Irish Dairy Industry Investigation

Assignment 2 | 22122

Brendan Crowley

Contents

[**Data Preparation & Visualization** 2](#_Toc104051667)

[**Acquiring raw data** 2](#_Toc104051668)

[**Exploratory Data Analysis** 3](#_Toc104051669)

[**Milk Statistics** 3](#_Toc104051670)

[**Heifer Cow** 3](#_Toc104051671)

[**Netherlands dataset** 4](#_Toc104051672)

[**Resolving the issues** 4](#_Toc104051673)

[**Goal of all transformations** 4](#_Toc104051674)

[**Raw Milk dataset** 4](#_Toc104051675)

[**Heifer cow** 5](#_Toc104051676)

[**Merging with Milk Statistics** 5](#_Toc104051677)

[**Dashboard** 6](#_Toc104051678)

[**Statistics** 7](#_Toc104051679)

[**Statistics on Irish data** 7](#_Toc104051680)

[**Raw milk values** 7](#_Toc104051681)

[**Normal distribution** 8](#_Toc104051682)

[**Comparisons between countries** 9](#_Toc104051683)

[**Tests** 9](#_Toc104051684)

[**Results** 10](#_Toc104051685)

[**Interpret results** 11](#_Toc104051686)

[**Machine Learning** 13](#_Toc104051687)

[**Sentiment analysis** 13](#_Toc104051688)

[**Data source** 13](#_Toc104051689)

[**Methodology** 13](#_Toc104051690)

[**Results** 14](#_Toc104051691)

[**Machine Learning** 15](#_Toc104051692)

[**Learning type** 15](#_Toc104051693)

[**Models used** 15](#_Toc104051694)

[**Hyper-parameter optimization** 15](#_Toc104051695)

[**Results** 16](#_Toc104051696)

[Table of Figures 20](#_Toc104051697)

[Bibliography 21](#_Toc104051698)

# **Data Preparation & Visualization**

For this project, the topic was to investigate a data analysis project related to an agricultural topic of my choosing. In this project, I have decided to investigate the variance in the value of raw milk, with respect to the inputs and outputs that could affect (e.g. price of feed, birth and selling of milking cows, production of dairy-derived products).

Code related to this project is available in the “DataViz.ipynb” and “MachineLearning.ipynb” (for the associated Dashboard code) Jupyter Notebook included for submission with this file.

## **Acquiring raw data**

The major source of statistical information stemmed from the data posted by the Central Statistics Office in Ireland (Office, n.d.), which detailed not only the specific quantities of milk produced and processed into other dairy products, but also the EU-wide figures for processed milk, as well as reports to rationalize and explain the variance in figures year-by-year. From this source, most of the milk statistical information was obtained.

To further the machine learning aspect of the project, it was decided to obtain further information related to the topic, due to the limited data provided by the Milk Statistic dataset solely. To obtain the data, it was decided to continue to use the CSO website for sources. The data.cso.ie portal provides access to the data provided on the main CSO website, in the form of downloadable files. These data tables can be interactively filtered and formatted on the site to select only the target data, as well as the amount of historic data required. From this, we obtained a dataset for Heifer Cow purchases (heifer cows being the primary source of dairy generation). (Office, n.d.)

For our international dataset for comparison, Netherlands was chosen as a comparing country. This is due to the similar production levels of dairy between the two countries, as well as the available information that was comparable to the data provided above. This data was obtained from the Centraal Bureau voor de Statistiek in the Netherlands, who have compiled data for the last 30 years with regards to milk production (Anon., n.d.)

For the final figure, the target value for machine learning was required. For this data, the European Commission provide a month-by-month breakdown of the average price (excluding V.A.T.) of raw milk in each country in the EU (with previous reports tracking United Kingdom figures). This data was

All datasets are stored in a singular Excel spreadsheet, with all original formatting kept.

Each dataset has been obtained under an open data license from each issuing body (CSO, CBS, European Commission)

## **Exploratory Data Analysis**

With the datasets selected, an evaluation of the data that was obtained for the project was to be made, to identify issues and investigate solutions into their problems.

### **Milk Statistics**

Milk Statistics is the dataset with the greatest amount of available data, containing 9 unique features for data. The format of this data varies, but falls under 3 distinct types: Thousand tonnes, Percent, Million Litres. The general format of the data is a float with one decimal place, with values of Percent having 2 decimal places instead.

The data is organized by year into a row of 12 values; Year, Description, Unit and the 12 figures reported monthly for each.

For this dataset, there are also some missing values for the features, owing to some figures remaining confidential for years after their obtaining, with the majority of these issues stemming from the latter years of the dataset.

However, a simple imputation of the missing variables is technically not a viable option for some features. For Imported milk intake, missing variables are present throughout the feature, most conglomerated in the last 2 years of data. However, upon evaluating the amount reported to the EU with regards to total raw milk processed for those 2 years, we can see that the value in the Domestic milk intake is equal to the figure the EU report. Thus, it is theorized that, for the latter years of the period, Ireland simply did not import any milk for processing, against it being missing data.

This logic matches real-world sentiment at the time; the destination for Ireland’s dairy imports prior to 2020 was the UK, with the latest-known figures showing that 87% of milk imports for 2018 came from the UK, with this exchange diminishing the closer the UK came to leaving the EU and thus losing the trade agreements commonplace between EU members. This theory is also bolstered by the temporary halt of shipping as a result of the COVID-19 pandemic.

For this project, the above theory will be assumed as fact, and the feature will not be estimated when imputing certain values.

### **Heifer Cow**

For the Heifer Cow dataset, no imputation is required; no value is omitted, and being a single-value dataset, there is no concern with regards to this.

The only processing required will be to format the dataset into one that matches the Milk Statistic dataset, through pivoting and melting the dataframe.

### **Netherlands dataset**

The data downloaded from the website can be filtered to only include specific values. In this context, I can choose to only download the values that have a corresponding value in the other datasets (e.g. I chose to remove a Whey Protein Powder feature, as there was no corresponding value in the other datasets).

One key difference in this dataset is how certain values are interpreted vs the Ireland dataset. Here, “Volume” denotes the total milk for processing (domestic + import), while concentrated milk sales denote the sum of whole milk and skimmed milk sales, all of which are separate features in the Milk Statistics dataset.

## **Resolving the issues**

For this section, the steps taken to format all datasets will be described and the logic and reasoning specified for the action taken. This section is to explain the logic and rationale behind the steps taken for the issues identified in the previous section. The code used to apply the logic is contained in the Jupyter notebook named “DataVis.ipynb”.

### **Goal of all transformations**

Initially, prior to merging datasets, all datasets should match the format of the Milk Statistics dataset; each category is separated by year, and each entry contains 12 values that represent the output in the month (January, February etc.). With the base datasets, it was imperative to use Pandas pivot and melt functions to transform the data to the required format. The following is the final format that each dataset should be in prior to merging with Milk Statistics.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Year** | **Category** | **Unit** | **Jan** | **Feb** | **Mar** | **Apr** | **May** | **Jun** | **Jul** | **Aug** | **Sep** | **Oct** | **Nov** | **Dec** |

Table 1 Descriptor of the headers of the targeted formatting for each table in the system

### **Raw Milk dataset**

1. For the Raw Milk dataset, the dataset was filtered to only contain the entries for Ireland and Netherlands. This was done because only that information was required at this moment for processing. (240 x 3)
2. Due to the Year being formatted incorrectly, it was necessary to format it to the right values. Using the built-in function split on the delimiting character, Year was converted to Year and Month, where it was possible to extract the Month value into its own feature. The rationale behind this is to A). convert the data into a format that matches the target, and B). allow us to use the feature for pivoting in future steps. (240 x 4)
3. After the above, the dataset was melted on the Year and Month features, converting it into a dataset where each row points to a Year, Month, a named country, and the value of raw milk. The rationale for this is so the data could be pivoted correctly in the future step, to allow it to match the format above. (480 x 4)
4. Finally, the dataset was pivoted on the Month feature, such that it could match the format of the Milk Statistics dataset. This was done to allow the dataset to be formatted correctly in the desired system. With this, each row now contains the correct values for the associated month, whilst having the information contained in a singular month, for merging with the MS dataset. (40 x 15)

### **Heifer cow**

1. Like the Raw Milk, the Heifer Cow had an incorrectly formatted Year feature that merged the Year and Month into a singular. The first step taken was to split this into 2 features, storing the month in the Month value. As before,
2. With no need to melt the dataset, due to already being in the format required for melting without other operations, a pivot is done on the Month feature with the Value feature as the value variable. With this, the data format as described above is reached.

### **Merging with Milk Statistics**

There were 2 datasets derived from these operations (one combining all 4 datasets, the second omitting Meal Price). Each dataset follows a similar operation set-up, and thus only the steps taken will be discussed here.

1. Concatenate the datasets with each other. This should include the Raw Milk (using only the Irish data), Heifer Cow, Meal Price (if applicable) and Milk Statistics.
2. Melt the datasets on Year, Category & Unit. As the intended outcome of the dataset is to be read such that each row contains Year, Month and the Category features as values, this first step allows us to divide the Category into 12 separate rows (one per month). This will allow us later to pivot the data and organize it by month per row.
3. Pivot the dataset. The index should be of the format Year-Month-Category (as all values are repeated multiple times, there would be no reference if only one was used). This is done as the pivot function does not contain methods to retain this information, and using these methods allows us to both pivot and keep the information referenced to the row. By pivoting this way, it initialises the target for the final dataset. Each row currently contains the index, and a feature of all the values in Category. For each row, every entry after the Index is np.NaN, save for the value that corresponds to the Category (e.g. 2002-01-Butter denotes the Butter feature being the only one currently filled.
4. Using the index, we split the Year and Month back into usable features.
5. Following this, we group all the rows by Year and Month, and sum the grouping. The resulting data would now represent a singular row for the month, containing the values of all the feature data from the rows from the pivot. This is done for dimensionality reduction and to logically display the information (e.g. for 2002-01 you will see all the data reported for that month in a singular row).
6. Finally, we impute the missing data, using the KNNImputer. For this imputation, we ignore the Import milk intake feature, as discussed earlier. We do imputation as the information omitted in the other features has simply not been reported by the governing bodies, and the assignment of where/when the data is not reported is not consistent enough to safely say the values are indeed 0.

## **Dashboard**

To attempt this task, an investigation into the best dashboard for displaying the results and systems used was made. With the constraints of the assignment and the limitations of the construction tools provided, a decision was made to instead use a module designed for simple execution with dashboard interface.

The module chosen was ExplainerDashboard, a dashboard module that took fitted models and target data as input, and broke down the calculations provided, the performance involved, as well as providing other features.

The highlighted feature provided by this Dashboard is an interactive tab that allows the user to input data to show the overall effect of certain features with regards to their effect on the predicted result, as well as the actual influence the feature had in that prediction. This feature would almost certainly conform to excellent descriptive nature for a presentation. In this example, a dashboard on the Irish dataset could allow us to show the overall effect on if the volume was increased/decreased.

This dashboard was designed such that the default appearance for all graphs is a 2-axis graph plotting 2 figures against each other. As such, this dashboard, without any modifications, can easily and clearly show the data required in the desired format.

For this project, both the ExplainderDashboard and the ExplainerHub methods will be used. The ExplainerDashboard methods will be shown as outputs of Jupyter notebook cells, while the ExplainerHub will initialise locally on the computer, prompting the user to launch a URL in a separate tab. This is done to showcase the multiple methods that can be used with this module.

# **Statistics**

In this section, the statistical analysis of the data studied and worked with in the previous Section will be reported. This will include an analysis of the target variable, non-parametric tests on the data with the comparison dataset and a conclusion on the above.

All calculations for this section are included in the “Statistics.ipynb” file submitted alongside this report.

## **Statistics on Irish data**

For the statistical analysis, I will primarily be looking at the target value of the Machine Learning section (i.e. Raw milk price), deriving statistics and comparisons with this data as the central component.

### **Raw milk values**

The following is a statistical breakdown of the Raw milk value feature, as obtained from the Irish dataset:

* 240 values recorded between January 2002 and December 2021
* Mean value of the period was **€32.36** per 100 litres
* Median of the period was **€31.78**
* Standard deviation from the mean is **€5.37**
* Lowest value in the series was **€21.83**
* Highest value in the series was **€48.65**
* Interquartile range is **€7.91**

With a mean greater than the median, the data is positively skewed. The next test is to verify the normality of the series.

Chart, box and whisker chart

Description automatically generated

Figure 1 Boxplot for the Ireland raw milk value (mean = 32.36, min = 21.83, max = 48.65)

### **Normal distribution**

Chart, line chart

Description automatically generated

Figure 2 Distribution of the Ireland Raw milk value from 2002-2021, with mean and standard deviation displayed

The above graph shows the distribution of the data in Gaussian format, with indicators for the mean, median and deviations as defined earlier.

The difference between the median and mean is .58, which would indicate, according to Pearson’s 2nd Coefficient of Skewness (Beri, 2010) a positive skew of rate:

* 3 (mean - median) /standard deviation =>
* 3 (32.36 - 31.78) / 5.37 =
* 3 (.58) / 5.37 = 1.74 / 5.37 = .32

To further test this, we check to see if the data follows the empirical rule, using the following methodology:

* 68% of data between 1 deviation
* 95% of data between 2 deviations
* 99.7% of data between 3 deviations

Using the Z-Score of all values in the domain to gather their deviation value, the following is the result obtained:

* **1 deviation** 174 (72.5%)
* **2 deviations** 234 (97.5%)
* **3 deviations** 238 (99.1%)

With this information, we can see that the data is not normally distributed, as it does not pass the empirical rule. The data, as shown in the histogram below, also does not follow a Gaussian distribution with regards to its value distribution, with the values lower than the mean primarily made up of values between 26 – 28.5, which would also explain the lower median. For this to be a normal distribution, the data should have followed the following:

* **1 SD** between 26.99 and 37.73
* **2 SD** between 21.62 and 43.1
* **3 SD** between 16.25 and 48.47

## **Comparisons between countries**

As a further investigation, I decided to compare the ordinal variance of my data with other data collected, to research the similarities and differences in the patterns each encountered. For a valuation of Raw milk, for example, as Glanbia attributed the 2009 crash primarily with the global market crash, and the 2016 crash with the surplus as a result of the removal of the EU milk cap, it was logical to compare these values between 2 EU countries (as both would theoretically have had encountered the same issues under the EU directives).

For each test performed, the suitability of the data was compared using the Shapiro evaluator for Gaussian distribution. This was primarily done on non-evaluated data from the previous part, as it has already been established that the Irish raw milk value does not follow a Gaussian distribution.

For this dataset, we have also imported the EU Average Milk Price dataset, as detailed in the previous chapter, for comparisons between Ireland and the other members of the EU.

### **Tests**

To determine which tests to use, I calculated what type of distributions each type fell under, using the Shapiro test to verify the distribution. Once it was selected, I chose my statistical test based on the number of data series desired to be evaluated.

For each test performed, no grouping of data passed the Shapiro analysis, meaning I evaluated based using non-parametric tests. For groupings of 2 data series, I evaluated using the Mann-Whitney U test, and used Kruskal-Wallis for the tests comparing multiple variables.

As a note, for the implementation of Python, any of the tests that obtain a p-value less than .05 is rejected, with values scoring higher considered to have been accepted.

### **Results**

The results of each test made where as follows:



Table 2 Short codes for the following table (e.g. in the following table, Data "A" denotes the Irish Raw milk value data series)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Data 1 | Data 2 | Test | alt | H0 | U | p | Result |
| A | B | Mann-Whitney U | two-sided | Raw milk valuations of Irish and Spanish milk are similar in rank | 24403 | 0.00381 | Reject |
| C | D | Mann-Whitney U | two-sided | No difference between the 2 distributions of milk production | 53558 | 1.1 x e-59 | Reject |
| E | F | Mann-Whitney U | two-sided | No overall difference between the fat content levels detected | 49728 | 3.642 x e-43 | Reject |
| A | G | Mann-Whitney U | two-sided | Raw milk valuations of Irish and German milk are similar in rank | 28337 | 0.76084 | Accept |
| A | H | Mann-Whitney U | two-sided | Raw milk valuations of Irish and Spanish milk are similar in rank | 27120 | 0.26901 | Accept |
| A | B, H | Kruskal-Wallis | two-sided | Raw milk valuations of the 3 countries are similar | 24.5 | 4.7 x e-6 | Reject |

Table 3 Table of tests performed and results

Due to the vast majority of the data not being a Gaussian distribution (as clarified using a Shapiro Walk test), it was required to perform non-parametric tests on the tests. With this, testing was performed using Mann-Whitney (when comparing 2 data sources) and Kruskal-Wallis (for the test that took 3 data sources).

As the table above shows, most of the hypotheses were rejected, with only 2 proposed theories passing. With these results, we can see that the difference in scale of some features with regards to the others is simply too great. Even though all values were scaled to the same format (e.g. for Volume was converted to Million litres to reduce its scale), the testing has yielded results that were so significantly small that Python nearly failed to interpret them. The feature that was expected to yield a positive acceptance (Raw milk values, where the expected variance was mitigated by the EU attempting to keep prices consistent across all members) failed between the primary comparison but passing with the Germany (the highest dairy-producing member of the EU) and Spain (the country with the closest yield to Ireland) would suggest that the hypothesis should succeed for the other countries.

One point of note to consider is the production limit. Comparing the production and processing variables, we can see a large difference with respect to the scale of the values. Dairy refinement is almost 3 times greater on average than the Irish equivalents, with the Cheese and Butter values being similar in scale to Ireland. This would match up with the population difference with Ireland, with the most recent estimates from the EU (https://ec.europa.eu/info/sites/default/files/food-farming-fisheries/farming/documents/agri-statistical-factsheet-nl\_en.pdf) placing the Netherlands as having over 17 million citizens (compared to Ireland’s 4.8 from the 2016 census), with Dairy being one of the more popular farming types in Netherlands (with the dairy farm output of the last 3 years being only eclipsed by the fact that Vegetables is treated as its own category).

One potential theory regarding this could be attributing to exporting. Ireland’s dairy exports tend to be of a finished product, with Cheese and Butter being well-valued and desired products across the world. However, Ireland does not export a similar amount of Raw milk, preferring to process the majority of the milk. With this theory, we would assume that the Netherlands typically has a surplus of dairy produce, and simply exports the raw milk more than Ireland, which could potentially increase its value (and thus would therefore increase the overall value of the milk), or the Dutch government could equally have provided a subsidy to farmers for the raw milk, which could also have affected the valuation.

### **Interpret results**

From these results, we can see the overall comparison between the Netherlands and Ireland. The scale of the Netherlands is different when compared to the other values, being on a higher production level as well as a higher level on a world-scale (Netherlands is among the top-5 producers in the world for dairy, while Ireland is only 6th in Europe).

However, taking the above tests in their own context would also be incorrect. With the other Mann-Whitney U tests performed without using the Netherlands data, we have determined that there can be a statistical comparison performed between these two values. On a comparison between Ireland and Spain on the same value, we determined that there is consistency in the similarity of the valuations, in contrast to the Netherlands.

A similar comparison made between Ireland and Germany also came to a similar conclusion. This is interesting, due to Germany’s output of dairy being closer to the Netherlands than Ireland (Germany also being in the top 5 countries for their production, being above Netherlands), when it would be assumed otherwise.

From these findings, I have determined that there is still a topic worth investigating here. I have established that the distribution and valuation of the month-to-month raw milk produced in the various countries do not share a similar distribution, and that these differences may be attributed to multiple valuations and inputs for consideration. From this, on top of the original Machine Learning test on the Ireland dataset that contains all productions and valuations of cattle, we will perform a 2nd test using the same models, on a dataset that contains all the Ireland data that has a comparable feature in the Ireland dataset.

# **Machine Learning**

In this section, the algorithm chosen for the machine learning portion of this project, as well as the modifications and adaptions performed, will be defined, and explained. The results of the machine learning using the defined algorithms will also be defined and discussed. As well as this, a discussion of the sentiment analysis undertook for the target subject will be included.

## **Sentiment analysis**

### **Data source**

For the sentiment analysis, I undertook an investigation of Twitter tweets to evaluate the overall sentiment of dairy milk farming in Ireland, as well as each of the individual keywords from that statement.

### **Methodology**

Having applied for Twitter Developer Elevated Status, I was afforded access to high-end functions that allowed me to search using my keywords to discover tweets. Using a function to identify permutations of my keywords (“dairy”, “milk”, “farming”, “Ireland), I implemented a recursive function to implement searches based on these keywords, due to the Twitter API not natively implementing this behaviour.

I performed this function on 3 iterations of the functions, under the following logic:

* **Search 1** 27/04/2022 – 12/05/2022
* **Search 2** 12/04/2022 – 27/05/2022
* **Search 3** Tweets prior to 12/04/2022

The reasoning behind this specific methodology is that Twitter divides the search functionalities into 2 categories; recent (posted at most 30 days prior to query) and historic (every other tweet), with caps on how many tweets from each category can be queried in total (5,000 historic, 25,000 recent). Without the ability to increase these limitations, the amount of data I could obtained was significantly reduced.

Once the tweets were obtained, they were processed to remove usernames and other identifications, as well as URLs. The dataset was then processed to remove any “duplicate” tweets (e.g. tweets from the same user, or retweets with no additional comments made to tweets already recorded). Finally, the TextBlob API was used on the processed output to garner the sentiment of that tweet, with the results saved into the dataset. The following is the results from this sentiment analysis.

For this analysis, we check the value of the positive sentiment against the negative and declare a tweet’s sentiment based on its scoring against the opposite. If both positive and negative are 0, the tweet is neutral, otherwise the tweet’s sentiment is the greater of the positive and negative sentiment calculation.

### **Results**

In this dataset, positive sentiment is the dominant value, with over 45% of the tweets scoring a positive value based on the above criteria. Negativity in the sector, based on these values, is the lowest sentiment, with only 17% tweets containing a negative sentiment, with the remainder of the tweets being negative.

Judging from the sum of residuals from the sentiment analysis, we can see the dataset overwhelming leans more towards neutrality than any other statement, with the scoring for neutrality (4216) far outscoring the values for the other 2 categories.

This would suggest that, overall, the subject is neutral on Twitter, which went against the initial reasoning used for choosing Twitter as a base for sentiment analysis, which theorized that Twitter’s unique ability to make points easily would make it a great area for analysis, as well as the character limitation forcing users to decisively make their point known, thus decreasing chances of confusion from the TextBlob API.

With these findings, we can say that, for this specific use, Twitter is not a useful database for discovering the sentiment of this specific topic. Upon investigating other projects that implemented a Twitter Sentiment Analysis and comparing the topics of those projects to my own, I can see that the major difference would be that, in general, the topics chosen in the other projects tended to be more controversial or topical compared to my own. The work done by TowardsDataScience (Yener, n.d.) investigated the sentiment towards the 2nd preventative COVID-19 measures taken by the United Kingdom government, a subject that would generate more commentary from those outside the relevant industry (i.e., politics), as the issue would naturally impact multiple industries and individuals. Another identifier for the issue is the timing of the testing. For example, an investigation between February and March 2022 would have included data surrounding the uncertainty concerning distribution, imports, and exports with respect to growing tensions between Ukraine and Russia.

## **Machine Learning**

### **Learning type**

The overall goal of the project is to see whether the Raw milk price of a country is influenced by the varying factors that affect its production and use (e.g. the average price of milking cows, the amount of produce generated), and as such is investigating the relationship between these variables and the target value. The data has been formatted to combine all these features into singular datasets ready for machine learning techniques.

Upon consideration of the data obtained and the overall goal of the project, it was determined that Supervised Learning models should be used for the predictions.

### **Models used**

2 sets of tests were performed: one that tested the Ireland data with regards to the data as described in the previous sections, and one that tested a combination of the matching data between the Ireland and Netherlands dataset.

Each test used the following models for testing:

* **Random Forest**
* **K-Nearest Neighbours Regression**
* **Decision Tree Regression**

Each model had a random state initialized for their execution (22122), and all models scored based on the “absolute error” criterion where possible.

### **Hyper-parameter optimization**

Hyper-parameter optimization was performed with the GridSearchCV algorithm, which would guarantee the best result generated from testing all possible combinations of the given parameter grid.

For each model, their parameters were investigated and evaluated based on their estimated importance to the performance of the model. Value ranges were chosen specifically but were either auto-generated (using list comprehension) or adapted directly from the documentation (if there were only a select number of values for it).

GridSearchCV operated quickly on DecisionTree and KNeighbors models, with execution times under 2 minutes for each model for each test, due to lower parameter distributions to test. RandomForest models had testing times over 2 hours each, due to a larger parameter distribution, and the overall impact of using absolute error as a scoring criterion (where squared error would execute faster but be less refined).

The generated models from GridSearchCV and parameter distributions are available in the Jupyter notebook attached for submission.

### **Results**

#### **Test 1**

The following for the experiments on the above test is as follows

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Algorithm | Method | Cross Validation | | Regular Testing |
| R2 | MAE | MAE |
| Random Forest | Base | 0.15344 | 3.29989 | 3.06952 |
| Random Forest | RFE performed on Base | 0.16505 | 3.29765 | 3.04383 |
| Random Forest | GridSearchCV | 0.17255 | 3.23942 | 2.99100 |
|  |  |  |  |  |
| K-Neighbors | Base | -0.21429 | 3.53936 | 3.36499 |
| K-Neighbors | GridSearchCV | -0.22780 | 3.48603 | 3.30980 |
|  |  |  |  |  |
| Decision Tree | Base | -0.62080 | 4.14739 | 3.27050 |
| Decision Tree | RFE performed on Base | -0.12887 | 3.69433 | 3.78034 |
| Decision Tree | GridSearchCV | -0.43302 | 3.75494 | 4.15134 |

Table 4 Details of tests performed for Test 1, referencing the Machine Learning algorithm, the enhancement to the model and the performances of cross validation and simple testing

##### **Results**

The models performed poorly when fitted with the data. Each model had a low R2 score during the cross-validation tests, with the values falling below -1 for non-Random Forest models (indicating that the model on predicting the dataset is not able to properly determine the value). The poor performance of the models, as well as the better-by-comparison performances of Random Forest models, means I will not be reporting on those datasets at this time.

Random Forest performed the best out of the models, across all its iterations. The R2 score was .17255 in the GridSearchCV model, achieving the best mean errors across all models tested. There was no feature selection performed on the best model, due to all generated models being identical to the best model here.

With a value of 3.3 in cross validation, and 2.99 on the stratified tests, the GridSearchCV Random Forest model performs the best out of all models tested for this dataset. However, the model’s R2 score is low enough to determine that the data is not a good fit for this model. The R2 score being lower than .5 means that the model is not a good predictor of the values from the data. However, the variance.

Calculating the average of the predictions, we can see that the average prediction was 91% accurate with regards to the true value it was attempting to predict. This would imply that the model, while not strong, was still able to generate good and near-accurate results with regards to its predictions.

Chart, scatter chart

Description automatically generated

Figure 3 Graph displaying the variance of the prediction against the true value. The y-axis plots the error from the true value: the closer the point is to 0, the more accurate the prediction is to the mean

From this graph, we can see the overall performance of the best Random Forest model’s predictions. The variance of the model is consistent with both ends. The predictions seem to trend upwards with regards to their predictions, (Yener, n.d.)the higher the true value, the higher the predicted value. The overall prediction distribution conforms to the data provided; 60% of the values fall within the average from the cross-validation tests, with outliers mostly occurring when the true value should be a value close to the minimum or maximum of the distribution. It is possible that the training data did not contain enough similar entries for the model to learn from, which led to this issue. Between the values of 25 and 35, there is only 3 distinct values that exceed 5, which means that data in that range might be easier for the model to understand and thus predict.

#### **Test 2**

The following are the results of the machine learning models on the second testing:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Algorithm | Method | Cross Validation | | Regular Testing |
| R2 | MAE | MAE |
| Random Forest | Base | -0.13564 | 4.15591 | 3.57198 |
| Random Forest | GridSearchCV | -0.10029 | 4.06098 | 3.34990 |
|  |  |  |  |  |
| K-Neighbors | Base | -0.32559 | 4.46343 | 4.26918 |
| K-Neighbors | GridSearchCV | -0.32365 | 4.47701 | 4.20655 |
|  |  |  |  |  |
| Decision Tree | Base | 0.57565 | 4.89806 | 3.71512 |
| Decision Tree | RFE performed on Base | -0.46156 | 4.71706 | 3.53713 |
| Decision Tree | GridSearchCV | -0.52581 | -4.80081 | 3.69992 |

Table 5 Breakdown of the models used for Machine Learning on Test 2, and results of the algorithm

##### **Discussion**

Like the previous test, the models performed poorly with regards to the accuracy of the model. No model fitted in this test had an R2 score greater than 0, implying the models are arbitrarily worse at prediction than using the mean than simply imputing based on the target value.

The best performing models in this system are Random Forest, although their performance is still substandard. While the value of R2 is the closest to 0 of the models tested, it is still below the minimum standard required for consideration.

One issue arising from this dataset would be the variance in the scale of the data. By combining the Ireland and Netherlands dataset, we effectively have 2 different distributions of data, one with greater production numbers than the other (milk production in the Netherlands is superior to the Irish production), and this is reflected similarly in the output of other areas. However, the scale of the variance is not in fact reflected in the target value, which is comparable in value to the Irish dataset. This may have caused the models to fail to learn the difference, as 2 sets of variables of vastly different scale could end up computing the exact same value (e.g. milk production of 200 for Ireland and 1000 for Netherlands could both equal €28.00 for the target value).

Chart, scatter chart

Description automatically generated

Figure 4 Graph of y\_test (the true values) vs the predictions. Y-axis represents the error of the prediction

Seeing the distribution of the best Random Forest model, we again see the variance in action. The residuals are dispersed across the graph for all values, with approximately 40% of all predictions falling greater/less than the mean prediction. Comparing this graph to the previous, we can see there is a similarity but otherwise greater difference. Here, the high variance of the errors contributes greatly to the mean value; while the model typically predicts close to true or near the mean, the extreme difference in the residuals is concerning and problematic.

As in the previous graph, there is an upward trend in the data. Here we see that the system has better performance predicting values that are closer to the mean of the test values; the values at the upper/lower limits tend to have the higher errors compared to those closer to the mean.

# Table of Figures

[Figure 1 Boxplot for the Ireland raw milk value (mean = 32.36, min = 21.83, max = 48.65) 7](#_Toc104051066)

[Figure 2 Distribution of the Ireland Raw milk value from 2002-2021, with mean and standard deviation displayed 7](#_Toc104051067)

[Figure 3 Graph displaying the variance of the prediction against the true value. The y-axis plots the error from the true value: the closer the point is to 0, the more accurate the prediction is to the mean 16](#_Toc104051068)

[Figure 4 Graph of y\_test (the true values) vs the predictions. Y-axis represents the error of the prediction 18](#_Toc104051069)

[Table 1 Descriptor of the headers of the targetted formatting for each table in the system 3](#_Toc104051075)

[Table 2 Shortcodes for the following table (e.g. in the following table, Data "A" denotes the Irish Raw milk value data series) 9](#_Toc104051076)

[Table 3 Table of tests performed and results 10](#_Toc104051077)

[Table 4 Details of tests performed for Test 1, referencing the Machine Learning algorithm, the enhancement to the model and the performances of cross validation and simple testing 15](#_Toc104051078)

[Table 5 Breakdown of the models used for Machine Learning on Test 2, and results of the algorithm 17](#_Toc104051079)

# Bibliography

Anon., n.d. *CBS Open data StatLine - Milk supply and dairy production by dairy factories.* [Online]   
Available at: https://opendata.cbs.nl/portal.html?\_la=en&\_catalog=CBS&tableId=7425eng&\_theme=1104

Anon., n.d. *Milk and milk product statistics.* [Online]   
Available at: https://ec.europa.eu/eurostat/statistics-explained/index.php?title=Milk\_and\_milk\_product\_statistics

AskPython, n.d. *Normal Distribution in Python.* [Online]   
Available at: https://www.askpython.com/python/normal-distribution

Beri, 2010. In: *Business Statistics.* s.l.:Tata McGraw-Hill Education.

Developers of scikit learn, n.d. *sklearn.ensemble.RandomForestRegressor.* [Online]   
Available at: https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html

MacConnell, S., 19/04/2009. Milk farmers furious at litre price cut by Glanbia to 20c. *The Irish Times.*

Office, C. S., n.d. *Archive - CSO - Central Statistics Office.* [Online]   
Available at: https://www.cso.ie/en/statistics/agriculture/archive/

Office, C. S., n.d. *Milk Statistics - CSO - Central Statistics Office Ireland.* [Online]   
Available at: https://www.cso.ie/en/statistics/agriculture/milkstatistics/

Tweepy developers, n.d. *API - tweepy 4.9.0 documentation.* [Online]   
Available at: https://docs.tweepy.org/en/stable/api.html

Yeder, Y., n.d. *Step by Step: Twitter Sentiment Analysis in Python.* [Online]   
Available at: https://towardsdatascience.com/step-by-step-twitter-sentiment-analysis-in-python-d6f650ade58d

Yener, Y., n.d. *Step by Step: Twitter Sentiment Analysis in Python.* [Online]   
Available at: https://towardsdatascience.com/step-by-step-twitter-sentiment-analysis-in-python-d6f650ade58d