	Integer
dayOfMonth	Integer
sum_gas	Double
avg_t_diff	Double

Table 2.8: Final dataset structure.

ID	heater_id	period	month	dayOfMonth	sum_gas	avg_t_diff
0	93059	3	4	14	2.7573	10.622500
1	93059	4	10	9	1.6920	8.964167
2	96265	5	1	11	6.0406	15.012917
3	66595	2	3	11	6.4874	11.985000
4	54477	4	10	30	5.6728	15.618750

Table 2.9: First rows of the final dataset.

2.2 Exploratory Data Analysis

The final dataset contains 6,886,234 records for 12,675 heaters from September 2015 until April 2020. The number of records for one heater is not necessarily equivalent to other heaters, meaning that some heaters were measured for longer heating periods than others.

Table 2.10 shows the descriptive statistics of the daily gas use and average temperature difference. Both, the daily gas consumption and the average temperature difference present extreme values on some occasions, while their median values are 4.66 and 12.10, respectively.

summary	sum_gas_use	avg_t_diff
mean	5.376	12.162
stddev	4.459	4.31
min	0.0	0.01
25%	1.723	8.912
50%	4.669	12.107
75%	7.821	14.99
max	74.567	33.44

Table 2.10: Descriptive Statistics.

To understand the relationship between these variables for the exploration, the Pearson Correlation Coefficient is computed and its value of 0.632 reveals that the daily gas use and the average temperature difference are positively correlated. As illustrated in figure 2.1, there is a moderately strong, positive, linear association with a few outliers.

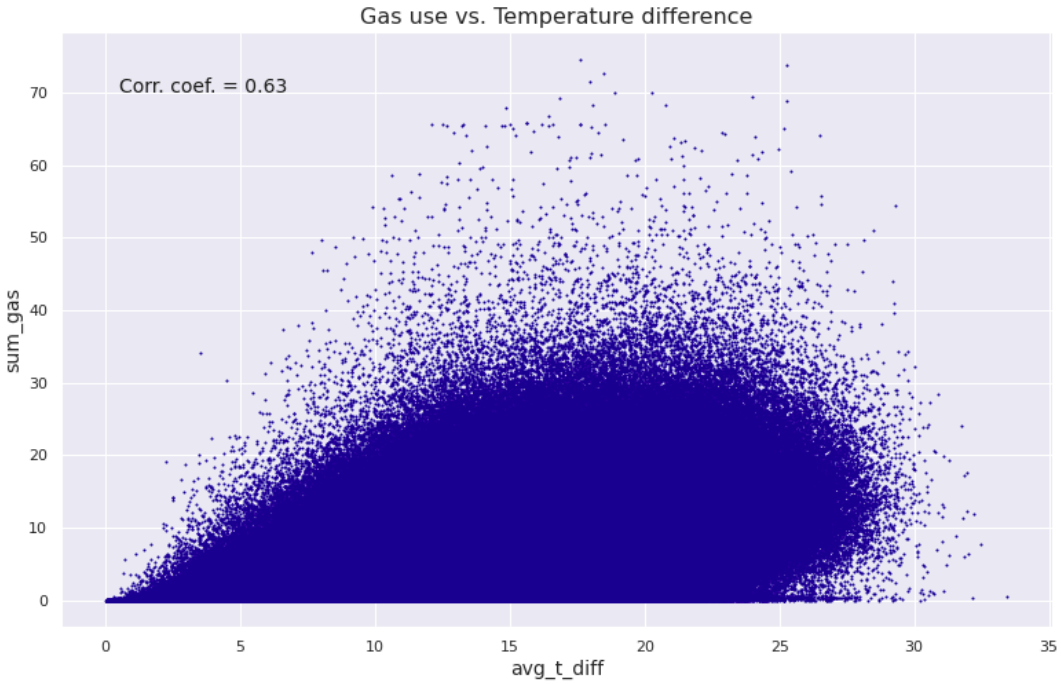

Figure 2.1: Scatter plot of gas use vs temperature difference

Furthermore, the 1st heating period contains the least data points - 5%, the 2nd heating period contains the second smallest share of the dataset - 14%. Data points from the 4th heating period exceed the rest at 31.7%, still those from the 5th and 3rd heating periods were nearly a quarter each, i.e., 25.2% and 24.1%, respectively. Therefore, the first period could not be perceived as a representative sample of the data, yet it was included in the three modelling approaches, as the objective of this analysis was to test how fast a change can be detected using the least possible amount of data.

The gas consumption is higher during the winter months and lower in April, September, and October. The same trend is observed for the temperature difference as well, while both cases suggest September to be the warmest month in the dataset(which does not include summer months), as it has the lowest gas use and temperature differences *(Figure 2.2)*.On the other hand, no pattern is detected for the gas use or temperature difference during the separate days of the months, which is a reasonable inference, and indicates uniformity across the daily behavior of the customers.

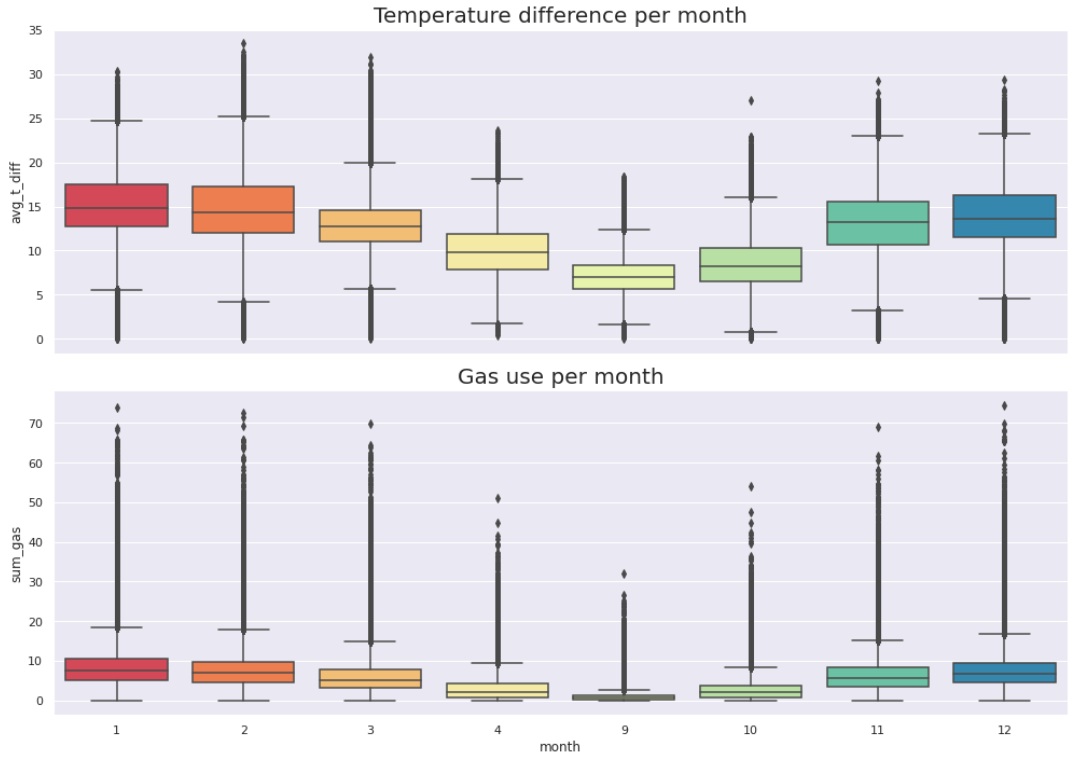

Figure 2.2: Boxplot of temperature difference and gas use per month

Some examples of heaters are selected for further investigation, as the initial aim is to distinguish those that present reduction in gas use, and then to examine how soon the distinction can be drawn. Figure 2.3, demonstrates three heaters of which two heaters *8180* and *27729* are potential houses that added insulation in the time frame of this dataset. *8180* seems to lower its gas use dramatically after the 2nd period, whereas *27729* appears to suddenly decrease after the 3rd period, and the gas use of both houses are stable after the decline. The gas use of *5924*, in contrast, remains quite stable through the different heating periods and thus, it can be assumed that this specific house did not improve its insulation.

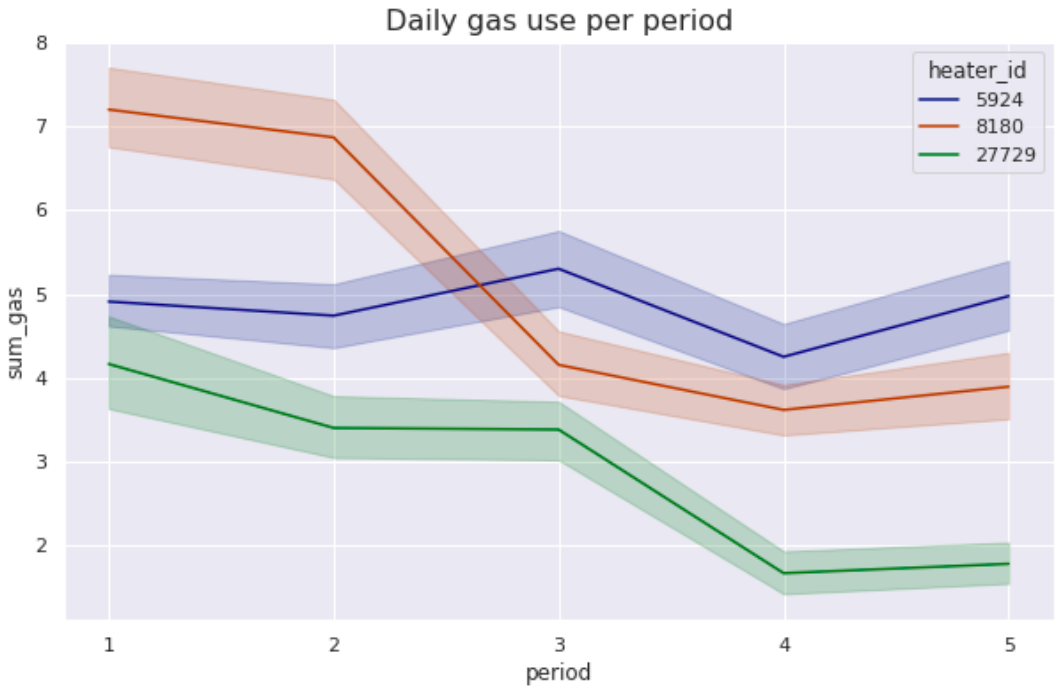

Figure 2.3: Line plot of average daily gas use per period

2.3 Ethical and Legal Consideration of Data

The data is collected by Intergas through various contracts with the customer. No individual household location can be tracked with the data available. Moreover, the neighbourhood information(Wijk) is also encoded with ID rather than names which makes it difficult to track any house. Furthermore, the dataset does not contain any information of the occupants of households that can be considered unethical including name, sex, race, age, etc.

3 Methods

3.1 Translation of the Research Question to a Data Science Question

The research question how soon can we say something about the new slope with a certain amount of certainty can be interpreted as detecting change in slope(gas use vs temperature difference) between two timeframes for a heater and identifying those heaters with a significant change. In this study, we are only interested in detecting a significant decrease to identify the addition of insulation at a home.

3.2 Motivated Selection of Method for Analysis

Regression analysis is a statistical technique used to determine the relationship between a dependent variable and one or more independent variables (Haris Lulic 2013). More specifically, simple regression analysis aids us in understanding the change in value of the dependent variable as the value of independent variable varies. A change in slope is to be detected to identify the change in insulation of a house. Linear association between two variables can be measured by correlation coefficient which has the value of 0.63 (between daily gas use and temperature difference) for our dataset. This implies that linear regression(LR) is one of the best ways to detect this change as it is also the most common form of regression analysis. This is also supplemented by figure 3.1 which depicts a linear relationship between temperature difference and gas use for heater_id 24171. We obtain a coefficient in Linear Regression for the feature temperature difference(avg_t_diff) which is exactly the slope of the regression line.

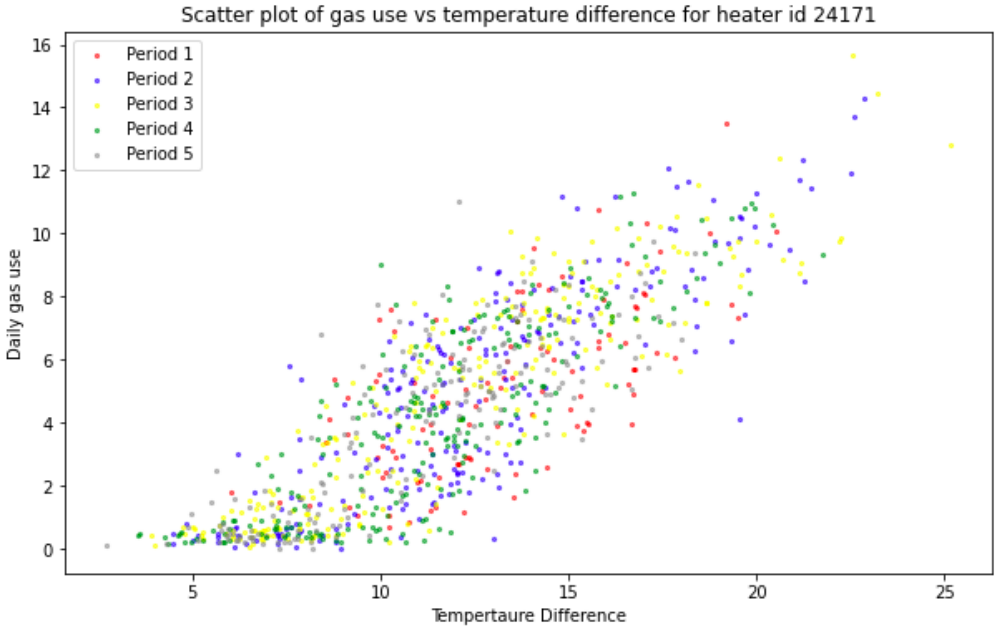

Figure 3.1: Scatter plot of temperature difference vs gas use for heater ID 24171

The first model does a regression for every month of each heating period for every heater. The maximum number of data points available per month are 31 (maximum number of days in a month). The slope is then compared with the slope for the same month in the previous heating period. Thus the differences for the first heating period are always zero. From the second heating period onward, the differences can be calculated. For any missing slope, null values are assigned and thus null values exist in the differences.

An empty dataframe monthly_slope_df is created to store these slopes. The algorithm iterates over three loops - first for each heater, second for each heating period, third for each month. A linear regression is fit for data points of each month. The co-efficient of avg_t_diff is stored in the slope_df with heater id, period, and month.

After having obtained the slope data, two new columns are added to monthly_slope_df. First is slope_diff which is the difference between slope of the month in current heating period and slope of same month in previous period. Second column is the percentage_diff which is the difference in percentage based on the formula (((slope_diff)/(previous_slope))*100). A null value is assigned to the new columns if there is no data available for previous or current slope.

The second model does a regression for every heater for every heating period. Thus a maximum of 5 slopes are obtained per heater - one for each heating period. The slope is then compared to the slope of same month in the previous period.

A new empty slope_df dataframe is created to store slopes and differences. The algorithm iterates over two loops - one for each heater and other for each heating period. A linear regression is fit for data points of each heating period. The slopes are stored for each heater in slope_df columns - slope_p1, slope_p2, slope_p3, slope_p4, slope_p5 which are slopes for period 1, 2 , 3, 4, and 5 respectively. A function is created to calculate the slope differences between consecutive periods for each heater. Each row of slope_df is passed to the function and slope differences and percentage of slope differences are added to the slope_df columns diff1, diff2, diff3, diff4, pdiff_1, pdiff_2, pdiff_3, and pdiff_4. diff1 represents slope difference between period 1 and period 2 (slope_p2 - slope_p1). pdiff_1 is calculated as (((diff_1)/(slope_p1))*100). If, for any heater, there is no data for a period, null value is assigned to the slope, thus assigning null values to slope differences and percentage differences.

3.3 Motivated Settings for Selected Method(s)

For monthly linear regression, pyspark ml library was utilised with default parameters. It has a maxIter = 100, and elasticNetParam = 0.0. Loss is calculated based on mean squared error.

Library scikit-learn was imported and utilised for period-wise linear regression. This is a simple linear regression with default parameters. Mean Squared Error minimisation is the task for this regression. Period-wise LR takes a little over 3 minutes as computation time for the entire dataset of 12675 heaters. Hence, it was assumed to keep all data points in the study without any sampling as it is computationally viable in this research.

4 | Results

The monthly regression results are stored in monthly_slope_df. It contains the heater_id, period, month, slope, slope_diff, percentage_diff columns. Table 4.1 shows some of these values for heater id 8736.

heater_id	period	month	slope	slope_diff	percentage_diff
8736	1	10	0.352	0.000	0.000
8736	1	1	0.337	0.000	0.000
8736	1	3	0.568	0.000	0.000
8736	2	9	0.037	0.000	0.000
8736	2	1	0.436	0.099	29.493
8736	2	4	-0.052	-0.274	-123.472
8736	3	4	0.343	0.395	-760.157
8736	4	2	0.403	0.036	9.936
8736	4	3	1.056	0.647	158.467
8736	5	9	0.019	-0.046	-70.364
8736	5	3	0.000	-1.056	-100.000

Table 4.1: Selected monthly linear regression slopes and differences

There are quite high variations in percentage_diff for the same heater. The slope for January in period 1 is lower than slope for October in the same period. This result is counter-intuitive as one would expect more heater usage in January than October. The slopes vary a lot within the same heating period and the percentage_diff is quite high at times. The monthly linear regression is not precise to compare the slopes due to high variance in the slope differences. This compels us to focus the research on linear regression for whole heating periods.

The period-wise regression results are stored in a separate dataframe slope_df. Table 4.2 and 4.3 show what this looks like.

heater_id	slope_p1	slope_p2	slope_p3	slope_p4	slope_p5
93059	NaN	NaN	0.544	0.546	0.506
96265	NaN	NaN	0.360	0.374	0.412
66595	NaN	0.343	0.655	0.615	0.604
54477	NaN	0.455	0.518	0.501	0.511
39755	NaN	0.737	0.703	0.691	0.726
...	...	...	...	...	...
50383	NaN	0.773	0.533	NaN	NaN
17168	1.014	NaN	NaN	NaN	NaN
20940	0.278	NaN	NaN	NaN	NaN
22093	NaN	NaN	NaN	3.486	NaN
72303	NaN	NaN	NaN	NaN	NaN

Table 4.2: Period-wise linear regression slopes and differences - Part 1

heater_id	diff_1	pdiff_1	diff_2	pdiff_2	diff_3	pdiff_3	diff_4	pdiff_4
93059	NaN	NaN	NaN	NaN	0.002	0.417	-0.040	-7.250
96265	NaN	NaN	NaN	NaN	0.014	3.840	0.037	9.992
66595	NaN	NaN	0.312	90.988	-0.040	-6.091	-0.011	-1.799
54477	NaN	NaN	0.063	13.844	-0.017	-3.316	0.011	2.115
39755	NaN	NaN	-0.035	-4.687	-0.012	-1.641	0.035	5.001
...	...	...	...	...	...	...	...	...
50383	NaN	NaN	-0.240	-31.019	NaN	NaN	NaN	NaN
17168	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
20940	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
22093	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
72303	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

Table 4.3: Period-wise linear regression slopes and differences - Part 2

Table 4.4 and 4.5 contain the summary of slope_df. All slopes below 0.1 have been removed because negative slopes do not logically make sense and could be an error possibly because of low number of data points available for regression. Quite small values (less than 0.1) do not hold any significant information and thus have been removed from further analysis. A total of 193 heaters are removed by this filtering.

The mean of slopes for all periods is similar with very little variation. The mean values are close to the correlation coefficient of 0.632 calculated during exploratory data analysis. Period 1 has least data only for 2781 heaters from a count of 12482 heaters. This also leads to high standard deviation for the slope of period 1.

The differences column have a mean value close to zero between all the consecutive periods. The standard deviation is also low for diff_2, diff_3, and diff_4.

Statistics	heaterid	slope_p1	slope_p2	slope_p3	slope_p4	slope_p5
count	12482.000	2781.000	5662.000	8594.000	11133.000	10182.000
mean	83829.211	0.593	0.631	0.680	0.676	0.659
std	51785.734	1.287	0.286	0.328	0.307	0.312
min	2036.000	0.103	0.101	0.101	0.100	0.100
25%	40273.500	0.347	0.442	0.470	0.470	0.460
50%	77486.000	0.483	0.589	0.633	0.637	0.620
75%	124954.000	0.646	0.774	0.833	0.831	0.807
max	204773.000	41.567	3.428	10.754	3.486	8.549

Table 4.4: Summary of slope_df - Part 1

Statistics	diff_1	pdiff_1	diff_2	pdiff_2	diff_3	pdiff_3	diff_4	pdiff_4
count	2665.000	2665.000	5184.000	5184.000	7844.000	7844.000	10022.000	10022.000
mean	0.051	33.446	0.038	11.759	-0.002	2.673	-0.024	-0.338
std	1.294	58.006	0.170	45.740	0.150	33.222	0.179	33.910
min	-41.172	-99.049	-0.928	-85.686	-1.498	-87.418	-2.906	-90.879
25%	0.044	9.089	-0.034	-5.540	-0.056	-8.546	-0.080	-11.578
50%	0.118	24.692	0.024	4.201	0.001	0.190	-0.019	-3.029
75%	0.203	44.034	0.092	16.352	0.056	8.996	0.038	6.348
max	2.778	941.974	1.870	635.008	2.695	597.630	7.635	836.270

Table 4.5: Summary of slope_df - Part 2

4.1 Density Plots

The percentage differences have been plotted as density plots. Fig 4.1 shows the density plot for pdiff_1. It has a high positive mean implying there was an increase of 33.446% gas use from period 1 to period 2 on average.This high value is due to less data points available for heating period 1. The one sigma range is from -24.56% to 91.452%. The minus two sigma value of pdiff_1 is -82.556%.

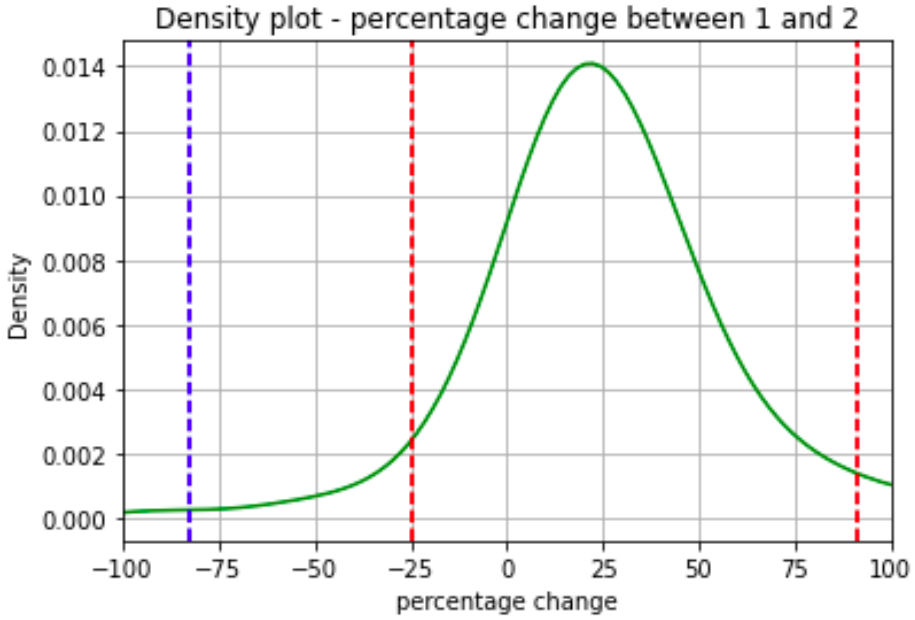

Figure 4.1: Density plot of percentage difference between period 1 and 2

Fig 4.2 shows the density plot for pdiff_2. It has a positive mean implying there was an increase of 11.759% gas use from period 2 to period 3 on average. The one sigma range is from -33.981% to 57.499%. The minus two sigma value of pdiff_1 is -79.721%.

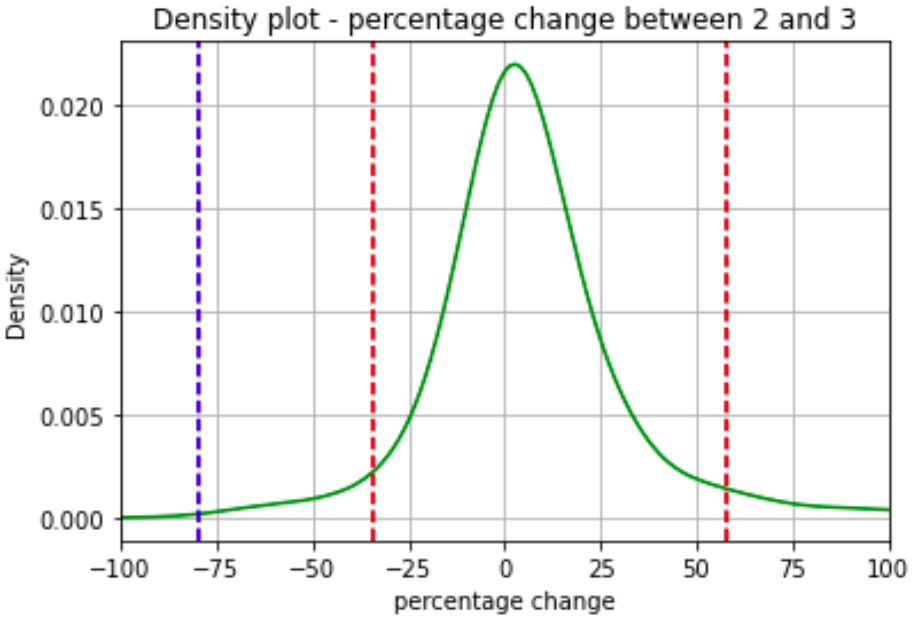

Figure 4.2: Density plot of percentage difference between period 2 and 3

Fig 4.3 shows the density plot for pdiff_3. It has a low positive mean implying there was an increase of only 2.673% gas use from period 3 to period 4 on average. The one sigma range is from -30.549% to 35.895%. The minus two sigma value of pdiff_1 is -63.771%.

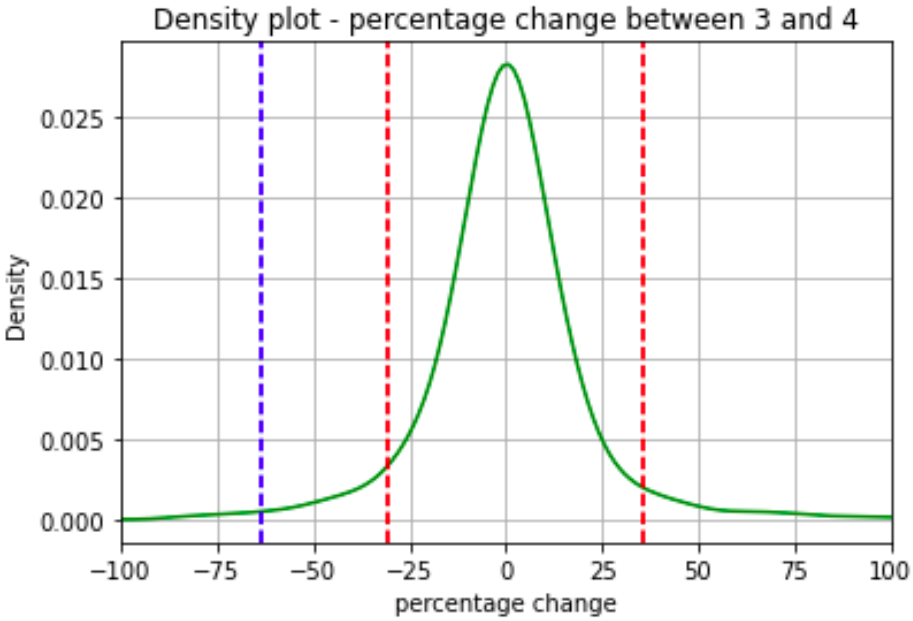

Figure 4.3: Density plot of percentage difference between period 3 and 4

Fig 4.4 shows the density plot for pdiff_4. It has a mean of almost 0% implying the gas use in period 4 and period 5 was the same on average. The one sigma range is from -34.248% to 33.572%. The minus two sigma value of pdiff_1 is -68.158%.

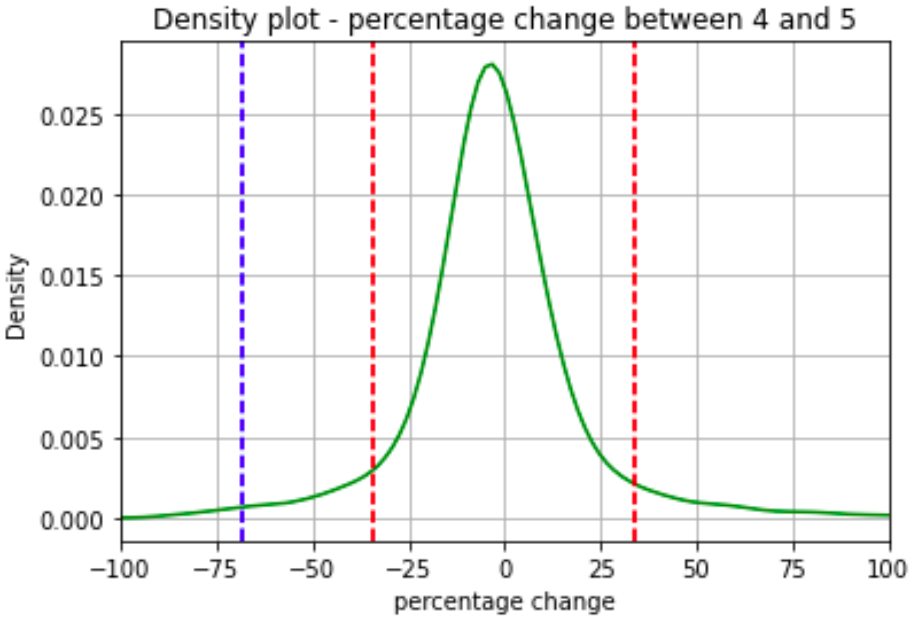

Figure 4.4: Density plot of percentage difference between period 4 and 5

4.2 Detection of Insulation Change

The final task to answer the research question is to detect which households might have had a change in insulation based on percentage differences. There are many possibilities to filter the data. The approach used in this study is that the percentage difference should be in the range (μ- 2σ, μ- 1σ) for one of the percentage differences and in the range (μ- 1σ, μ+ 1σ) for all other percentage differences. The other percentage differences can be null values as well. Heaters beyond μ- 2σare considered outliers in this study because they are assumed to have too high decrease in percentage change.

With this filter policy, a total of 795 heaters are detected which had an insulation change between two consecutive heating periods. Additionally, these heaters did not have any significant change between other heating periods.

4.3 Some Heater Examples

There are plots for four heaters in this section. These four heaters have different scenarios and each will be analysed based on their slopes for the five heating periods. The plots for slope comparison ignore the intercept values of the linear regression as this study is focused only on comparing slopes for gas use. Hence, all the lines start from (0,0) which is not the case for actual regression lines. This research focuses on change in slope of gas use vs temperature difference and not on the change in actual gas use. For this purpose, the offset is not important and only the slopes are considered.

Fig 4.5 shows the slopes for heater id 25007 for all periods. The slope remains similar for first 4 periods in the range of (μ- 1σ, μ+ 1σ) and decreases significantly in period 5 - in the range of (μ- 2σ, μ- 1σ). This heater is filtered in the final list of heaters with insulation change. Fig 4.6 shows the percentage differences and it can be seen that the values are close to zero for first 3 differences and decreases to -41% for pdiff_4. This household seems to be correctly detected with insulation change.

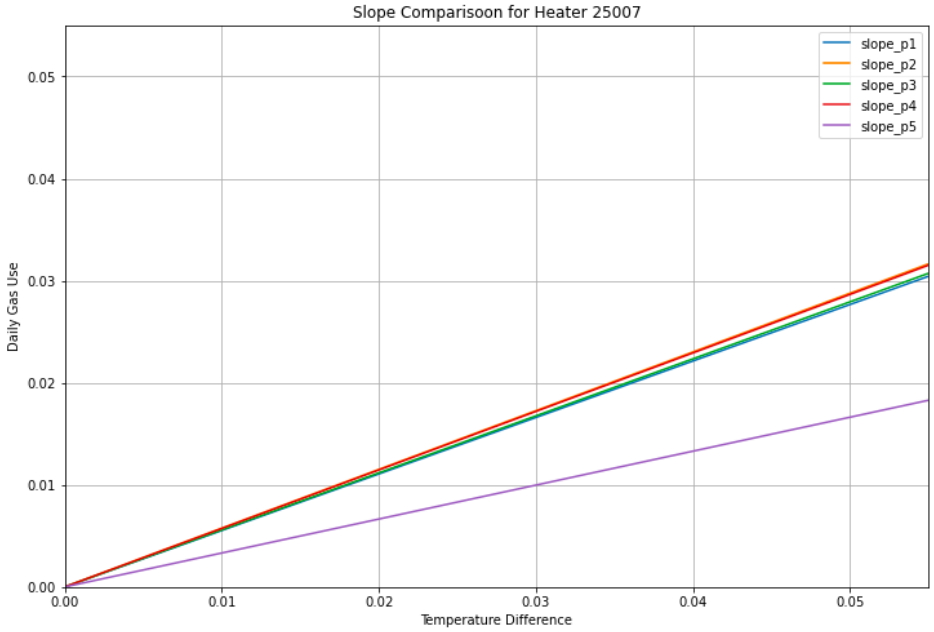

Figure 4.5: Slopes comparison for heater id 25007

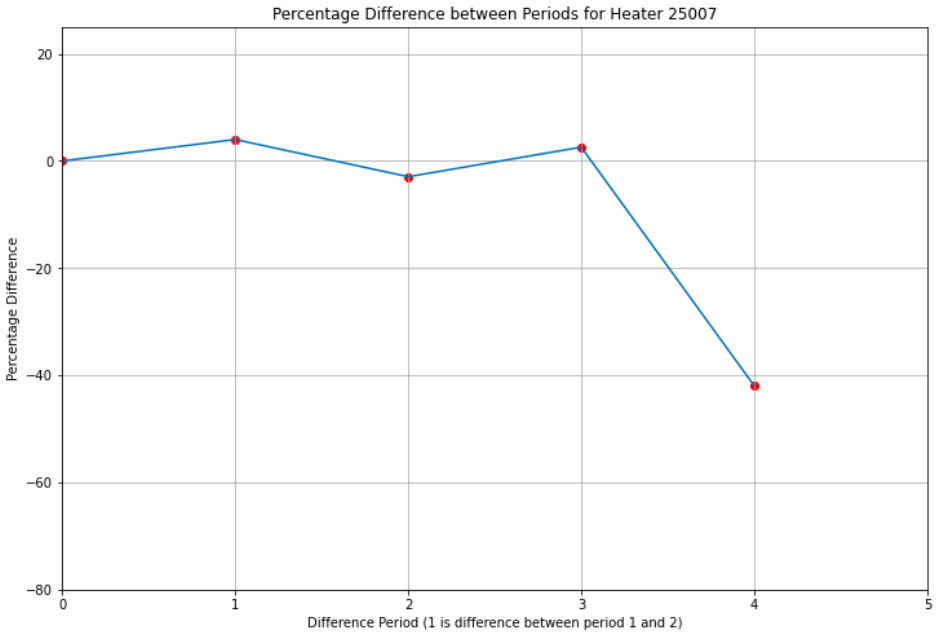

Figure 4.6: Percentage Difference comparison for heater id 25007

Fig 4.7 shows the slopes for heater id 30549 for all periods. The slope remains similar for first 3 periods in the range of (μ- 1σ, μ+ 1σ) and decreases significantly in period 4 - in the range of (μ- 2σ, μ- 1σ). It then remains stable for period 5. This heater is filtered in the final list of heaters with insulation change. Fig 4.8 shows the percentage differences and it can be seen that the values are close to zero for first 2 differences and decreases to -38% for pdiff_3. It is again close to zero for pdiff_4. This household seems to be correctly detected with insulation change which is reaffirmed by the value of pdiff_4 indicating consistency of the new slope.

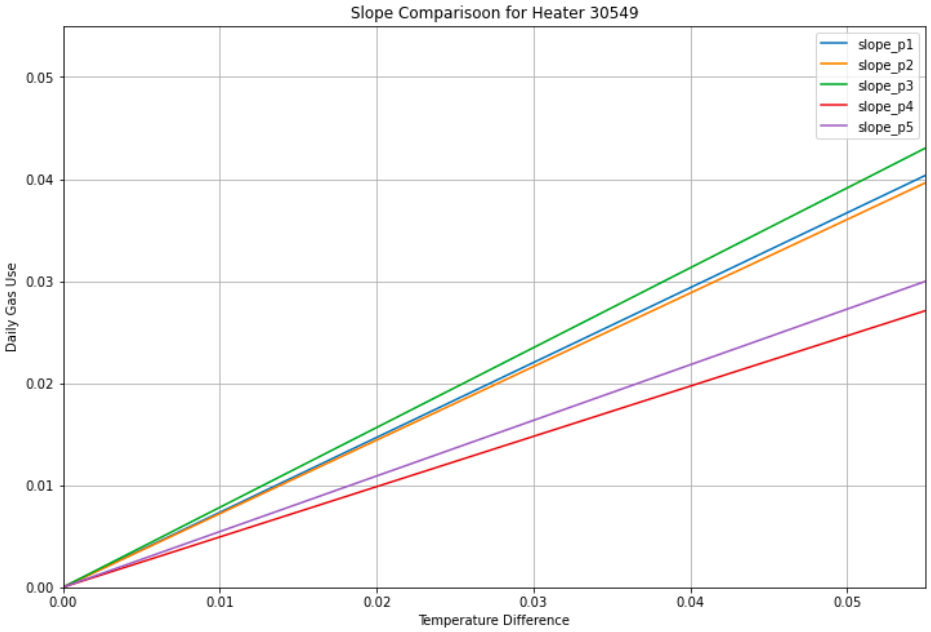

Figure 4.7: Slopes comparison for heater id 30549

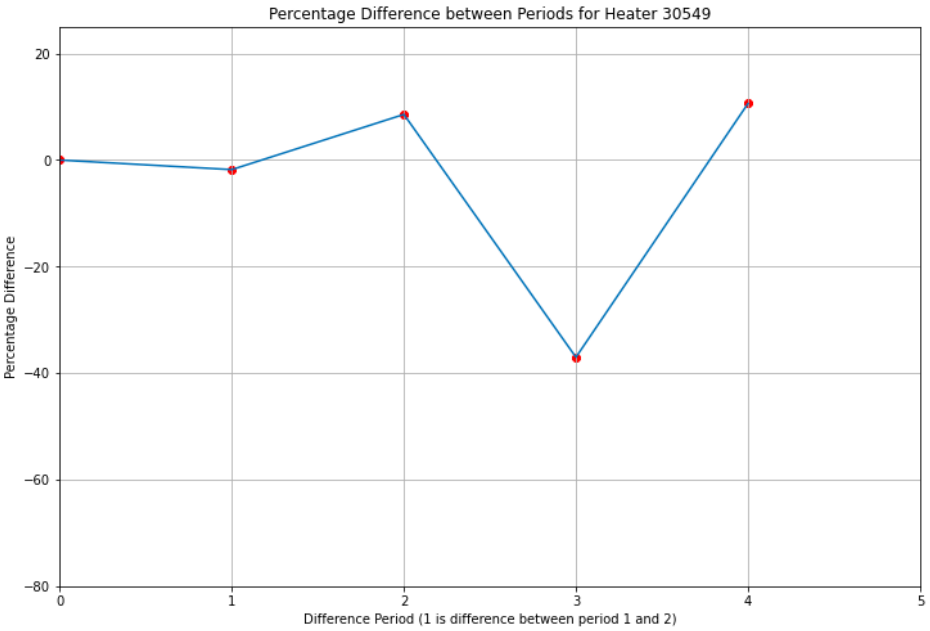

Figure 4.8: Percentage Difference comparison for heater id 30549

Fig 4.9 shows the slopes for heater id 18680 for all periods. The slope is similar for all periods except period 3 where it is higher than the others. Fig 4.10 shows the percentage difference and it can be seen that the value for pdiff_2 is high, however, it is still in the range of (μ- 1σ, μ+ 1σ) and thus not considered a significant increase. The value for pdiff_3 is -31% and is considered in the range of (μ- 2σ, μ- 1σ). Thus, this heater is filtered in the final list of detected heaters. However, we can see that the slopes in period 4 and 5 are similar to slopes in period 1 and 2. Thus, this house seems to be incorrectly detected with insulation change.

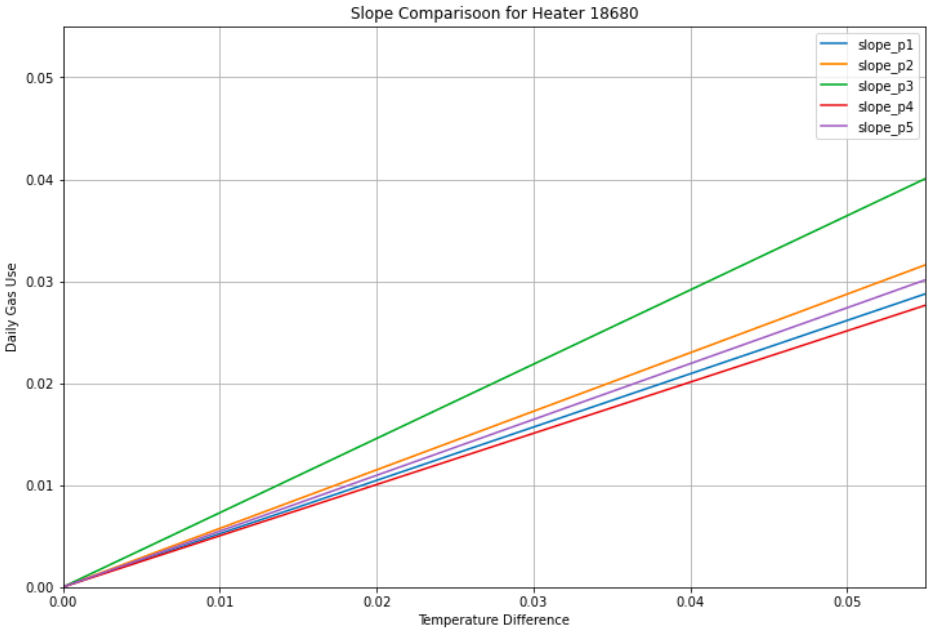

Figure 4.9: Slopes comparison for heater id 18680

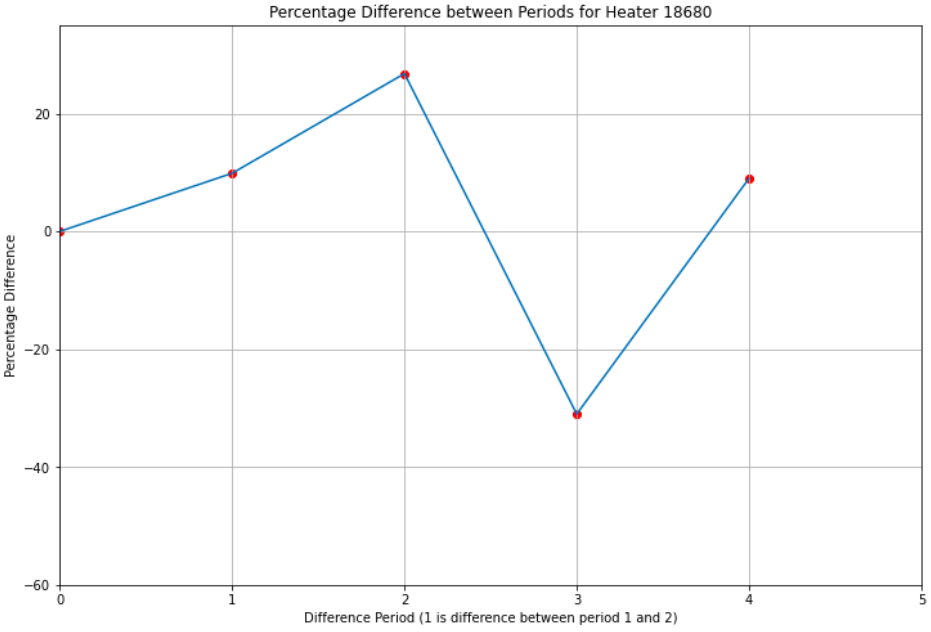

Figure 4.10: Percentage Difference comparison for heater id 18680

Fig 4.11 shows the slopes for heater id 27729 for all periods. The slope decreases significantly between period 4 and 5. It would be expected that this heater is filtered in the final list of detected heaters. Fig 4.12 shows the percentage difference and it can be seen that the value for pdiff_3 is -64% which is outside the cutoff value of μ- 2σ(-63%) and thus this change is considered an outlier in this study and this heater is filtered out. This heater represents a potential household with insulation change and should have been included in the final list of detected heaters.

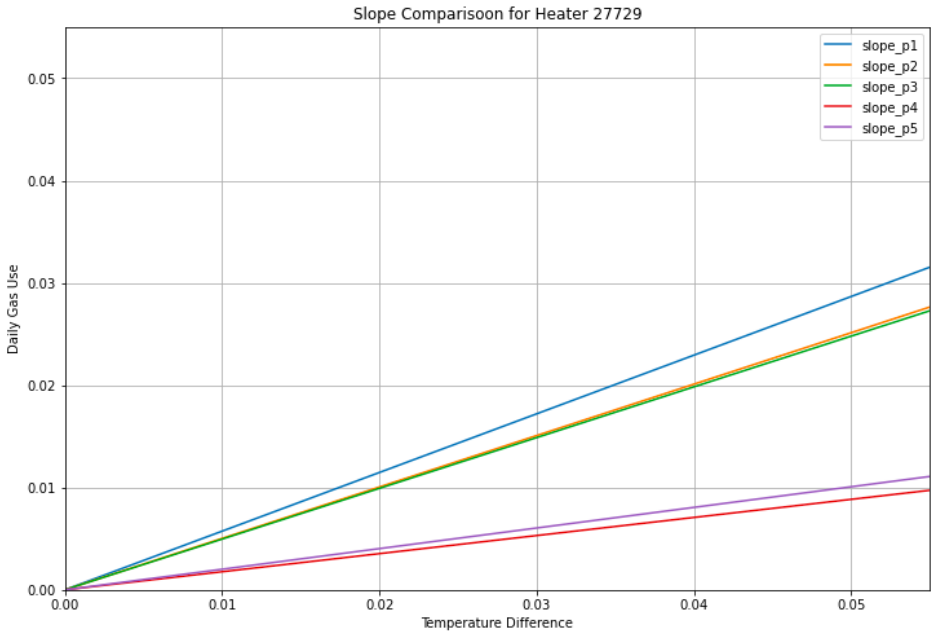

Figure 4.11: Slopes comparison for heater id 27729

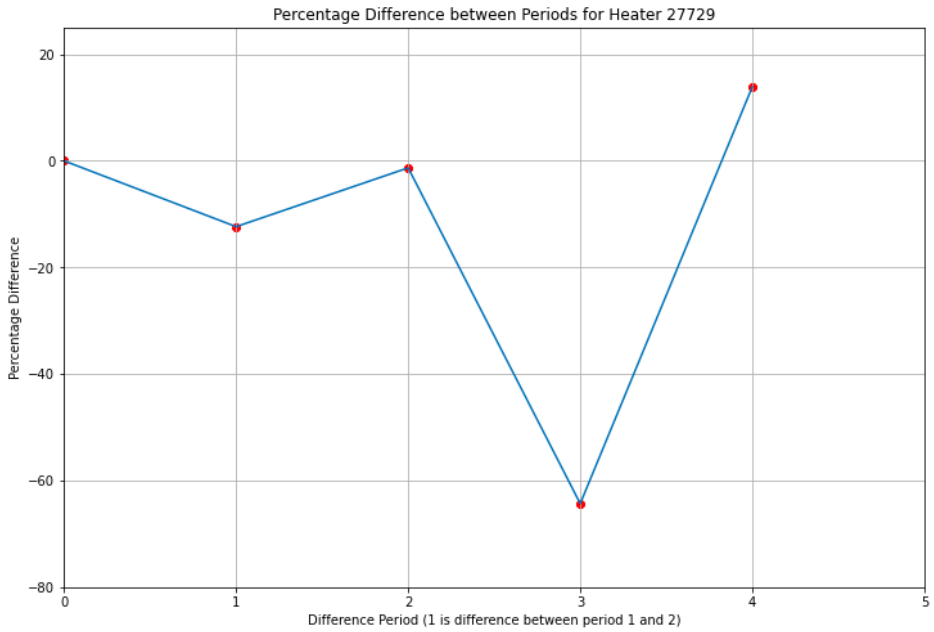

Figure 4.12: Percentage Difference comparison for heater id 27729

5 Discussion

5.1 Answering the Research Question

The slopes of linear relation between daily gas use and temperature difference are obtained with the linear regression approach. This model is fit per heating period for every heater. The percentage differences are calculated to identify significant decrease in slope compared to previous heating period. The filters based on mean and standard deviation of percentage differences aided in identifying the heaters with a significant decrease. The decrease in slope can be identified based on data availability for two heating periods. However, to add a certainty that the new slope remains stable, data for a third heating period would be required.

This research aimed to find the answer to the question : How soon can we say something about the new slope (gas consumption per temperature difference between outside and inside) with a certain amount of certainty? We can say something about the new slope with data for two heating periods. To add certainty to the results, data for a third period can be utilised to validate consistency of the new slope.

5.2 Ethical Implications

Intergas can deploy this research to identify households with added insulation. They can further communicate with the households to confirm the change in insulation and gather details about insulation. These insulation details can be provided as advice to other households with high gas consumption. Leakage in the heating system or insulation can also be identified with this research by focusing the research on households where gas consumption has drastically increased.

With this research, the households can reduce energy consumption with added insulation and save energy costs. They will also be contributing in the fight against the climate crisis.

5.3 Limitations and Future Research

The linear regression model has some limitations in detecting the households with insulation change. The model only considers one independent variable - temperature difference(avg_t_diff). Moreover, the model assumes that the insulation change occurs only in between different heating periods(in summer months). The model detects some households with insulation change where the new slope is similar to slopes of the previous period(as seen in figure 4.9 for heater id 18680). This suggests that some households are incorrectly detected for insulation change. Some households (for example heater id 27729 in figure 4.11) potentially have an insulation change but are not detected due to the filters for percentage difference based on mean and standard deviation. The slope can be affected by the characteristics of heater itself - like the quality, type of heater, transfer system, controller system. The decrease in slope would also be affected by the type of insulation and occupant's behaviour and can vary per household (Paula van den Brom 2019). This research also assumes that the decrease in the gas use is due to insulation change, however, other factors might be the reason for this decrease. Moreover, this research focuses on change in slope and not on the change in actual gas consumption by heaters. Thus, it does not give any insight on the decrease in actual gas consumption by households.

For future research, multiple independent features can be utilised in the regression model like weather attributes (rain, sun, wind) and house details (surface area). The households have distinct characteristics and it could be a nice approach to cluster similar houses together and then fit different models for clusters. Separate filters can be tried on percentage difference to detect maximum households with insulation change. One of the approaches to sooner detect the change is to compare the slopes over rolling period(for example 4 months rolling period - slope 1 for months 1-4, slope 2 for months 2-5, and so on). Another approach in identifying heaters could be checking the mean squared error of predictions for new heating period on the LR of previous heating period. This would eliminate the need for percentage differences and introduce a possibility that heaters with lower training error are easier to detect the slope change. The density plots of percentage difference(figure 4.1 - 4.4) represent an almost symmetric spread. A separate research could focus on the causes for increase in gas use.

5.4 Comparison of Models

Three different approaches by three students were a part of this research. Two approaches focused on regression model. This paper discusses the approach of linear regression(LR). The other regression approach is Support Vector Regression(SVR)(Muenten 2022). SVR is quite similar to LR as it calculates slopes and percentages differences in a similar manner using support vector machine. The difference is in the filtering of results, while LR implements the filters based on μ and σ, the SVR approach filters in the lowest and highest 5% heaters from the density plot of percentage differences. Additionally, SVR has a density plot which is combined for all the five heating periods which makes it less detailed than LR. The results of LR and SVR overlap, however, the count of households finally detected by the SVR approach is lower than LR owing to the fact that the density plot is combined for all five heating periods in SVR.

The Random Forest(RF)(Fakou 2022) is a tree-based approach. It sets its hyperparameters and runs the algorithm for all five heating periods. There is no slope obtained with this approach, hence, it plots the training and test error. RF is computationally intensive and requires further research to find a solution to detect households with insulation change.

5.5 Conclusion

This research explores methods to detect households where insulation has been added in between 2015 - 2020. It is implemented with the available gas use data from Intergas. Linear Regression is utilised to understand the relationship between gas use and temperature difference. The change in slope of regression lines between different heating periods for each heater helped us identify 795 households with potential insulation change. This research would act as a building block and help Intergas to dive deeper into detecting insulation change from available gas use data and provide assistance to their clients in reducing gas consumption.

Reference

BPIE (2022), 'How building insulation can reduce fossil fuel imports and boost eu energy security', *Putting a stop to energy waste* .

Consumption of energy (2016), https://ec-europa-eu.proxy.library.uu.nl/eurostat/statistics-explained/index.php?title=Consumption_of_energy. [Online; accessed 29-June-2022].

Delft CE, Hinicio, I. I. E. C. D.-G. f. E. (2015), 'Financing the energy renovation of buildings with cohesion policy funding : technical guidance: final report', *Publications Office of the European Union* .

Dotzauer, E. (2002), 'Simple model for prediction of loads in district-heating systems', *Applied Energy* .

Faidra Filippidou, Nico Nieboer, H. V. (2018), 'Effectiveness of energy renovations: a reassessment based on actual consumption savings', *Effectiveness of energy renovations: a reassessment based on actual consumption savings* .

Fakou, M. (2022), 'Doing more with less - 1', *Doing More with Less - 1* .

Haris Lulic, Adnan Civic, M. P. A. O.-E. D. (2013), 'Optimization of thermal insulation and regression analysis of fuel consumption', *24th DAAAM International Symposium on Intelligent Manufacturing and Automation, 2013* .

Insulation and gas prices (2022), https://eciu.net/analysis/briefings/heating/insulation-and-gas-prices. [Online; accessed 29-June-2022].

Insulation: the Missing Key to Energy Efficiency (2018), https://insulation.org/io/articles/insulation-the-missing-key-to-energy-efficiency/. [Online; accessed 29-June-2022].

Intergas (2022), https://www.intergas-verwarming.nl/en/consumer/. [Online; accessed 29-June-2022].

Majcen, D. (2016), 'Actual and theoretical gas consumption in dutch dwellings', *Predicting energy consumption and savings in the housing stock* .

Muenten, M. (2022), 'Doing more with less - 1', *Doing More with Less - 1* .

Paula van den Brom, Anders Rhiger Hansen, K. G.-H. A. M. H. V. (2019), 'Variances in residential heating consumption', *Applied Energy* .

Spark (2022), https://spark.apache.org/. [Online; accessed 29-June-2022].

Appendix

A | Scripts of analyses

A.1 Data Preprocessing

```
#Read raw data
gasuse_hourly = session.read.options(header = True, delimiter=';').csv(hdfs_url
    + '/user/intergas/datasets/ig-gasuse-hourly.csv')
gasuse_df = gasuse_hourly.select(gasuse_hourly.gas_use.cast("double"),
                     gasuse_hourly.heater_id.cast("int"),
                     gasuse_hourly.oppervlakteverblijfsobject.cast("int"),
                     gasuse_hourly.t_set.cast("double"),
                     gasuse_hourly.t_act.cast("double"),
                     gasuse_hourly.TimeKey
                    )

knmi_hourly = session.read.options(header = True).json(hdfs_url + '/user/
   intergas/datasets/od-knmi-hourly-wijken-v2.json' )
knmi_hourly_df = knmi_hourly.select(knmi_hourly.TimeKey,
                                    knmi_hourly.Wijk.cast("int"),
                                    knmi_hourly.rain.cast("double"),
                                    knmi_hourly.sun.cast("double"),
                                    knmi_hourly.temp.cast("double"),
                                    knmi_hourly.wind.cast("double")
                                   )


house_prop = session.read.options(header = True).csv('/home/jovyan/files/
   House_prop.csv')
house_prop_df = house_prop.select(house_prop.HEATER_ID.cast("int"),
                                    house_prop.WONING_TYPE,
                                    )

heater_info = session.read.csv(hdfs_url + '/user/intergas/datasets/ig-heater-
   info-nl-2.csv', header=True, inferSchema=True)
```

Figure A.1: Read Data

```
#Filter surface area, t_set, t_act
gasuse_df = gasuse_df.filter((gasuse_df.oppervlakteverblijfsobject >= 40) & (gasuse_df.oppervlakteverblijfsobject <= 400) & (gasuse_df.t_set <= 26)
                     & (gasuse_df.t_set >= 0) & (gasuse_df.t_act <= 30) & (gasuse_df.t_act >= 10))



#Drop summer months(May-August) from gasuse and knmi
gasuse_df = gasuse_df.withColumn('month', substring('TimeKey', 5,2).cast('int'))
gasuse_no_summer = gasuse_df.filter((gasuse_df.month > 8) | (gasuse_df.month < 5)).drop('month')

knmi_hourly_df = knmi_hourly_df.withColumn('month', substring('TimeKey', 5,2).cast('int'))
knmi_no_summer = knmi_hourly_df.filter((knmi_hourly_df.month > 8) | (knmi_hourly_df.month < 5)).drop('month')

#Drop rows with no woning_type
house_prop_df = house_prop_df.na.drop()

#Drop rows with no wijk or building year
heater_info = heater_info.na.drop(subset=['pandbouwjaar', 'wijk']).select('HEATER_ID', 'pandbouwjaar', 'wijk')

#Filter building year after 1950
heater_info = heater_info.filter(heater_info.pandbouwjaar>1950)
```

Figure A.2: Data preprocessing - Stage 1

```
#Join the tables
join_1 = gasuse_no_summer.join(heater_info, gasuse_no_summer.heater_id == heater_info.HEATER_ID, "inner").drop(heater_info.HEATER_ID)
join_2 = join_1.join(knmi_no_summer, ['Wijk', 'TimeKey'], "inner")
df_joined = join_2.join(house_prop_df, gasuse_with_knmi.heater_id == house_prop_df.HEATER_ID, "inner").drop(house_prop_df.HEATER_ID)
```

Figure A.3: Data preprocessing - Stage 2

```
#Drop duplicate rows in the joined_df
df_joined = df_joined.dropDuplicates()

# Drop duplicate heater IDs for same timestamp as one heater ID can be linked to multiple houses
duplicate_id = df_joined.groupby(['heater_id', 'TimeKey']).count() \
    .where('count > 1').select('heater_id').distinct()
duplicate_id = [row[0] for row in duplicate_id.select('heater_id').collect()]

df = df_joined.filter(~df_joined.heater_id.isin(duplicate_id))

# add temperature difference
df = df.withColumn("temp", df.temp/10)
df = df.withColumn("t_diff", df.t_act - df.temp)

# Replace 24 hours in time to 00 hours
df = df.withColumn('TimeKey', when(df.TimeKey.endswith('24'),F.regexp_replace(df.TimeKey,'(\\d{8})(\\d+)','$1\\00'))\
                              .otherwise(df.TimeKey))

# Converting TimeKey to Timestamp datatype
df = df.withColumn("TimeKey",
                               F.regexp_replace(F.col("TimeKey"), "(\\d{4})(\\d{2})(\\d{2})(\\d+)", "$1-$2-$3 $4:00:00.000")
                              )
df = df.withColumn("TimeKey", to_timestamp(df.TimeKey))

#Extract Day, month, year, hour,
df = df.withColumn('dayOfWeek', dayofweek(col('TimeKey')))
df = df.withColumn('month', month(col('TimeKey')))
df = df.withColumn('year', year(col('TimeKey')))
df = df.withColumn('hour', hour(col('TimeKey')))
df = df.withColumn('dayOfMonth', dayofmonth(col('TimeKey')))

#More filters
df = df.filter(df.t_diff>0)
```

Figure A.4: Data preprocessing - Stage 3

```
#New Period column
df = df.withColumn("period", \
                   when((df.year == 2015) & (df.month > 8), lit("1")) \
                   .when((df.year == 2016) & (df.month < 5), lit("1")) \
                   .when((df.year == 2016) & (df.month > 8), lit("2")) \
                   .when((df.year == 2017) & (df.month < 5), lit("2")) \
                   .when((df.year == 2017) & (df.month > 8), lit("3")) \
                   .when((df.year == 2018) & (df.month < 5), lit("3")) \
                   .when((df.year == 2018) & (df.month > 8), lit("4")) \
                   .when((df.year == 2019) & (df.month < 5), lit("4")) \
                   .when((df.year == 2019) & (df.month > 8), lit("5")) \
                   .when((df.year == 2020) & (df.month < 5), lit("5")) \
      .otherwise(lit("0")) \
  )

#Filter heaters where count of periods >=2
group_1 = df.groupBy('heater_id', 'period').count()
heaters = [row[0] for row in group_1.groupBy('heater_id').count().where('count␣
    >␣1').select('heater_id').collect()]

df = df.filter(df.heater_id.isin(heaters))


#Select columns relevant to further research
df = df.select('heater_id','period','month','dayOfMonth','gas_use','t_diff')

#Aggregate daily data
df = df_test.groupBy('heater_id','period','month','dayOfMonth')\
.agg(
    sum('gas_use').alias('sum_gas'), \
    avg('t_diff').alias('avg_t_diff')
    )

#Save the dataframe to a file
writer = DataFrameWriter(df)
writer.csv('/home/jovyan/files/sum_data.csv.gz', mode="overwrite", header=True,
    compression="gzip")
```

Figure A.5: Data preprocessing - Aggregate Data

A.2 Data Exploration

```
df = session.read.csv('/home/jovyan/files/sum_data.csv.gz', header=True,
    inferSchema=True)

#Show summary
df.summary().show()
df_1 = df.toPandas()
#Number of heaters
len(df_1.heater_id.unique())
#Correlation coefficient between gas use and temperature difference
df.stat.corr('sum_gas', 'avg_t_diff')

#Scatter Plot of gas use and temperature difference
fig=plt.figure(figsize=(10,6))
ax=fig.add_axes([0,0,1,1])
ax.scatter(df_1['avg_t_diff'],df_1['sum_gas'], color='darkblue', s=1.)
ax.set_xlabel('avg_t_diff', fontsize=14)
ax.set_ylabel('sum_gas', fontsize=14)
plt.text(0.5, 70, 'Corr. coef. = {}'.format(round(df.stat.corr('sum_gas', '
    avg_t_diff'), 2)),
         fontsize = 14, color = 'k')
plt.title('Gas use vs. Temperature difference', fontsize=16)
plt.show()

#Pie chart of data distribution in periods
plt.figure(figsize=(10,6))
periods = df_1.period.value_counts().reset_index()
periods.rename({'index':'period',  'period':'count'}, axis=1, inplace=True)
periods['period'] = periods['period'].astype(str)
plt.pie(periods['count'], labels=periods['period'], autopct="%1.1f%%")
plt.legend()

#Line plot of selected heaters
sns.set_theme(style="darkgrid")
plt.figure(figsize=(10,6))
df_4 = df_1[df_1.heater_id.isin([8180, 27729, 5924])]
sns.lineplot(x='period', y='sum_gas', hue='heater_id',  data=df_4, palette='
    dark')
plt.ylabel('sum_gas')
plt.xticks([1, 2, 3, 4, 5])
plt.title('Daily gas use per period', fontsize=16)
plt.show()

#Box plot of period wise gas use and temp. diff.
fig, axs = plt.subplots(2, figsize=(14,10), sharex=True)
sns.boxplot(x='month', y='avg_t_diff', data=df_1, palette='Spectral', ax=axs
    [0])
axs[0].set_title('Temperature difference per month', fontsize=20)
axs[0].set_xlabel('')
sns.boxplot(x='month', y='sum_gas', data=df_1, palette='Spectral', ax=axs[1])
axs[1].set_title('Gas use per month', fontsize=20)
plt.tight_layout()

#Box plot of day-wise gas use
plt.figure(figsize=(14,6))
sns.boxplot(x='dayOfMonth', y='avg_t_diff', data=df_1, palette='Spectral')
plt.show()
```


Figure A.6: Data Exploration

A.3 Linear Regression - Monthly

```
df = session.read.csv("/home/jovyan/files/sum_data.csv.gz", header = True,
    inferSchema = True)
emp_RDD = session.sparkContext.emptyRDD()
columns1 = StructType([StructField('heater_id', StringType(), False),
                       StructField('period', IntegerType(), False),
                       StructField('month', IntegerType(), False),
                       StructField('slope', DoubleType(), False),
                       StructField('slope_diff', DoubleType(), False),
                       StructField('percentage_diff', DoubleType(), False),
                      ])
monthly_slope_df = session.createDataFrame(data=emp_RDD,
                                           schema=columns1
                                          )

#Calculate slopes with slope difference and percentage difference

#Create a list of months
lm = [9,10,11,12,1,2,3,4]

#Create a list of periods available in the heater
periods = df.select('period').distinct().orderBy('period')
periods = periods.select('period').rdd.flatMap(lambda x: x).collect()

heaters = [8736,6361,2508,27729,18680]
```

Figure A.7: Monthly LR - Part 1

```
#Loop over periods, then loop over every month in the period
for heater in heaters:
    df_h = df.filter(df.heater_id == heater)
    for period in periods:
        df_p = df_h.filter(df_h.period == period)
        vectorAssembler = VectorAssembler(inputCols = ['avg_t_diff'], outputCol
            = 'features')
        lr_df = vectorAssembler.transform(df_p)
        lr = LinearRegression(featuresCol = 'features', labelCol='sum_gas')

        for month in lm:
            df_m = lr_df.filter(lr_df.month == month)
            #If number of rows in month is not zero, fit the model
            if(df_m.count() > 0):
                lr_model = lr.fit(df_m)
                #Calculate slope difference if slope for same month in previous
                    period is available, else slope_diff = 0
                if(monthly_slope_df.filter((monthly_slope_df.period == period
                   -1) & (monthly_slope_df.month == month)).count() > 0):
                    prev_slope = monthly_slope_df.filter((monthly_slope_df.
                       period == period-1) & (monthly_slope_df.month == month))
                       .select('slope').rdd.flatMap(lambda x: x).collect()
                    prev_slope = [float(i) for i in prev_slope] #float(
                       prev_slope[0])
                    current_slope = lr_model.coefficients[0]
                    slope_diff = current_slope - prev_slope[0]

                    #If previous slope = 0, set percentage change to 100
                    if (prev_slope[0] != 0):
                        percent_diff = (slope_diff/prev_slope[0])*100
                        rows = [[heater, period, month, str(lr_model.
                           coefficients[0]), str(slope_diff), str(percent_diff)
                           ]]
                        columns = ['heater_id', 'period','month','slope','
                           slope_diff','percentage_diff']
                        second_df = session.createDataFrame(rows, columns)
                        monthly_slope_df = monthly_slope_df.union(second_df)
                    else:
                        percent_diff = 100
                        rows = [[heater, period, month, str(lr_model.
                           coefficients[0]), str(slope_diff), str(percent_diff)
                           ]]
                        columns = ['heater_id', 'period','month','slope','
                           slope_diff','percentage_diff']
                        second_df = session.createDataFrame(rows, columns)
                        monthly_slope_df = monthly_slope_df.union(second_df)

                else:
                    rows = [[heater, period, month, str(lr_model.coefficients
                       [0]), 0, 0]]
                    columns = ['heater_id', 'period','month','slope','
                       slope_diff','percentage_diff']
                    second_df = session.createDataFrame(rows, columns)
                    monthly_slope_df = monthly_slope_df.union(second_df)
```

Figure A.8: Monthly LR - Part 2

```
#Density plot of percentage_diff
percent_diff = monthly_slope_df.select(monthly_slope_df.percentage_diff.cast("
    float"))
pdf_slope = percent_diff.toPandas()
pdf_slope.plot.density()
```

Figure A.9: Monthly LR - Part 3

A.4 Linear Regression - Period-wise

```
df_main = session.read.csv("/home/jovyan/files/sum_data.csv.gz", header = True,
    inferSchema = True)

df = df_main.toPandas()

df_plot = df[df['heater_id'] == 24171]
df_plot1 = df_plot[df_plot['period'] == 1]
df_plot2 = df_plot[df_plot['period'] == 2]
df_plot3 = df_plot[df_plot['period'] == 3]
df_plot4 = df_plot[df_plot['period'] == 4]
df_plot5 = df_plot[df_plot['period'] == 5]


fig=plt.figure(figsize=(10,6))
plt.scatter(df_plot1.avg_t_diff, df_plot1.sum_gas, s= 5, color = 'red', alpha =
    0.5, label = 'Period 1')
plt.scatter(df_plot2.avg_t_diff, df_plot2.sum_gas, s= 5, color = 'blue', alpha
    = 0.5, label = 'Period 2')
plt.scatter(df_plot3.avg_t_diff, df_plot3.sum_gas, s= 5, color = 'yellow',
    alpha = 0.5, label = 'Period 3')
plt.scatter(df_plot4.avg_t_diff, df_plot4.sum_gas, s= 5, color = 'green', alpha
    = 0.5, label = 'Period 4')
plt.scatter(df_plot5.avg_t_diff, df_plot5.sum_gas, s= 5, color = 'grey', alpha
    = 0.5, label = 'Period 5')
plt.legend()
plt.title("Scatter plot of gas use vs temperature difference for heater id
    24171")
plt.xlabel('Tempertaure Difference')
plt.ylabel('Daily gas use')
plt.show()

#Obtain a list of heaters
heaters = df['heater_id'].unique().tolist()
```

Figure A.10: Period LR - Part 1

```
#Create an empty list to store slopes
slope = []
#Loop over heaters and periods to calculate slope for each period
for heater_id in heaters:
    df_h = df[df['heater_id'] == heater_id]
    for period in range(1,6):
        df_p = df_h[df_h['period'] == period]
        if(df_p.shape[0]>0):
            x = df_p[["avg_t_diff"]]
            y = df_p[["sum_gas"]]
            model = LinearRegression().fit(x, y)
            if(period == 1 ):
                if(abs(model.coef_[0][0]) >= 0.1 ):
                    sp1 = model.coef_[0][0]
                else:
                    sp1 = None
            elif(period == 2):
                if(abs(model.coef_[0][0]) >= 0.1 ):
                    sp2 = model.coef_[0][0]
                else:
                    sp2 = None
            elif(period == 3):
                if(abs(model.coef_[0][0]) >= 0.1 ):
                    sp3 = model.coef_[0][0]
                else:
                    sp3 = None
            elif(period == 4):
                if(abs(model.coef_[0][0]) >= 0.1 ):
                    sp4 = model.coef_[0][0]
                else:
                    sp4 = None
            elif(period == 5):
                if(abs(model.coef_[0][0]) > 0.1 ):
                    sp5 = model.coef_[0][0]
                else:
                    sp5 = None
        else:
            if(period == 1):
                sp1 = None
            elif(period == 2):
                sp2 = None
            elif(period == 3):
                sp3 = None
            elif(period == 4):
                sp4 = None
            elif(period == 5):
                sp5 = None
    L = [heater_id, sp1, sp2, sp3, sp4, sp5]
    slope.append(L)

cols = ['heaterid', 'slope_p1', 'slope_p2', 'slope_p3', 'slope_p4','slope_p5']
slope_df = pd.DataFrame(slope, columns=cols)
slope_df = slope_df.apply(pd.to_numeric)
```

Figure A.11: Period LR - Part 2

```
#Function to calculate differences between slopes of periods for a heater
def differences (row) :
    if(row['slope_p1'] == None or row['slope_p2'] == None):
        row['diff_1'] = None
    else:
        row['diff_1'] = row['slope_p2'] - row['slope_p1']
        if(row['slope_p1'] == 0):
            row['pdiff_1'] = None
        else:
            row['pdiff_1'] = (row['diff_1']/row['slope_p1'])*100
    if(row['slope_p2'] == None or row['slope_p3'] == None):
        row['diff_2'] = None
    else:
        row['diff_2'] = row['slope_p3'] - row['slope_p2']
        if(row['slope_p2'] == 0):
            row['pdiff_2'] = None
        else:
            row['pdiff_2'] = (row['diff_2']/row['slope_p2'])*100
    if(row['slope_p3'] == None or row['slope_p4'] == None):
        row['diff_3'] = None
    else:
        row['diff_3'] = row['slope_p4'] - row['slope_p3']
        if(row['slope_p3'] == 0):
            row['pdiff_3'] = None
        else:
            row['pdiff_3'] = (row['diff_3']/row['slope_p3'])*100
    if(row['slope_p5'] == None or row['slope_p4'] == None):
        row['diff_4'] = None
    else:
        row['diff_4'] = row['slope_p5'] - row['slope_p4']
        if(row['slope_p4'] == 0):
            row['pdiff_4'] = None
        else:
            row['pdiff_4'] = (row['diff_4']/row['slope_p4'])*100

    return row

slope_df = slope_df.apply(lambda row: differences(row), axis=1)

slope_df.to_csv('/home/jovyan/varoon/slopes.csv', header=True, na_rep = "null")
```

Figure A.12: Period LR - Part 3

A.5 Evaluating Results

```
detected heaters = df = pd.read_csv("/home/jovyan/varoon/slopes.csv")

#Filter positive slopes
df = df[ ((df['slope_p1'] > 0) | (df.slope_p1.isnull()))
            & ((df['slope_p2'] > 0) | (df.slope_p2.isnull()))
            & ((df['slope_p3'] > 0) | (df.slope_p3.isnull()))
            & ((df['slope_p4'] > 0) | (df.slope_p4.isnull()))
            & ((df['slope_p5'] > 0) | (df.slope_p5.isnull()))
            ]

tdf = df.describe()
tdf.round(3)
```

Figure A.13: Results Summary

```
#Density plot with filtering between -100 and +100
df.pdiff_1.plot.density(color = "green")
plt.title('Density plot - percentage change between 1 and 2')
plt.xlim(-100,100)
plt.grid()
plt.xlabel('percentage change')
plt.axvline(df.pdiff_1.mean() - df.pdiff_1.std(), color = 'r', linestyle = '
    dashed')
plt.axvline(df.pdiff_1.mean() + df.pdiff_1.std(), color = 'r', linestyle = '
    dashed')
plt.axvline(df.pdiff_1.mean() - 2*df.pdiff_1.std(), color = 'b', linestyle = '
    dashed')
plt.show()
```

Figure A.14: Density Plot 1

```
#Density plot with filtering between -100 and +100
df.pdiff_2.plot.density(color = "green")
plt.title('Density plot - percentage change between 2 and 3')
plt.xlim(-100,100)
plt.grid()
plt.xlabel('percentage change')
plt.axvline(df.pdiff_2.mean() - df.pdiff_2.std(), color = 'r', linestyle = '
    dashed')
plt.axvline(df.pdiff_2.mean() + df.pdiff_2.std(), color = 'r', linestyle = '
    dashed')
plt.axvline(df.pdiff_2.mean() - 2*df.pdiff_2.std(), color = 'b', linestyle = '
    dashed')
plt.show()
```

Figure A.15: Density Plot 2

```
#Density plot with filtering between -100 and +100
df.pdiff_3.plot.density(color = "green")
plt.title('Density plot - percentage change between 3 and 4')
plt.xlim(-100,100)
plt.grid()
plt.xlabel('percentage change')
plt.axvline(df.pdiff_3.mean() - df.pdiff_3.std(), color = 'r', linestyle = '
    dashed')
plt.axvline(df.pdiff_3.mean() + df.pdiff_3.std(), color = 'r', linestyle = '
    dashed')
plt.axvline(df.pdiff_3.mean() - 2*df.pdiff_3.std(), color = 'b', linestyle = '
    dashed')
plt.show()
```

Figure A.16: Density Plot 3

```
#Density plot with filtering between -100 and +100
df.pdiff_4.plot.density(color = "green")
plt.title('Density plot - percentage change between 4 and 5')
plt.xlim(-100,100)
plt.grid()
plt.xlabel('percentage change')
plt.axvline(df.pdiff_4.mean() - df.pdiff_4.std(), color = 'r', linestyle = '
    dashed')
plt.axvline(df.pdiff_4.mean() + df.pdiff_4.std(), color = 'r', linestyle = '
    dashed')
plt.axvline(df.pdiff_4.mean() - 2*df.pdiff_4.std(), color = 'b', linestyle = '
    dashed')
plt.show()
```

Figure A.17: Density Plot 4

```
#Filter based on mean - 1std and mean + 1std
change4 = df[((df.pdiff_4 < (df.pdiff_4.mean() - df.pdiff_4.std())) & (df.pdiff_4 > (df.pdiff_4.mean() - 2*df.pdiff_4.std())) )
            & ( ((df.pdiff_1 > (df.pdiff_1.mean() - df.pdiff_1.std())) & (df.pdiff_1 < (df.pdiff_1.mean() + df.pdiff_1.std()))) | (df.pdiff_1.isnull()) )
            & ( ((df.pdiff_2 > (df.pdiff_2.mean() - df.pdiff_2.std())) & (df.pdiff_2 < (df.pdiff_2.mean() + df.pdiff_2.std()))) | (df.pdiff_2.isnull()) )
            & ( ((df.pdiff_3 > (df.pdiff_3.mean() - df.pdiff_3.std())) & (df.pdiff_3 < (df.pdiff_3.mean() + df.pdiff_3.std()))) | (df.pdiff_3.isnull()) )
           ]


change3 = df[((df.pdiff_3 < (df.pdiff_3.mean() - df.pdiff_3.std())) & (df.pdiff_3 > (df.pdiff_3.mean() - 2*df.pdiff_3.std()))  )
            & ( ((df.pdiff_1 > (df.pdiff_1.mean() - df.pdiff_1.std())) & (df.pdiff_1 < (df.pdiff_1.mean() + df.pdiff_1.std()))) | (df.pdiff_1.isnull()) )
            & ( ((df.pdiff_2 > (df.pdiff_2.mean() - df.pdiff_2.std())) & (df.pdiff_2 < (df.pdiff_2.mean() + df.pdiff_2.std()))) | (df.pdiff_2.isnull()) )
            & ( ((df.pdiff_4 > (df.pdiff_4.mean() - df.pdiff_4.std())) & (df.pdiff_4 < (df.pdiff_4.mean() + df.pdiff_4.std()))) | (df.pdiff_4.isnull()) )
           ]

change2 = df[((df.pdiff_2 < (df.pdiff_2.mean() - df.pdiff_2.std())) & (df.pdiff_2 > (df.pdiff_2.mean() - 2*df.pdiff_2.std()))  )
            & ( ((df.pdiff_1 > (df.pdiff_1.mean() - df.pdiff_1.std())) & (df.pdiff_1 < (df.pdiff_1.mean() + df.pdiff_1.std()))) | (df.pdiff_1.isnull()) )
            & ( ((df.pdiff_3 > (df.pdiff_3.mean() - df.pdiff_3.std())) & (df.pdiff_3 < (df.pdiff_3.mean() + df.pdiff_3.std()))) | (df.pdiff_3.isnull()) )
            & ( ((df.pdiff_4 > (df.pdiff_4.mean() - df.pdiff_4.std())) & (df.pdiff_4 < (df.pdiff_4.mean() + df.pdiff_4.std()))) | (df.pdiff_4.isnull()) )
           ]

change1 = df[((df.pdiff_1 < (df.pdiff_1.mean() - df.pdiff_1.std())) & (df.pdiff_1 > (df.pdiff_1.mean() - 2*df.pdiff_1.std()))  )
            & ( ((df.pdiff_2 > (df.pdiff_2.mean() - df.pdiff_2.std())) & (df.pdiff_2 < (df.pdiff_2.mean() + df.pdiff_2.std()))) | (df.pdiff_2.isnull()) )
            & ( ((df.pdiff_3 > (df.pdiff_3.mean() - df.pdiff_3.std())) & (df.pdiff_3 < (df.pdiff_3.mean() + df.pdiff_3.std()))) | (df.pdiff_3.isnull()) )
            & ( ((df.pdiff_4 > (df.pdiff_4.mean() - df.pdiff_4.std())) & (df.pdiff_4 < (df.pdiff_4.mean() + df.pdiff_4.std()))) | (df.pdiff_4.isnull()) )
           ]

detected_heaters = pd.concat([change1,change2,change3,change4])

detected_heaters.to_csv('/home/jovyan/varoon/detected_heaters.csv', header=True, na_rep = "null")
```


Figure A.18: Filter and Save Detected Heaters

```
#For heater id 25007
df_25007 = detected_heaters[detected_heaters['heaterid'] == 25007]
df_25007

fig, ax = plt.subplots()
ax.axline((0, 0), slope=float(df_25007.slope_p1), color='C0', label='slope_p1')
ax.axline((0, 0), slope=float(df_25007.slope_p2), color='C1', label='slope_p2')
ax.axline((0, 0), slope=float(df_25007.slope_p3), color='C2', label='slope_p3')
ax.axline((0, 0), slope=float(df_25007.slope_p4), color='C3', label='slope_p4')
ax.axline((0, 0), slope=float(df_25007.slope_p5), color='C4', label='slope_p5')
ax.set_xlim(0, )
plt.title("Slope Comparisoon for Heater 25007")
plt.xlabel("Temperature Difference")
plt.ylabel("Daily Gas Use")
ax.set_ylim(0, )
ax.legend()
plt.grid()
fig.set_figheight(8.0)
fig.set_figwidth(12.0)

fig, ax = plt.subplots()
x = [0,1,2,3,4]
y = [0,float(df_25007.pdiff_1), float(df_25007.pdiff_2), float(df_25007.pdiff_3
    ), float(df_25007.pdiff_4)]
plt.plot(x,y)
plt.scatter(x,y,color='red')
plt.title("Percentage Difference between Periods for Heater 25007")
plt.xlabel("Difference Period (1 is difference between period 1 and 2)")
plt.ylabel("Percentage Difference")
ax.set_xlim(0, 5)
ax.set_ylim(-80, 25)
plt.grid()
fig.set_figheight(8.0)
fig.set_figwidth(12.0)
```

Figure A.19: Evaluate Heater 25007

```
#For heater id 30549
df_30549 = detected_heaters[detected_heaters['heaterid'] == 30549]
df_30549

fig, ax = plt.subplots()
ax.axline((0, 0), slope=float(df_30549.slope_p1), color='C0', label='slope_p1')
ax.axline((0, 0), slope=float(df_30549.slope_p2), color='C1', label='slope_p2')
ax.axline((0, 0), slope=float(df_30549.slope_p3), color='C2', label='slope_p3')
ax.axline((0, 0), slope=float(df_30549.slope_p4), color='C3', label='slope_p4')
ax.axline((0, 0), slope=float(df_30549.slope_p5), color='C4', label='slope_p5')
ax.set_xlim(0, )
plt.title("Slope Comparisoon for Heater 30549")
plt.xlabel("Temperature Difference")
plt.ylabel("Daily Gas Use")
ax.set_ylim(0, )
ax.legend()
plt.grid()
fig.set_figheight(8.0)
fig.set_figwidth(12.0)

fig, ax = plt.subplots()
x = [0,1,2,3,4]
y = [0,float(df_30549.pdiff_1), float(df_30549.pdiff_2), float(df_30549.pdiff_3
    ), float(df_30549.pdiff_4)]
plt.plot(x,y)
plt.scatter(x,y,color='red')
plt.title("Percentage Difference between Periods for Heater 30549")
plt.xlabel("Difference Period (1 is difference between period 1 and 2)")
plt.ylabel("Percentage Difference")
ax.set_xlim(0, 5)
ax.set_ylim(-80, 25)
plt.grid()
fig.set_figheight(8.0)
fig.set_figwidth(12.0)
```

Figure A.20: Evaluate Heater 30549

```
#For heater id 18680
df_18680 = detected_heaters[detected_heaters['heaterid'] == 18680]
df_18680

fig, ax = plt.subplots()
ax.axline((0, 0), slope=float(df_18680.slope_p1), color='C0', label='slope_p1')
ax.axline((0, 0), slope=float(df_18680.slope_p2), color='C1', label='slope_p2')
ax.axline((0, 0), slope=float(df_18680.slope_p3), color='C2', label='slope_p3')
ax.axline((0, 0), slope=float(df_18680.slope_p4), color='C3', label='slope_p4')
ax.axline((0, 0), slope=float(df_18680.slope_p5), color='C4', label='slope_p5')
ax.set_xlim(0, )
plt.title("Slope Comparisoon for Heater 18680")
plt.xlabel("Temperature Difference")
plt.ylabel("Daily Gas Use")
ax.set_ylim(0, )
ax.legend()
plt.grid()
fig.set_figheight(8.0)
fig.set_figwidth(12.0)

fig, ax = plt.subplots()
x = [0,1,2,3,4]
y = [0,float(df_18680.pdiff_1), float(df_18680.pdiff_2), float(df_18680.pdiff_3
    ), float(df_18680.pdiff_4)]
plt.plot(x,y)
plt.scatter(x,y,color='red')
plt.title("Percentage Difference between Periods for Heater 18680")
plt.xlabel("Difference Period (1 is difference between period 1 and 2)")
plt.ylabel("Percentage Difference")
ax.set_xlim(0, 5)
ax.set_ylim(-60, 35)
plt.grid()
fig.set_figheight(8.0)
fig.set_figwidth(12.0)
```

Figure A.21: Evaluate Heater 18680

```
#For heater id 14237
df_14237 = detected_heaters[detected_heaters['heaterid'] == 14237]
df_14237

fig, ax = plt.subplots()
ax.axline((0, 0), slope=float(df_14237.slope_p1), color='C0', label='slope_p1')
ax.axline((0, 0), slope=float(df_14237.slope_p2), color='C1', label='slope_p2')
ax.axline((0, 0), slope=float(df_14237.slope_p3), color='C2', label='slope_p3')
ax.axline((0, 0), slope=float(df_14237.slope_p4), color='C3', label='slope_p4')
ax.axline((0, 0), slope=float(df_14237.slope_p5), color='C4', label='slope_p5')
ax.set_xlim(0, )
plt.title("Slope Comparisoon for Heater 14237")
plt.xlabel("Temperature Difference")
plt.ylabel("Daily Gas Use")
ax.set_ylim(0, )
ax.legend()
plt.grid()
fig.set_figheight(8.0)
fig.set_figwidth(12.0)

fig, ax = plt.subplots()
x = [0,1,2,3,4]
y = [0,float(df_14237.pdiff_1), float(df_14237.pdiff_2), float(df_14237.pdiff_3
    ), float(df_14237.pdiff_4)]
plt.plot(x,y)
plt.scatter(x,y,color='red')
plt.title("Percentage Difference between Periods for Heater 14237")
plt.xlabel("Difference Period (1 is difference between period 1 and 2)")
plt.ylabel("Percentage Difference")
ax.set_xlim(0, 5)
ax.set_ylim(-60, 45)
plt.grid()
fig.set_figheight(8.0)
fig.set_figwidth(12.0)
```

Figure A.22: Evaluate Heater 14237

```
#For heater id 27729
df_27729 = df[df['heaterid'] == 27729]
df_27729

fig, ax = plt.subplots()
ax.axline((0, 0), slope=float(df_27729.slope_p1), color='C0', label='slope_p1')
ax.axline((0, 0), slope=float(df_27729.slope_p2), color='C1', label='slope_p2')
ax.axline((0, 0), slope=float(df_27729.slope_p3), color='C2', label='slope_p3')
ax.axline((0, 0), slope=float(df_27729.slope_p4), color='C3', label='slope_p4')
ax.axline((0, 0), slope=float(df_27729.slope_p5), color='C4', label='slope_p5')
ax.set_xlim(0, )
plt.title("Slope Comparisoon for Heater 27729")
plt.xlabel("Temperature Difference")
plt.ylabel("Daily Gas Use")
ax.set_ylim(0, )
ax.legend()
plt.grid()
fig.set_figheight(8.0)
fig.set_figwidth(12.0)

fig, ax = plt.subplots()
x = [0,1,2,3,4]
y = [0,float(df_27729.pdiff_1), float(df_27729.pdiff_2), float(df_27729.pdiff_3
    ), float(df_27729.pdiff_4)]
plt.plot(x,y)
plt.scatter(x,y,color='red')
plt.title("Percentage Difference between Periods for Heater 27729")
plt.xlabel("Difference Period (1 is difference between period 1 and 2)")
plt.ylabel("Percentage Difference")
ax.set_xlim(0, 5)
ax.set_ylim(-80, 25)
plt.grid()
fig.set_figheight(8.0)
fig.set_figwidth(12.0)
```

Figure A.23: Evaluate Heater 27729

B Full Data Exploration Results

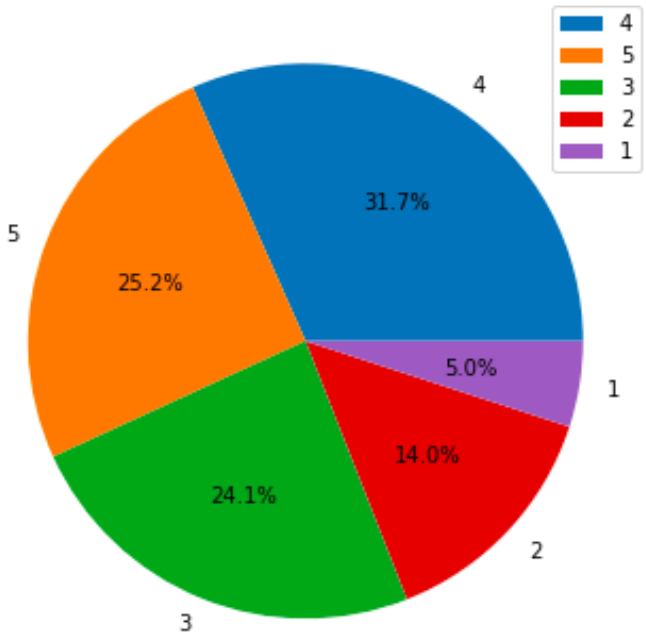

Figure B.1: Distribution of data by heating periods.

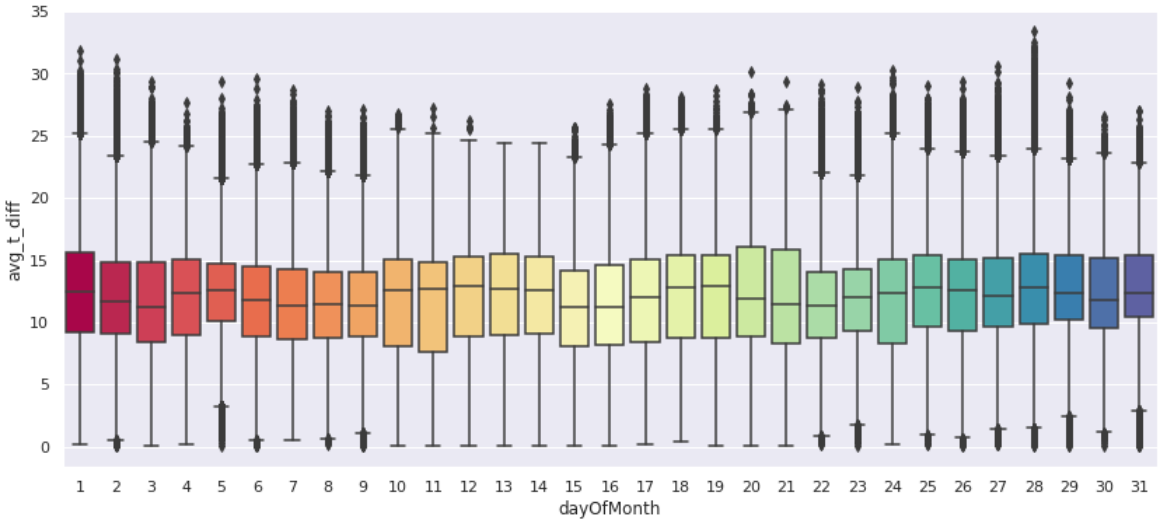

Figure B.2: Boxplots of temperature difference by day of month.

C Full Analysis Results

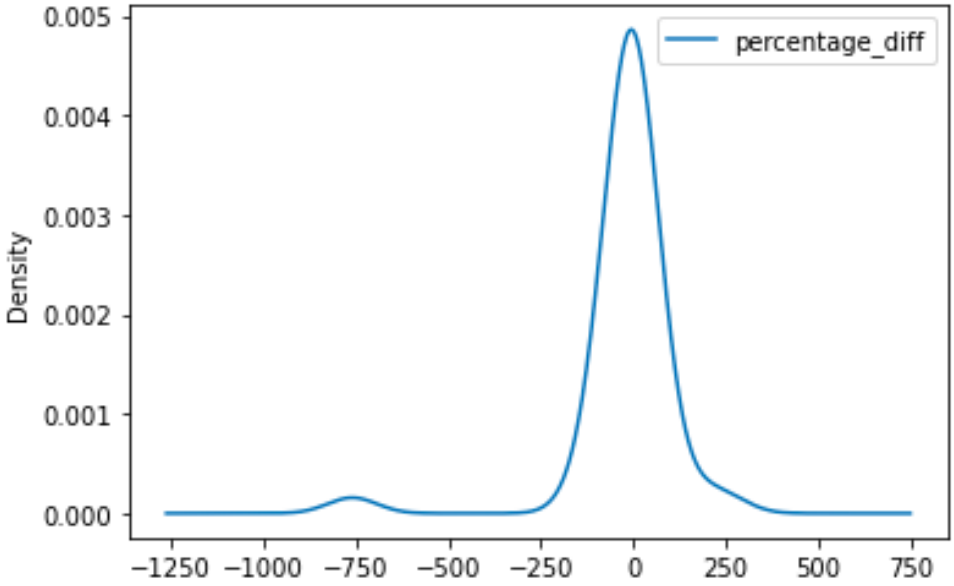

Figure C.1: Density plot of Percentage Difference for Monthly LR.

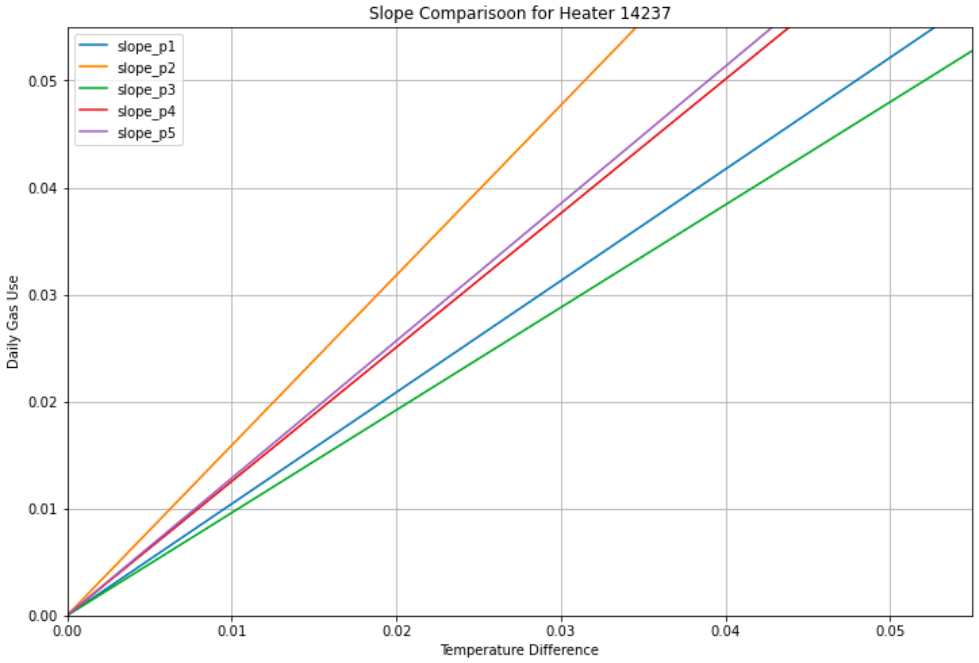

Figure C.2: Slope comparison for heater ID 14237.

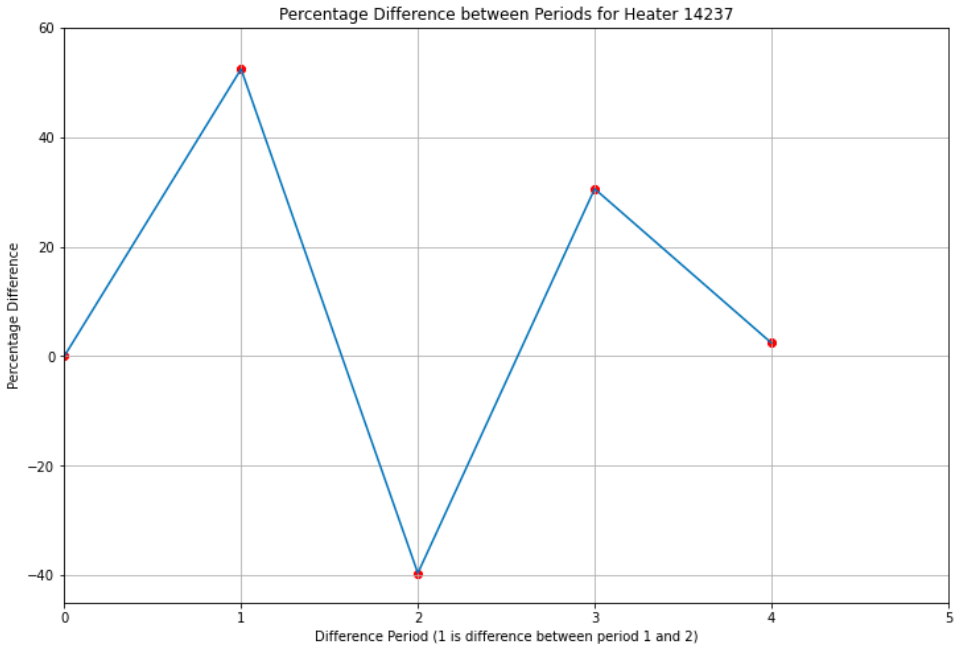

Figure C.3: Percentage Difference comparison for heater ID 14237.

heater_id	period	month	slope	slope_diff	percentage_diff
8736	1	10	0.35249986054487253	0.0	0.0
8736	1	11	0.40335912189386447	0.0	0.0
8736	1	12	0.43153232925545515	0.0	0.0
8736	1	1	0.33665134679458264	0.0	0.0
8736	1	2	0.3693107313248875	0.0	0.0
8736	1	3	0.5680570726775284	0.0	0.0
8736	1	4	0.22163216762777968	0.0	0.0
8736	2	9	0.037372804714753294	0.0	0.0
8736	2	10	0.19811884360330306	-0.15438101694156947	-43.7960505013922
8736	2	11	0.37441568271931075	-0.02894343917455372	-7.1756005017706235
8736	2	12	0.35416737243571517	-0.07736495681973998	-17.92796311535261
8736	2	1	0.43593882334379874	0.0992874765492161	29.492671719444857
8736	2	2	0.3195771507461685	-0.04973358057871902	-13.46659502698494
8736	2	3	0.29690888330283455	-0.2711481893746938	-47.732561113382665
8736	2	4	-0.05202072062030315	-0.2736528882480828	-123.47164726903243
8736	3	9	0.12931255206457437	0.09193974734982108	246.00708470115683
8736	3	10	0.30132973963080745	0.10321089602750438	52.09544642515955
8736	3	11	0.3178618294465116	-0.05655385327279916	-15.104563158802312
8736	3	12	0.46794267679940954	0.11377530436369437	32.12472780347538
8736	3	1	0.2757090534576732	-0.16022976988612553	-36.755104456425514
8736	3	2	0.3669660808377483	0.047388930091579795	14.828635270366854
8736	3	3	0.40850100639838227	0.11159212309554772	37.58463601836004
8736	3	4	0.34341845194410636	0.3954391725644095	-760.157044825853
8736	4	9	0.0649668346623583	-0.06434571740221608	-49.7598387587958
8736	4	10	0.2510522582103076	-0.05027748142049987	-16.68520388399114
8736	4	11	0.35454150629165376	0.03667967684514217	11.539503471999764
8736	4	12	0.3494398734482123	-0.11850280335119723	-25.324213675427426
8736	4	1	0.394598938028412	0.11888988457073879	43.12150184396855
8736	4	2	0.4034285688243085	0.03646248798656021	9.936201161513308
8736	4	3	1.0558416106042139	0.6473406042058316	158.46732176089816
8736	4	4	0.27730357612304624	-0.06611487582106013	-19.251987028297698
8736	5	9	0.019253409517664452	-0.04571342514469384	-70.36424874672268
8736	5	10	0.13345304034471833	-0.11759921786558925	-46.84252541838356
8736	5	11	0.2575162240772426	-0.09702528221441115	-27.366409995053154
8736	5	12	0.43490899051357784	0.08546911706536553	24.458890802005804
8736	5	1	0.4862907836005553	0.09169184557214327	23.236718788518697
8736	5	2	0.5090234571171844	0.10559488829287594	26.17437049651833
8736	5	3	0.0	-1.0558416106042139	-100.0

Table C.1: Complete results of Monthly LR for heater id 8736