Table of Contents

[Introduction 2](#_Toc104141337)

[Data Understanding 2](#_Toc104141338)

[Data Preparation 3](#_Toc104141339)

[Shaping the datasets 3](#_Toc104141340)

[Missing values and data types 3](#_Toc104141341)

[Numerical features statistical summary 3](#_Toc104141342)

[Data Visualisation 4](#_Toc104141343)

[Graph’s design 4](#_Toc104141344)

[Distribution of the numeric features 4](#_Toc104141345)

[Outlier detection 4](#_Toc104141346)

[Inferential statistics report 5](#_Toc104141347)

[Correlation 5](#_Toc104141348)

[Variance 5](#_Toc104141349)

[Covariance 5](#_Toc104141350)

[Summary statistics 5](#_Toc104141351)

[Plotly scatter plots 5](#_Toc104141352)

[Producer Price Index comparison 6](#_Toc104141353)

[Visualisation with dashboards 6](#_Toc104141354)

[Statistical Significance Tests 6](#_Toc104141355)

[Livestock comparison for the two countries 6](#_Toc104141356)

[Parametric Statistical Significance Tests 7](#_Toc104141357)

[Student’s t-Test 7](#_Toc104141358)

[Paired Student’s t-Test 7](#_Toc104141359)

[Analysis of Variance Test (ANOVA) 7](#_Toc104141360)

[Nonparametric Statistical Significance Tests 8](#_Toc104141361)

[Shapiro-Wilk test 8](#_Toc104141362)

[Mann-Whitney U Test 8](#_Toc104141363)

[Wilcoxon signed-rank test 8](#_Toc104141364)

[Modelling 8](#_Toc104141365)

[Normalization 9](#_Toc104141366)

[Encoding 9](#_Toc104141367)

[Dependent and Independent variables 9](#_Toc104141368)

[Evaluation 9](#_Toc104141369)

[Sentiment Analysis 10](#_Toc104141370)

[Gathering the tweets 10](#_Toc104141371)

[Content cleaning 10](#_Toc104141372)

[Tfidf Vectorizer 11](#_Toc104141373)

[Sentiment analysis 11](#_Toc104141374)

[Machine learning models 11](#_Toc104141375)

[Cross Validation 12](#_Toc104141376)

[Deployment 12](#_Toc104141377)

[Github repository 12](#_Toc104141378)

[References 13](#_Toc104141379)

[Appendix 16](#_Toc104141380)

[Figures 16](#_Toc104141381)

[Tables 20](#_Toc104141382)

# Introduction

In this report I tried to find and prove correlation between export unit value index and producer price index (PPI), livestock value and different export indicators through the main four livestock items (chicken, pig, sheep, cattle) compare two countries figures (Ireland and Hungary) on the same time scale (from 1990 till 2014).

Meat accounts for over 40% of Ireland’s gross agricultural output, dominated by beef, followed by pig meat and sheep meat. Ireland’s meat and livestock exports account for one third of all food and drink exports[[1]](#footnote-1). Agri-food exports account for over 11% of total Irish merchandise exports. The Hungarian animal production includes 2.9 million pigs and poultry flocks of 40 million birds.  The number of cattle of all types is approximately 885,000 and the share of agricultural exports was 8.7 percent in total exports of Hungary[[2]](#footnote-2).

At this stage I presumed there won’t be high bias between the two-country based on the share of the agricultural export in total export figures by each country.

# Data Understanding

The datasets I have used in my study were gathered from Food and Agriculture Organization of the United Nations (FAO) website. To create a more complex dataset I had to combine three different datasets where the countries, items were the same, but they had extra features in it such as livestock value or producer price index. After I merged the datasets, I got 193 observations and 10 features. This final dataset contains features in regards of export price and quantity indexes and values, producer price index, livestock value and some categorical features such as country, or item.

Table

Description automatically generated

Figure Final dataset before pre-processing

# Data Preparation

## Shaping the datasets

The original datasets I have found were not on the structure I wanted, especially the first one, so first I have deleted the missing values as I only had a few in the top and the last two rows. Deleting these values didn’t cause major information loss in my dataset. I created a list of new column names, and I passed this list to my header where I only kept the important features. Now the frame of the first dataset was ready so I engineered the variables as well. Using *lambda* function I removed the unnecessarily characters from the values then I sort them by country and year. I did the same with the next two dataset as well where I only kept the feature, I meant important to add to my first dataset and additional three features (item, country, year) what I intended to use to merge the three datasets together. For merging the datasets, I used lambda function again where I defied the left and right merge, the matching features will be merged on the left whiles the “extra” features will be added to the original dataset from the right doing this row by row (*reduce* function) using the datasets indexes.

## Missing values and data types

Using *info* function I checked the data types of the features in my dataset. I got back object even for some of the features they looked like numerical. I selected the features I didn’t want to change the data types and only converted the remaining ones to numeric and then to integer. Then I converted year feature to datetime.

I have checked the missing values using *isnull* function and there are none in the dataset.

## Numerical features statistical summary

Using *describe* function we can display a summary of the numerical features in the dataset. This summary shows some important information about features mean, standard deviation, minimum-maximum values, and quartiles. Based on the figures we can see that PPI and Export Unit/Value Index seems to be normally distributed as standard deviation values are smaller than the mean value. The smaller the standard deviation the less spread out the datapoints are. The rest of the features are probably skewed, but I will check this more detailed in detailed statistical summary later.

# Data Visualisation

## Graph’s design

When I created my graphs no matter if that was a scatter plot or a histogram I aimed to create and easily understandable graph where the main information what the data holds could be clearly visible. I have couple of rules what I have followed through my report in visualisation.

**Colors:** - If there was a comparison colors had to distinguish the categories or features from each other. I tried to use colors that won’t evoke emotions but when I combined them, they still could highlight the information. This will be important on histograms where I used lines to represent mean and median. I also tried not to use too harsh, violent colors what human brains could hardly associate with.

**Font size and weight**: - Labels, axis tickers, titles have to be easily readable and padded from the graph. Title has the largest font size as it stands what the graph intends to show for the reader. Axis labels has to be slightly larger than the tickers to clearly shown what the particular axis represents.

**Figure size and structure:** - I used single graphs and subplots in some of my graphs, but in both cases I tried to define an appropriate graph size where all the information what the data carries can be easily visible. I also aimed to use similar graph size through my study give a kind of a standard frame to my visualisation.

## Distribution of the numeric features

To visualise the distribution of the numeric features I used histograms where I defined a blue curve line represents the density, red dashed line represents the mean, and the yellow is the median. Alpha is specifying the transparency of each histogram. The legend shows the mean and standard deviation for each features. As I have already thought according to the statistical summary above PPI and Export Unit/Value Index follows some kind of normal distribution as their standard deviation are small compare to their mean values. The rest of the numeric feature have positive skew that indicates that the tail is on the right side of the distribution, which extends towards more positive values.

## Outlier detection

An outlier of a dataset is defined as a value that is more than 3 standard deviations from the mean. Some of the machine learning algorithms are sensitive for outliers so most of the time it is advised to remove outliers from the data. I used plotly (***Figure 2 Boxplot for outlier detection***) boxplot to detect outliers in my dataset, and as boxplot graphs show I have some suspicious datapoints. When I remove them “cattle” value will disappear from Item category. This means those datapoints are not outliers rather belongs to a category which has higher values therefore makes the dataset imbalanced.

I decided to not to remove these outliers as they might carries underlying information in regards of my observation.

## Inferential statistics report

### Correlation

Correlation heatmap gives an excellent solution to visualize correlation between features. I used *seaborn* to plot my correlation heatmap where I decided to work with Pearson correlation which summarize the strength of the linear relationship between two data samples[[3]](#footnote-3). The graph shows in a glance which variables are correlated, to what degree (***Table 1 Correlation strength guidance****)* in which direction. As the graph shows Export Value Index and Export Quantity Index have the strongest correlation.

### Variance

Variance is a measure of how much the data for a variable varies from it's mean.

### Covariance

Covariance is a measure of relationship between 2 variables that is scale dependent, in other words, how much will a variable change when another variable changes. As the table shows (***Table 2 Covariance test for numeric features***) eg. Export Value Base Price will respond the most for Export Quantity Index changes.

### Summary statistics

With *aggregation* function I selected the statistical parameters I wanted to compute and passed them to the index column in my df\_num data frame. This summary gives the most detailed statistical info information about the numeric features in the dataset. We can see the minimum, maximum values, the value count, mean, median, standard deviation, and skewness of each feature (***Table 3 Statistical summary for numeric features***).

## Plotly scatter plots

Based on the correlation observation I thought important to visualise those feature pairs more detailed whom had the strongest correlation according to the heatmap. I used plotly scatter plot and I defined the two feature I wanted to plot against each other, then selected different colors by Item groups, and finally I draw a straight linear line to represent the overall trend for both features datapoints. As I wanted to see these graphs for the two countries separately therefore, I used side by side plots with and shared the *y axis.*

On the first graph(***Figure 3 Scatter plot for Export Quality Index vs. Export Value Index***) the datapoints are very close to the trend line and they are pretty much grouped together whilst on the second graph (***Figure 4 Scatter plot for Export Value Base Quantity vs. Value Base Price)*** only the Hungarian plot is close to the trend live and the Irish datapoints are spread out, furthermore datapoints belongs to cattle category are fare from the trend line and from any other groups. This fact just proves the assumption of imbalance data.

## Producer Price Index comparison

The other extra feature I added to my original dataset is PPI what I want to visualise more carefully. I thought to check item categories on a timeline to see what values had PPI for both countries between 1990 and 2014. I used line graph where each category had different colored line. On the plot I used subplots again but with vertical integration at time as the graphs had to be wide enough to draw nice lines with their braking points.

## Visualisation with dashboards

Dashboard helped to give option to plot different combinations of the dedicated features. First, I created a function where I defined the main parameters of my graph. I used ipywidgets to interact between the features I selected. I listed the features for both axis what I want being plotted against each other than separated the datapoints by country and item categories. Ipywidget will provide the function what slightly changes the input parameters by the listed options on the axis and plot the selected features against each other (***Figure 6 Scatter plot with Dashboard for numeric features****).*

On the second graph I followed the same idea as on the first one but at this time I used barchart. The function structure is the same, but the interaction is different as I only had to define selectable features for y axis. The function will plot the selected numerical feature against producer price index creating two separate charts for the two countries (***Figure 7 Dashboard for barchart***).

# Statistical Significance Tests

Tests for statistical significance are used to estimate the probability that a relationship observed in the data occurred only by chance; the probability that the variables are unrelated in the population.

I decided to use livestock value for the tests I will perform later but first I have to create the same sized samples for both countries.

## Livestock comparison for the two countries

I wanted to compare livestock between Hungary and Ireland and to achieve that I used five number summary where I have added mean and standard deviation as an extra parameter. Both countries have similar values on this feature, and both has positive skew.

To visualise the distribution, I used plotly again (***Figure 5 Distribution of Livestock by countries***) but now I plotted two subplots. On the graph green dashed line represents the mean value, and with column parameter I defined the location of each graph.

## Parametric Statistical Significance Tests

Parametric statistical methods often mean those methods that assume the data samples have a Gaussian distribution[[4]](#footnote-4).

### Student’s t-Test

The student’s t-test compares the mean of the two samples I created called from livestock feature by the observed countries. In the unpaired t-test the operator assumes that population distributions are normal (Gaussian), the standard deviations are equal, and the assays are independent.

My hypothesis are:

H0: Sample distributions are equal

H1: Sample distributions are not equal

The result of the test is to fail to reject the hypothesis as the sample means are equal with the 5% significance level.

### Paired Student’s t-Test

The paired sample t-test, sometimes called the dependent sample t-test, is a statistical procedure used to determine whether the mean difference between two sets of observations is zero. In a paired sample t-test, each subject or entity is measured twice, resulting in pairs of observations.

The null hypothesis assumes that the true mean difference between the paired samples is zero. Alternative hypothesis is the paired sample distributions are not equal. The result of the test is to fail to reject the hypothesis as the sample means are equal with the 5% significance level.

### Analysis of Variance Test (ANOVA)

For ANOVA test I used one-way ANOVA. One-way ANOVA is used to compare two means from two independent (unrelated) groups using the F-distribution. The function I used took these two data samples as arguments and returned the test statistic and f-value.

The null hypothesis for the test is that the two means are equal. Therefore, a significant result means that the two means are unequal.

H0: All sample distributions are equal.

H1: One or more sample distributions are not equal.

The interpretation of the p-value in ANOVA test accepts the null hypothesis indicating that the samples means are not differ.

## Nonparametric Statistical Significance Tests

### Shapiro-Wilk test

The Shapiro-Wilk test is a test of normality. It is used to determine whether a sample comes from a normal distribution. If the p-value is below a certain significance level, then we have sufficient evidence to say that the sample data does not come from a normal distribution. As pvalue less then 0.05I conclude that both samples of data have normality in their distribution[[5]](#footnote-5).

### Mann-Whitney U Test

Mann-Whithey U test is used to test the null hypothesis that two samples come from the same population (i.e. have the same median) or, alternatively, whether observations in one sample tend to be larger than observations in the other[[6]](#footnote-6).

As the p value obtained from the Mann-Whitney U test I conclude that the yield of the two samples is not different from each other.

### Wilcoxon signed-rank test

Wilcoxon test is recommended when the data violates the assumption of normality. The Wilcoxon signed rank test assumes that there is information in the magnitudes and signs of the differences between paired observations[[7]](#footnote-7).

H0: Sample distributions are equal.

H1: Sample distributions are not equal.

The p-value is interpreted that the samples distributions are not equal.

# Modelling

The goal in my study was to see how strong the correlation is between my target variable Export Unit Value Index and the export price indexes, and livestock value.

The import and export price indexes represent one of the key metrics that help to measuring the change in prices of goods and services in the economy. The other metrics is the Producer Price Index (PPI), which measures price changes in commercial goods sold. The indexes are used to help measure inflation in products that are traded globally. That is why I found important to observe the correlation between this metrics.

This is a classic Regression problem as my target variable is continuous. The machine learning algorithms I tend to use are:

* Linear Regression
* Ridge Regression
* Elastic Net
* Random Forest Regressor
* Extra Trees Regressor
* Support Vector Machine

## Normalization

As my input features values are differ greatly between their ranges, I had to normalize them before I build my models. I used *StandardScaler* what removes the mean and scales the data to the unit variance.

## Encoding

My dataset still has categorical values that have to convert to numeric format as my regression models are require numeric inputs. I used ordinal encoder to do the job. Ordinal encoding, each unique category value is assigned an integer value. The integer values have a natural ordered relationship between each other, and machine learning algorithms may be able to understand and harness this relationship. The dataset now ready for build the models (***Table 4 Normalized and encoded dataset)***.

## Dependent and Independent variables

My independent variables are all the features in my dataset except Export Unit Value index as it would be my dependent (output) variable. These independent variables will help me to determine the dependent variable. After I defined the input and output features, I split my dataset to training and test parts. As my dataset is quite small, I will use 33% of my data for testing.

I created two functions to count metrics that will help me measure the quality of my regression models. These parameters are:

* Mean Squared Error (MSE)
* Mean Absolute Error (MAE)
* coefficient of determination (R²)
* Root Mean Square Error (RMSE)

The second function will help to evaluate the calculated metrics.

# Evaluation

According to the models results I can state that most of my regression models are performed way better on training than on testing data, what shows that my models are overfitted. MSE scores were relatively high what also proves the overfitted model as the distance between the regression line and the set points are high.

Ridge Regression model performed poor both on training and testing as well, and MSE score were the highest compared to other models.

Extra Trees Regression model performed the best compared to other regression models in my study if we dispense with the overfitted matter (***Figure 8 MLA comparison before hyperparameter tuning***).

To gain better performance of my models I used Grid Search to optimize the hyperparameters of my machine learning models and get better scores.

Using Grid Search I defined the best parameters on each model to get the best score using my input values. These parameters are different in each model. First, I defined these parameters by each model then I passed the parameters and printed out the best parameters what used I can reach the best score with the model. The final step to evaluate the model with the best parameters.

After the hyperparameter tuning Ridge Regression performed way better, R² score are much higher but still slightly overfitted based on testing and training score difference and high MSE score.

The rest of the models are performed very well, there are no sign of overfitting. Comparing the R² score Elastic Net model performed the best (***Figure 9 MLA comparison after hyperparameter tuning***).

# Sentiment Analysis

## Gathering the tweets

Sentiment Analysis is a sub-field of NLP that measures the inclination of people’s opinions (Positive/Negative/Neutral) within the unstructured text[[8]](#footnote-8). I used *Search Scraper* to gather tweets from Twitter. Search Scaper creates an unofficial Twitter API to extract tweets, retweets, replies, favourites, and conversation threads with no Twitter API limits. I gathered tweets in regards of livestock in Ireland with a maximum limit of 5000 tweets.

I got 4987 observations and 27features in my dataset. As I mainly wanted to focus on the tweets, I dropped all features but content and ID.

## Content cleaning

For the first look I saw many characters such as @, numbers, underscore, and symbols. To remove them from the content I created two functions that will do the job for me.

The first function I defined the emojis and symbols I wanted to remove, and with the second function I will convert all characters to lower case, remove the numbers, remove all the special characters such as ?,!, #, etc. The function will split the text by the defined punctuation marks and will re-join it without them.

I used *apply* function to use those functions on my content.

To remove stopwords from my content I created a lambda function what split the text, remove the stopwords that has been defined in the library I imported and re-join the text again. I listed the 10 most frequent words in the content, and I used lambda function to iterate through the content feature and remove these words. I did the same with the 10 less frequent words as well.

To check the correct spellings in content feature I used *Textblob* library and iterated through its *correct* function on my content feature. Then I divided the text into sequences of words and removed the suffix of the words with *stem* function and brought them to a base word. After that I used *lemmatize* function to return the base or dictionary form of a word.

## Tfidf Vectorizer

The TfidfVectorizer will tokenize documents, learn the vocabulary and inverse document frequency weightings, and allow you to encode new documents[[9]](#footnote-9). Tfidf is focuses on the frequency of words present in the corpus and provides the importance of the words.

## Sentiment analysis

Sentiment analysis focuses on the polarity of a text (positive, negative, neutral) but it also goes beyond polarity to detect specific feelings and emotions (angry, happy, sad, etc), urgency (urgent, not urgent) and even intentions (interested v. not interested).

I used TextBlob *sentiment* function to iterate through on my content feature with a lambda function and return a numeric value that refers to a sentiment if the tweet was positive, negative, or neutral. I created another function to label the tweets by their sentiments where 1 will represent the positive, 0 the neutral, and -1 the negative sentiment.

I used *WordCloud* function to plot the negative and the positive words from content feature and I also plotted seaborn catplot to visualise the sentiments distribution in my data.

Now the dataset ready for machine learning (***Table 5 Pre-processed dataset for ML***).

## Machine learning models

First step I defined the input and the target variables. I have a multilabel classification problem as I have three class in my target variable. Then I will be building a simple sentiment analysis classifier on top of livestock reviews, that will classify if the user review of the matter was positive, negative, or neutral. I will use Logistic Regression, Decision Tree Classifier and Support Vector Machine algorithms.

Logistic Regression and Support Vector Machine performed good but based on their training and testing accuracy it seems the models are slightly overfitted. Recall score is not too high which means models are identifying True Positives average. F1-score is the harmonic mean of the precision (training accuracy) and Recall. It could be a bit higher but as my precision is slightly low then it makes F1 score bit lower.

Decision Tree Classifier model is overfitted as the model performed much better on training data and poor on test data.

The best performed ML model is Support Vector Machine.

## Cross Validation

Cross-validation is a statistical method used to estimate the performance of machine learning models. It is also helps to avoid overfitting issue. I used Repeated Kfold Cross Validation what provides a way to improve the estimated performance of a machine learning model. This involves simply repeating the cross-validation procedure multiple times and reporting the mean result across all folds from all runs[[10]](#footnote-10).

After defined Cross Validation parameters with 10 splits and 3repeats I applied it for my models. Logistic Regression performed a bit better after cross validation. Decision Tree Classifier and Support Vector Machine dropped a bit of the accuracy after cross validation. After using Grid Search on Support Vector Machine and run the model again with the best parameters I got the same accuracy score what I got without cross validation.

# Deployment

After all statistical observations and, visualisations and machine learning models I can state that Export Unit Value Index value is strongly affected by other export indexes, and it is predictable with high accuracy based on the regression models I have presented in my study.

# Github repository

<https://github.com/zsolt-adam/CA2-CCT.git>

# References

Liberto D. (2021), Investopedia, Import and Export Price Indexes (MXP),

Availabel at: <https://www.investopedia.com/terms/i/import-export-prices.asp>

Stratis K. (2022), Real Python, Combining Data in Pandas With merge(), .join(), and concat(),

Available at: <https://realpython.com/pandas-merge-join-and-concat/>

Lee A. (2019), Towards Data Science, Why And How To Use Merge With Pandas in Python,

Available at: <https://towardsdatascience.com/why-and-how-to-use-merge-with-pandas-in-python-548600f7e738>

(2009) Washington, Export and Import Price Index Manual, International Monetary Fund,

Available at: <https://stats.gov.bb/wp-content/uploads/2020/05/xipim-Manual.pdf>

Plotly|Graphing Libraries, Linear and Non-Linear Trendlines in Python,

Available at: <https://plotly.com/python/linear-fits/>

Plotly|Graphing Libraries, Box Plots in Python,

Available at: <https://plotly.com/python/box-plots/>

(2022), Geek for Geeks, Enumerate() in Python,

Available at: <https://www.geeksforgeeks.org/enumerate-in-python/>

Willems K. (2020), DataCamp, Python Functions Tutorial,

Available at: <https://www.datacamp.com/tutorial/functions-python-tutorial>

Jupyter meets the Earth, Jupyter Widgets

Available at: <https://jupytearth.org/jupyter-resources/ecosystem/widgets.html>

Jupyter widgets, Widget list

Available at: <https://ipywidgets.readthedocs.io/en/latest/examples/Widget%20List.html>

Koehrsen W. (2019), Towards Data Science, Interactive Controls in Jupyter Notebooks

Available at: <https://towardsdatascience.com/interactive-controls-for-jupyter-notebooks-f5c94829aee6>

Laerd statistics, Independent t-test for two samples

Available at: <https://statistics.laerd.com/statistical-guides/independent-t-test-statistical-guide.php>

Python for data science, Wilcoxon Sign-Ranked Test

Available at: <https://pythonfordatascienceorg.wordpress.com/wilcoxon-sign-ranked-test-python/>

(2018), Data Vedas, Inferential statistics in python

Available at: <https://www.datavedas.com/inferential-statistics-in-python/>

Brownlee J. (2020), Machine Learning Mastery, Ordinal and One-Hot Encodings for Categorical Data

Available at: <https://machinelearningmastery.com/one-hot-encoding-for-categorical-data/>

Boyle T. (2019), Towards Data Science, Linear Regression Models

Available at: <https://towardsdatascience.com/linear-regression-models-4a3d14b8d368>

Chauhan G. (2021), Machine Learning HD, Gridsearchcv for regression

Available at: <https://machinelearninghd.com/gridsearchcv-hyperparameter-tuning-sckit-learn-regression-classification/>

(2021), Geek for Geeks, SVM Hyperparameter Tuning using GridSearchCV | ML

Available at: <https://www.geeksforgeeks.org/svm-hyperparameter-tuning-using-gridsearchcv-ml/>

Shin T. (2020), Towards Data Science, All Machine Learning Models Explained in 6 Minutes

Available at: <https://towardsdatascience.com/all-machine-learning-models-explained-in-6-minutes-9fe30ff6776a>

Karajgi A. (2021)., Towards Data Science, Evaluating Multi-label Classifiers

Available at: <https://towardsdatascience.com/evaluating-multi-label-classifiers-a31be83da6ea>

Bharathi (2021), Analytics Vidhya, Confusion Matrix for Multi-Class Classification

Available at: <https://www.analyticsvidhya.com/blog/2021/06/confusion-matrix-for-multi-class-classification/>

Mogyorosi M. (2021), Real Python, Sentiment Analysis: First Steps With Python's NLTK Library

Available at: <https://realpython.com/python-nltk-sentiment-analysis/>

Brownlee J. (2020), Machine Learning Mastery, How to Develop Ridge Regression Models in Python

Available at: <https://machinelearningmastery.com/ridge-regression-with-python/>

Saeed M., StackAbuse, Calculating Pearson Correlation Coefficient in Python with Numpy,

Available at: <https://stackabuse.com/calculating-pearson-correlation-coefficient-in-python-with-numpy/>

Shier R. (2004), Mathematics Learning Support Centre, Statistics: 2.3 The Mann-Whitney U Test, Introduction, p.1., Available at: <https://www.statstutor.ac.uk/resources/uploaded/mannwhitney.pdf>

Zach (2020), Statology, How to perform a Shapiro-Wilk test in Python, Available at: <https://www.statology.org/shapiro-wilk-test-python/>

Hayes A. (2021), Investopedia, Wilcoxon test, Available at: <https://www.investopedia.com/terms/w/wilcoxon-test.asp>

Bonthu H. (2021), Analytics Vidhya, Rule-Based Sentiment Analysis in python, Available at: <https://www.analyticsvidhya.com/blog/2021/06/rule-based-sentiment-analysis-in-python/>

Brownlee J. (2017), Machine Learning Mastery, How to encode text data for machine learning with scikit-learn, Available at: <https://machinelearningmastery.com/prepare-text-data-machine-learning-scikit-learn/>

Brownlee J. (2020), Machine Learning Mastery, Repeated k-fold cross-validation for model evaluation in pyhton, Available at: <https://machinelearningmastery.com/repeated-k-fold-cross-validation-with-python/>

# Appendix

## Figures

Chart

Description automatically generated

Figure Boxplot for outlier detection

Chart, line chart

Description automatically generated

Figure Scatter plot for Export Quality Index vs. Export Value Index

Chart

Description automatically generated

Figure Scatter plot for Export Value Base Quantity vs. Value Base Price

Chart, histogram

Description automatically generated

Figure Distribution of Livestock by countries

Chart

Description automatically generated

Figure Scatter plot with Dashboard for numeric features

Chart, bar chart

Description automatically generated

Figure Dashboard for barchart

Chart, bar chart

Description automatically generated

Figure 8 MLA comparison before hyperparameter tuning

Chart, bar chart

Description automatically generated

Figure 9 MLA comparison after hyperparameter tuning

## Tables

Table

Description automatically generated

Table Correlation strength guidance

Graphical user interface

Description automatically generated

Table 2 Covariance test for numeric features

Table

Description automatically generated

Table 3 Statistical summary for numeric features

Table

Description automatically generated

Table 4 Normalized and encoded dataset

Graphical user interface, table

Description automatically generated with medium confidence

Table 5 Pre-processed dataset for ML

1. Enterprise Ireland, Start a business in Ireland

   Available at: <https://www.enterprise-ireland.com/en/start-a-business-in-ireland/food-investment-from-outside-ireland/key-sectors/meat-and-livestock/> [↑](#footnote-ref-1)
2. Dr.Golya G. (2016), 2016 Exporter guide Hungary, Market overview, p.4.

   Available at: <https://apps.fas.usda.gov/newgainapi/api/report/downloadreportbyfilename?filename=Exporter%20Guide_Budapest_Hungary_12-9-2016.pdf> [↑](#footnote-ref-2)
3. # Saeed M., StackAbuse, Calculating Pearson Correlation Coefficient in Python with Numpy,

   Available at: <https://stackabuse.com/calculating-pearson-correlation-coefficient-in-python-with-numpy/> [↑](#footnote-ref-3)
4. # Brownlee J. (2018), Machine Learning Mastery, How to Calculate Parametric Statistical Hypothesis Tests in Python, Available at: <https://machinelearningmastery.com/parametric-statistical-significance-tests-in-python/>

   [↑](#footnote-ref-4)
5. Zach (2020), Statology, How to perform a Shapiro-Wilk test in Python, Available at: <https://www.statology.org/shapiro-wilk-test-python/> [↑](#footnote-ref-5)
6. Shier R. (2004), Mathematics Learning Support Centre, Statistics: 2.3 The Mann-Whitney U Test, Introduction, p.1., Available at: <https://www.statstutor.ac.uk/resources/uploaded/mannwhitney.pdf> [↑](#footnote-ref-6)
7. Hayes A. (2021), Investopedia, Wilcoxon test, Available at: <https://www.investopedia.com/terms/w/wilcoxon-test.asp> [↑](#footnote-ref-7)
8. Bonthu H. (2021), Analytics Vidhya, Rule-Based Sentiment Analysis in python, Available at: <https://www.analyticsvidhya.com/blog/2021/06/rule-based-sentiment-analysis-in-python/> [↑](#footnote-ref-8)
9. Brownlee J. (2017), Machine Learning Mastery, How to encode text data for machine learning with scikit-learn, Available at: <https://machinelearningmastery.com/prepare-text-data-machine-learning-scikit-learn/> [↑](#footnote-ref-9)
10. Brownlee J. (2020), Machine Learning Mastery, Repeated k-fold cross-validation for model evaluation in pyhton, Available at: <https://machinelearningmastery.com/repeated-k-fold-cross-validation-with-python/> [↑](#footnote-ref-10)