National College of Ireland

Configuration Manual

MSc Research Project
Data Analytics

Muskaan Kapoor
Student ID: X22105476

School of Computing
National College of Ireland

Supervisor: Teerath Kumar Menghwar

National College of Ireland
Project Submission Sheet
School of Computing

National College of Ireland

Student Name:	Muskaan Kapoor
Student ID:	22105476
Programme:	Data Analytics
Year:	2023
Module:	MSc Research Project
Supervisor:	Teerath Kumar Menghwar
Submission Due Date:	14/12/2023
Project Title:	Configuration Manual
Word Count:	1001
Page Count:	9

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	Muskaan Kapoor
Date:	14th December 2023

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	☐
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	☐
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	☐

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Configuration Manual

Muskaan Kapoor
22105476

1 Introduction

This configuration manual document consists enlist all the hardware and software requirements and the steps that were performed during the research "Supply and Disappearance of food grains in USA". The main objective of this project was to analyze the trends for demand of food grains such as Corn, Barley, Sorghum and Oat in U.S. The following sections of the handbook will discuss all the hardware and software specifications, environment setup, data cleaning and transformation steps opted during the study.

2 Hardware and Software Specifications

This section of the manual will address all the hardware and software specifications.

2.1 Hardware Specifications

1. Device: HP Pavilion
2. Operating System : Windows 11
3. Processor: AMD Ryzen 5 5625U with Radeon Graphics
4. RAM: 16.0 GB
5. System: 64-bit operating system, x64-based processor
6. SSD: 476 GB

2.2 Software Specifications

1. Programming Language: Python 3.9.13
2. web browser: Google chrome
3. Softwares Used: Jupyter notebook

3 Environment Setup

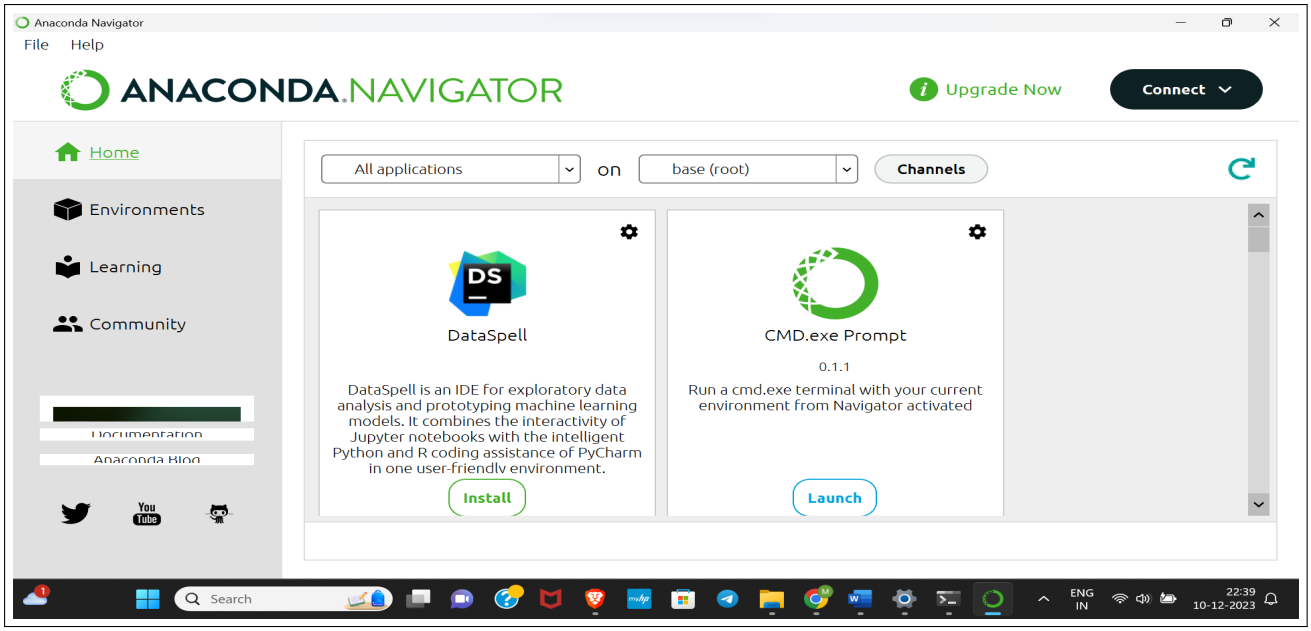

Figure 1: Interface for Anaconda

This section will go through all the steps that are required to run the code smoothly and efficiently. The first step is to install Anaconda[1] software and Jupyter[2] notebook. Fig.1 shows the interface for Anaconda software. Jupyter Notebook is used to run the code and python libraries are utilised for further analysis.

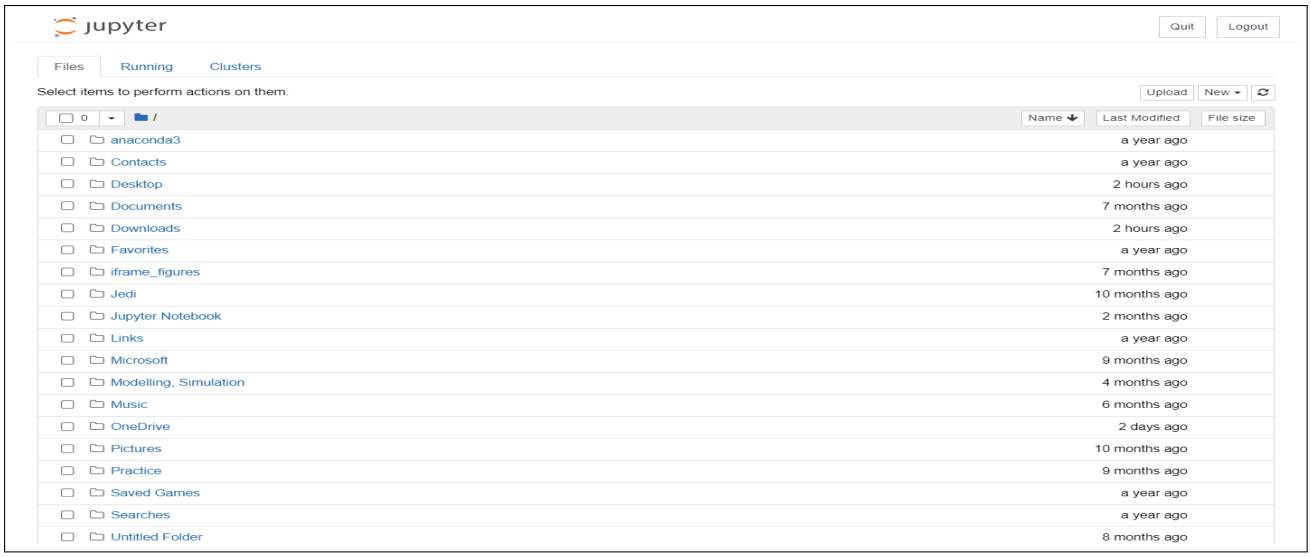

Figure 2: Interface for Anaconda

[1] https://www.anaconda.com/
[2] https://jupyter.org/install

Once jupyter notebook is installed, then click on the new button to create a new notebook. Fig.2 shows the main page of Jupyter notebook.

4 Data Preparation and Preprocessing

For this study the dataset is acquired from U.S. DEPARTMENT OF AGRICULTURE website and is in .CSV format. The next step is to import all the necessary libraries that are required for the study such as pandas, numpy, seaborn, matplotlib, sklearn and so on refer Fig.3.

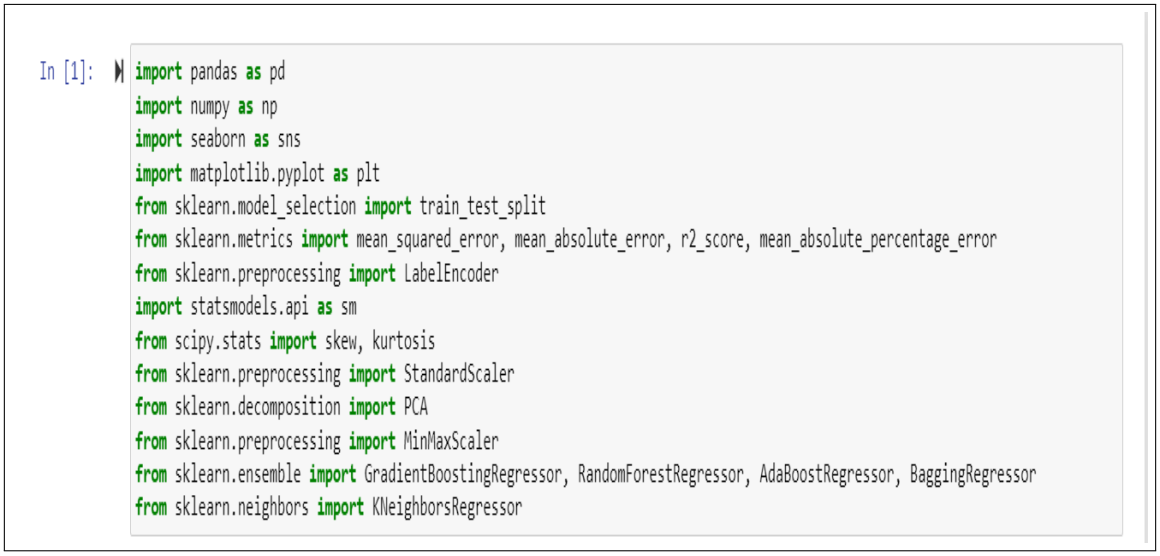

Figure 3: Importing libraries

After importing the libraries then data loading is done and data transformation was initiated. The original dataset consisted of the double column header to convert them into single column header column mapping was done so that further data preprocessing can be done effectively. After that columns were renamed and all the non available rows were removed from the dataset as shown in Fig.4.

```python
def transform_sheet(sheet_name, column_mapping, years_series_full):
    """Function to load and transform data from the specified sheet."""
    # Load the sheet with the specified header row
    data = pd.read_excel("Feed Grains Yearbook Tables-All Years.xlsx", sheet_name=sheet_name, header=3)

    # Rename columns based on the observed structure
    data_renamed = data.rename(columns=column_mapping)

    # Drop rows with NaN in the 'Quearter' column
    data_cleaned = data_renamed.dropna(subset=['Quearter'])

    # Extract the years from the original data and then assign them to the cleaned data
    data_cleaned['Year'] = [year for year in years_series_full if not pd.isnull(year)][0:len(data_cleaned)]

    return data_cleaned

# Column mapping for renaming
column_mapping_05 = {
    'Unnamed: 0': 'Year',
    'Unnamed: 1': 'Quearter',
    'Unnamed: 2': 'Supply Beginning stocks',
    'Unnamed: 3': 'Supply Production',
    'Unnamed: 4': 'Supply Imports',
    'Unnamed: 5': 'Supply Total  2/',
    'Domestic use': 'Disappearance Domestic use Food, alcohol, and industrial use',
    'Unnamed: 7': 'Disappearance Seed use',
    'Unnamed: 8': 'Disappearance Feed and residual use',
    'Unnamed: 9': 'Disappearance Total domestic use 2/',
    'Unnamed: 10': 'Disappearance Exports',
    'Unnamed: 11': 'Disappearance Total  2/',
    'Unnamed: 12': 'Ending stocks'
}
```

Figure 4: Column mapping and data loading

	Beginning stocks	Production	Imports	Total supply 2/	Food, alcohol, and industrial use	Seed use	Feed and residual use	Total domestic use 2/	Exports	Total disappearance 2/	Endin stock
count	239.000000	239.000000	239.00000	239.000000	239.000000	239.000000	239.000000	239.000000	239.000000	239.000000	239.00000
mean	4344.299979	4076.523950	7.35849	8428.182418	1255.745523	9.500996	2013.499230	3278.745749	769.399611	4048.145360	4380.03705
std	2994.457411	5292.137197	13.97842	3861.688901	1551.767176	11.246408	1605.784874	2889.658782	611.137762	3420.264353	2983.71931
min	425.942000	0.000000	0.00300	1720.749000	114.400000	0.000000	246.966000	765.580000	150.897000	956.392000	425.94200
25%	1717.549000	0.000000	0.85600	5225.860000	349.400000	0.000000	952.497500	1459.265500	418.881500	1965.165000	1724.52900
50%	3799.541000	0.000000	3.41500	8057.562000	639.000000	1.922000	1305.189000	2195.344000	503.245000	2678.233000	3848.20000
75%	6535.272000	8875.453000	8.14850	11077.903500	1620.332000	20.050000	2200.961000	3577.542000	668.453500	4073.645000	6567.20800
max	12566.501000	15265.000000	159.94600	16942.164000	7027.145000	31.000000	6131.649000	12484.280000	2746.941000	14955.669000	12566.50100

Figure 5: Displaying first 5 rows of Corn dataset

To print the first 5 rows of the dataset .head() was used refer Fig.5. The figure shows the head for Corn dataset.

After that basic function were applied on the dataset to check the size, shape, datatype of the dataset. To check if the dataset consist of any null values .isnull() function was utilized as shown in Fig.6.

```
In [7]: df_fgyearbook04_Corn.isnull().sum()

Out[7]: Year                                0
        Quearter                            0
        Beginning stocks                    0
        Production                          0
        Imports                             0
        Total supply 2/                     0
        Food, alcohol, and industrial use   0
        Seed use                            0
        Feed and residual use               0
        Total domestic use 2/               0
        Exports                             0
        Total disappearance 2/              0
        Ending stocks                       0
        dtype: int64
```

Figure 6: Checking for null values

5 Feature Selection

Feature Extraction is one of the vital step that is used to transform raw data into group of features that makes dataset more manageable as shown in Fig.7 It improves model performance and reduces the processing time. For this study PCA is used in Feature Extraction.

```
# Feature Extraction
# Using PCA for feature extraction
pca = PCA(n_components=5)  # the number of components should be less than the number of original features
principal_components = pca.fit_transform(df_scaled_Corn)
principal_df_Corn = pd.DataFrame(data=principal_components, columns=['PC1', 'PC2', 'PC3', 'PC4', 'PC5'])
df_fgyearbook04_Corn.tail(5)
```

Figure 7: Feature Selection using PCA

6 Exploratory Data Analysis

EDA is one of the crucial step in a machine learning project that helps to learn about the trends, patterns or any anomalies in the dataset. It gives a summary about data and

its attributes. Below are the EDA performed on the Oats and Barley dataset.

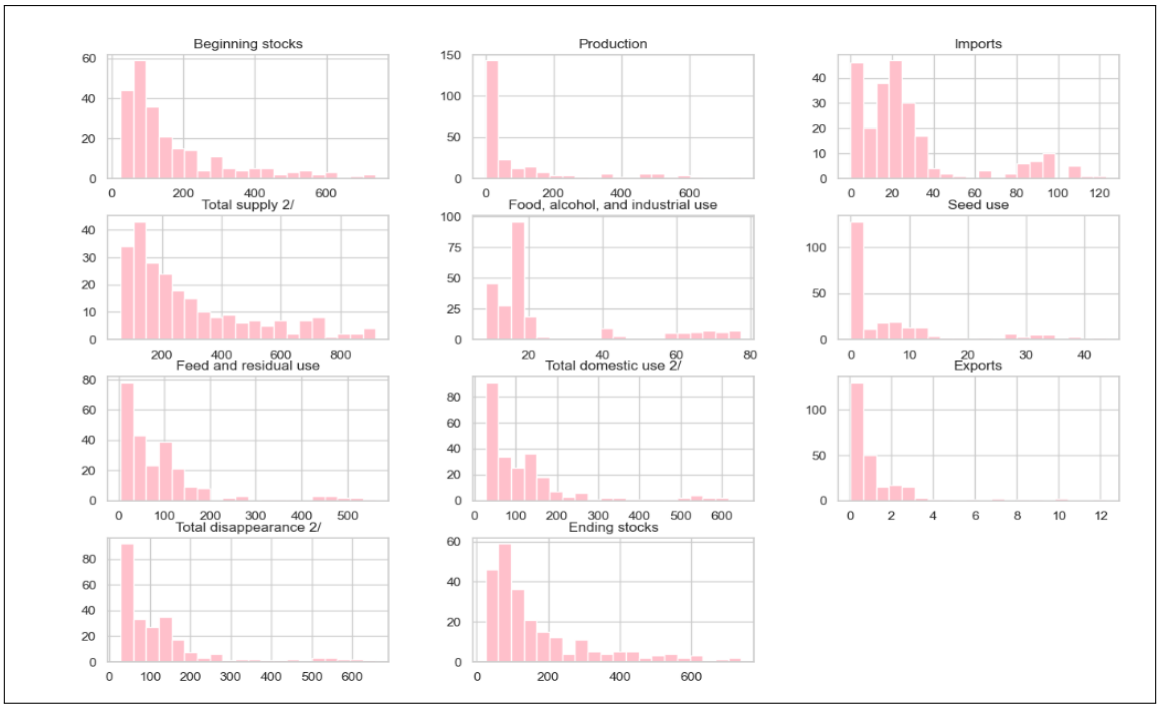

Figure 8: Histogram for all numerical columns for Oats

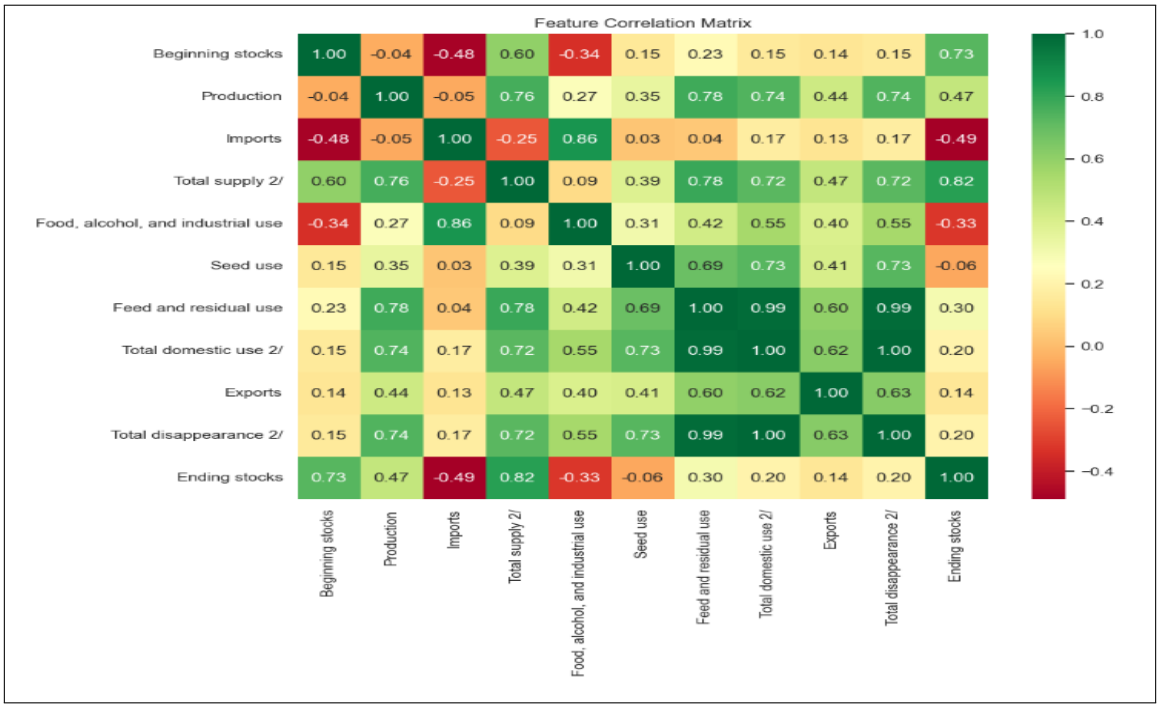

Figure 9: Correlation plot for Oats dataset

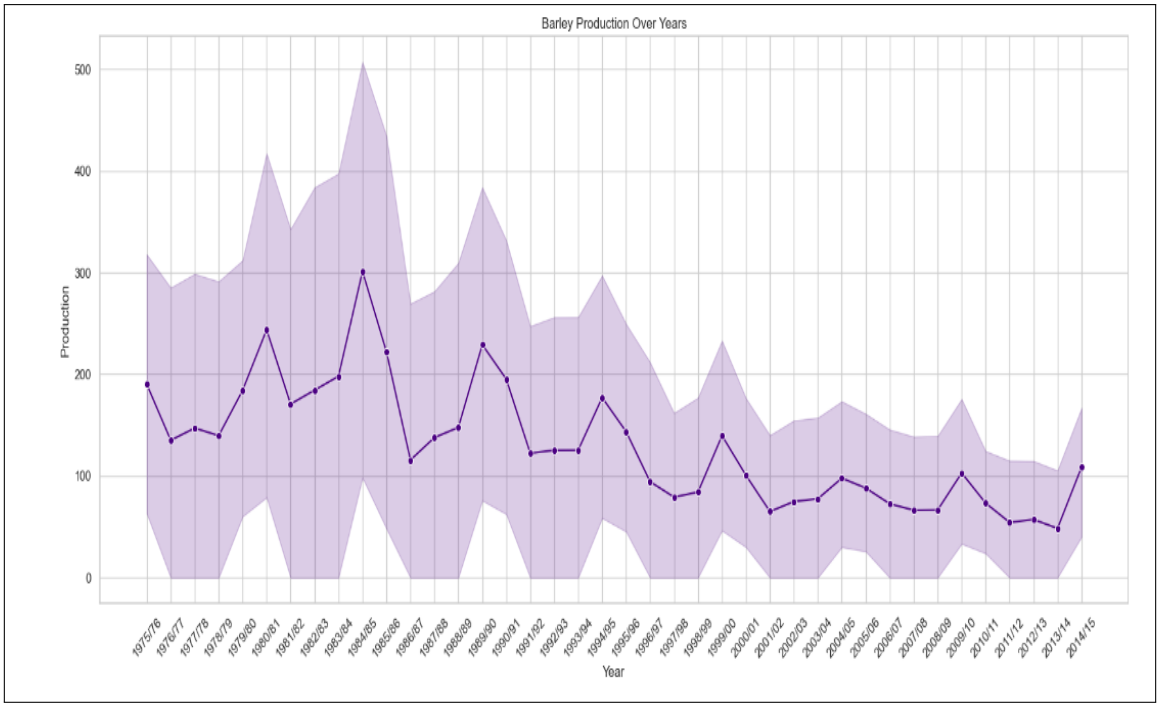

Figure 10: Barley Production over years

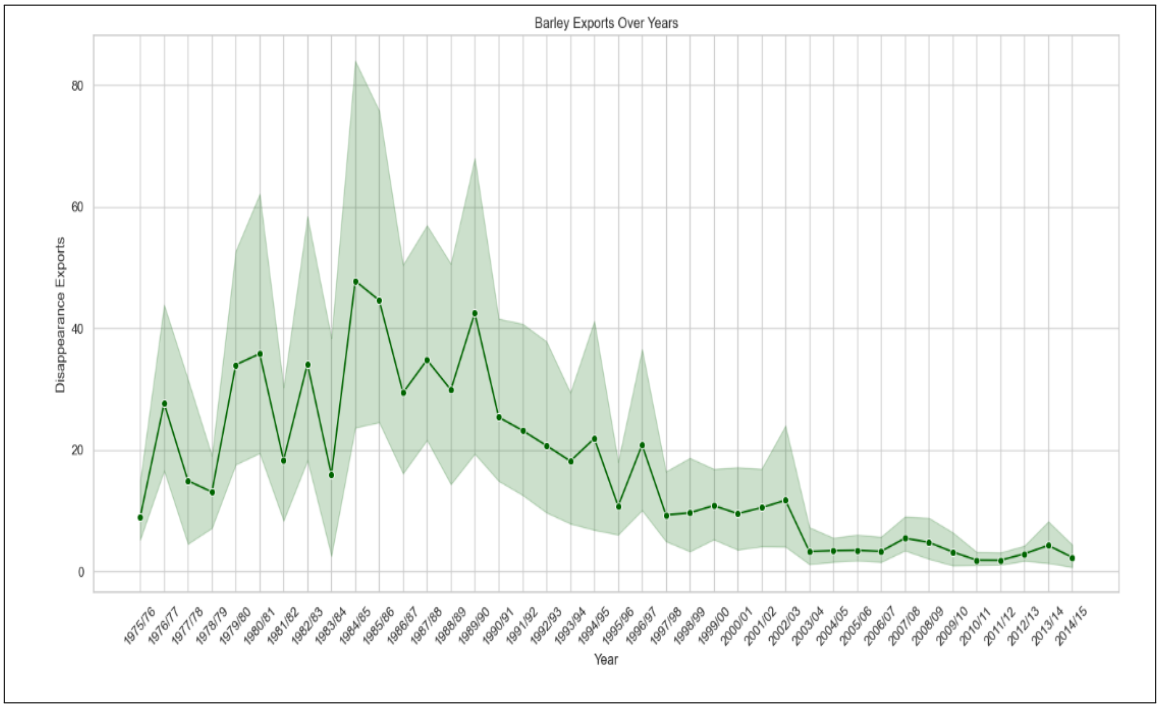

Figure 11: Barley Export over years

7 Model Building

Now the final step of research is building the model. In this stage lazy predict library was used, it is a machine learning library that makes predictions simple and efficient. This

library helps to simplify model selection without the need for extensive coding, enabling researchers to focus on selecting the best model for their data refer Fig.12.

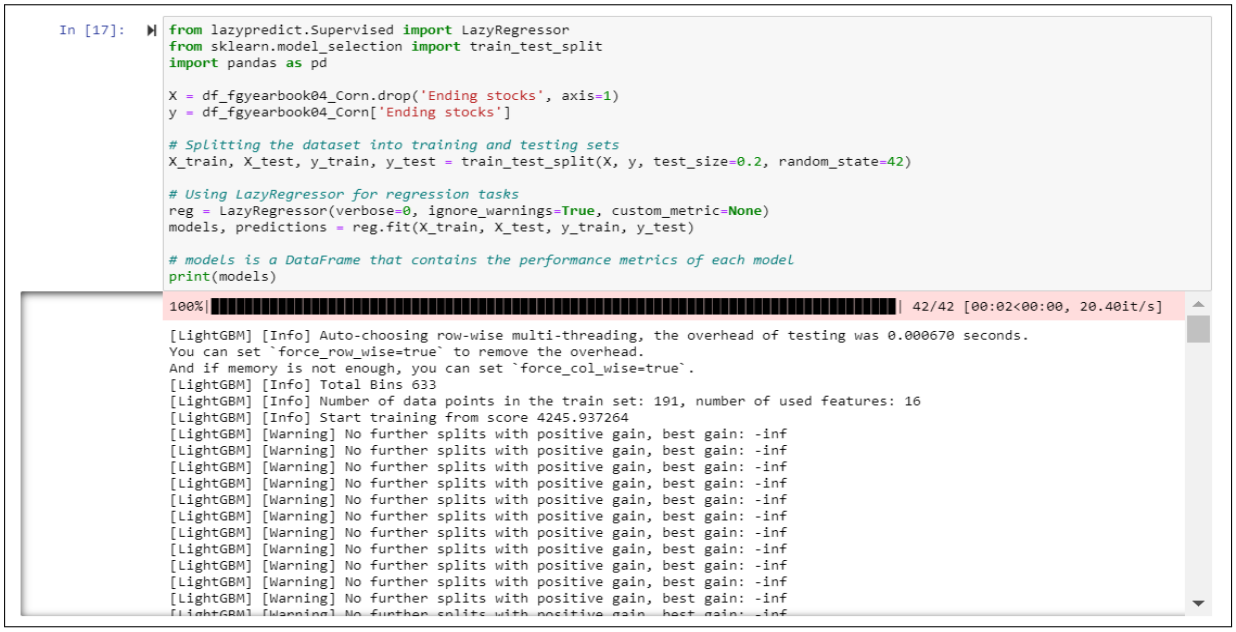

Figure 12: Lazy predict library

First the dataset is split into training and testing data in 80% to 20% ratio split which can be seen in the Fig.13.

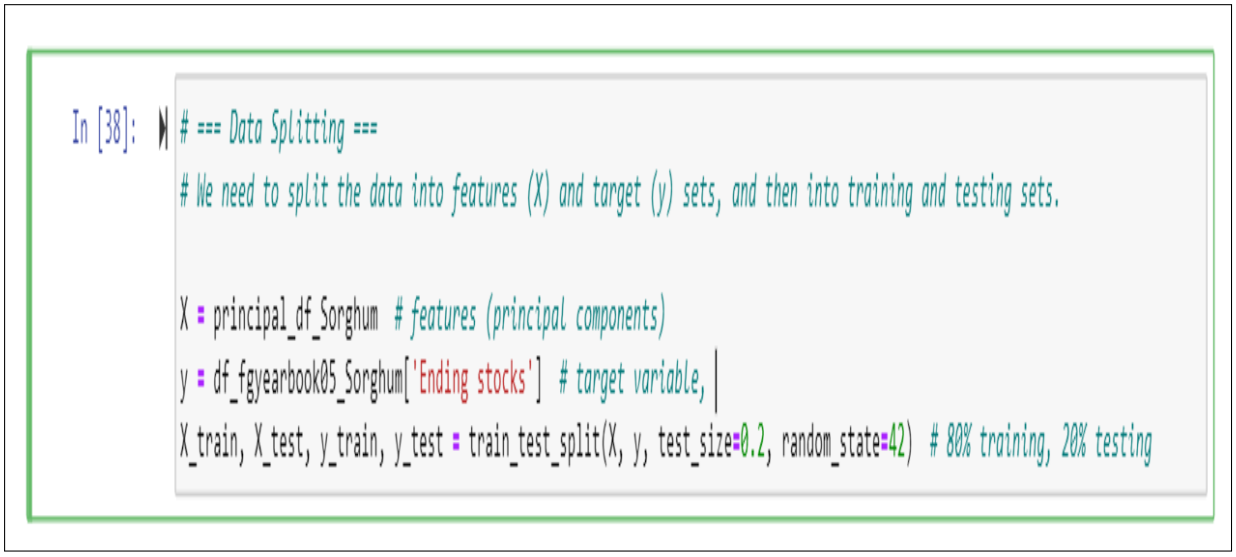

Figure 13: Testing-Training split on Sorghum dataset

In this research Gradient Boosting, Bagging, Random Forest and KNN regressor are utilized. Then the trained dataset will be evaluated based on on different evaluation metrices such as RMSE, MSE, MAE, MAPE, R-Squared, NMSE and all the results will be stored in the from of a dataframe as shown in Fig.14.

```python
from sklearn.metrics import mean_squared_error, mean_absolute_error, r2_score, mean_absolute_percentage_error
# Initialize models as before
models = {
    "Gradient Boosting Regressor": GradientBoostingRegressor(),
    "Bagging Regressor": BaggingRegressor(),
    "Random Forest Regressor": RandomForestRegressor(),
    "AdaBoost Regressor": AdaBoostRegressor(),
    "K-Neighbors Regressor": KNeighborsRegressor()
}

results = {}

# Define feature transformation
scaler = StandardScaler()

# Train and evaluate each model
for name, model in models.items():
    model.fit(X_train, y_train)
    y_pred = model.predict(X_test)

    mse = mean_squared_error(y_test, y_pred)
    rmse = np.sqrt(mse)
    mae = mean_absolute_error(y_test, y_pred)
    mape = mean_absolute_percentage_error(y_test, y_pred)
    r2 = r2_score(y_test, y_pred)
    nmse = mse / np.var(y_test)

    results[name] = {'RMSE': rmse, 'MAE': mae, 'MAPE': mape, 'R-squared': r2, 'NMSE': nmse}

# Creating a DataFrame from the results
results_df = pd.DataFrame(results).T

# Display the results table
print(results_df)
```

Figure 14: Models Applied