TAYLOR'S UNIVERSITY

Wisdom · Integrity · Excellence

TAYLOR'S UWE DUAL AWARDS PROGRAMMES

JANUARY 2024 SEMESTER

MACHINE LEARNING AND PARALLEL COMPUTING (ITS66604)

Individual Assignment (20%)

STUDENT DECLARATION

1. *I confirm that I am aware of the University's Regulation Governing Cheating in a University Test and Assignment and of the guidance issued by the School of Computing and IT concerning plagiarism and proper academic practice, and that the assessed work now submitted is in accordance with this regulation and guidance.*
2. *I understand that, unless already agreed with the School of Computing and IT, assessed work may not be submitted that has previously been submitted, either in whole or in part, at this or any other institution.*
3. *I recognize that should evidence emerge that my work fails to comply with either of the above declarations, then I may be liable to proceedings under Regulation.*

Student Name	Student ID	Date	Signature	Score
Aryan Kushwaha	**0363237**	**18th March 2024**		

Table of Contents

Introduction

Livestock and commodity production are critical components of Nepal's agricultural landscape, contributing considerably to the country's economy and people's livelihoods. In this practical exercise, we look at a large dataset encompassing 75 districts in Nepal, concentrating on different elements of livestock and commodity production. This dataset includes a wide variety of livestock populations, including horses, yak-nak-chauri, and rabbits, as well as major commodities including meat, cotton, eggs, wool, and animal milk.

Background

The agricultural landscape of Nepal, which is strongly anchored in its mountainous environment, is supported by a varied range of livestock and commodity production. Many rural populations rely on livestock, ranging from traditional breeds to rare Himalayan species, for food security and income. In this context, a dataset spanning 75 districts is a great resource for study. This dataset, which includes information on animal numbers and commodity output, provides a chance to identify trends and difficulties in Nepal's agricultural sector. By investigating aspects such as climatic variability and market demand, stakeholders may utilize data-driven insights to support strategic actions targeted at increasing agricultural productivity and sustainability, ultimately contributing to the country's economic development and rural well-being.

Objective

The goal of this practical test is to undertake a detailed study of a dataset concentrating on livestock and commodity output in 75 districts throughout Nepal. Our major goal is to thoroughly grasp the dataset's structure and derive useful insights through analysis and visualization. We want to identify patterns, trends, and linkages in the data by diving into many areas such as the population of different livestock species such as horses, yak-nak-chauri, rabbits, and the production levels of meat, cotton, eggs, wool, and animal milk. These insights will not only provide a clearer picture of the dynamics of livestock and commodity production throughout Nepal's many regions but will also be useful for strategic agricultural decision-making. Furthermore, we intend to evaluate regional differences.

Engaging with this study provides students with a diverse learning experience. Initially, they focus on practical data analysis abilities, traveling through real-world statistics to derive useful insights. Students obtain a sophisticated understanding of agricultural methods by investigating the dynamics of livestock and commodity production in Nepal's varied districts, including regional variances and market needs. Students improve their presentation abilities by successfully displaying facts.

Furthermore, the multidisciplinary aspect of agricultural research is revealed when students examine the socioeconomic ramifications of production patterns, developing critical thinking and problem-solving skills. Ethical aspects in agriculture, such as animal welfare and sustainability, improve the learning experience and spark conversations about responsible agricultural methods. Finally, manage the study endeavor from data gathering to analysis.

Research Goal

The primary goal of this study is to undertake a thorough examination of the livestock and commodity production dataset, which includes 75 districts in Nepal. The goal is to get a thorough understanding of Nepal's agricultural environment, with a special emphasis on livestock numbers and production levels of major commodities including meat, cotton, eggs, wool, and animal milk.

1. Identifying Production Trends: This research tries to investigate the factors that influence livestock and commodity production in Nepal. This involves investigating the effect of regional differences, meteorological variables, infrastructural availability, and market demand on output levels. Understanding these elements is critical for developing successful ways to increase agricultural output and resilience.

2. Regional Disparities Analysis: Another goal is to determine regional inequalities in livestock and commodity production among the 75 districts. By comparing production levels and agricultural methods across regions, the study hopes to discover possible areas of concern or opportunities for focused interventions. This research can help policymakers and stakeholders understand the importance of fair resource allocation and development initiatives.

3. Insights for Sustainable Development: The project aims to provide insights that might guide sustainable development strategies in Nepal's agriculture sector. By identifying sustainable practices, resource-efficient technology, and market-driven initiatives, the study hopes to promote environmentally friendly and socially inclusive agricultural growth.

4. Policy Recommendations: Finally, the research aims to give evidence-based policy recommendations to policymakers, government agencies, and other stakeholders. These suggestions will be based on the empirical findings from the dataset analysis and are intended to influence decision-making processes relating to agricultural policy formation, resource allocation, and investment priorities.

By achieving these research objectives, the project hopes to contribute to a better knowledge of Nepal's agricultural dynamics and give practical insights to help the sector expand, resilient, and sustainable.

Related Works

1. Situation of Livestock, Production, and its Products in Nepal (Research Paper)

Nepal's livestock sector plays a key role in its agricultural economy, with animals outnumbering people and serving as an essential aspect of rural life. The geographically diverse landscape of Nepal, which varies from high Himalayan to subtropical zones, has an impact on livestock farming and agricultural methods in its provinces and districts. The nation's secular constitution notwithstanding, cultural and religious beliefs are strong, and creatures like cows are regarded as sacred entities. The foundation of Nepalese agriculture is animal husbandry, which provides vital goods like milk, meat, and fertilizers as well as supporting rural livelihoods. Nonetheless, the industry faces several difficulties, such as a lack of funding, a lack of technical know-how, and cultural norms that have an impact on productivity and livestock husbandry.

The article explores the various issues that Nepal's livestock industry faces, including the effects of modernization, concerns about animal welfare, and prevalent superstitions. Animals' place in agriculture changed as a result of modern farming practices that replaced antiquated techniques with machinery. Animal welfare is also seriously threatened by problems like improper slaughter techniques, animal abandonment, and superstitious customs like animal sacrifice at festivals. To effectively address violations of animal rights, comprehensive policies, and public awareness campaigns are essential. These issues are made more difficult by the lack of strong regulations and enforcement.

The analysis's findings demonstrate how urgent actions are required to raise Nepalese livestock welfare and productivity. Animal productivity is not at its best when traditional farming methods, low literacy rates, and poor management techniques are combined. As a result, the nation must spend a large amount of money each year to meet the demand for animal products. However, nutritional deficiencies could be reduced, and animal productivity could be raised by taking advantage of Nepal's diverse climate and geography and encouraging sustainable livestock management techniques. Enforcing animal welfare laws, addressing cultural attitudes, and raising public awareness seem to be crucial steps in ensuring that Nepal's livestock resources are used sustainably and responsibly.

In conclusion, Nepal's livestock industry is at a crossroads with a plethora of obstacles to overcome but also a great deal of room to grow and prosper. Nepal can unlock its animals' latent productivity by putting into practice focused interventions to enhance management techniques, increase technical knowledge, and feed livestock properly. Promoting a more moral and sustainable method of managing livestock also requires addressing cultural attitudes and upholding laws pertaining to animal welfare. Nepal can take advantage of its rich agricultural history to create a livestock industry that is resilient and prosperous through coordinated efforts and wise investments.

2. Sustainable Livestock Production in Nepal: A Focus on Animal Nutrition Strategies (From Journals).

Nepal's livestock industry is essential to the country's economy because it generates jobs, revenue, and food. Its low productivity, however, is a problem that is made worse by changes in the environment, socioeconomics, and demography. This review evaluates the current state of Nepalese livestock production systems and suggests strategies tailored to individual species to improve sustainability and productivity. Improved feed quality and utilization are necessary for ruminants, which may involve adding additives like urea, molasses, and enzymes to improve digestibility. Furthermore, technologies that maintain anti-methanogenic nutrients and seasonal forages are essential for sustainability. The insects raised on plant leftovers are one example of a novel protein feed ingredient that could benefit monogastric livestock and promote the circular bio-economy. The review highlights the necessity of developing research infrastructure, increasing capacity, and fostering cooperation between the industry and research sectors to establish.

This paper explores the complexities of the livestock production industry in Nepal, analyzing the problems it faces today and suggesting creative solutions to improve sustainability and productivity. The article emphasizes the critical issues of low productivity and environmental impact amid shifting socio-economic and environmental dynamics while acknowledging the sector's pivotal role in Nepal's economy and its ability to provide essential food sources, income, and employment opportunities. It looks at strategies that are species-specific and promote better feed quality and utilization for ruminants, including the use of technologies and additives to increase digestibility and reduce methane emissions. It also talks about new feed ingredients for monogastric animals like poultry, like insects raised on plant leftovers. The article highlights the significance of stakeholder collaboration, capacity building, and policy interventions.

The results of the paper present a thorough evaluation of Nepal's livestock production situation today and offer suggestions for future strategies to raise sustainability and productivity. The study identifies the main obstacles and difficulties the Nepalese livestock industry faces by analyzing data from governmental and international organizations as well as the body of scientific literature that is currently available. These include problems like low output, deteriorating environmental conditions, and scarce resources. The article also projects future milk and meat consumption in Nepal, emphasizing the livestock industry's growing significance in supplying the country's population's nutritional needs in the face of rising urbanization and income levels. Additionally, the study provides a thorough analysis of the methodologies used to assess the trends in the production and demand for milk and meat.

In conclusion, the Nepalese livestock sectors are required to improve productivity and sustainability, and the livestock industry in Nepal needs focused interventions. Sustainable development can be aided by tactics like raising awareness of the circular bio-economy for monogastric animals, implementing cutting-edge feed additives, and enhancing feed quality. Furthermore, developing research infrastructure, fostering stakeholder collaboration, and increasing capacity are necessary for putting policies into action. Nepal can create a robust and productive livestock industry that promotes livelihoods, food security, and environmental conservation by tackling these issues and implementing sustainable practices.

Methodology

Research Design:

A quantitative research design is most suitable given the nature of the task. This is because we are working with numerical data about the numbers of livestock and the production of commodities in various Nepalese districts. The objective is to examine and forecast the trends and patterns in these datasets using machine learning techniques.

Data Collection Methods:

It is not necessary to use any additional data collection techniques because the data is already available in CSV files. However, in order to clean up and get the data ready for analysis, preprocessing steps will be required. This includes operations like managing missing values, coding categorical variables, and, if required, scaling features.

Sampling Strategy:

The selection of a sampling strategy is contingent upon both the research objectives and the dataset's representativeness. Simple random sampling might be used if the dataset is thought to be representative of all districts in Nepal's population. However, a stratified sampling approach might be more appropriate if certain districts are known to have unique characteristics or if there are particular research, questions related to specific regions. This guarantees that the sample accurately reflects the population's diversity and makes it possible to make better generalizations and predictions.

Data Analysis:

For data analysis, several machine-learning approaches will be used for data analysis, depending on the goals of the study. Continuous variables, like the levels of commodity production, can be predicted using regression models. Livestock types can be categorized using classification algorithms according to a range of features. Through the use of clustering techniques, patterns or groups within the data can be found, revealing regional differences in production methods. In order to maximize the effectiveness of the machine learning models and guarantee the accuracy of the outcomes, feature selection strategies, model evaluation approaches, and hyperparameter tuning will also be applied.

Ethical Considerations:

Research ethics are extremely important, especially when working with data that could have an impact on people or communities. The ethical implications of this analysis are limited because the data is publicly available and collected at the district level. If the study involves human subjects or sensitive data, it is imperative to guarantee data privacy and confidentiality, follow ethical guidelines for data usage, and secure the required approvals. Maintaining ethical standards and the reliability of research practices also depends on transparency in the methods and results reported.

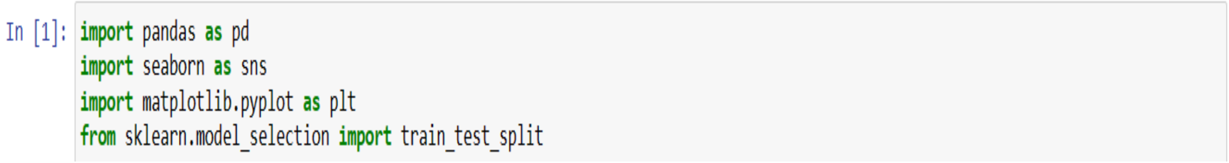
Feasibility and Resources:

The chosen methodology of secondary data analysis is feasible and requires minimal resources compared to primary data collection methods, to ensure that the project is completed successfully, it is essential to evaluate the analysis's resources and viability. This involves taking into factors like time constraints, financial limitations, computing resource accessibility, and machine learning technique proficiency. The implementation will be done in Python using libraries like matplotlib for visualization, sci-kit-learn for machine learning algorithms, and pandas for data manipulation. Depending on the size and complexity of the dataset, appropriate computing resources might be needed to train machine learning models.

Implementation

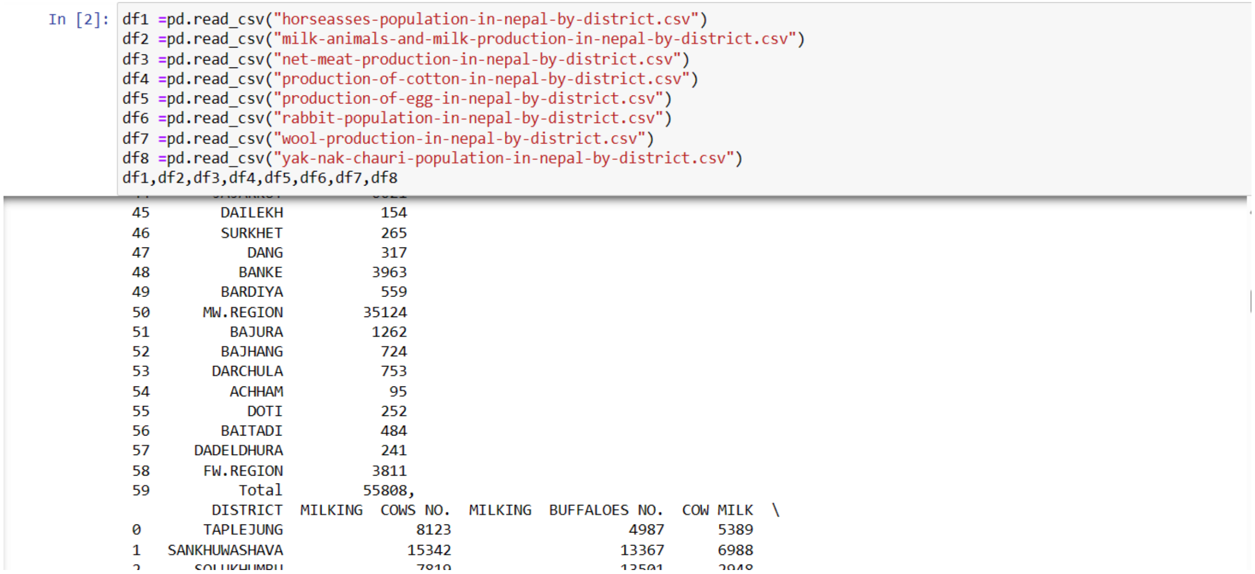
Section 1

Importing Libraries



```
In [1]: import pandas as pd
        import seaborn as sns
        import matplotlib.pyplot as plt
        from sklearn.model_selection import train_test_split
```

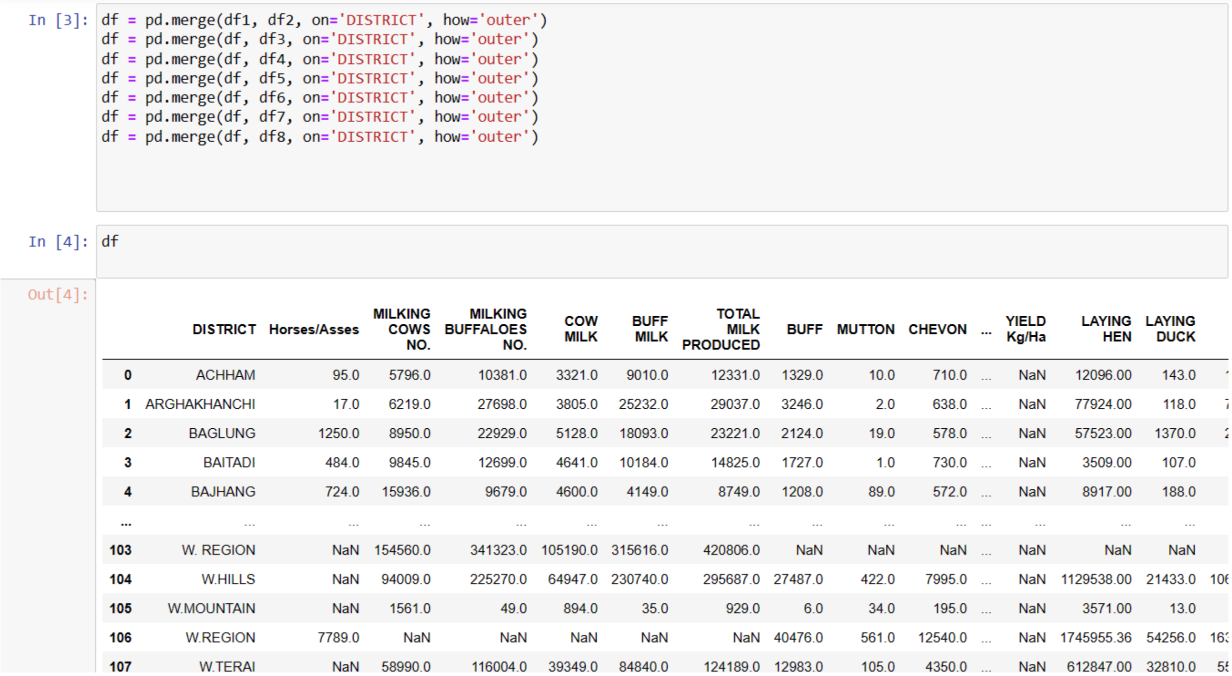
Reading CSV files



```
In [2]: df1 =pd.read_csv("horseasses-population-in-nepal-by-district.csv")
        df2 =pd.read_csv("milk-animals-and-milk-production-in-nepal-by-district.csv")
        df3 =pd.read_csv("net-meat-production-in-nepal-by-district.csv")
        df4 =pd.read_csv("production-of-cotton-in-nepal-by-district.csv")
        df5 =pd.read_csv("production-of-egg-in-nepal-by-district.csv")
        df6 =pd.read_csv("rabbit-population-in-nepal-by-district.csv")
        df7 =pd.read_csv("wool-production-in-nepal-by-district.csv")
        df8 =pd.read_csv("yak-nak-chauri-population-in-nepal-by-district.csv")
        df1,df2,df3,df4,df5,df6,df7,df8
```

```
 45        DAILEKH         154
 46        SURKHET         265
 47           DANG         317
 48          BANKE        3963
 49        BARDIYA         559
 50      MW.REGION       35124
 51         BAJURA        1262
 52        BAJHANG         724
 53       DARCHULA         753
 54         ACHHAM          95
 55           DOTI         252
 56        BAITADI         484
 57     DADELDHURA         241
 58      FW.REGION        3811
 59          Total       55808,
          DISTRICT  MILKING  COWS NO.  MILKING  BUFFALOES NO.  COW MILK  \
 0       TAPLEJUNG              8123                     4987      5389   
 1    SANKHUWASHAVA            15342                    13367      6988   
 2       SOLUKHUMBU             7819                    13501      2948   
```

Merging datasets



```
In [3]: df = pd.merge(df1, df2, on='DISTRICT', how='outer')
        df = pd.merge(df, df3, on='DISTRICT', how='outer')
        df = pd.merge(df, df4, on='DISTRICT', how='outer')
        df = pd.merge(df, df5, on='DISTRICT', how='outer')
        df = pd.merge(df, df6, on='DISTRICT', how='outer')
        df = pd.merge(df, df7, on='DISTRICT', how='outer')
        df = pd.merge(df, df8, on='DISTRICT', how='outer')
```

```
In [4]: df
```

Out[4]:

	DISTRICT	Horses/Asses	MILKING COWS NO.	MILKING BUFFALOES NO.	COW MILK	BUFF MILK	TOTAL MILK PRODUCED	BUFF	MUTTON	CHEVON	...	YIELD Kg/Ha	LAYING HEN	LAYING DUCK	
0	ACHHAM	95.0	5796.0	10381.0	3321.0	9010.0	12331.0	1329.0	10.0	710.0	...	NaN	12096.00	143.0	1
1	ARGHAKHANCHI	17.0	6219.0	27698.0	3805.0	25232.0	29037.0	3246.0	2.0	638.0	...	NaN	77924.00	118.0	7
2	BAGLUNG	1250.0	8950.0	22929.0	5128.0	18093.0	23221.0	2124.0	19.0	578.0	...	NaN	57523.00	1370.0	2
3	BAITADI	484.0	9845.0	12699.0	4641.0	10184.0	14825.0	1727.0	1.0	730.0	...	NaN	3509.00	107.0	
4	BAJHANG	724.0	15936.0	9679.0	4600.0	4149.0	8749.0	1208.0	89.0	572.0	...	NaN	8917.00	188.0	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
103	W. REGION	NaN	154560.0	341323.0	105190.0	315616.0	420806.0	NaN	NaN	NaN	...	NaN	NaN	NaN	
104	W.HILLS	NaN	94009.0	225270.0	64947.0	230740.0	295687.0	27487.0	422.0	7995.0	...	NaN	1129538.00	21433.0	106
105	W.MOUNTAIN	NaN	1561.0	49.0	894.0	35.0	929.0	6.0	34.0	195.0	...	NaN	3571.00	13.0	
106	W.REGION	7789.0	NaN	NaN	NaN	NaN	NaN	40476.0	561.0	12540.0	...	NaN	1745955.36	54256.0	163
107	W.TERAI	NaN	58990.0	116004.0	39349.0	84840.0	124189.0	12983.0	105.0	4350.0	...	NaN	612847.00	32810.0	55

Displaying rows

```
In [5]: pd.set_option("display.max_rows", None)
```

```
In [6]: df
```

```
25     DHANUSHA     NaN   19150.0   17700.0   14056.0   22710.0   36766.0   2538.0     2.0    1566.0 ...   NaN    103976.00   3
26     DOLAKHA      NaN   12344.0   13805.0    5544.0   10155.0   15699.0   1704.0    40.0     576.0 ...   NaN     67464.00   1
27     DOLPA     4115.0    2043.0     713.0    1100.0     514.0    1614.0     74.0   112.0     270.0 ...   NaN      1706.00
28     DOTI       252.0   15432.0   10381.0    7843.0    9010.0   16853.0   1625.0     2.0     840.0 ...   NaN     22999.00
29     Dang         NaN       NaN       NaN       NaN       NaN       NaN      NaN     NaN       NaN ... 700.0          NaN
30     E. REGION    NaN  332384.0  292178.0  196708.0  263199.0  459907.0      NaN     NaN       NaN ...   NaN          NaN
31     E.HILLS      NaN  123976.0  109431.0   74587.0   95837.0  170424.0  15440.0   146.0    5484.0 ...   NaN    549366.00   5
32     E.MOUNTAIN   NaN   31284.0   31855.0   15324.0   20339.0   35663.0   3376.0   100.0    1865.0 ...   NaN    135548.00   1
33     E.REGION  7616.0       NaN       NaN       NaN       NaN       NaN  41220.0   269.0   15729.0 ...   NaN   1780554.06  60
34     E.TERAI      NaN  177124.0  150892.0  106797.0  147023.0  253820.0  22404.0    23.0    8380.0 ...   NaN   1095640.00  53
35     FW. REGION   NaN  130595.0  132257.0   87936.0  112438.0  200374.0      NaN     NaN       NaN ...   NaN          NaN
36     FW.HILLS     NaN   45036.0   39569.0   22850.0   33505.0   56355.0   5692.0    14.0    3103.0 ...   NaN     48735.00
```

```
In [7]: df = df.drop(df.index[[1, 34, 48, 49, 47, 11, 72, 5, 13, 18, 50, 58, 59, 63, 60, 66, 67, 70, 71, 80, 85, 86, 87, 89, 90, 91, 92,
        #df.reset_index(drop=True, inplace=True)
```

Removing Total rows from the district column

```
In [8]: df = df.drop(['AREA (Ha.)', 'PROD. (Mt.)', 'YIELD Kg/Ha'], axis=1)
```

```
In [9]: df.rename(columns={"DISTRICT":"District"}, inplace=True)
        df.rename(columns={'COW MILK':'Cows Milk'}, inplace=True)
        df.rename(columns={'BUFF MILK':'Buffaloes Milk'}, inplace=True)
        df.rename(columns={'TOTAL MILK PRODUCED':'Total Milk Production'}, inplace=True)
        df.rename(columns={'MILKING  COWS NO.':'Milking Cows Number'}, inplace=True)
        df.rename(columns={'MILKING  BUFFALOES NO.':'Milking Buffaloes Number'}, inplace=True)
        df.rename(columns={'BUFF':'Buff'}, inplace=True)
        df.rename(columns={'MUTTON':'Mutton'}, inplace=True)
        df.rename(columns={'CHEVON':'Chevon'}, inplace=True)
        df.rename(columns={'PORK ':'Pork'}, inplace=True)
        df.rename(columns={'CHICKEN':'Chicken'}, inplace=True)
        df.rename(columns={'DUCK MEAT':'Duck Meat'}, inplace=True)
        df.rename(columns={'TOTAL MEAT':'Total Meat'}, inplace=True)
        df.rename(columns={'SHEEPS NO.':'Sheeps Number'}, inplace=True)
        df.rename(columns={'SHEEP WOOL PRODUCED':'Sheep Wool Produced'}, inplace=True)
        df.rename(columns={'YAK/NAK/CHAURI':'Yak/Nak/Chauri'}, inplace=True)
        df.rename(columns={'LAYING HEN':'Laying Hen'}, inplace = True)
        df.rename(columns={'LAYING DUCK':'Laying Duck'}, inplace = True)
        df.rename(columns={'HEN EGG':'Hen Egg'}, inplace = True)
        df.rename(columns={'DUCK EGG':'Duck Egg'}, inplace = True)
        df.rename(columns={'TOTAL EGG':'Total Egg'}, inplace = True)
```

```
In [10]: df
```

```
43    ILLAM     2815.0  26821.0   5759.0  19735.0  15261.0  34996.0  1974.0    1.0   870.0 ...  3365.0   26781.00   332.0  665
44    JAJARKOT  6021.0   5869.0   9102.0   3002.0   8763.0  11765.0   975.0   83.0   617.0 ...  1925.0   14589.00    91.0  229
45    JHAPA       42.0  36068.0  19327.0  29667.0  32457.0  62124.0  5241.0    0.0  1447.0 ...  9237.0  199044.00  3200.0  2861
46    JUMLA     6769.0   6206.0    585.0   2554.0    507.0   3061.0   106.0  240.0   202.0 ...   593.0    2605.00   303.0  45
```

Capitalizing columns

In [11]:
```
df['District'] = df['District'].str.capitalize()
```

In [12]:
```
df
```

```
29        Dang     NaN       NaN       NaN       NaN       NaN       NaN      NaN    NaN      NaN ...      NaN         NaN      NaN       NaN
30   E. region     NaN  332384.0  292178.0  196708.0  263199.0  459907.0      NaN    NaN      NaN ...      NaN         NaN      NaN       NaN
31     E.hills     NaN  123976.0  109431.0   74587.0   95837.0  170424.0  15440.0  146.0   5484.0 ...  28738.0   549366.00   5916.0   78878.0
32  E.mountain     NaN   31284.0   31855.0   15324.0   20339.0   35663.0   3376.0  100.0   1865.0 ...   7362.0   135548.00   1180.0   10271.0
33    E.region  7616.0       NaN       NaN       NaN       NaN       NaN  41220.0  269.0  15729.0 ...  76313.0  1780554.06  60193.0  214241.0
35  Fw. region     NaN  130595.0  132257.0   87936.0  112438.0  200374.0      NaN    NaN      NaN ...      NaN         NaN      NaN       NaN
36    Fw.hills     NaN   45036.0   39569.0   22850.0   33505.0   56355.0   5692.0   14.0   3103.0 ...   9133.0    48735.00    535.0    7717.0
37 Fw.mountain     NaN   37637.0   23773.0   14035.0   16380.0   30415.0   2684.0  223.0   1460.0 ...   4545.0    25484.00    487.0    2615.0
38   Fw.region  3811.0       NaN       NaN       NaN       NaN       NaN  18154.0  335.0   6893.0 ...  29107.0   537737.00   6372.0   40743.0
39    Fw.terai     NaN   47922.0   68915.0   51051.0   62553.0  113604.0   9778.0   98.0   2330.0 ...  15429.0   463517.80   5350.0   30411.0
40      Gorkha   854.0   14927.0   17250.0    7653.0   10468.0   18121.0   3147.0  100.0    815.0 ...   4951.0    78538.00    425.0    4598.0
41       Gulmi    54.0    6860.0   11742.0    4346.0   11233.0   15579.0   1446.0   23.0    821.0 ...   2723.0    33577.00    266.0    3837.0
42       Humla  4523.0    1677.0     496.0     961.0     421.0    1382.0     64.0   96.0    125.0 ...    297.0     2835.00     40.0     446.0
```

Section -2

Correcting the district actual name

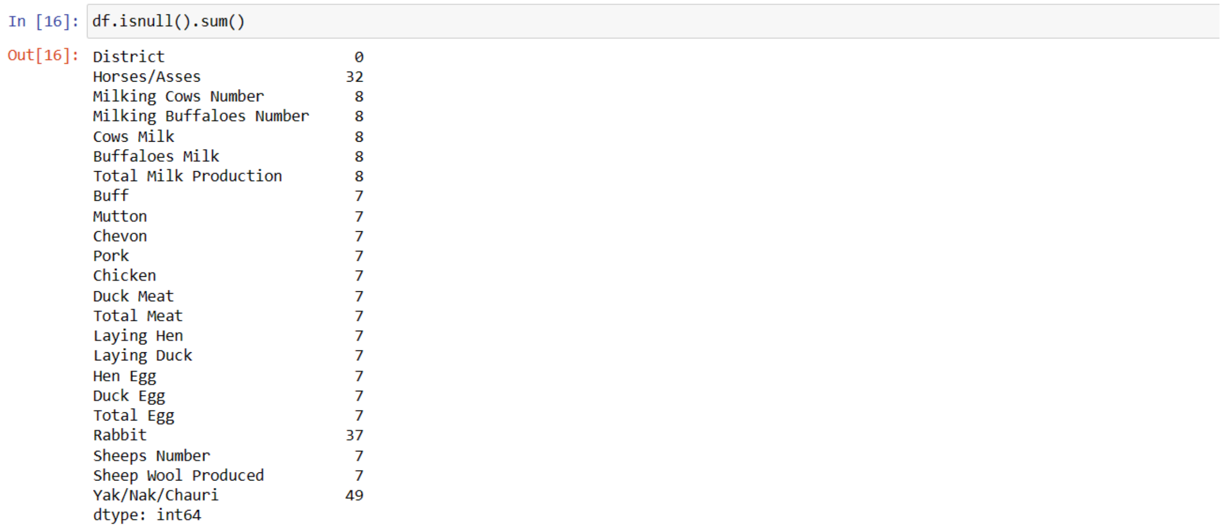
In [13]:
```
corrections = {
    "Illam": "Ilam",
    "Kapilbastu": "Kapilvastu"
}
```

In [14]:
```
df['District'] = df['District'].replace(corrections, regex=True)
```

In [15]:
```
df
```

```
12     Bardiya     NaN       NaN       NaN       NaN       NaN       NaN      NaN    NaN      NaN ...       NaN         NaN      NaN       NaN
14     C.hills     NaN  125519.0  187803.0   78958.0  187149.0  266107.0  23305.0  147.0   6777.0 ...   46715.0  3222902.00  17417.0  276415.0
15  C.mountain     NaN   21380.0   32607.0   13173.0   30261.0   43434.0   4486.0   85.0   1821.0 ...    7785.0   232271.00   3423.0   16482.0
16    C.region  1468.0       NaN       NaN       NaN       NaN       NaN  50244.0  256.0  16893.0 ...  100620.0  7118554.32  49572.0  756783.0
17     C.terai     NaN  116829.0  157331.0   85684.0  141074.0  226758.0  22453.0   24.0   8295.0 ...   46120.0  3663381.00  28732.0  463886.0
19  Dadeldhura   241.0   13963.0    6108.0    7045.0    5301.0   12346.0   1011.0    1.0    823.0 ...    1934.0    10131.00    205.0    1596.0
20     Dailekh   154.0    9438.0   24351.0    5408.0   13942.0   19350.0   2145.0   50.0    578.0 ...    3082.0    43020.00    313.0    6776.0
21        Dang   317.0   18630.0   31882.0    9984.0   18043.0   28027.0   3507.0  115.0   3352.0 ...    9261.0   411349.00   3223.0   54787.0
22    Darchula   753.0    9682.0    8560.0    4548.0    7430.0   11978.0    768.0   68.0    437.0 ...    1313.0     6723.00    101.0     778.0
23     Dhading     NaN   24068.0   36469.0   13791.0   24416.0   38207.0   3105.0   16.0    314.0 ...    6819.0   604699.00   3576.0   28205.0
24    Dhankuta     NaN   12523.0    5391.0    8176.0    4579.0   12755.0   1521.0    2.0    405.0 ...    2863.0   120878.00   1400.0    1878.0
25    Dhanusha     NaN   19150.0   17700.0   14056.0   22710.0   36766.0   2538.0    2.0   1566.0 ...    4721.0   103976.00   3885.0    8650.0
```

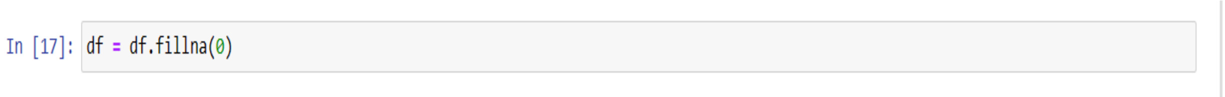
Checking for null values



In [16]:
```
df.isnull().sum()
```

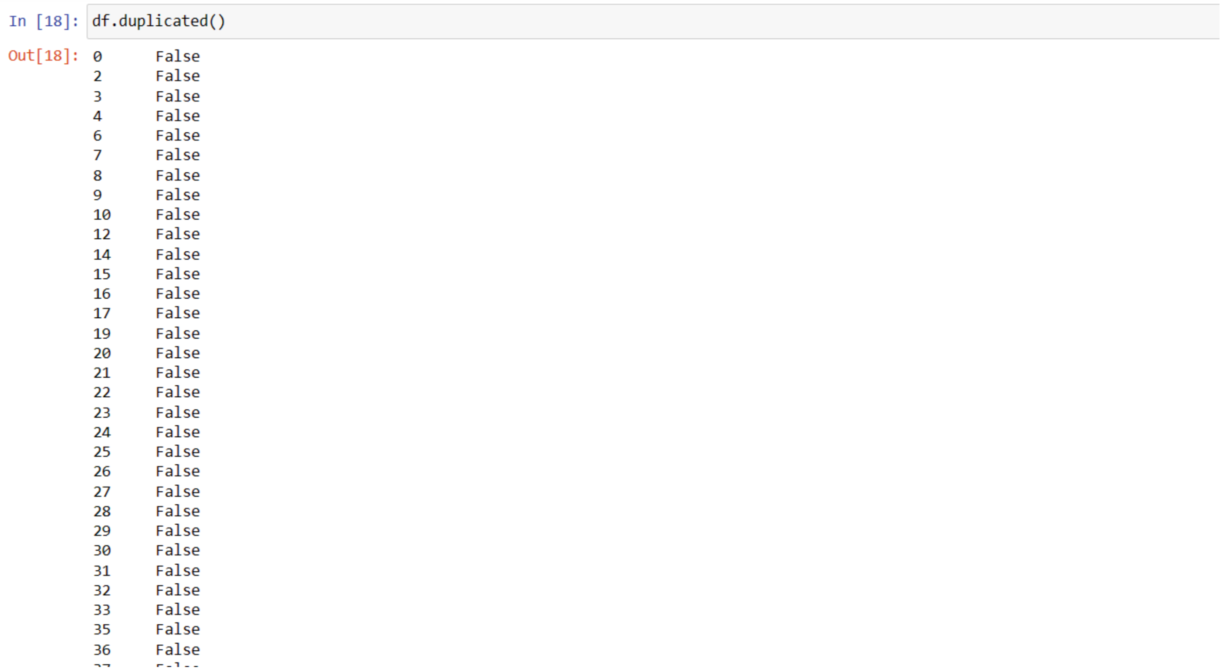
Out[16]:
```
District                      0
Horses/Asses                 32
Milking Cows Number           8
Milking Buffaloes Number      8
Cows Milk                     8
Buffaloes Milk                8
Total Milk Production         8
Buff                          7
Mutton                        7
Chevon                        7
Pork                          7
Chicken                       7
Duck Meat                     7
Total Meat                    7
Laying Hen                    7
Laying Duck                   7
Hen Egg                       7
Duck Egg                      7
Total Egg                     7
Rabbit                       37
Sheeps Number                 7
Sheep Wool Produced           7
Yak/Nak/Chauri               49
dtype: int64
```

Replacing null values with 0



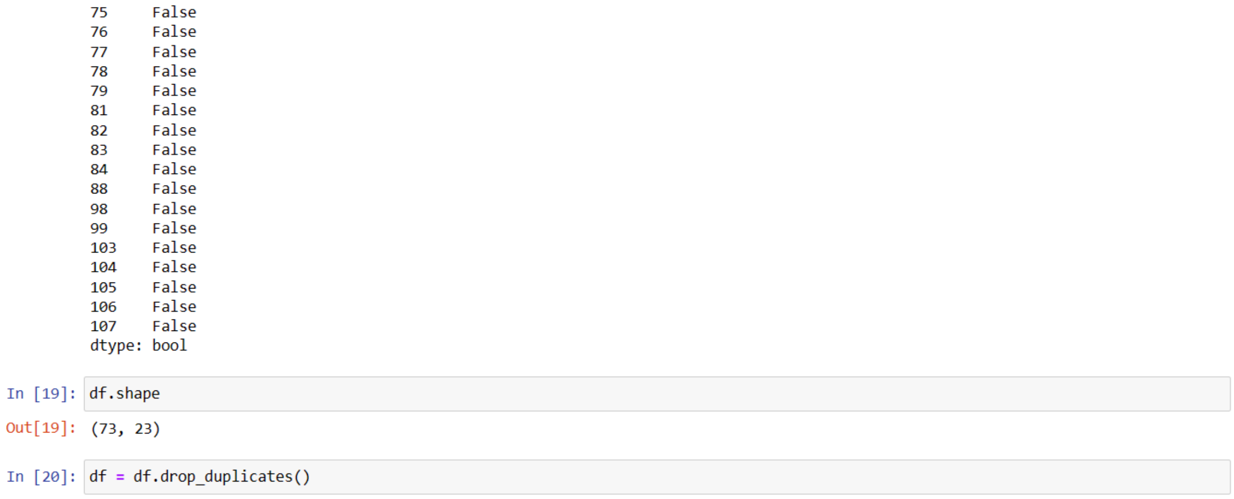
In [17]:
```
df = df.fillna(0)
```

Checking for duplicate values



In [18]:
```
df.duplicated()
```

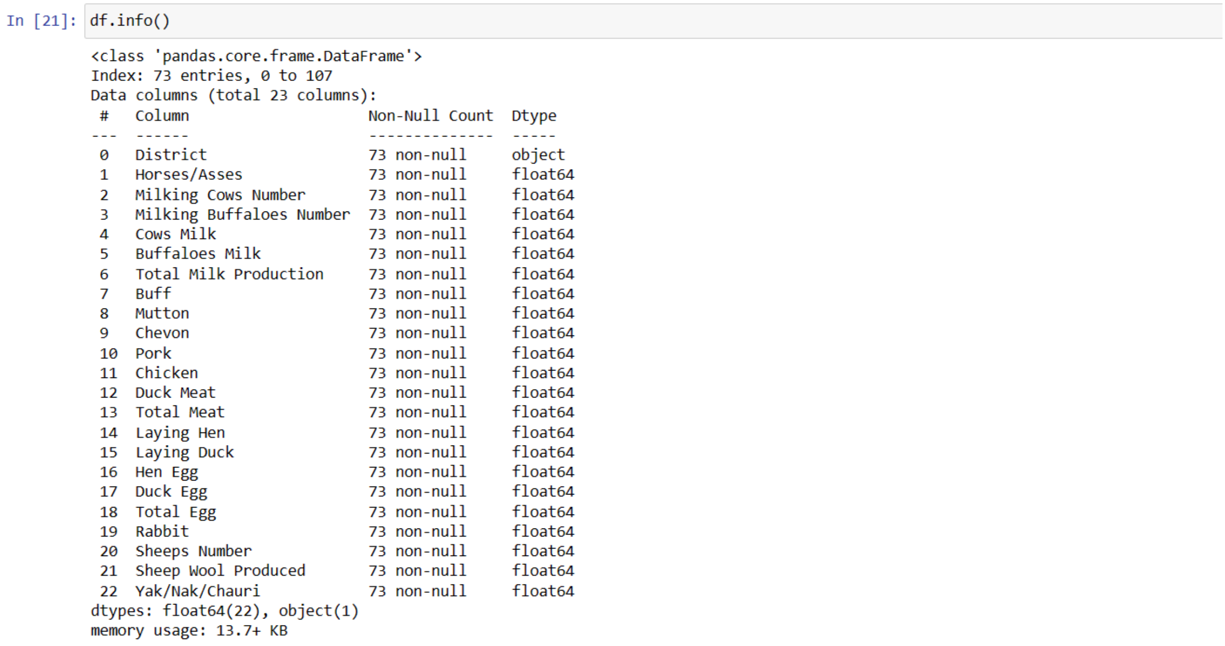
Out[18]:
```
0      False
2      False
3      False
4      False
6      False
7      False
8      False
9      False
10     False
12     False
14     False
15     False
16     False
17     False
19     False
20     False
21     False
22     False
23     False
24     False
25     False
26     False
27     False
28     False
29     False
30     False
31     False
32     False
33     False
35     False
36     False
37     False
```



```
75     False
76     False
77     False
78     False
79     False
81     False
82     False
83     False
84     False
88     False
98     False
99     False
103    False
104    False
105    False
106    False
107    False
dtype: bool
```

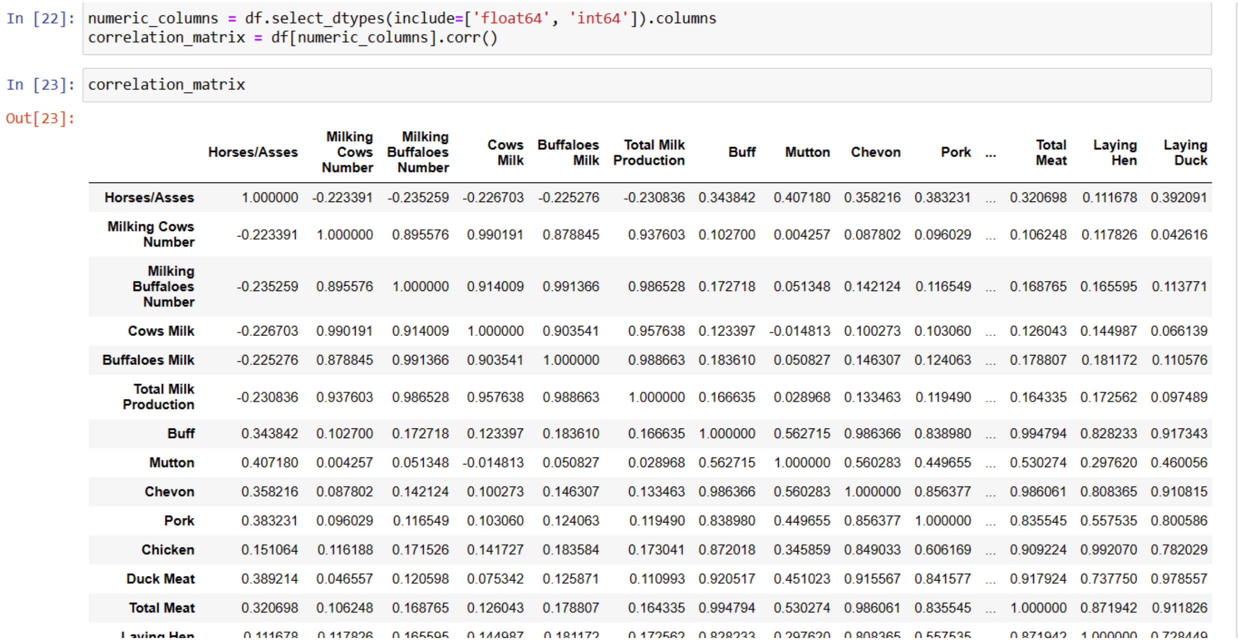
```
In [19]: df.shape

Out[19]: (73, 23)

In [20]: df = df.drop_duplicates()
```

Checking the datatypes



```
In [21]: df.info()

<class 'pandas.core.frame.DataFrame'>
Index: 73 entries, 0 to 107
Data columns (total 23 columns):
 #   Column                    Non-Null Count  Dtype  
---  ------                    --------------  -----  
 0   District                  73 non-null     object 
 1   Horses/Asses              73 non-null     float64
 2   Milking Cows Number       73 non-null     float64
 3   Milking Buffaloes Number  73 non-null     float64
 4   Cows Milk                 73 non-null     float64
 5   Buffaloes Milk            73 non-null     float64
 6   Total Milk Production     73 non-null     float64
 7   Buff                      73 non-null     float64
 8   Mutton                    73 non-null     float64
 9   Chevon                    73 non-null     float64
 10  Pork                      73 non-null     float64
 11  Chicken                   73 non-null     float64
 12  Duck Meat                 73 non-null     float64
 13  Total Meat                73 non-null     float64
 14  Laying Hen                73 non-null     float64
 15  Laying Duck               73 non-null     float64
 16  Hen Egg                   73 non-null     float64
 17  Duck Egg                  73 non-null     float64
 18  Total Egg                 73 non-null     float64
 19  Rabbit                    73 non-null     float64
 20  Sheeps Number             73 non-null     float64
 21  Sheep Wool Produced       73 non-null     float64
 22  Yak/Nak/Chauri            73 non-null     float64
dtypes: float64(22), object(1)
memory usage: 13.7+ KB
```

Section-3



In [22]:
```
numeric_columns = df.select_dtypes(include=['float64', 'int64']).columns
correlation_matrix = df[numeric_columns].corr()
```

In [23]:
```
correlation_matrix
```

Out[23]:

	Horses/Asses	Milking Cows Number	Milking Buffaloes Number	Cows Milk	Buffaloes Milk	Total Milk Production	Buff	Mutton	Chevon	Pork	...	Total Meat	Laying Hen	Laying Duck
Horses/Asses	1.000000	-0.223391	-0.235259	-0.226703	-0.225276	-0.230836	0.343842	0.407180	0.358216	0.383231	...	0.320698	0.111678	0.392091
Milking Cows Number	-0.223391	1.000000	0.895576	0.990191	0.878845	0.937603	0.102700	0.004257	0.087802	0.096029	...	0.106248	0.117826	0.042616
Milking Buffaloes Number	-0.235259	0.895576	1.000000	0.914009	0.991366	0.986528	0.172718	0.051348	0.142124	0.116549	...	0.168765	0.165595	0.113771
Cows Milk	-0.226703	0.990191	0.914009	1.000000	0.903541	0.957638	0.123397	-0.014813	0.100273	0.103060	...	0.126043	0.144987	0.066139
Buffaloes Milk	-0.225276	0.878845	0.991366	0.903541	1.000000	0.988663	0.183610	0.050827	0.146307	0.124063	...	0.178807	0.181172	0.110576
Total Milk Production	-0.230836	0.937603	0.986528	0.957638	0.988663	1.000000	0.166635	0.028968	0.133463	0.119490	...	0.164335	0.172562	0.097489
Buff	0.343842	0.102700	0.172718	0.123397	0.183610	0.166635	1.000000	0.562715	0.986366	0.838980	...	0.994794	0.828233	0.917343
Mutton	0.407180	0.004257	0.051348	-0.014813	0.050827	0.028968	0.562715	1.000000	0.560283	0.449655	...	0.530274	0.297620	0.460056
Chevon	0.358216	0.087802	0.142124	0.100273	0.146307	0.133463	0.986366	0.560283	1.000000	0.856377	...	0.986061	0.808365	0.910815
Pork	0.383231	0.096029	0.116549	0.103060	0.124063	0.119490	0.838980	0.449655	0.856377	1.000000	...	0.835545	0.557535	0.800586
Chicken	0.151064	0.116188	0.171526	0.141727	0.183584	0.173041	0.872018	0.345859	0.849033	0.606169	...	0.909224	0.992070	0.782029
Duck Meat	0.389214	0.046557	0.120598	0.075342	0.125871	0.110993	0.920517	0.451023	0.915567	0.841577	...	0.917924	0.737750	0.978557
Total Meat	0.320698	0.106248	0.168765	0.126043	0.178807	0.164335	0.994794	0.530274	0.986061	0.835545	...	1.000000	0.871942	0.911826
Laying Hen	0.111678	0.117826	0.165595	0.144987	0.181172	0.172562	0.828233	0.297620	0.808365	0.557535	...	0.871942	1.000000	0.728449

Existence of correlations

Positive correlation: coefficient close to 1| Negative correlation: coefficient close to -1

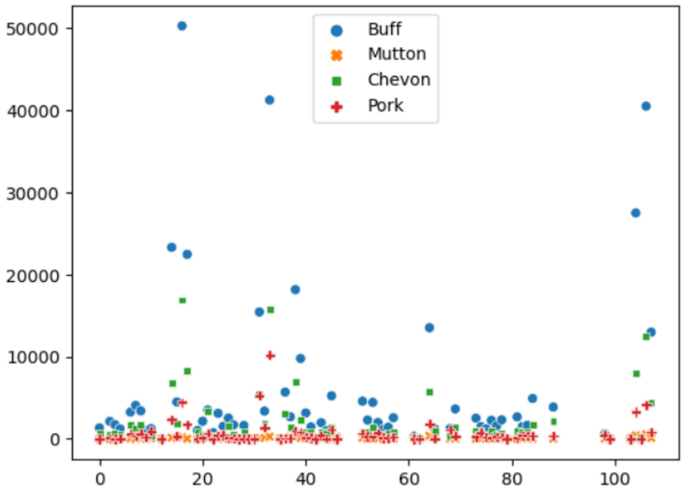
How strong are they?

- 0.00 - 0.19: Very weak , 0.20 - 0.39: Weak, 0.40 - 0.59: Moderate, 0.60 - 0.79: Strong, 0.80 - 1.00: Very strong

five Data Visualization techniques # use column accordingly for visualizations

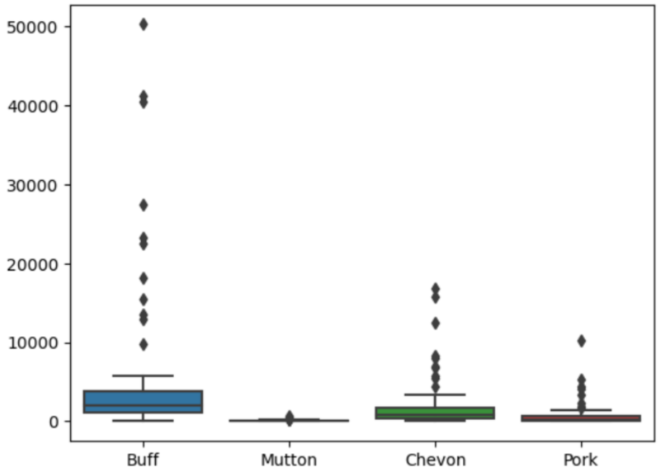
In [24]:
```
sns.scatterplot(df[['Buff', 'Mutton', 'Chevon', 'Pork' ]]) # For single column

# sns.scatterplot(data=df) # for entire dataset
```

Out[24]:
```
<Axes: >
```



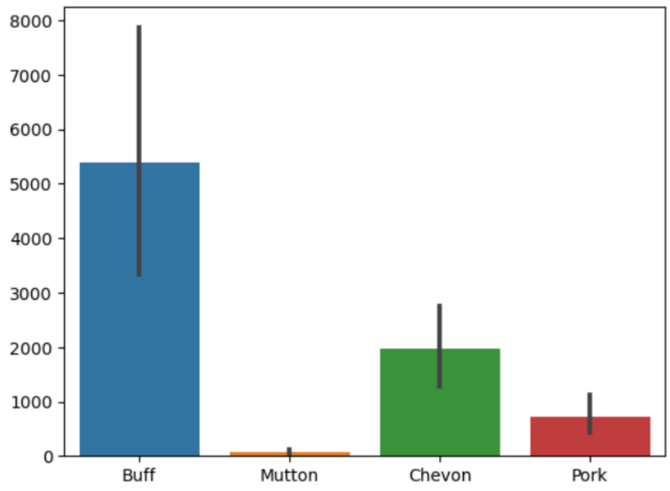
In [25]:
```
sns.boxplot(df[['Buff', 'Mutton', 'Chevon', 'Pork' ]])
```

Out[25]:
```
<Axes: >
```



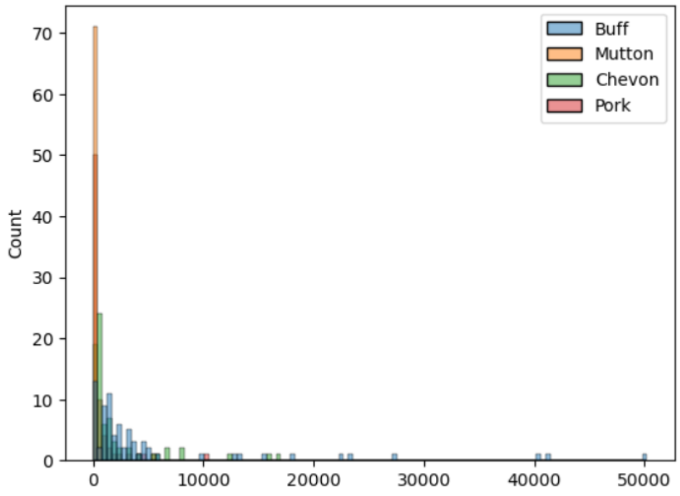
```
In [26]: sns.barplot(df[['Buff', 'Mutton', 'Chevon', 'Pork' ]])
```

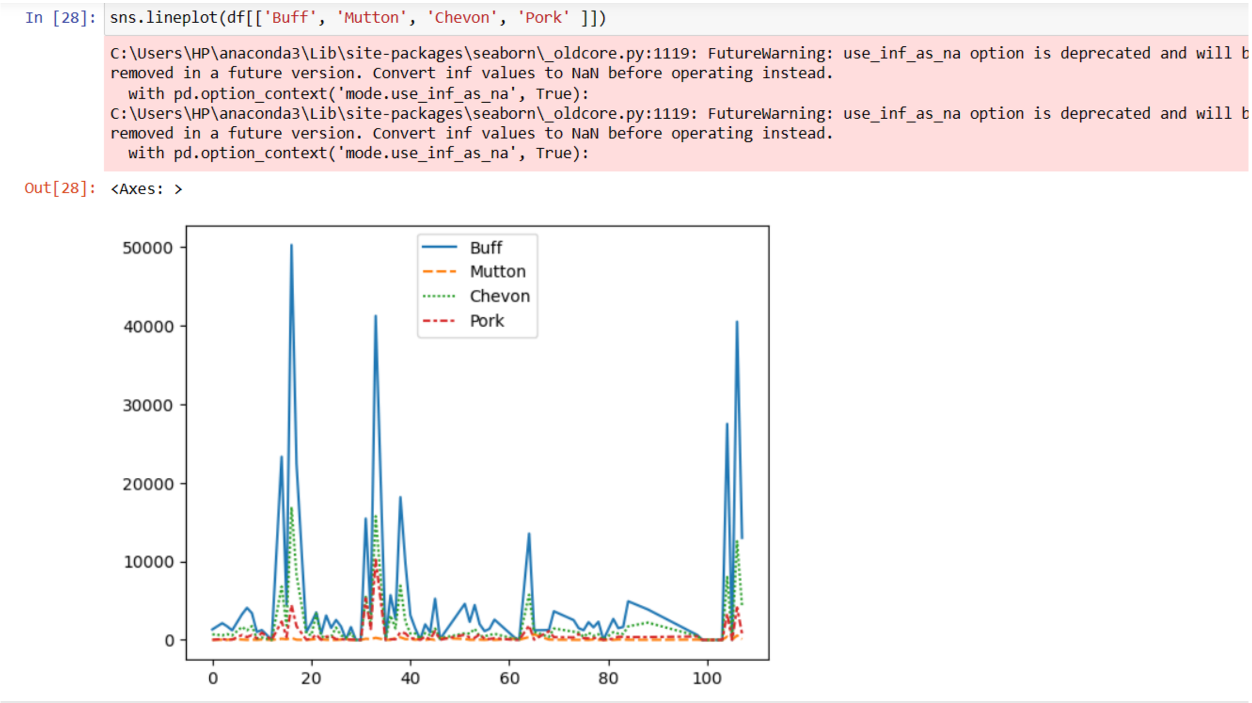
```
Out[26]: <Axes: >
```



```
In [27]: sns.histplot(df[['Buff', 'Mutton', 'Chevon', 'Pork' ]])
```

```
Out[27]: <Axes: ylabel='Count'>
```

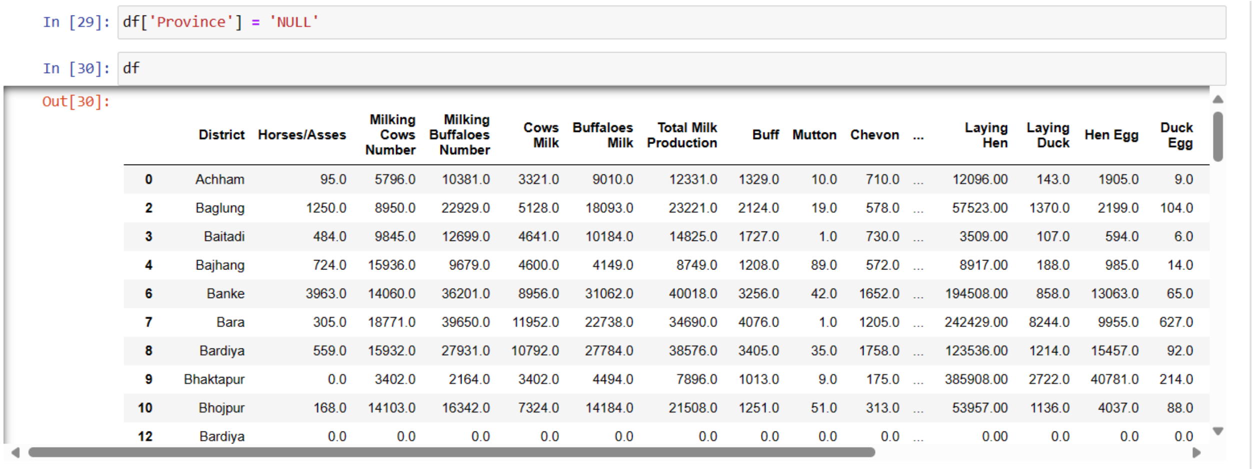




```
In [28]: sns.lineplot(df[['Buff', 'Mutton', 'Chevon', 'Pork' ]])

C:\Users\HP\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1119: FutureWarning: use_inf_as_na option is deprecated and will b
removed in a future version. Convert inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
C:\Users\HP\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1119: FutureWarning: use_inf_as_na option is deprecated and will b
removed in a future version. Convert inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):

Out[28]: <Axes: >
```

Section-4

Adding Province



```
In [29]: df['Province'] = 'NULL'

In [30]: df

Out[30]:
```

	District	Horses/Asses	Milking Cows Number	Milking Buffaloes Number	Cows Milk	Buffaloes Milk	Total Milk Production	Buff	Mutton	Chevon	...	Laying Hen	Laying Duck	Hen Egg	Duck Egg
0	Achham	95.0	5796.0	10381.0	3321.0	9010.0	12331.0	1329.0	10.0	710.0	...	12096.00	143.0	1905.0	9.0
2	Baglung	1250.0	8950.0	22929.0	5128.0	18093.0	23221.0	2124.0	19.0	578.0	...	57523.00	1370.0	2199.0	104.0
3	Baitadi	484.0	9845.0	12699.0	4641.0	10184.0	14825.0	1727.0	1.0	730.0	...	3509.00	107.0	594.0	6.0
4	Bajhang	724.0	15936.0	9679.0	4600.0	4149.0	8749.0	1208.0	89.0	572.0	...	8917.00	188.0	985.0	14.0
6	Banke	3963.0	14060.0	36201.0	8956.0	31062.0	40018.0	3256.0	42.0	1652.0	...	194508.00	858.0	13063.0	65.0
7	Bara	305.0	18771.0	39650.0	11952.0	22738.0	34690.0	4076.0	1.0	1205.0	...	242429.00	8244.0	9955.0	627.0
8	Bardiya	559.0	15932.0	27931.0	10792.0	27784.0	38576.0	3405.0	35.0	1758.0	...	123536.00	1214.0	15457.0	92.0
9	Bhaktapur	0.0	3402.0	2164.0	3402.0	4494.0	7896.0	1013.0	9.0	175.0	...	385908.00	2722.0	40781.0	214.0
10	Bhojpur	168.0	14103.0	16342.0	7324.0	14184.0	21508.0	1251.0	51.0	313.0	...	53957.00	1136.0	4037.0	88.0
12	Bardiya	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.00	0.0	0.0	0.0

Adding province according to district

In [31]:

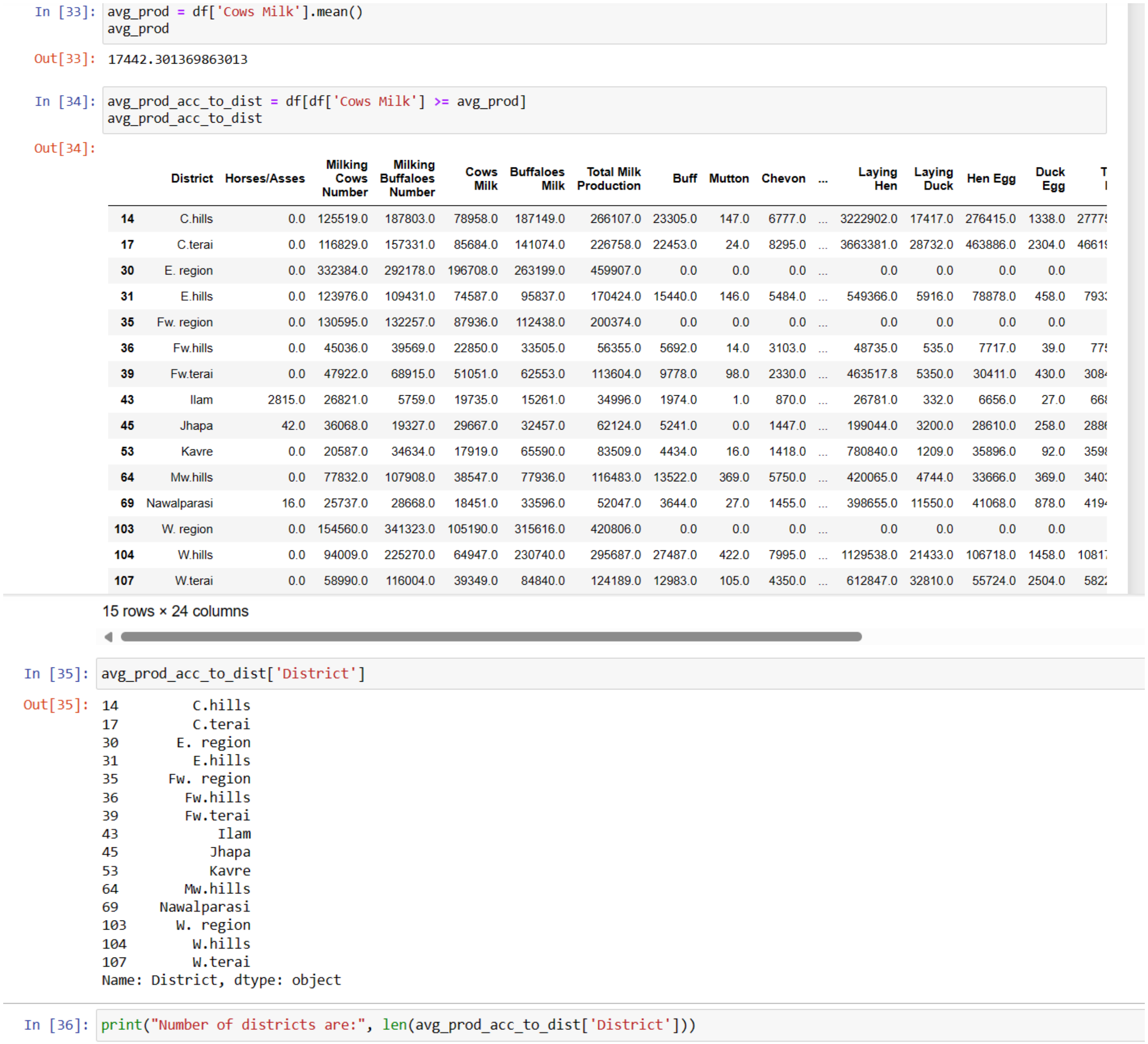
```
 df= df.assign(Province = [
"Province 1",
"Province 1",
"Province 1",
"Province 1",
"Province 1",
"Province 1",
"Province 1",
"Province 1",
"Province 1",
"Province 1",
"Province 3",
"Province 3",
"Province 3",
"Province 3",
"Province 4",
"Province 4",
"Province 4",
"Province 4",
"Province 4",
"Province 4",
"Province 4",
"Province 4",
"Province 4",
"Province 4",
"Province 4",
"Province 4",
"Province 5",
"Province 5",
"Province 5",
"Province 6",
"Province 6",
"Province 6",
```

In [32]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 73 entries, 0 to 107
Data columns (total 24 columns):
 #   Column                    Non-Null Count  Dtype  
---  ------                    --------------  -----  
 0   District                  73 non-null     object 
 1   Horses/Asses              73 non-null     float64
 2   Milking Cows Number       73 non-null     float64
 3   Milking Buffaloes Number  73 non-null     float64
 4   Cows Milk                 73 non-null     float64
 5   Buffaloes Milk            73 non-null     float64
 6   Total Milk Production     73 non-null     float64
 7   Buff                      73 non-null     float64
 8   Mutton                    73 non-null     float64
 9   Chevon                    73 non-null     float64
 10  Pork                      73 non-null     float64
 11  Chicken                   73 non-null     float64
 12  Duck Meat                 73 non-null     float64
 13  Total Meat                73 non-null     float64
 14  Laying Hen                73 non-null     float64
 15  Laying Duck               73 non-null     float64
 16  Hen Egg                   73 non-null     float64
 17  Duck Egg                  73 non-null     float64
 18  Total Egg                 73 non-null     float64
 19  Rabbit                    73 non-null     float64
 20  Sheeps Number             73 non-null     float64
 21  Sheep Wool Produced       73 non-null     float64
 22  Yak/Nak/Chauri            73 non-null     float64
 23  Province                  73 non-null     object 
dtypes: float64(22), object(2)
memory usage: 14.3+ KB
```

Counting the districts where the cow’s milk production is more/equal to the average production.



```
In [33]: avg_prod = df['Cows Milk'].mean()
         avg_prod

Out[33]: 17442.301369863013

In [34]: avg_prod_acc_to_dist = df[df['Cows Milk'] >= avg_prod]
         avg_prod_acc_to_dist

Out[34]:
```

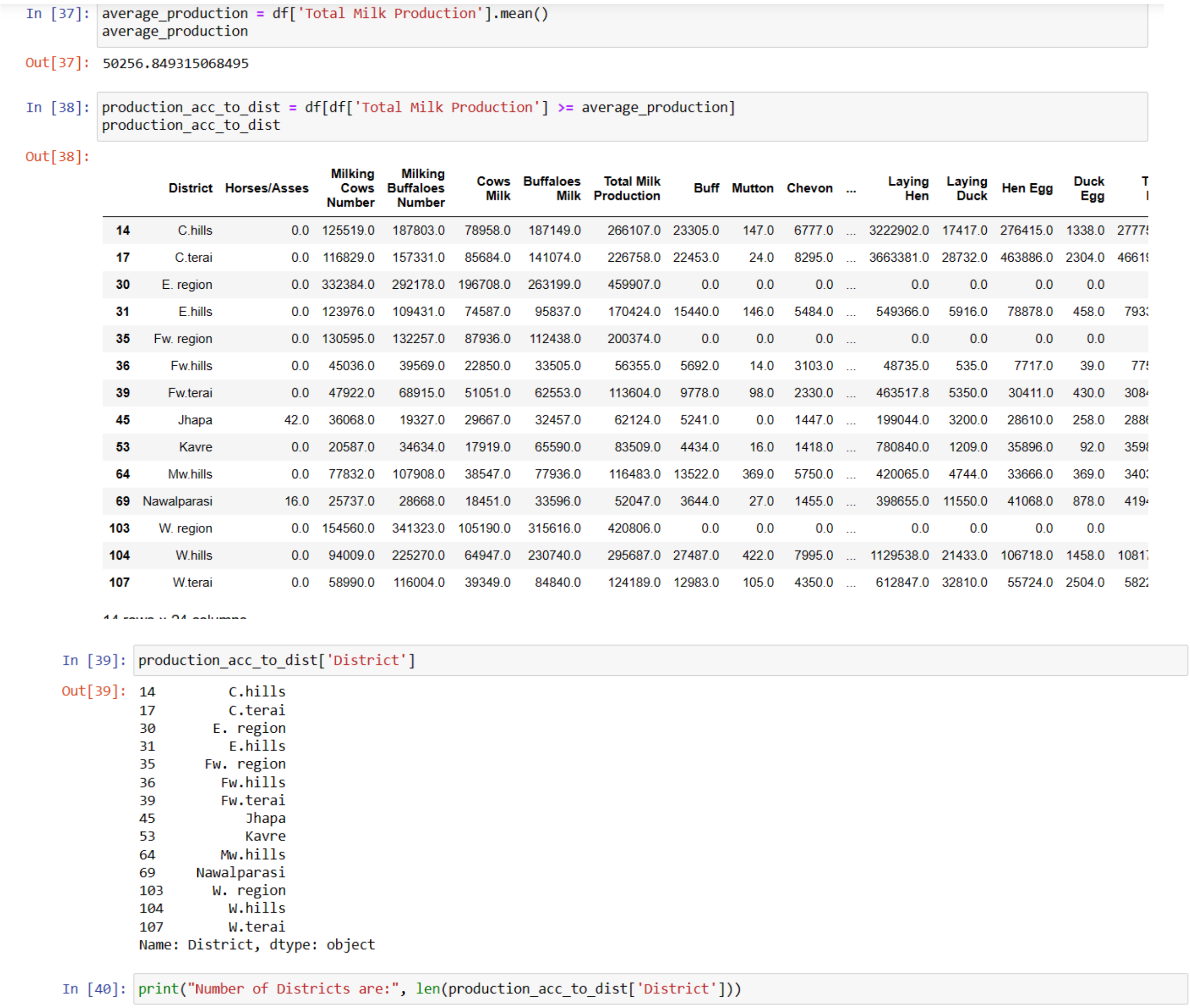
	District	Horses/Asses	Milking Cows Number	Milking Buffaloes Number	Cows Milk	Buffaloes Milk	Total Milk Production	Buff	Mutton	Chevon	...	Laying Hen	Laying Duck	Hen Egg	Duck Egg	T I
14	C.hills	0.0	125519.0	187803.0	78958.0	187149.0	266107.0	23305.0	147.0	6777.0	...	3222902.0	17417.0	276415.0	1338.0	2777
17	C.terai	0.0	116829.0	157331.0	85684.0	141074.0	226758.0	22453.0	24.0	8295.0	...	3663381.0	28732.0	463886.0	2304.0	4661
30	E. region	0.0	332384.0	292178.0	196708.0	263199.0	459907.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	
31	E.hills	0.0	123976.0	109431.0	74587.0	95837.0	170424.0	15440.0	146.0	5484.0	...	549366.0	5916.0	78878.0	458.0	793
35	Fw. region	0.0	130595.0	132257.0	87936.0	112438.0	200374.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	
36	Fw.hills	0.0	45036.0	39569.0	22850.0	33505.0	56355.0	5692.0	14.0	3103.0	...	48735.0	535.0	7717.0	39.0	77
39	Fw.terai	0.0	47922.0	68915.0	51051.0	62553.0	113604.0	9778.0	98.0	2330.0	...	463517.8	5350.0	30411.0	430.0	308
43	Ilam	2815.0	26821.0	5759.0	19735.0	15261.0	34996.0	1974.0	1.0	870.0	...	26781.0	332.0	6656.0	27.0	66
45	Jhapa	42.0	36068.0	19327.0	29667.0	32457.0	62124.0	5241.0	0.0	1447.0	...	199044.0	3200.0	28610.0	258.0	288
53	Kavre	0.0	20587.0	34634.0	17919.0	65590.0	83509.0	4434.0	16.0	1418.0	...	780840.0	1209.0	35896.0	92.0	359
64	Mw.hills	0.0	77832.0	107908.0	38547.0	77936.0	116483.0	13522.0	369.0	5750.0	...	420065.0	4744.0	33666.0	369.0	340
69	Nawalparasi	16.0	25737.0	28668.0	18451.0	33596.0	52047.0	3644.0	27.0	1455.0	...	398655.0	11550.0	41068.0	878.0	419
103	W. region	0.0	154560.0	341323.0	105190.0	315616.0	420806.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	
104	W.hills	0.0	94009.0	225270.0	64947.0	230740.0	295687.0	27487.0	422.0	7995.0	...	1129538.0	21433.0	106718.0	1458.0	1081
107	W.terai	0.0	58990.0	116004.0	39349.0	84840.0	124189.0	12983.0	105.0	4350.0	...	612847.0	32810.0	55724.0	2504.0	582

15 rows × 24 columns

```
In [35]: avg_prod_acc_to_dist['District']

Out[35]: 14          C.hills
         17          C.terai
         30        E. region
         31          E.hills
         35       Fw. region
         36         Fw.hills
         39         Fw.terai
         43             Ilam
         45            Jhapa
         53            Kavre
         64         Mw.hills
         69      Nawalparasi
         103       W. region
         104         W.hills
         107         W.terai
         Name: District, dtype: object

In [36]: print("Number of districts are:", len(avg_prod_acc_to_dist['District']))
```

The number of districts is 15.

Milk production from cows and buffaloes



```
In [37]: average_production = df['Total Milk Production'].mean()
         average_production

Out[37]: 50256.849315068495
```

```
In [38]: production_acc_to_dist = df[df['Total Milk Production'] >= average_production]
         production_acc_to_dist

Out[38]:
```

	District	Horses/Asses	Milking Cows Number	Milking Buffaloes Number	Cows Milk	Buffaloes Milk	Total Milk Production	Buff	Mutton	Chevon	...	Laying Hen	Laying Duck	Hen Egg	Duck Egg	T I
14	C.hills	0.0	125519.0	187803.0	78958.0	187149.0	266107.0	23305.0	147.0	6777.0	...	3222902.0	17417.0	276415.0	1338.0	2777
17	C.terai	0.0	116829.0	157331.0	85684.0	141074.0	226758.0	22453.0	24.0	8295.0	...	3663381.0	28732.0	463886.0	2304.0	4661
30	E. region	0.0	332384.0	292178.0	196708.0	263199.0	459907.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	
31	E.hills	0.0	123976.0	109431.0	74587.0	95837.0	170424.0	15440.0	146.0	5484.0	...	549366.0	5916.0	78878.0	458.0	793
35	Fw. region	0.0	130595.0	132257.0	87936.0	112438.0	200374.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	
36	Fw.hills	0.0	45036.0	39569.0	22850.0	33505.0	56355.0	5692.0	14.0	3103.0	...	48735.0	535.0	7717.0	39.0	77
39	Fw.terai	0.0	47922.0	68915.0	51051.0	62553.0	113604.0	9778.0	98.0	2330.0	...	463517.8	5350.0	30411.0	430.0	308
45	Jhapa	42.0	36068.0	19327.0	29667.0	32457.0	62124.0	5241.0	0.0	1447.0	...	199044.0	3200.0	28610.0	258.0	288
53	Kavre	0.0	20587.0	34634.0	17919.0	65590.0	83509.0	4434.0	16.0	1418.0	...	780840.0	1209.0	35896.0	92.0	359
64	Mw.hills	0.0	77832.0	107908.0	38547.0	77936.0	116483.0	13522.0	369.0	5750.0	...	420065.0	4744.0	33666.0	369.0	340
69	Nawalparasi	16.0	25737.0	28668.0	18451.0	33596.0	52047.0	3644.0	27.0	1455.0	...	398655.0	11550.0	41068.0	878.0	419
103	W. region	0.0	154560.0	341323.0	105190.0	315616.0	420806.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	
104	W.hills	0.0	94009.0	225270.0	64947.0	230740.0	295687.0	27487.0	422.0	7995.0	...	1129538.0	21433.0	106718.0	1458.0	1081
107	W.terai	0.0	58990.0	116004.0	39349.0	84840.0	124189.0	12983.0	105.0	4350.0	...	612847.0	32810.0	55724.0	2504.0	582

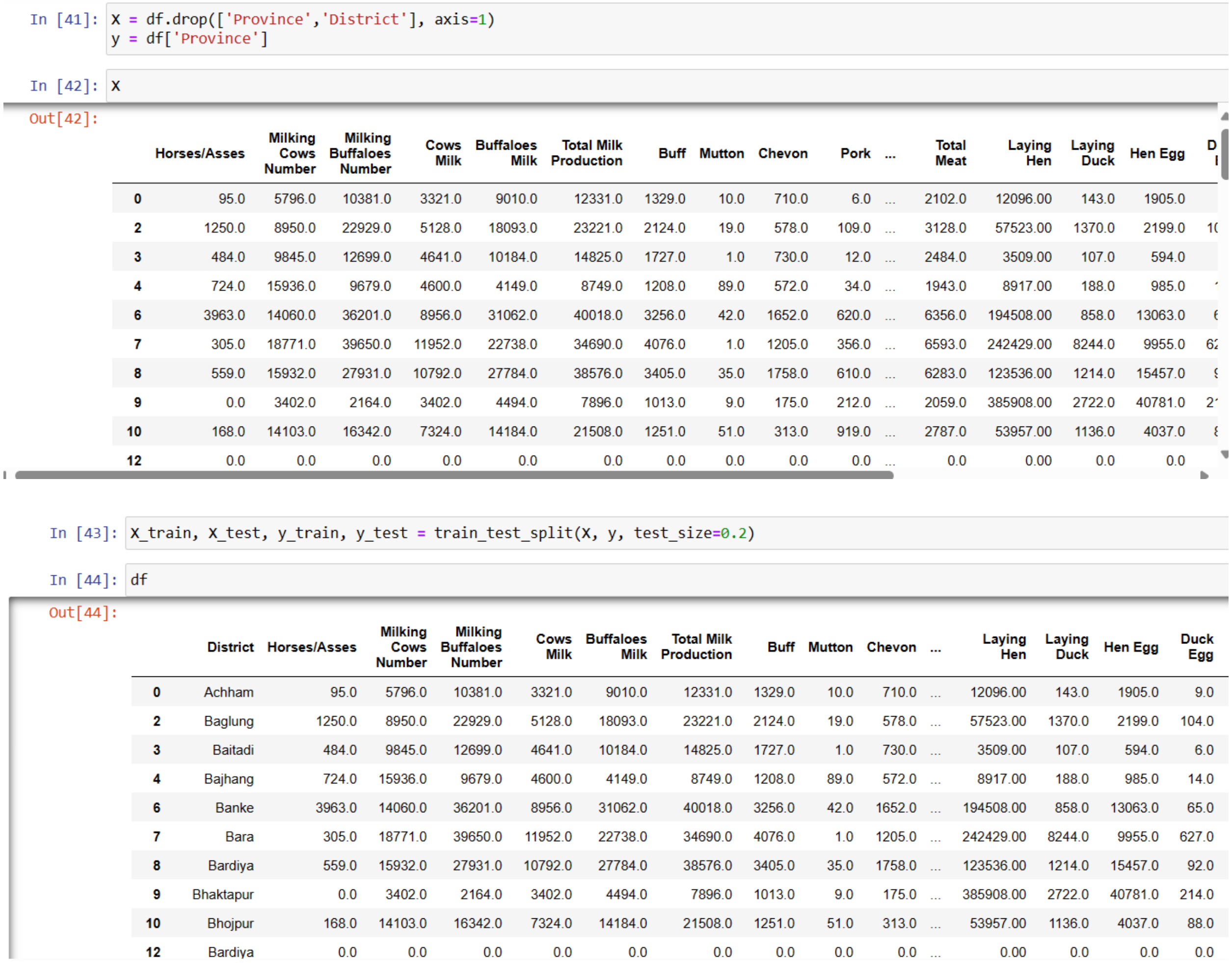
14 rows × 24 columns

```
In [39]: production_acc_to_dist['District']

Out[39]: 14           C.hills
         17           C.terai
         30         E. region
         31           E.hills
         35        Fw. region
         36          Fw.hills
         39          Fw.terai
         45             Jhapa
         53             Kavre
         64          Mw.hills
         69       Nawalparasi
         103        W. region
         104          W.hills
         107          W.terai
         Name: District, dtype: object
```

```
In [40]: print("Number of Districts are:", len(production_acc_to_dist['District']))
```

The number of Districts is 14.

80% training and 20% testing data



```
In [41]: X = df.drop(['Province','District'], axis=1)
         y = df['Province']
```

```
In [42]: X
```

Out[42]:

	Horses/Asses	Milking Cows Number	Milking Buffaloes Number	Cows Milk	Buffaloes Milk	Total Milk Production	Buff	Mutton	Chevon	Pork	...	Total Meat	Laying Hen	Laying Duck	Hen Egg	D
0	95.0	5796.0	10381.0	3321.0	9010.0	12331.0	1329.0	10.0	710.0	6.0	...	2102.0	12096.00	143.0	1905.0	
2	1250.0	8950.0	22929.0	5128.0	18093.0	23221.0	2124.0	19.0	578.0	109.0	...	3128.0	57523.00	1370.0	2199.0	10
3	484.0	9845.0	12699.0	4641.0	10184.0	14825.0	1727.0	1.0	730.0	12.0	...	2484.0	3509.00	107.0	594.0	
4	724.0	15936.0	9679.0	4600.0	4149.0	8749.0	1208.0	89.0	572.0	34.0	...	1943.0	8917.00	188.0	985.0	1
6	3963.0	14060.0	36201.0	8956.0	31062.0	40018.0	3256.0	42.0	1652.0	620.0	...	6356.0	194508.00	858.0	13063.0	6
7	305.0	18771.0	39650.0	11952.0	22738.0	34690.0	4076.0	1.0	1205.0	356.0	...	6593.0	242429.00	8244.0	9955.0	62
8	559.0	15932.0	27931.0	10792.0	27784.0	38576.0	3405.0	35.0	1758.0	610.0	...	6283.0	123536.00	1214.0	15457.0	9
9	0.0	3402.0	2164.0	3402.0	4494.0	7896.0	1013.0	9.0	175.0	212.0	...	2059.0	385908.00	2722.0	40781.0	21
10	168.0	14103.0	16342.0	7324.0	14184.0	21508.0	1251.0	51.0	313.0	919.0	...	2787.0	53957.00	1136.0	4037.0	8
12	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.00	0.0	0.0	

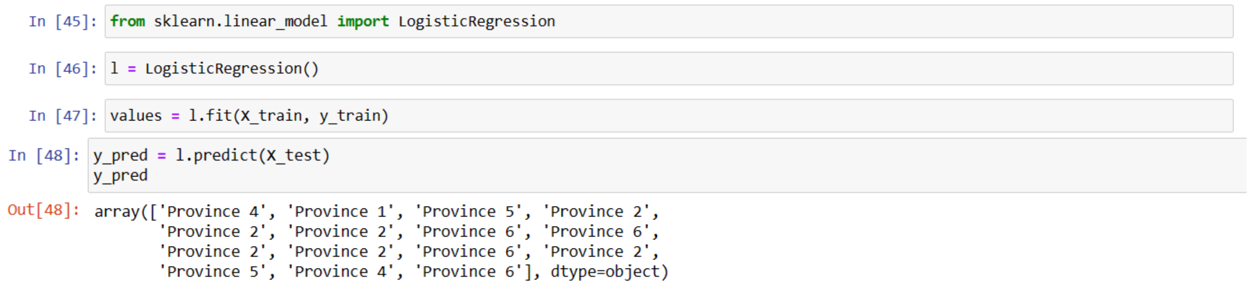
```
In [43]: X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2)
```

```
In [44]: df
```

Out[44]:

	District	Horses/Asses	Milking Cows Number	Milking Buffaloes Number	Cows Milk	Buffaloes Milk	Total Milk Production	Buff	Mutton	Chevon	...	Laying Hen	Laying Duck	Hen Egg	Duck Egg
0	Achham	95.0	5796.0	10381.0	3321.0	9010.0	12331.0	1329.0	10.0	710.0	...	12096.00	143.0	1905.0	9.0
2	Baglung	1250.0	8950.0	22929.0	5128.0	18093.0	23221.0	2124.0	19.0	578.0	...	57523.00	1370.0	2199.0	104.0
3	Baitadi	484.0	9845.0	12699.0	4641.0	10184.0	14825.0	1727.0	1.0	730.0	...	3509.00	107.0	594.0	6.0
4	Bajhang	724.0	15936.0	9679.0	4600.0	4149.0	8749.0	1208.0	89.0	572.0	...	8917.00	188.0	985.0	14.0
6	Banke	3963.0	14060.0	36201.0	8956.0	31062.0	40018.0	3256.0	42.0	1652.0	...	194508.00	858.0	13063.0	65.0
7	Bara	305.0	18771.0	39650.0	11952.0	22738.0	34690.0	4076.0	1.0	1205.0	...	242429.00	8244.0	9955.0	627.0
8	Bardiya	559.0	15932.0	27931.0	10792.0	27784.0	38576.0	3405.0	35.0	1758.0	...	123536.00	1214.0	15457.0	92.0
9	Bhaktapur	0.0	3402.0	2164.0	3402.0	4494.0	7896.0	1013.0	9.0	175.0	...	385908.00	2722.0	40781.0	214.0
10	Bhojpur	168.0	14103.0	16342.0	7324.0	14184.0	21508.0	1251.0	51.0	313.0	...	53957.00	1136.0	4037.0	88.0
12	Bardiya	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.00	0.0	0.0	0.0

Creating a model using a classification algorithm.



```
In [45]: from sklearn.linear_model import LogisticRegression
```

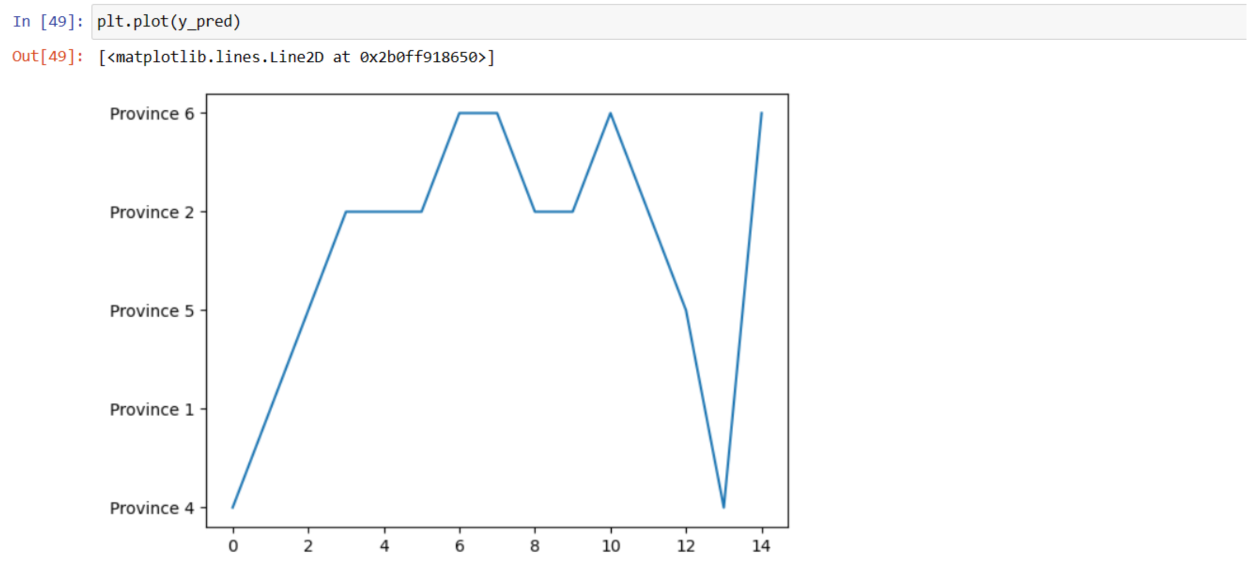
```
In [46]: l = LogisticRegression()
```

```
In [47]: values = l.fit(X_train, y_train)
```

```
In [48]: y_pred = l.predict(X_test)
         y_pred
```

```
Out[48]: array(['Province 4', 'Province 1', 'Province 5', 'Province 2',
                'Province 2', 'Province 2', 'Province 6', 'Province 6',
                'Province 2', 'Province 2', 'Province 6', 'Province 2',
                'Province 5', 'Province 4', 'Province 6'], dtype=object)
```

Visualization output



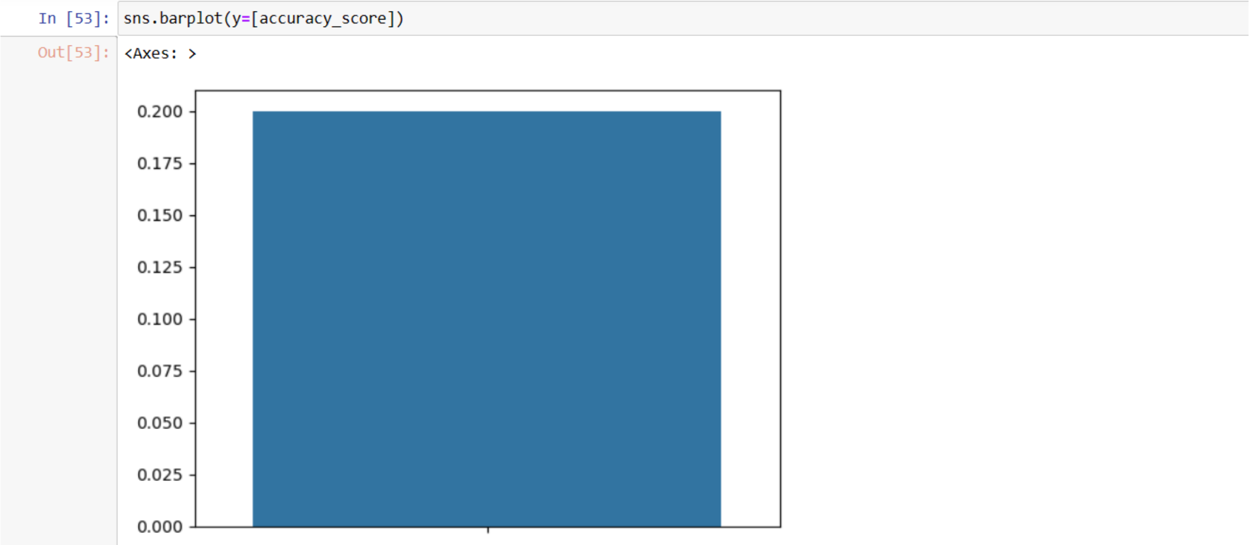
```
In [49]: plt.plot(y_pred)

Out[49]: [<matplotlib.lines.Line2D at 0x2b0ff918650>]
```

Section-5

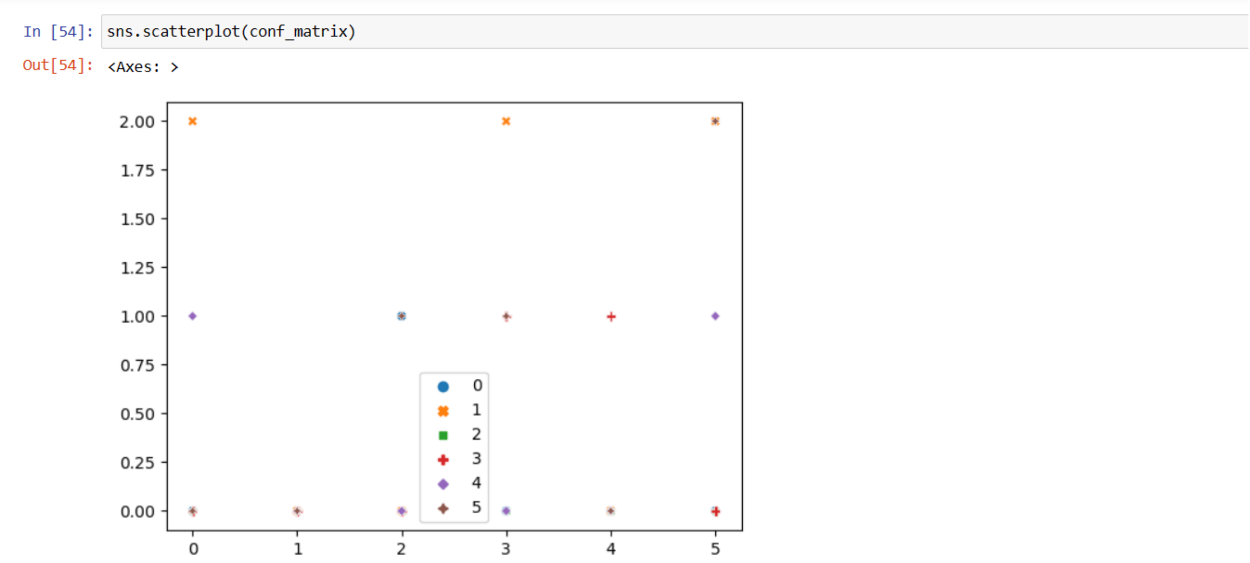
```
In [50]: from sklearn.metrics import accuracy_score, confusion_matrix
```

```
In [51]: accuracy_score = accuracy_score(y_test, y_pred)
         accuracy_score

Out[51]: 0.2
```

```
In [52]: conf_matrix = confusion_matrix(y_test, y_pred)
         conf_matrix

Out[52]: array([[0, 2, 0, 0, 1, 0],
                [0, 0, 0, 0, 0, 0],
                [1, 0, 0, 0, 0, 1],
                [0, 2, 0, 1, 0, 1],
                [0, 0, 0, 1, 0, 0],
                [0, 2, 0, 0, 1, 2]], dtype=int64)
```



```
In [53]: sns.barplot(y=[accuracy_score])

Out[53]: <Axes: >
```



```
In [54]: sns.scatterplot(conf_matrix)

Out[54]: <Axes: >
```

Analysis & Recommendations:

Social Impacts:

- **Identify Potential Social Impacts:** The research outcomes can have significant implications for agricultural practices, resource allocation, and policy formulation in Nepal. By providing insights into livestock and commodities production trends, the research can inform strategies to enhance food security, rural livelihoods, and economic development.
- **Positive and Negative Implications:** Positively, the findings can contribute to optimizing agricultural productivity, reducing poverty, and promoting sustainable development in rural communities. However, there may be negative implications if the research findings are not effectively translated into actionable policies or if certain communities disproportionately benefit from interventions.
- **Stakeholder Involvement:** Relevant stakeholders, including government agencies, agricultural organizations, farmers' associations, and research institutions, should be engaged throughout the research process. Their involvement ensures that the research outcomes are relevant, accessible, and effectively utilized to benefit the target communities.

Ethical Issues:

- **Informed Consent:** Since the research involves secondary data analysis, informed consent is not applicable. However, ethical considerations include obtaining permission for data usage from the original data providers and ensuring proper attribution.
- **Privacy and Confidentiality:** Data privacy and confidentiality are paramount. Measures such as anonymization of data and secure storage must be implemented to protect participants' privacy.

- **Harm and Risk Mitigation:** The research poses minimal risks to participants as it primarily involves analyzing aggregated data. However, measures should be in place to mitigate potential harm, such as ensuring data security and minimizing the risk of re-identification.
- **Compliance with Regulations:** The research must comply with ethical guidelines and regulations governing data usage and research conduct. This includes obtaining necessary approvals from institutional review boards (IRBs) or data ethics committees, if applicable, to ensure the ethical conduct of the study.

Recommendations:

Foster collaboration with local stakeholders, including government agencies, NGOs, and community groups, to ensure the relevance and applicability of research findings.

Disseminate research findings through accessible channels, such as policy briefs, workshops, and community meetings, to facilitate evidence-based decision-making and community empowerment.

Prioritize transparency and accountability in data management and research processes to build trust among stakeholders and uphold ethical standards.

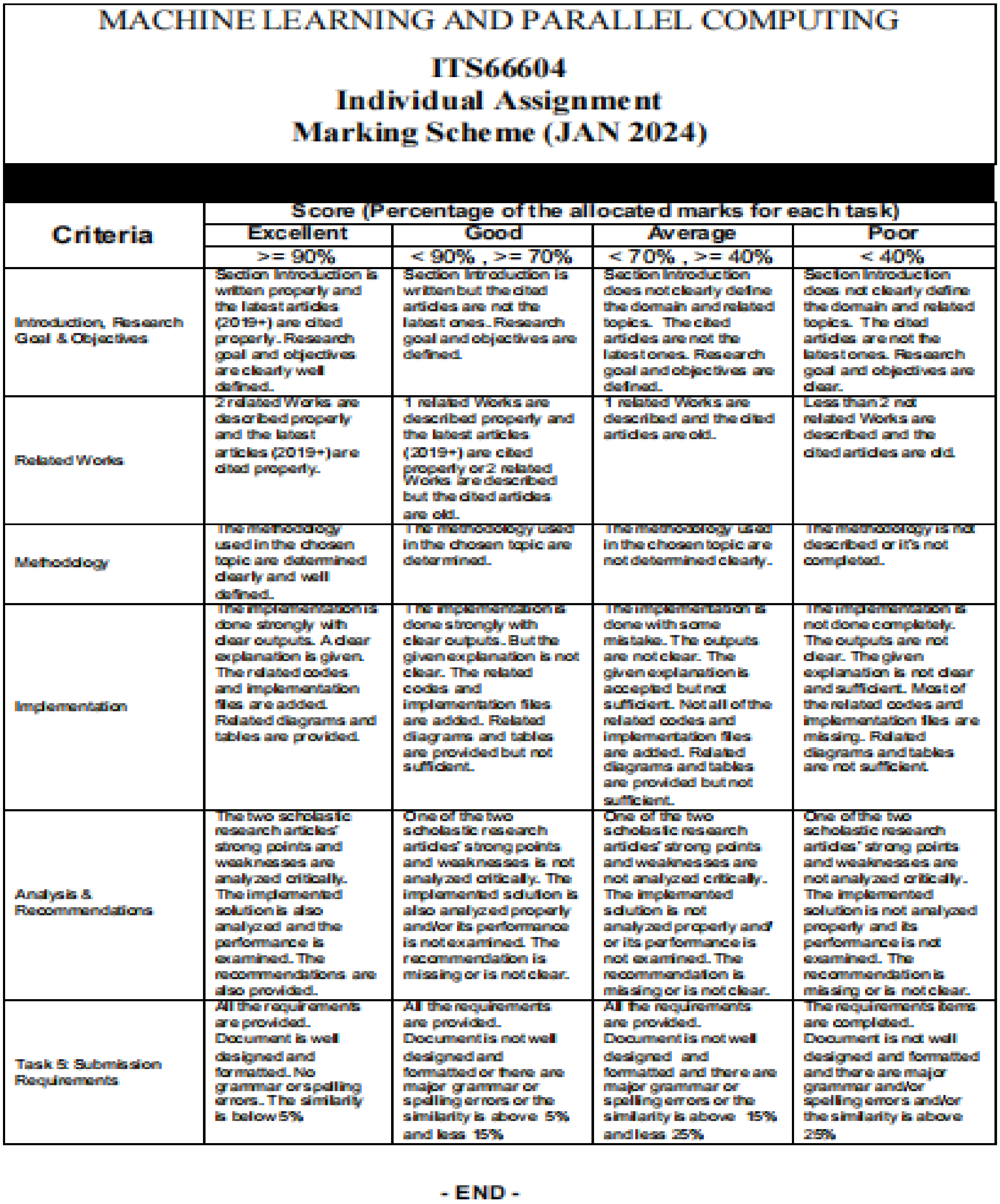
Continuously monitor and evaluate the social impacts of the research to identify opportunities for improvement and address any unintended consequences.

Advocate for policies that promote equitable access to resources and opportunities, based on the insights generated from the research, to address social inequalities and promote inclusive development in Nepal's agricultural sector.

References:

1.https://www.researchgate.net/publication/337003597_Situation_of_Livestock_Production_and_its_Products_in_Nepal

2. https://www.mdpi.com/2077-0472/12/5/679



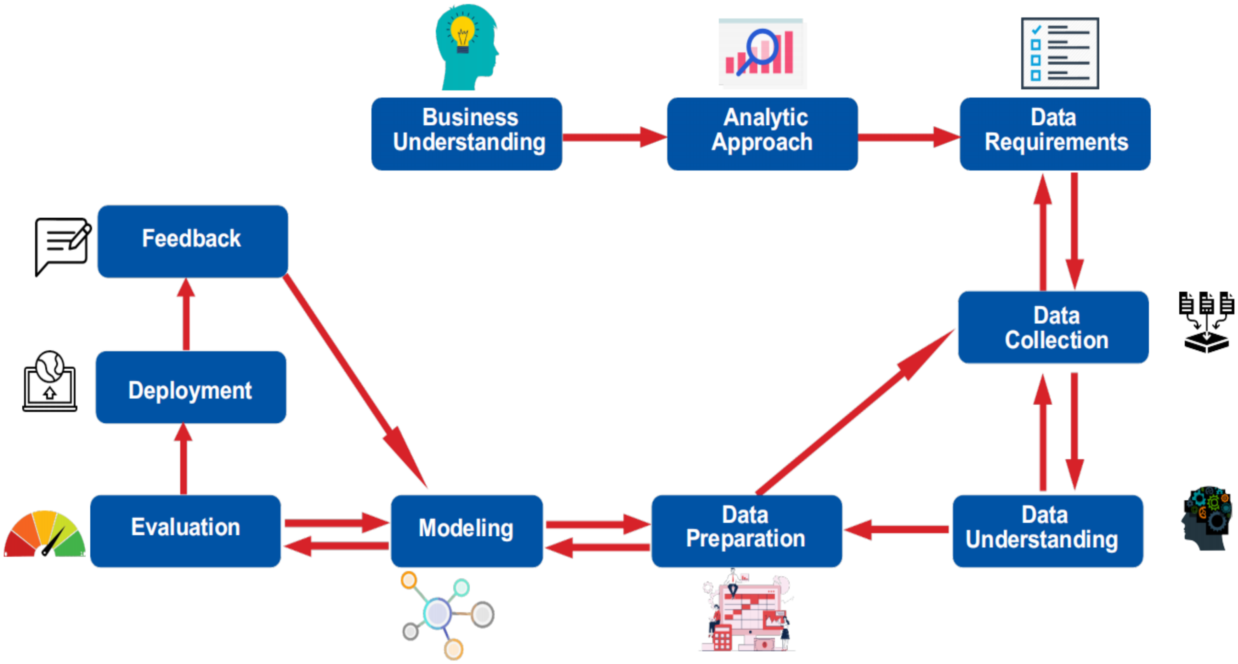
MACHINE LEARNING AND PARALLEL COMPUTING

ITS66604
Individual Assignment
Marking Scheme (JAN 2024)

[redacted]

Criteria	Score (Percentage of the allocated marks for each task)			
	Excellent	Good	Average	Poor
	>= 90%	< 90% , >= 70%	< 70% , >= 40%	< 40%
Introduction, Research Goal & Objectives	Section Introduction is written properly and the latest articles (2019+) are cited properly. Research goal and objectives are clearly well defined.	Section Introduction is written but the cited articles are not the latest ones. Research goal and objectives are defined.	Section Introduction does not clearly define the domain and related topics. The cited articles are not the latest ones. Research goal and objectives are defined.	Section Introduction does not clearly define the domain and related topics. The cited articles are not the latest ones. Research goal and objectives are clear.
Related Works	2 related Works are described properly and the latest articles (2019+)are cited properly.	1 related Works are described properly and the latest articles (2019+) are cited properly or 2 related Works are described but the cited articles are old.	1 related Works are described and the cited articles are old.	Less than 2 not related Works are described and the cited articles are old.
Methodology	The methodology used in the chosen topic are determined clearly and well defined.	The methodology used in the chosen topic are determined.	The methodology used in the chosen topic are not determined clearly.	The methodology is not described or it's not completed.
Implementation	The implementation is done strongly with clear outputs. A clear explanation is given. The related codes and implementation files are added. Related diagrams and tables are provided.	The implementation is done strongly with clear outputs. But the given explanation is not clear. The related codes and implementation files are added. Related diagrams and tables are provided but not sufficient.	The implementation is done with some mistake. The outputs are not clear. The given explanation is accepted but not sufficient. Not all of the related codes and implementation files are added. Related diagrams and tables are provided but not sufficient.	The implementation is not done completely. The outputs are not clear. The given explanation is not clear and sufficient. Most of the related codes and implementation files are missing. Related diagrams and tables are not sufficient.
Analysis & Recommendations	The two scholastic research articles' strong points and weaknesses are analyzed critically. The implemented solution is also analyzed and the performance is examined. The recommendations are also provided.	One of the two scholastic research articles' strong points and weaknesses is not analyzed critically. The implemented solution is also analyzed properly and/or its performance is not examined. The recommendation is missing or is not clear.	One of the two scholastic research articles' strong points and weaknesses are not analyzed critically. The implemented solution is not analyzed properly and/ or its performance is not examined. The recommendation is missing or is not clear.	One of the two scholastic research articles' strong points and weaknesses are not analyzed critically. The implemented solution is not analyzed properly and its performance is not examined. The recommendation is missing or is not clear.
Task 5: Submission Requirements	All the requirements are provided. Document is well designed and formatted. No grammar or spelling errors. The similarity is below 5%	All the requirements are provided. Document is not well designed and formatted or there are major grammar or spelling errors or the similarity is above 5% and less 15%	All the requirements are provided. Document is not well designed and formatted and there are major grammar or spelling errors or the similarity is above 15% and less 25%	The requirements items are completed. Document is not well designed and formatted and there are major grammar and/or spelling errors and/or the similarity is above 25%

- END -


Business Understanding
Analytic Approach
Data Requirements
Feedback
Data Collection
Deployment
Evaluation
Modeling
Data Preparation
Data Understanding