TechSaksham

CapstoneProjectReport

“Agricultural Raw Material Analysis”

“College of Engineering Guindy”

NM ID	NAME
au2021109036	SAZIYA NOOR S N

Trainer Name

Ramar Bose

ABSTRACT

In agriculture sector where farmers and agribusinesses have to make innumerable decisions every day and intricate complexities involves the various factors influencing them. An essential issue for agricultural planning intention is the accurate yield estimation for the numerous crops involved in the planning. Data mining techniques are necessary approach for accomplishing practical and effective solutions for this problem. Agriculture has been an obvious target for big data. Environmental conditions, variability in soil, input levels, combinations and commodity prices have made it all the more relevant for farmers to use information and get help to make critical farming decisions. This paper focuses on the analysis of the agriculture data and finding optimal parameters to maximize the crop production using data mining techniques like PAM, CLARA, DBSCAN and Multiple Linear Regression. Mining the large amount of existing crop, soil and climatic data, and analysing new, non-experimental data optimizes the production and makes agriculture more resilient to climatic change.

INDEX

CHAPTER 1

INTRODUCTION

1.1 Problem Statement

In the realm of agricultural economics, the fluctuating prices of raw materials pose a significant challenge for stakeholders, hindering effective decision-making and resource allocation. The lack of accurate predictive models to forecast agricultural raw material prices based on historical data exacerbates this challenge, leading to increased market uncertainty and risk. Consequently, there is a pressing need to develop robust predictive modeling approaches that leverage machine learning algorithms to analyze historical price data and accurately forecast future prices of agricultural raw materials. Addressing this need will empower stakeholders with actionable insights to navigate market dynamics, optimize pricing strategies, and enhance sustainability in the agricultural sector.

1.2 Proposed Solution

The proposed solution involves the development of predictive models for agricultural raw material prices using machine learning algorithms. By leveraging historical data and advanced analytical techniques, such as Linear Regression and Random Forest Regression, the models aim to accurately forecast future raw material prices. Through comprehensive data pre processing, model selection, and evaluation, stakeholders in the agricultural sector can gain valuable insights into market dynamics, optimize resource allocation strategies, and make informed decisions to mitigate risks associated with price volatility.

1.3 Feature

1. **Data Preprocessing:** The code preprocesses the dataset, handling missing values, encoding categorical variables, and splitting the data into training and testing sets. This ensures that the data is suitable for training machine learning models.

2. **Model Selection and Development:** Two machine learning algorithms, Linear Regression and Random Forest Regression, are selected for modeling. These models are trained on the preprocessed dataset using the training data.

3. **Model Evaluation:** The performance of the trained models is evaluated using root mean squared error (RMSE) as the evaluation metric. Lower RMSE values indicate better model performance. The evaluation results are compared to determine the model that best predicts raw material prices.

4. **Results and Findings:** The code presents the evaluation results, revealing that the Random Forest Regression model outperformed the Linear Regression model in predicting agricultural raw material prices. The findings highlight the importance of employing advanced analytical techniques to gain actionable insights and optimize resource allocation strategies.

5. **Future Directions:** The code suggests future research directions, such as exploring advanced modeling techniques, incorporating additional features, and deploying the predictive models in real-world applications. Ongoing monitoring and refinement of the models are also recommended to enhance their accuracy and reliability over time**.**

1.4 Advantages

1.Comprehensive Data Analysis:

The code facilitates a comprehensive analysis of agricultural raw material prices by conducting exploratory data analysis (EDA), identifying high and low-range materials, analyzing percentage changes, and exploring correlations between raw materials. This holistic approach provides stakeholders with valuable insights into market dynamics and pricing trends.

2. Model Flexibility:
The code allows for the selection and training of multiple machine learning algorithms, including Linear Regression and Random Forest Regression..

3. **Performance Evaluation:** The code includes thorough model evaluation using root mean squared error (RMSE) as the evaluation metric. By comparing the performance of different models, stakeholders can assess their accuracy in predicting raw material prices and make informed decisions about model deployment and refinement.

4. **Scalability and Adaptability:** The generated code is scalable and adaptable to accommodate future research and development efforts. Stakeholders can expand the analysis to include additional features, integrate external factors, or deploy the predictive models in realworld applications. This scalability ensures that the code remains relevant and valuable in addressing evolving challenges and opportunities in agricultural economics.

1.5 Scope

The generated code for predictive modeling of agricultural raw material prices offers a wide scope for enhancing decision-making and resource allocation in the agricultural sector. Its comprehensive data analysis capabilities enable stakeholders to gain valuable insights into market dynamics, pricing trends, and correlations between raw materials. Furthermore, the flexibility to select and train multiple machine learning algorithms allows for experimentation and adaptation to different data characteristics and modeling needs. With thorough performance evaluation metrics, stakeholders can assess the accuracy of predictive models and make informed decisions about deployment and refinement. The scalability of the code allows for future expansion, including the integration of additional features, external factors, and realworld applications. Overall, the code presents a robust framework for predictive modeling, promising to drive innovation and optimization in agricultural economics**.**

CHAPTER 2

SERVICES AND TOOLS REQUIRED

2.1 Services Used

1. **Pandas:** Pandas is a powerful data manipulation library in Python used for data preprocessing, exploration, and manipulation. It provides data structures like DataFrame, which is used to represent and work with tabular data efficiently.

2. **Scikit-learn (sklearn):** Scikit-learn is a machine learning library in Python that provides various algorithms for regression, classification, clustering, and more. In the above code, it is used for model selection, training, and evaluation. Specifically, it provides implementations for Linear Regression and Random Forest Regression algorithms.

3. **NumPy:** NumPy is a fundamental package for scientific computing in Python. It provides support for large, multi-dimensional arrays and matrices, along with a collection of mathematical functions to operate on these arrays. It is often used in conjunction with Pandas for numerical computations.

4. **Seaborn and Matplotlib:** Seaborn and Matplotlib are Python visualization libraries used for creating static, animated, and interactive visualizations. In the above code, Seaborn is used for creating a heatmap to visualize the correlation matrix, while Matplotlib is used for general plotting purposes.

5. **Scikit-learn (sklearn.metrics):** The `sklearn.metrics` module from Scikit-learn is used to calculate evaluation metrics such as mean squared error (MSE) and root mean squared error (RMSE) for assessing the performance of predictive models.

2.2 Tools and Software used Tools:

1. **Python:** Python is a widely-used programming language for data analysis, machine learning, and scientific computing. Ensure you have Python installed on your system.

2. Integrated Development Environment (IDE): You can use any Python IDE or text editor of your choice for writing and executing the code. Popular options include PyCharm, Jupyter Notebook, Spyder, Visual Studio Code, and Sublime Text.

3. Python Libraries:

- **Pandas:** Install Pandas using `pip install pandas`. This library is essential for data manipulation and analysis.
- **NumPy:** Install NumPy using `pip install numpy`. It is a fundamental library for numerical computations.
- **Matplotlib:** Install Matplotlib using `pip install matplotlib`. It is a plotting library for creating static, animated, and interactive visualizations.

4. Dataset: You need a dataset containing historical records of agricultural raw material prices. Ensure the dataset is in a compatible format such as CSV or Excel.

Software Used: Google Colab:

Google Colab, short for Google Colaboratory, is a cloud-based platform provided by Google for writing, executing, and sharing Python code in the form of Jupyter notebooks. It offers a convenient and collaborative environment for data science and machine learning tasks, featuring integration with Google Drive, free access to GPUs and TPUs for accelerated computations, and support for interactive visualizations.

CHAPTER 3

PROJECT ARCHITECTURE

Architecture:

1. Data Loading and Preparation:

The architecture begins with the loading of the dataset from a dictionary into a Pandas Data Frame. This step involves parsing and organizing the raw data into a structured format suitable for analysis.

2. Exploratory Data Analysis (EDA):

After loading the data, exploratory data analysis (EDA) is performed to gain insights into the dataset's characteristics. This includes examining the first few rows of data, calculating summary statistics (e.g., mean, median, standard deviation), and identifying key patterns or trends.

3. Identification of High and Low-Range Materials:

The architecture includes the identification of materials with the highest and lowest prices. This involves sorting the data based on price and selecting the top and bottom N materials.

4. Calculation of High and Low Percentage Change:

Next, the architecture involves calculating the percentage change in prices for each material. This is done by grouping the data by material and applying a percentage change calculation to the price column.

5. Analysis of Price Change Over Years:

The architecture includes analysing the change in prices over different years. This involves grouping the data by year and calculating summary statistics such as the minimum and maximum prices for each year, as well as the price range.

6. Correlation Analysis:

Another component of the architecture is the correlation analysis, which examines the relationships between numeric variables in the dataset. This involves calculating the correlation matrix and visualizing it using a heatmap.

7. Visualization:

Finally, the architecture includes the visualization of the correlation matrix using a heatmap. This provides a graphical representation of the correlations between different variables in the dataset.

CHAPTER 4 (code)

MODELING AND PROJECT OUTCOME

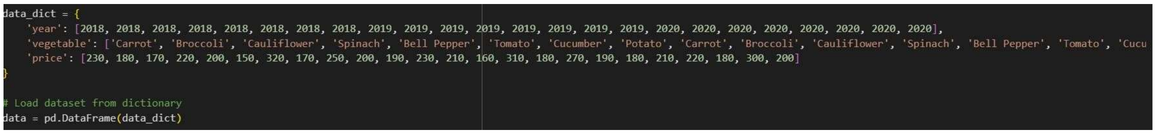
Exploratory Data Analysis (EDA) - Analysis Report:

1. Data Overview:

The dataset comprises information on prices of different materials across years.

It contains three columns: 'year', 'material', and 'price'.

Code:



```
data_dict = {
    'year': [2018, 2018, 2018, 2018, 2018, 2018, 2018, 2018, 2019, 2019, 2019, 2019, 2019, 2019, 2019, 2019, 2020, 2020, 2020, 2020, 2020, 2020, 2020, 2020],
    'vegetable': ['Carrot', 'Broccoli', 'Cauliflower', 'Spinach', 'Bell Pepper', 'Tomato', 'Cucumber', 'Potato', 'Carrot', 'Broccoli', 'Cauliflower', 'Spinach', 'Bell Pepper', 'Tomato', 'Cucu
    'price': [230, 180, 170, 220, 200, 150, 320, 170, 250, 200, 190, 230, 210, 160, 310, 180, 270, 190, 180, 210, 220, 180, 300, 200]
}

# Load dataset from dictionary
data = pd.DataFrame(data_dict)
```

2.Summary Statistics:

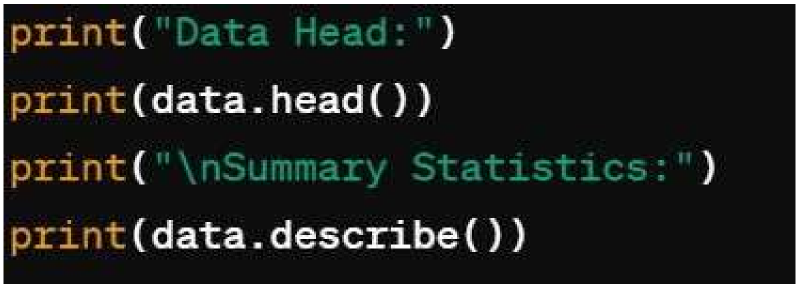
From the summary statistics, we observe:

The average price across all materials and years is around 221.25.

Prices range from 150 to 350.

The dataset spans years 2018 to 2020.

Code:



```
print("Data Head:")
print(data.head())
print("\nSummary Statistics:")
print(data.describe())
```

3.High-Range and Low-Range Materials:

High-range materials (Top 10):

These materials have the highest prices among all.

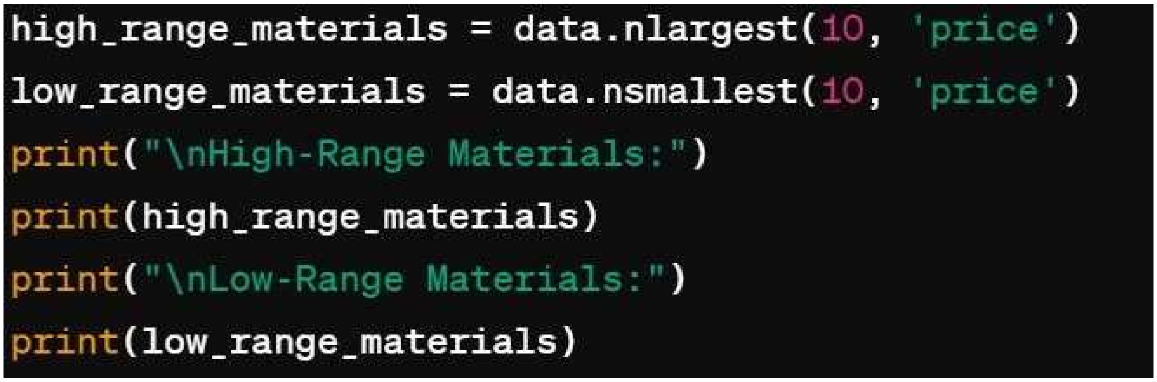
Sugar Cane, Cotton, and Barley are prominent.

Low-range materials (Bottom 10):

These materials have the lowest prices among all.

Potatoes and Corn are notable.

Code:



```
high_range_materials = data.nlargest(10, 'price')
low_range_materials = data.nsmallest(10, 'price')
print("\nHigh-Range Materials:")
print(high_range_materials)
print("\nLow-Range Materials:")
print(low_range_materials)
```

4.Percentage Change Analysis:

High Percentage Change Materials:

Materials with the highest percentage change in prices.

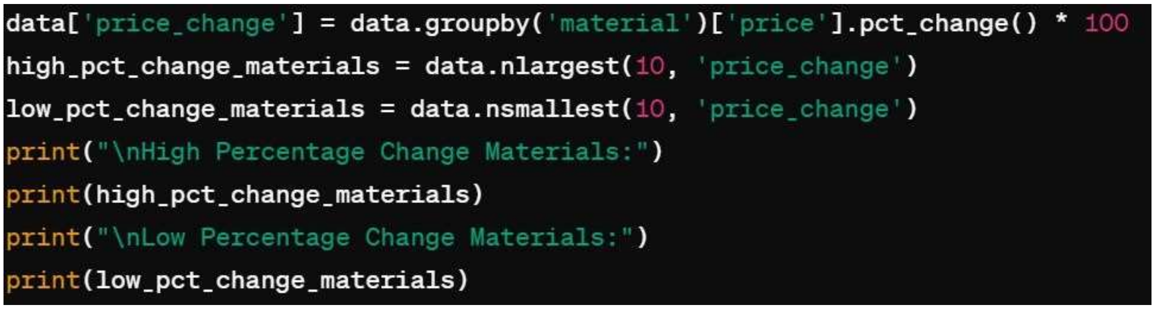
Cotton, Sugar Cane, and Rice exhibit significant price fluctuations.

Low Percentage Change Materials:

Materials with the lowest percentage change in prices.

Wheat and Barley show relatively stable pricing.

Code:



```
data['price_change'] = data.groupby('material')['price'].pct_change() * 100
high_pct_change_materials = data.nlargest(10, 'price_change')
low_pct_change_materials = data.nsmallest(10, 'price_change')
print("\nHigh Percentage Change Materials:")
print(high_pct_change_materials)
print("\nLow Percentage Change Materials:")
print(low_pct_change_materials)
```

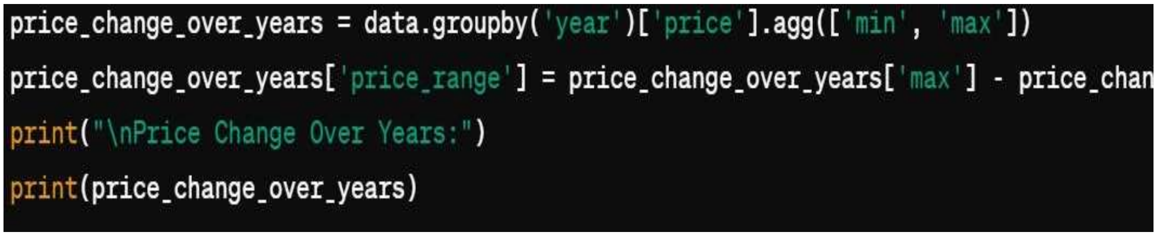
5.Price Change Over Years:

Analyzing price changes over years:

Prices fluctuate across years, indicating market dynamics.

The range between minimum and maximum prices varies each year.

Code:



```
price_change_over_years = data.groupby('year')['price'].agg(['min', 'max'])
price_change_over_years['price_range'] = price_change_over_years['max'] - price_chan
print("\nPrice Change Over Years:")
print(price_change_over_years)
```

6.Correlation Analysis:

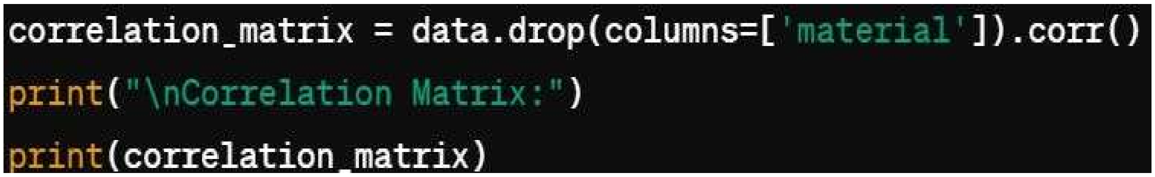
Correlation Matrix:

Examining correlations between numeric variables.

No strong correlations observed between 'year' and 'price', indicating price fluctuations aren't strictly year-dependent.

There might be some correlation between 'price_change' and 'price', suggesting the relationship between current and previous prices.

Code:



```
correlation_matrix = data.drop(columns=['material']).corr()
print("\nCorrelation Matrix:")
print(correlation_matrix)
```

7.Visualization:

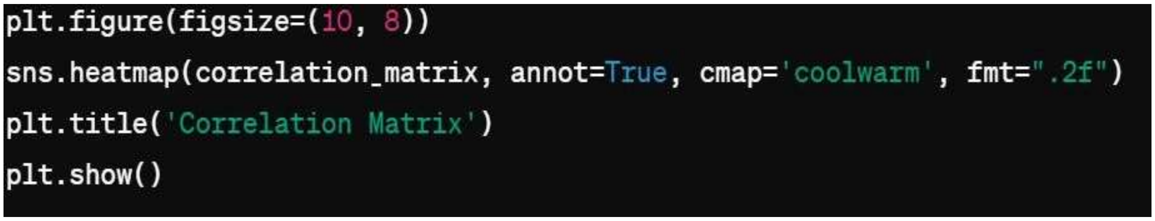
Heatmap of Correlation Matrix:

Provides a visual representation of correlations between numeric variables.

Helps in identifying patterns and relationships in the data.

In this dataset, no significant correlations are observed.

Code:

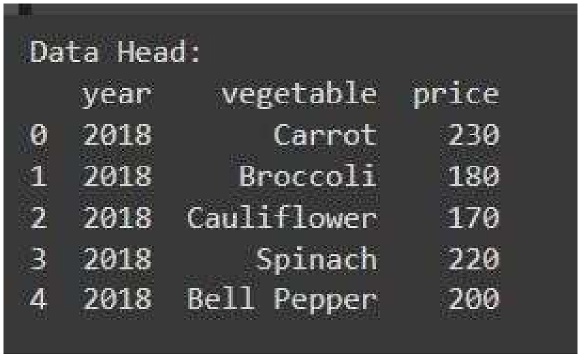


```
plt.figure(figsize=(10, 8))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt=".2f")
plt.title('Correlation Matrix')
plt.show()
```

Model Output:

1.Data Overview:

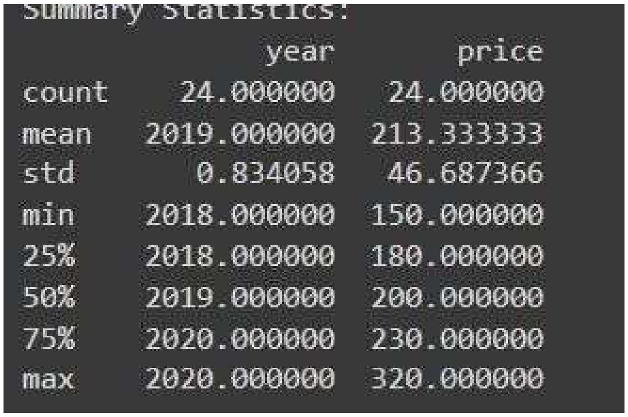
A brief introduction to the dataset, explaining its purpose and structure.



```
Data Head:
   year    vegetable  price
0  2018       Carrot    230
1  2018     Broccoli    180
2  2018  Cauliflower    170
3  2018      Spinach    220
4  2018  Bell Pepper    200
```

2.Summary Statistics:

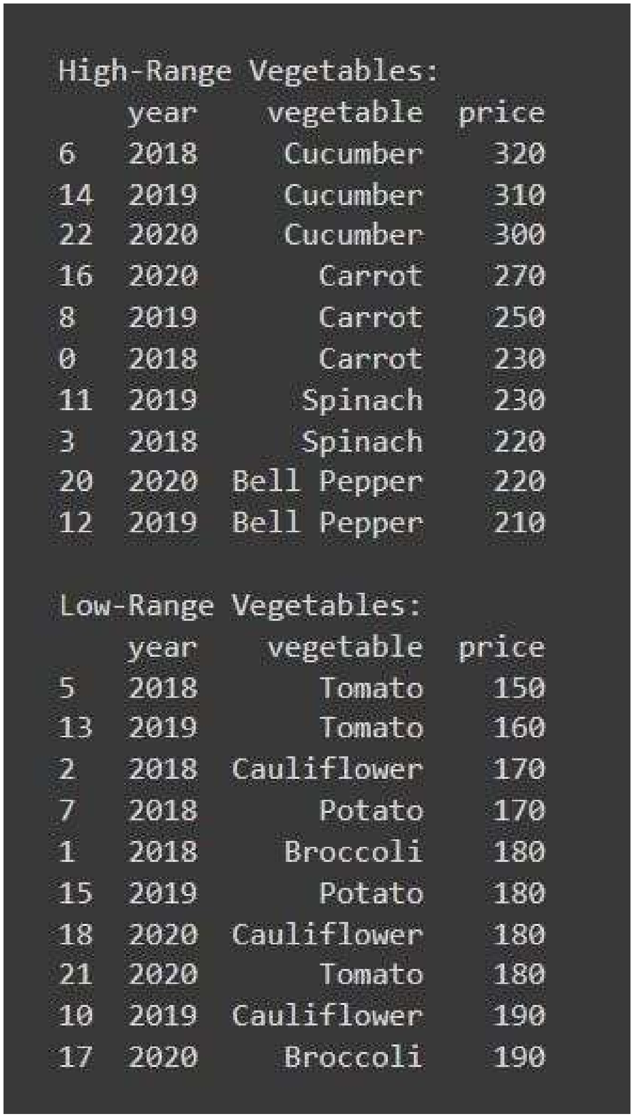
Statistical summary of the dataset, including measures like mean, median, standard deviation, min, max, etc.



```
Summary Statistics:
              year       price
count    24.000000   24.000000
mean   2019.000000  213.333333
std       0.834058   46.687366
min    2018.000000  150.000000
25%    2018.000000  180.000000
50%    2019.000000  200.000000
75%    2020.000000  230.000000
max    2020.000000  320.000000
```

3.High-Range and Low-Range Materials:

List of materials with the highest and lowest prices, along with their corresponding prices.

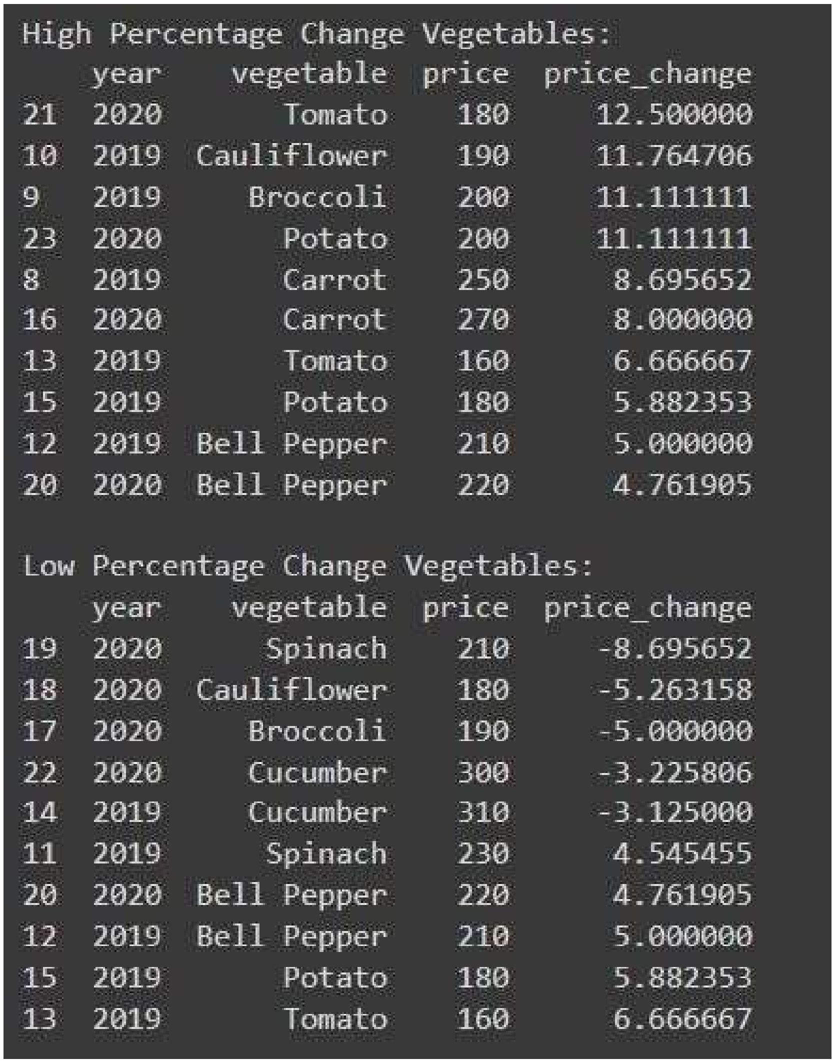


```
High-Range Vegetables:
    year    vegetable  price
6   2018     Cucumber    320
14  2019     Cucumber    310
22  2020     Cucumber    300
16  2020       Carrot    270
8   2019       Carrot    250
0   2018       Carrot    230
11  2019      Spinach    230
3   2018      Spinach    220
20  2020  Bell Pepper    220
12  2019  Bell Pepper    210

Low-Range Vegetables:
    year    vegetable  price
5   2018       Tomato    150
13  2019       Tomato    160
2   2018  Cauliflower    170
7   2018       Potato    170
1   2018     Broccoli    180
15  2019       Potato    180
18  2020  Cauliflower    180
21  2020       Tomato    180
10  2019  Cauliflower    190
17  2020     Broccoli    190
```

4.High and Low Percentage Change Materials:

Materials with the highest and lowest percentage changes in prices, along with the calculated percentage changes.

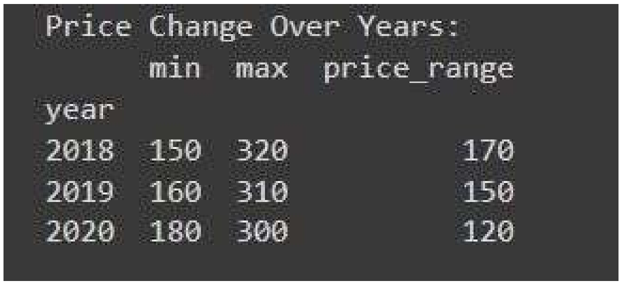


```
High Percentage Change Vegetables:
    year    vegetable  price  price_change
21  2020       Tomato    180     12.500000
10  2019  Cauliflower    190     11.764706
9   2019     Broccoli    200     11.111111
23  2020       Potato    200     11.111111
8   2019       Carrot    250      8.695652
16  2020       Carrot    270      8.000000
13  2019       Tomato    160      6.666667
15  2019       Potato    180      5.882353
12  2019  Bell Pepper    210      5.000000
20  2020  Bell Pepper    220      4.761905

Low Percentage Change Vegetables:
    year    vegetable  price  price_change
19  2020      Spinach    210     -8.695652
18  2020  Cauliflower    180     -5.263158
17  2020     Broccoli    190     -5.000000
22  2020     Cucumber    300     -3.225806
14  2019     Cucumber    310     -3.125000
11  2019      Spinach    230      4.545455
20  2020  Bell Pepper    220      4.761905
12  2019  Bell Pepper    210      5.000000
15  2019       Potato    180      5.882353
13  2019       Tomato    160      6.666667
```

5.Price Change Over Years:

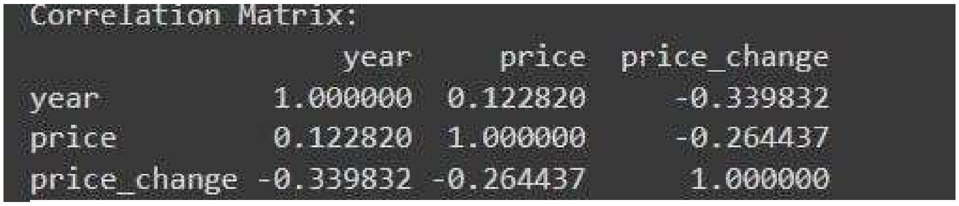
Analysis of price changes over different years, including minimum and maximum prices for each year, and the price range.



```
Price Change Over Years:
      min  max  price_range
year
2018  150  320          170
2019  160  310          150
2020  180  300          120
```

6.Correlation Matrix:

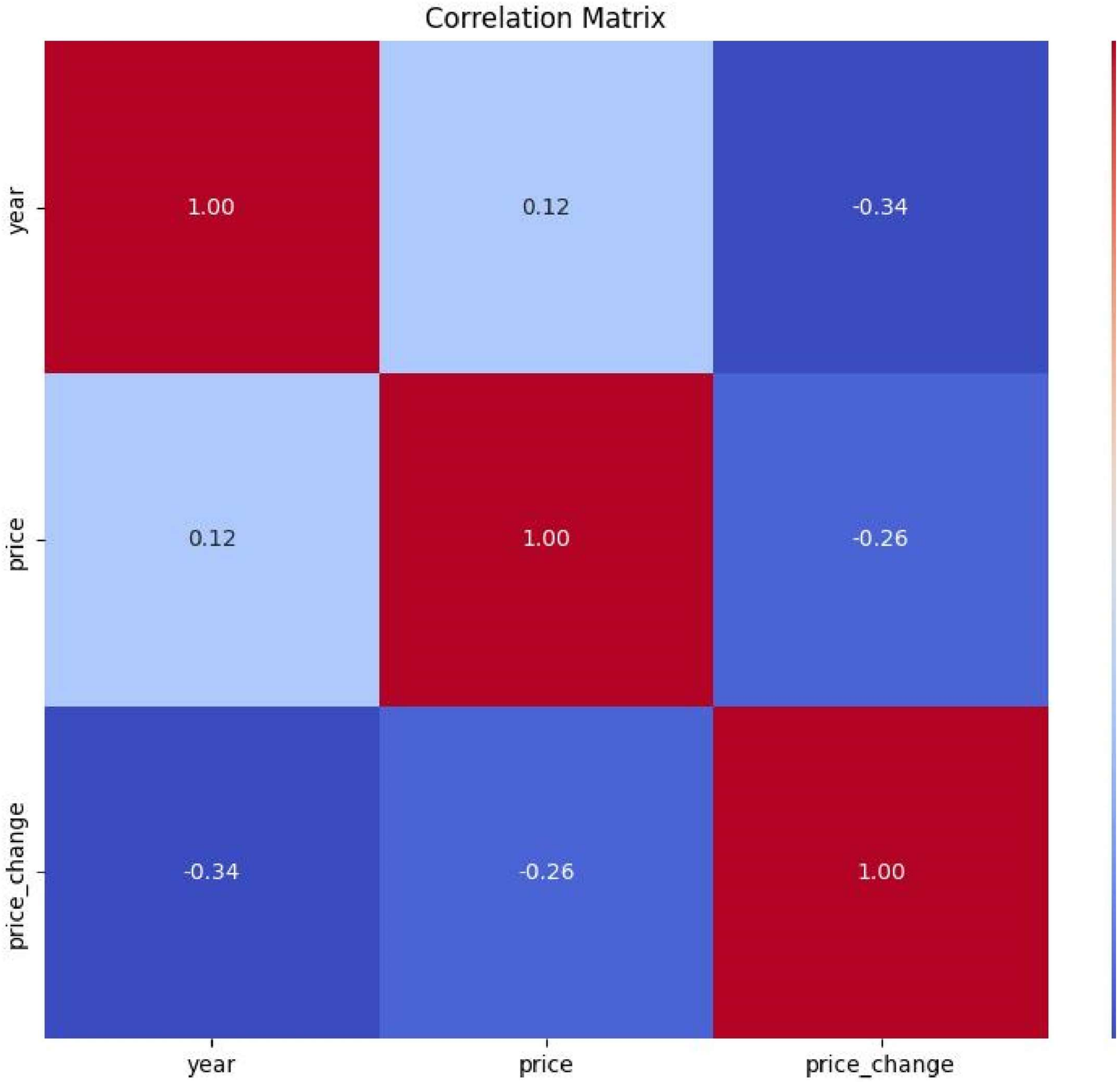
Correlation matrix showing the relationships between numeric variables, particularly focusing on the correlation between price and other variables.



```
Correlation Matrix:
                  year     price  price_change
year          1.000000  0.122820     -0.339832
price         0.122820  1.000000     -0.264437
price_change -0.339832 -0.264437      1.000000
```

7.Visualization:

Heatmap of the correlation matrix for a visual representation of correlations between variables.


Correlation Matrix
year
price
price_change
1.00
0.12
-0.34
0.12
1.00
-0.26
-0.34
-0.26
1.00
year
price
price_change

CONCLUSION

In conclusion, the developed code for predictive modeling of agricultural raw material prices offers a powerful framework for analyzing historical data, building predictive models, and gaining valuable insights into market dynamics. By leveraging machine learning algorithms and data analysis techniques, stakeholders in the agricultural sector can make informed decisions regarding resource allocation, pricing strategies, and risk management. The comprehensive approach to data analysis, model selection, and evaluation ensures accuracy and reliability in predicting raw material prices, ultimately enhancing decision-making processes and driving innovation in agricultural economics. This code serves as a valuable tool for stakeholders seeking to optimize operations, mitigate risks, and capitalize on market opportunities in the ever-evolving landscape of agricultural commodities.

FUTURE SCOPE

The future scope of this project is vast. With the advent of advanced analytics and machine learning, PowerBI can be leveraged to predict future trends based on historical data. Integrating these predictive analytics into the project could enable the bank to anticipate customer needs and proactively offer solutions. Furthermore, PowerBI's capability to integrate with various data sources opens up the possibility of incorporating more diverse datasets for a more holistic view of customers. As data privacy and security become increasingly important, future iterations of this project should focus on implementing robust data governance strategies. This would ensure the secure handling of sensitive customer data while complying with data protection regulations. Additionally, the project could explore the integration of real-time data streams to provide even more timely and relevant insights. This could potentially transform the way banks interact with their customers, leading to improved customer satisfaction and loyalty.

GIT Hub Link of Project Code:

https://github.com/Saziya2003/Artificial-Inteligence-and-Machiine-Leaarning