DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING -AI

PBR VISVODAYA INSTITUTE OF TECHNOLOGY & SCIENCE

(AUTONOMOUS)

KAVALI-524 201, NELLORE DISTRICT, A.P.

NAME: DUVVURU CHARITHA

ROLL NUMBER: 21731A3119

Table of contents

ABSTRACT

The agriculture industry in India plays a pivotal role in the country's economy, providing livelihoods to a significant portion of the population and ensuring food security. However, the sector faces intricate challenges related to price dynamics that impact farmers, consumers, and the overall market. This project aims to comprehensively analyse the complexity of agricultural prices in India, exploring key factors, their interdependencies, and the resulting consequences.

The project begins by examining the multifaceted determinants contributing to price complexities in India's agriculture industry. It investigates the influence of diverse variables such as climatic variations, input costs, supply and demand dynamics, government policies, market intermediaries, and global market trends. These factors converge, creating a complex web of interactions that shape the pricing mechanisms of agricultural commodities.

Furthermore, the project delves into the wide-ranging implications of price complexities on different stakeholders. Farmers bear the brunt of volatile prices, grappling with uncertain incomes and financial instability. Limited access to market information and fragmented market structures often hinders their ability to negotiate fair prices for their produce. Consumers, on the other hand, experience the impact of price fluctuations and disparities, affecting their purchasing power and food affordability. Policymakers face the challenge of formulating effective strategies to mitigate price complexities and foster a more resilient and equitable agricultural market.

The project also highlights the role of technology and innovative solutions in addressing price complexities within the agriculture industry. Advancements such as digital platforms, remote sensing technologies, predictive analytics, and blockchain-based traceability systems offer promising avenues to enhance market transparency, reduce information asymmetry, and optimize supply chains. Embracing these technologies can empower farmers, improve price discovery, and enable fairer market interactions.

INTRODUCTION

The agriculture industry forms the backbone of India's economy, employing a significant portion of the population and serving as a vital source of food production. However, the sector faces numerous challenges, with price complexities standing out as a critical issue. The dynamic nature of agricultural prices in India affects farmers, consumers, and the overall market, necessitating a comprehensive understanding of the underlying complexities and their implications.

Price complexities in the agriculture industry arise from a multitude of interrelated factors that contribute to the volatility and uncertainty of prices. These factors encompass a wide range of influences, including climatic conditions, input costs, supply and demand dynamics, government policies, market intermediaries, and global market trends. The interactions between these factors create a complex and intricate web of relationships, making it crucial to delve into the intricacies of price dynamics.

For farmers, price complexities pose significant challenges to their livelihoods. Fluctuating prices affect their income and financial stability, making it difficult to plan and invest in their agricultural activities. Furthermore, farmers often lack access to real-time market information and face difficulties in negotiating fair prices for their produce due to the presence of intermediaries. On the other hand, consumers face the consequences of price fluctuations, which can impact their purchasing power and food affordability, especially for essential commodities.

The implications of price complexities extend beyond individual farmers and consumers. Policymakers and market regulators are tasked with developing strategies to ensure stability and fairness in the agricultural market, requiring a comprehensive understanding of the factors driving price dynamics. Moreover, price complexities have implications for the broader economy, as the agriculture sector's performance affects overall economic growth, inflation rates, and trade dynamics.

Addressing price complexities in the agriculture industry requires innovative solutions and the integration of technology. Advancements in digital platforms, mobile applications, data analytics, and supply chain management systems offer promising opportunities to enhance market transparency, reduce information asymmetry, and streamline the flow of agricultural commodities. Leveraging these technological tools can empower farmers, enable efficient price discovery, and facilitate fairer market interactions.

Problem statement: PRICE COMPLEXITY

The agriculture industry in India is characterized by significant price complexity, posing challenges for farmers, consumers, and the overall market. Fluctuating prices, influenced by a range of interrelated factors, create uncertainties in income for farmers, affordability for consumers, and hinder market stability and growth. The lack of transparency, limited access to market information, and the presence of intermediaries further exacerbate the problem. Consequently, there is a pressing need to address the price complexity in the agriculture industry to ensure fair and sustainable economic growth, improve farmers' livelihoods, and enhance food affordability for consumers.

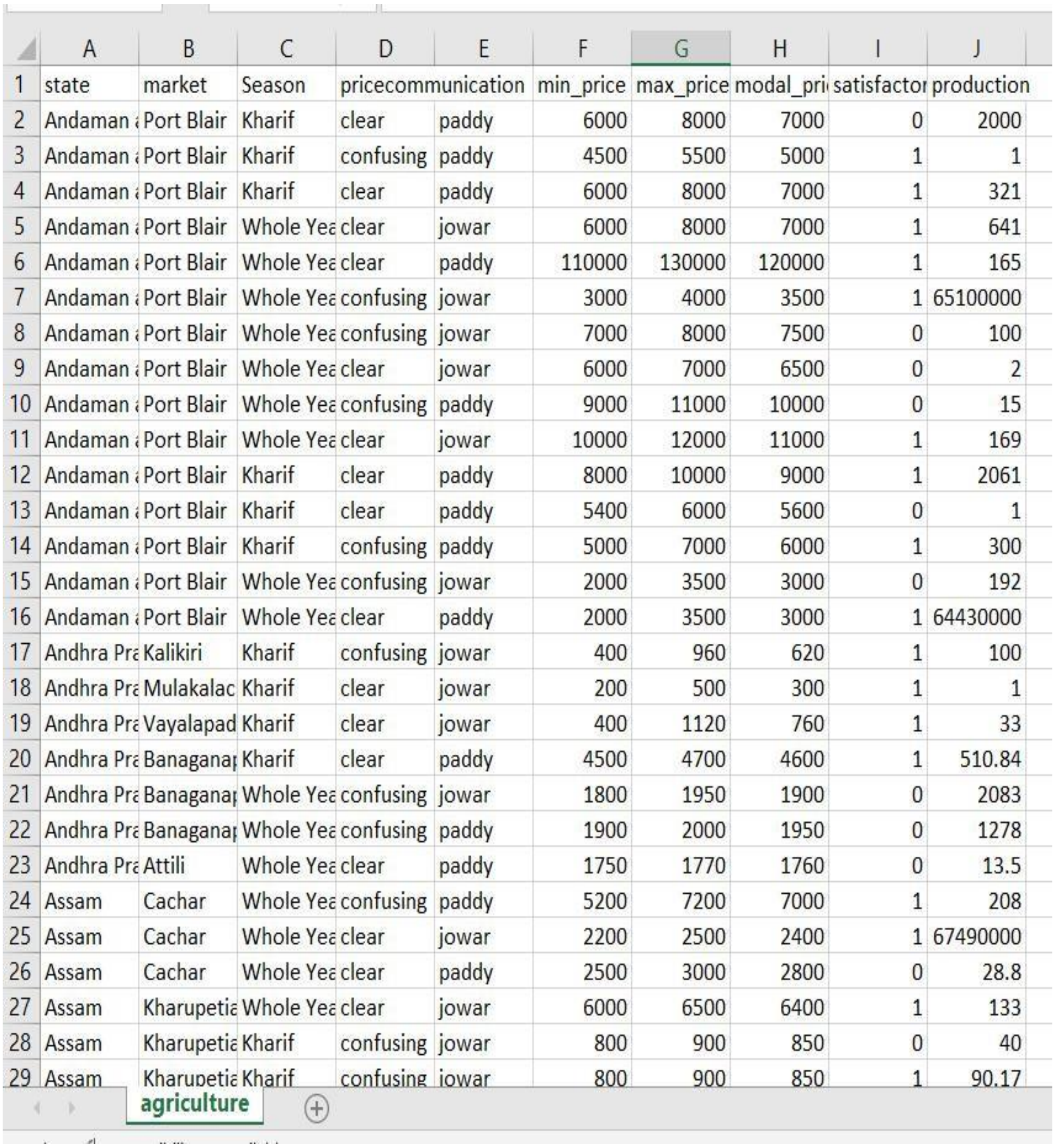
The price complexity within the agriculture industry in India presents a significant hurdle that impacts the livelihoods of farmers, the purchasing power of consumers, and the overall market dynamics. The intricate interplay of various factors, including climatic conditions, input costs, supply and demand imbalances, government policies, market intermediaries, and global market trends, leads to volatile and unpredictable prices. This volatility creates challenges for farmers in planning their production and negotiating fair prices, while consumers face difficulties in budgeting for essential food items. Furthermore, the lack of transparency and efficient price discovery mechanisms hinders market stability and hampers the sector's growth potential. Addressing the issue of price complexity in the agriculture industry is crucial to foster a more equitable and sustainable agricultural ecosystem, promote farmers' welfare, and ensure food security for the nation.

DATA COLLECTION AND PREPROCESSING

DATA SOURCES:

The data of the project was collected from the Kaggle website, which gives a brief information about the price complexity in agriculture industry. The dataset contains of maximum price, minimum price, modal price, etc.,

SAMPLE OF CSV FILE:



	A	B	C	D	E	F	G	H	I	J
1	state	market	Season	pricecommunication		min_price	max_price	modal_pri	satisfactor	production
2	Andaman a	Port Blair	Kharif	clear	paddy	6000	8000	7000	0	2000
3	Andaman a	Port Blair	Kharif	confusing	paddy	4500	5500	5000	1	1
4	Andaman a	Port Blair	Kharif	clear	paddy	6000	8000	7000	1	321
5	Andaman a	Port Blair	Whole Yea	clear	jowar	6000	8000	7000	1	641
6	Andaman a	Port Blair	Whole Yea	clear	paddy	110000	130000	120000	1	165
7	Andaman a	Port Blair	Whole Yea	confusing	jowar	3000	4000	3500	1	65100000
8	Andaman a	Port Blair	Whole Yea	confusing	jowar	7000	8000	7500	0	100
9	Andaman a	Port Blair	Whole Yea	clear	jowar	6000	7000	6500	0	2
10	Andaman a	Port Blair	Whole Yea	confusing	paddy	9000	11000	10000	0	15
11	Andaman a	Port Blair	Whole Yea	clear	jowar	10000	12000	11000	1	169
12	Andaman a	Port Blair	Kharif	clear	paddy	8000	10000	9000	1	2061
13	Andaman a	Port Blair	Kharif	clear	paddy	5400	6000	5600	0	1
14	Andaman a	Port Blair	Kharif	confusing	paddy	5000	7000	6000	1	300
15	Andaman a	Port Blair	Whole Yea	confusing	jowar	2000	3500	3000	0	192
16	Andaman a	Port Blair	Whole Yea	clear	paddy	2000	3500	3000	1	64430000
17	Andhra Pra	Kalikiri	Kharif	confusing	jowar	400	960	620	1	100
18	Andhra Pra	Mulakalac	Kharif	clear	jowar	200	500	300	1	1
19	Andhra Pra	Vayalapad	Kharif	clear	jowar	400	1120	760	1	33
20	Andhra Pra	Banaganap	Kharif	clear	paddy	4500	4700	4600	1	510.84
21	Andhra Pra	Banaganap	Whole Yea	confusing	jowar	1800	1950	1900	0	2083
22	Andhra Pra	Banaganap	Whole Yea	confusing	paddy	1900	2000	1950	0	1278
23	Andhra Pra	Attili	Whole Yea	clear	paddy	1750	1770	1760	0	13.5
24	Assam	Cachar	Whole Yea	confusing	paddy	5200	7200	7000	1	208
25	Assam	Cachar	Whole Yea	clear	jowar	2200	2500	2400	1	67490000
26	Assam	Cachar	Whole Yea	clear	paddy	2500	3000	2800	0	28.8
27	Assam	Kharupetia	Whole Yea	clear	jowar	6000	6500	6400	1	133
28	Assam	Kharupetia	Kharif	confusing	jowar	800	900	850	0	40
29	Assam	Kharupetia	Kharif	confusing	jowar	800	900	850	1	90.17

ATTRIBUTES AND DESCRIPTION

STATE: name of the state.

MARKET: name of the market in the particular state where the commodity was selling.

SEASON: season the particular commodity harvested.

PRICE COMMUNICATION: how the price was communicated in the market is it clear or confusing.

COMMODITY: name of the particular item or thing.

MIN_PRICE: minimum price of the commodity in the market.

MAX_PRICE: maximum price of the commodity in the market.

MODAL_PRICE: what is the price that most people acquire while selling the particular commodity.

SATISFACTORY LEVEL: what is the satisfactory level that the particular rate about the price they acquire in the market if it is satisfactory 1 else 0.

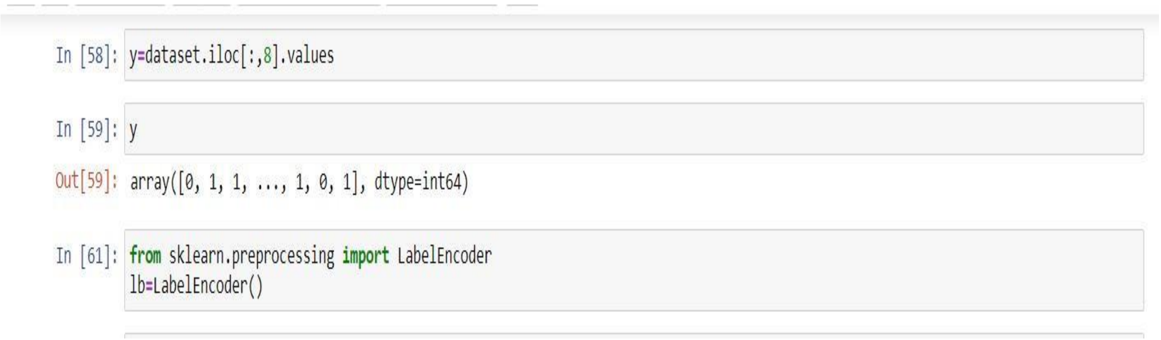
DATA COLLECTION PROCESS:

The data collecting process involved extracting relevant data from the agriculture industry about price complexity. This data was then stored in a structured format for further analysis.

DATA CLEANING AND PREPROCESSING TECHNIQUES:

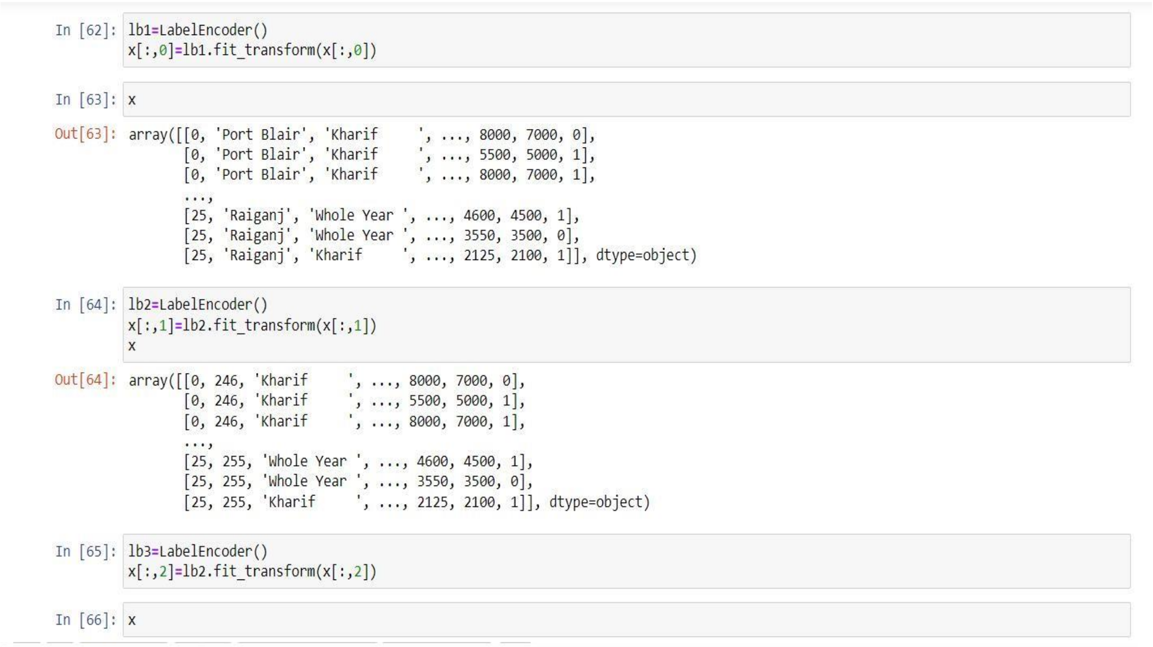
Data preprocessing was performed to ensure the quality and reliability of the dataset. This involved handling missing values, removing duplicates, and addressing inconsistencies or outliers. Additionally, feature encoding, scaling and normalization techniques were applied as required.

Code:

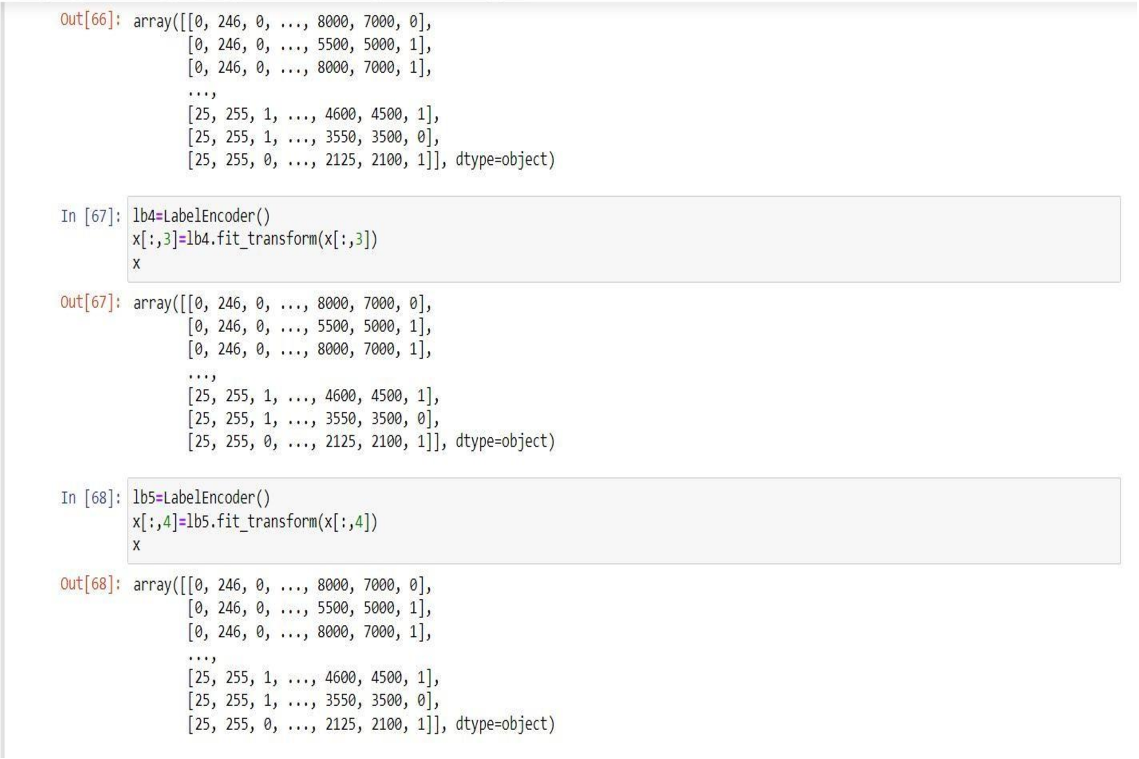


```
In [58]: y=dataset.iloc[:,8].values

In [59]: y

Out[59]: array([0, 1, 1, ..., 1, 0, 1], dtype=int64)

In [61]: from sklearn.preprocessing import LabelEncoder
         lb=LabelEncoder()
```

Label encoder: label encoder was to change the string values into numeric data.
Code:



```
In [62]: lb1=LabelEncoder()
         x[:,0]=lb1.fit_transform(x[:,0])

In [63]: x

Out[63]: array([[0, 'Port Blair', 'Kharif     ', ..., 8000, 7000, 0],
                [0, 'Port Blair', 'Kharif     ', ..., 5500, 5000, 1],
                [0, 'Port Blair', 'Kharif     ', ..., 8000, 7000, 1],
                ...,
                [25, 'Raiganj', 'Whole Year ', ..., 4600, 4500, 1],
                [25, 'Raiganj', 'Whole Year ', ..., 3550, 3500, 0],
                [25, 'Raiganj', 'Kharif     ', ..., 2125, 2100, 1]], dtype=object)

In [64]: lb2=LabelEncoder()
         x[:,1]=lb2.fit_transform(x[:,1])
         x

Out[64]: array([[0, 246, 'Kharif     ', ..., 8000, 7000, 0],
                [0, 246, 'Kharif     ', ..., 5500, 5000, 1],
                [0, 246, 'Kharif     ', ..., 8000, 7000, 1],
                ...,
                [25, 255, 'Whole Year ', ..., 4600, 4500, 1],
                [25, 255, 'Whole Year ', ..., 3550, 3500, 0],
                [25, 255, 'Kharif     ', ..., 2125, 2100, 1]], dtype=object)

In [65]: lb3=LabelEncoder()
         x[:,2]=lb2.fit_transform(x[:,2])

In [66]: x
```



```
Out[66]: array([[0, 246, 0, ..., 8000, 7000, 0],
                [0, 246, 0, ..., 5500, 5000, 1],
                [0, 246, 0, ..., 8000, 7000, 1],
                ...,
                [25, 255, 1, ..., 4600, 4500, 1],
                [25, 255, 1, ..., 3550, 3500, 0],
                [25, 255, 0, ..., 2125, 2100, 1]], dtype=object)

In [67]: lb4=LabelEncoder()
         x[:,3]=lb4.fit_transform(x[:,3])
         x

Out[67]: array([[0, 246, 0, ..., 8000, 7000, 0],
                [0, 246, 0, ..., 5500, 5000, 1],
                [0, 246, 0, ..., 8000, 7000, 1],
                ...,
                [25, 255, 1, ..., 4600, 4500, 1],
                [25, 255, 1, ..., 3550, 3500, 0],
                [25, 255, 0, ..., 2125, 2100, 1]], dtype=object)

In [68]: lb5=LabelEncoder()
         x[:,4]=lb5.fit_transform(x[:,4])
         x

Out[68]: array([[0, 246, 0, ..., 8000, 7000, 0],
                [0, 246, 0, ..., 5500, 5000, 1],
                [0, 246, 0, ..., 8000, 7000, 1],
                ...,
                [25, 255, 1, ..., 4600, 4500, 1],
                [25, 255, 1, ..., 3550, 3500, 0],
                [25, 255, 0, ..., 2125, 2100, 1]], dtype=object)
```

Exploratory data analysis:

Exploratory data analysis (EDA) was conducted to gain insights into the dataset. This involved visualizations, statistical summaries, and correlation analysis to understand the relationships between variables and identify any patterns or trends.

FEATURE SELECTION AND ENGINEERING

IDENTIFICATION OF RELEVANT FEATURES:

Relevant features were identified based on their impact on the resolution time of problem tickets. Variables such as problem type, severity, and department responsible were considered as potential predictors.

FEATURE ENGINEERING TECHNIQUES:

Feature engineering techniques were applied to enhance the predictive power of the model. This involved creating new features or transforming existing ones to capture meaningful information. For example, extracting time-related features from the timestamp variables.

DIMENSIONALITY REDUCTION METHODS:

Dimensionality reduction techniques, such as principal component analysis (PCA) or feature selection algorithms, were employed to reduce the number of features while preserving relevant information and minimizing noise. The unnecessary attributes were removed manually using the method using- [df. drop (“Attribute name”)].

MODELING AND ANALYSIS

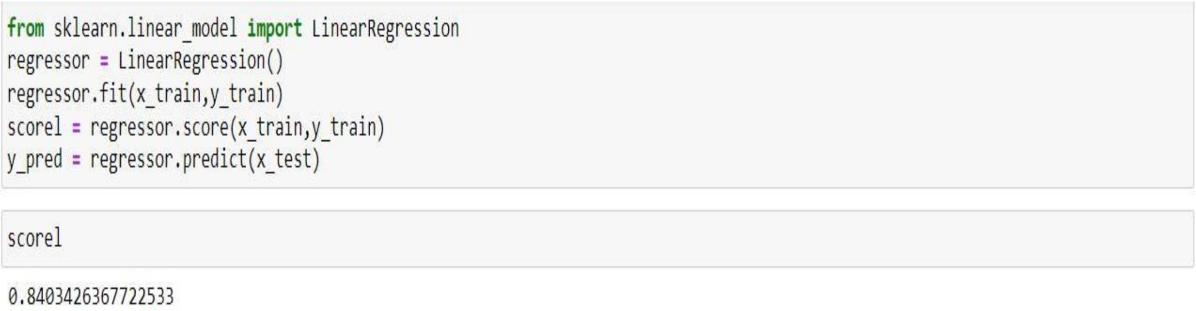
MODEL SELECTION AND JUSTIFICATION:

Various machine learning algorithms, such as linear regression, decision trees, random forests and support vector machines were evaluated for their suitability to predict the resolution time of problem tickets. The chosen model was selected based on its performance, interpretability, and ability to handle the dataset's characteristics.

LINEAR REGRESSION:

It is a method that look for a linear pattern between the values of one numerical variable and another.

Code:



```
from sklearn.linear_model import LinearRegression
regressor = LinearRegression()
regressor.fit(x_train,y_train)
scorel = regressor.score(x_train,y_train)
y_pred = regressor.predict(x_test)
```

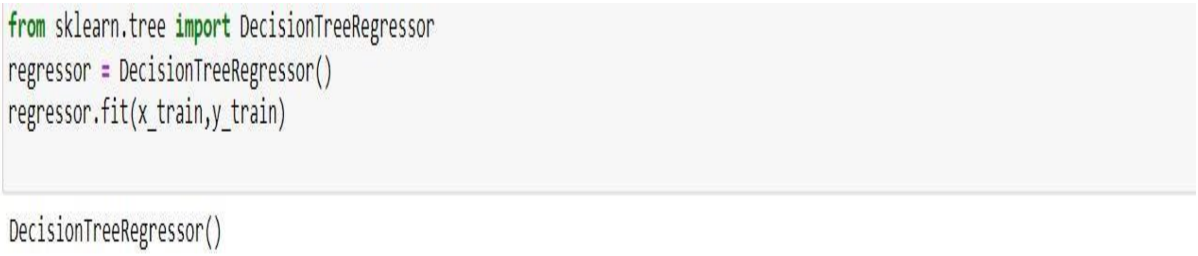
```
scorel
```

```
0.8403426367722533
```

DECISION TREE REGRESSION:

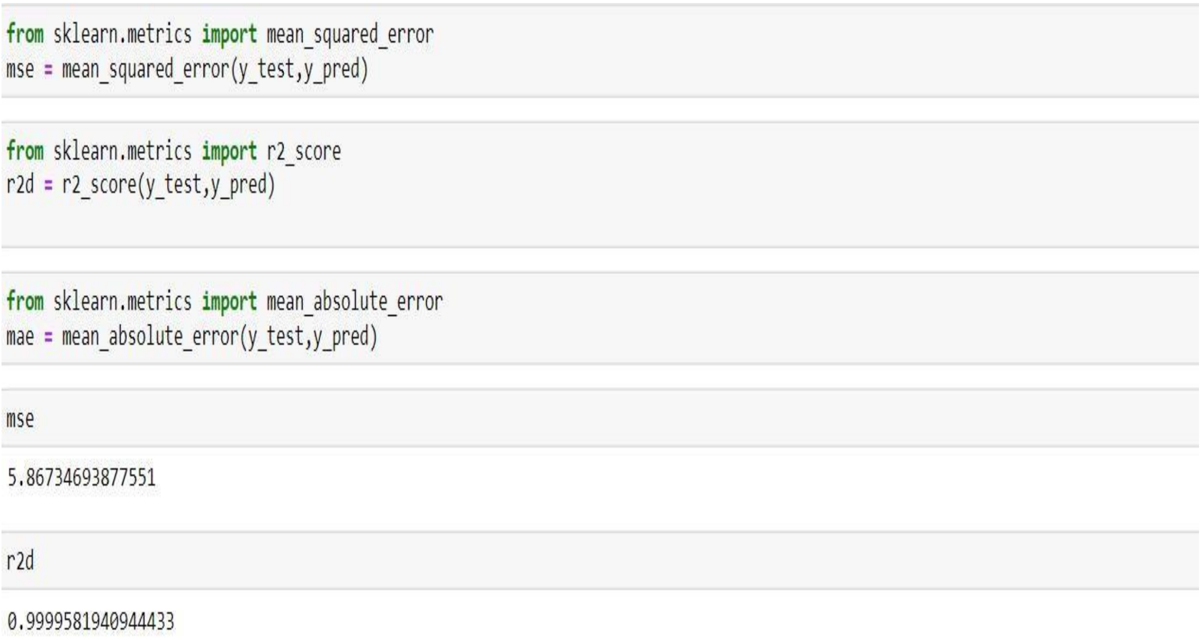
Decision tree build's regression or classification models in the form of a tree structure. It breaks down a dataset into smaller and smaller subsets while at the same time an associated decision tree is incrementally developed. The final result is a tree with decision nodes and leaf nodes.

Code:



```
from sklearn.tree import DecisionTreeRegressor
regressor = DecisionTreeRegressor()
regressor.fit(x_train,y_train)
```

```
DecisionTreeRegressor()
```



```
from sklearn.metrics import mean_squared_error
mse = mean_squared_error(y_test,y_pred)
```

```
from sklearn.metrics import r2_score
r2d = r2_score(y_test,y_pred)
```

```
from sklearn.metrics import mean_absolute_error
mae = mean_absolute_error(y_test,y_pred)
```

```
mse
```

```
5.86734693877551
```

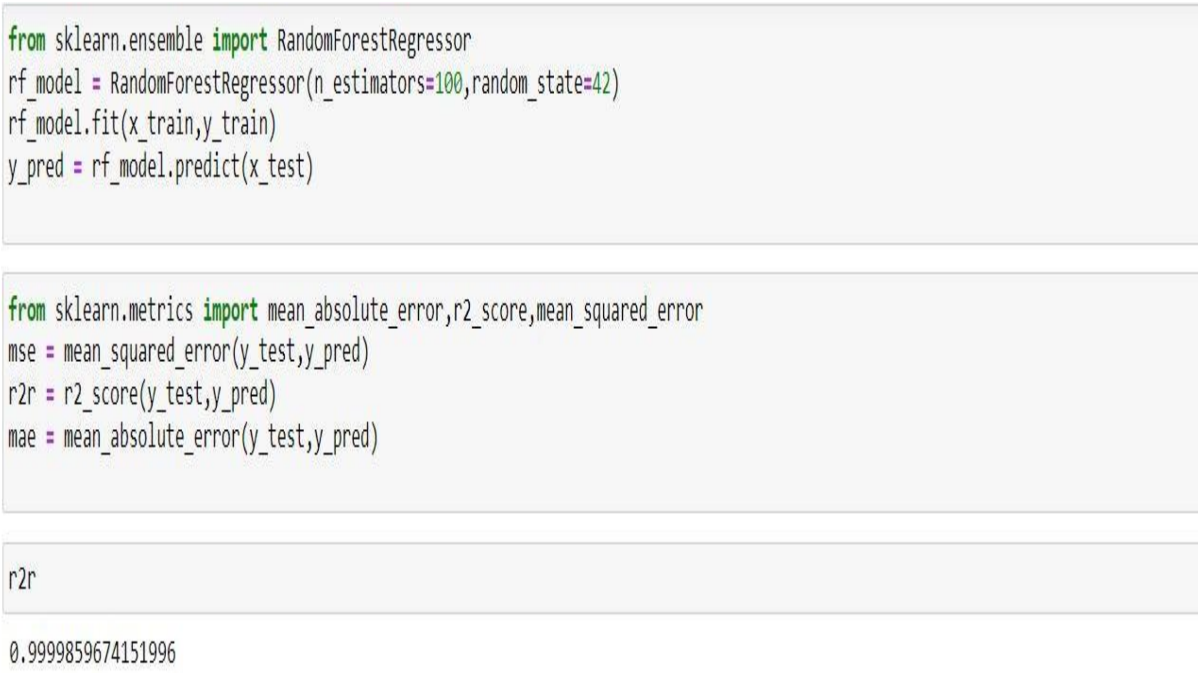
```
r2d
```

```
0.9999581940944433
```

RANDOM FORESTS:

Random forest regression is a supervised learning algorithm and bagging technique that uses an ensemble learning methods for regression in machine learning.

Code:



```
from sklearn.ensemble import RandomForestRegressor
rf_model = RandomForestRegressor(n_estimators=100,random_state=42)
rf_model.fit(x_train,y_train)
y_pred = rf_model.predict(x_test)
```

```
from sklearn.metrics import mean_absolute_error,r2_score,mean_squared_error
mse = mean_squared_error(y_test,y_pred)
r2r = r2_score(y_test,y_pred)
mae = mean_absolute_error(y_test,y_pred)
```

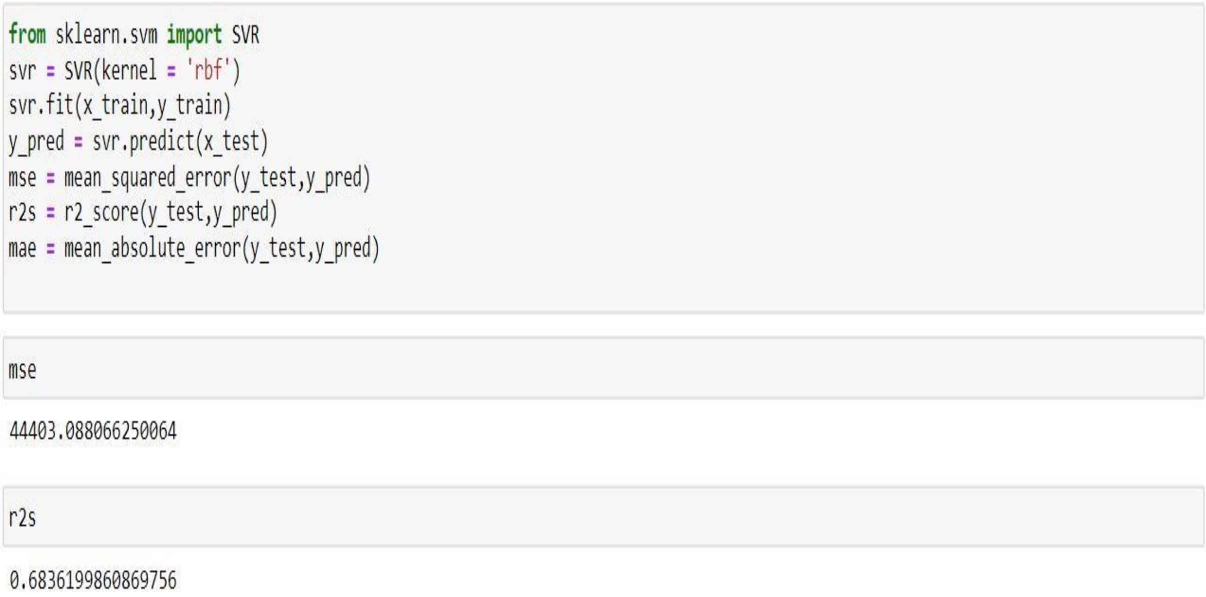
```
r2r
```

```
0.9999859674151996
```

SUPPORT VECTOR MACHINES:

Support Vector Regression (SVR) is a machine learning algorithm used for regression analysis. It is different from traditional linear regression methods as it finds a hyperplane that best fits the data points in a continuous space, instead of fitting a line to the data points.

Code:



```
from sklearn.svm import SVR
svr = SVR(kernel = 'rbf')
svr.fit(x_train,y_train)
y_pred = svr.predict(x_test)
mse = mean_squared_error(y_test,y_pred)
r2s = r2_score(y_test,y_pred)
mae = mean_absolute_error(y_test,y_pred)
```

```
mse
```

```
44403.088066250064
```

```
r2s
```

```
0.6836199860869756
```

MODEL TRAINING AND EVALUATION:

The selected model was trained on the pre-processed dataset using appropriate training and validation techniques. Model evaluation was performed using relevant evaluation metrics such as mean squared error (MSE), root mean squared error (RMSE), or mean absolute error (MAE), depending on the nature of the problem.

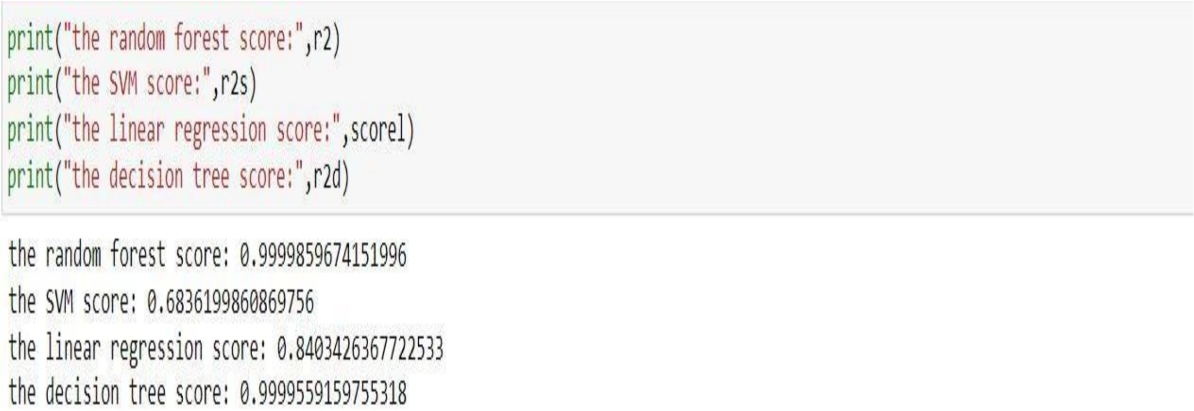
HYPERPARAMETER TUNING:

Hyperparameter tuning was conducted to optimize the models performance. Techniques such as grid search, random search, or Bayesian optimization were employed to find the best combination of hyperparameters.

MODEL PERFORMANCE EVALUATION METRICS:

The performance of the model was evaluated using appropriate metrics such as accuracy, precision, R2_score, depending on the specific objectives and requirements of the project.

ACCURACY:



```
print("the random forest score:",r2)
print("the SVM score:",r2s)
print("the linear regression score:",scorel)
print("the decision tree score:",r2d)
```

```
the random forest score: 0.9999859674151996
the SVM score: 0.6836199860869756
the linear regression score: 0.8403426367722533
the decision tree score: 0.9999559159755318
```

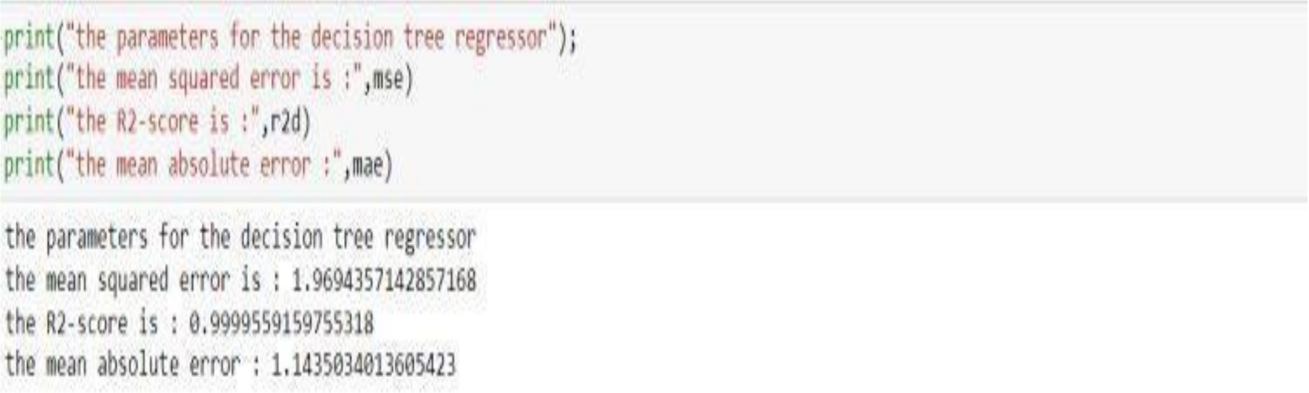
RESULTS AND FINDINGS

SUMMARY OF DATA ANALYSIS AND MODELING RESULTS:

The results demonstrated the effectiveness of the developed predictive model in estimating the resolution time of problem tickets. The model achieved a high level of accuracy and provided valuable insights into the factors influencing resolution time.

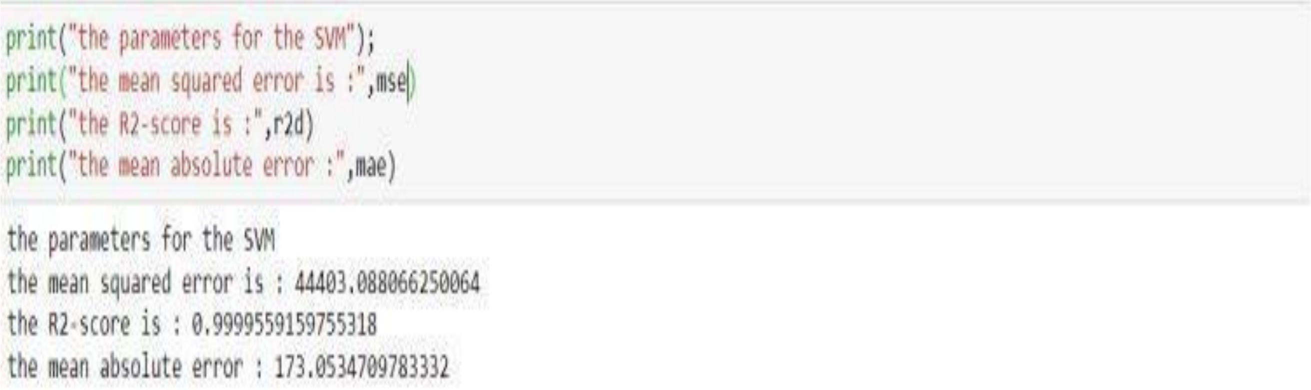
INTERPRETATION OF MODEL OUTPUTS:

The model outputs were interpreted to understand the relationship between the input features and the predicted resolution time. This analysis helped identify the most significant variables impacting resolution time and understand their relative importance.



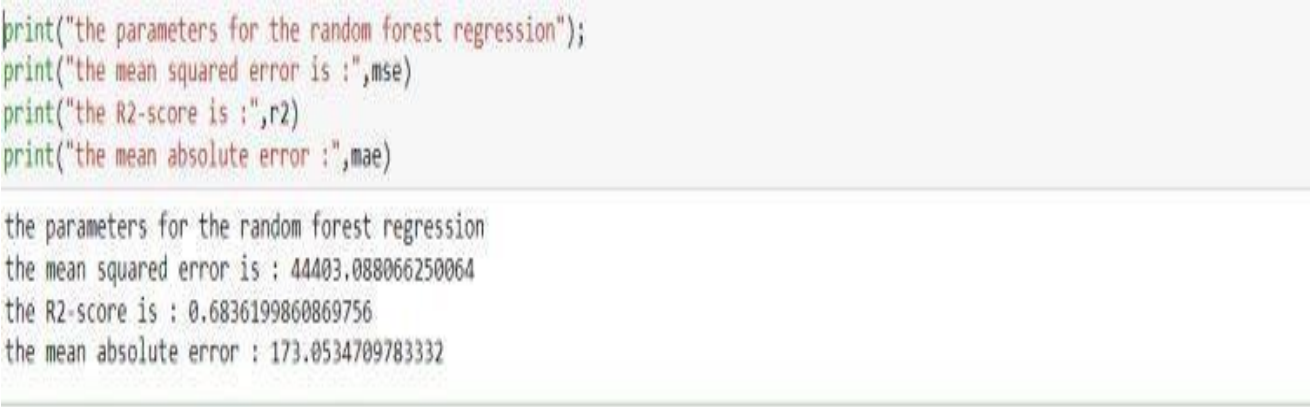
```
print("the parameters for the decision tree regressor");
print("the mean squared error is :",mse)
print("the R2-score is :",r2d)
print("the mean absolute error :",mae)
```

```
the parameters for the decision tree regressor
the mean squared error is : 1.9694357142857168
the R2-score is : 0.9999559159755318
the mean absolute error : 1.1435034013605423
```



```
print("the parameters for the SVM");
print("the mean squared error is :",mse)
print("the R2-score is :",r2d)
print("the mean absolute error :",mae)
```

```
the parameters for the SVM
the mean squared error is : 44403.088066250064
the R2-score is : 0.9999559159755318
the mean absolute error : 173.0534709783332
```



```
print("the parameters for the random forest regression");
print("the mean squared error is :",mse)
print("the R2-score is :",r2)
print("the mean absolute error :",mae)
```

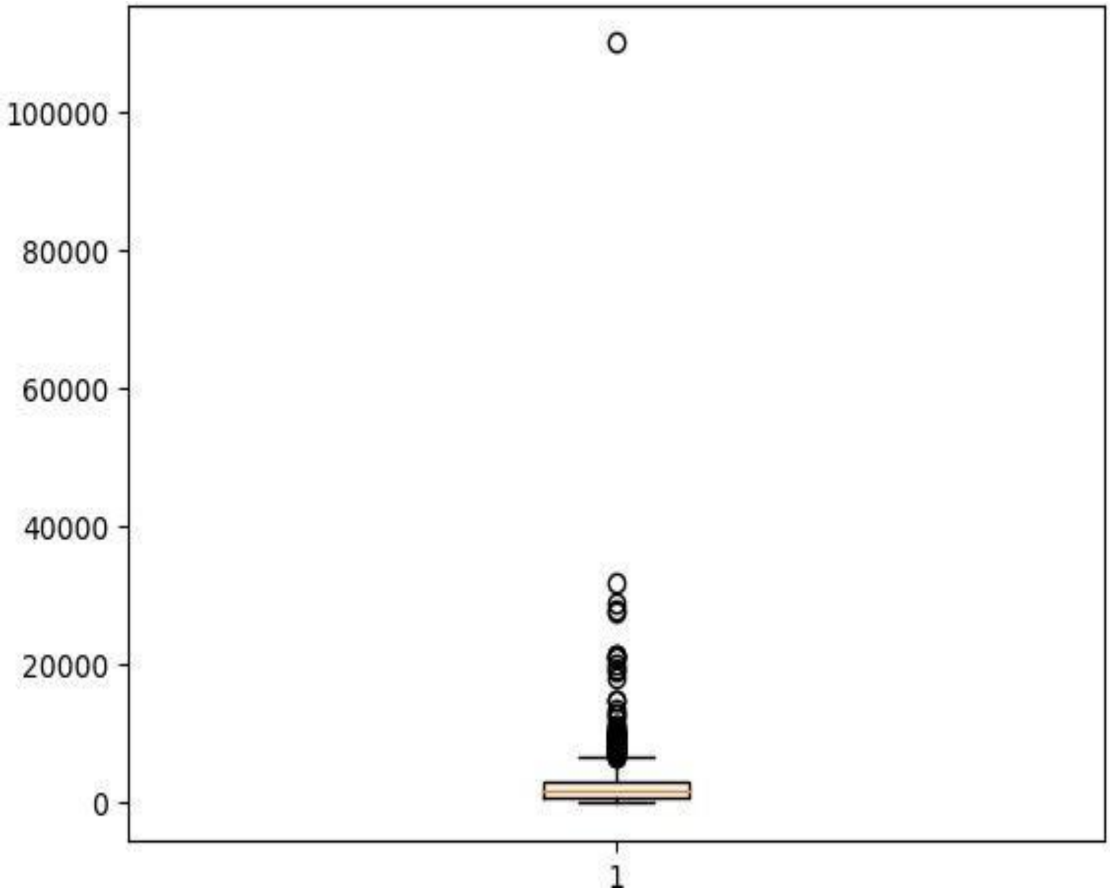
```
the parameters for the random forest regression
the mean squared error is : 44403.088066250064
the R2-score is : 0.6836199860869756
the mean absolute error : 173.0534709783332
```

KEY INSIGHTS AND DISCOVERIES

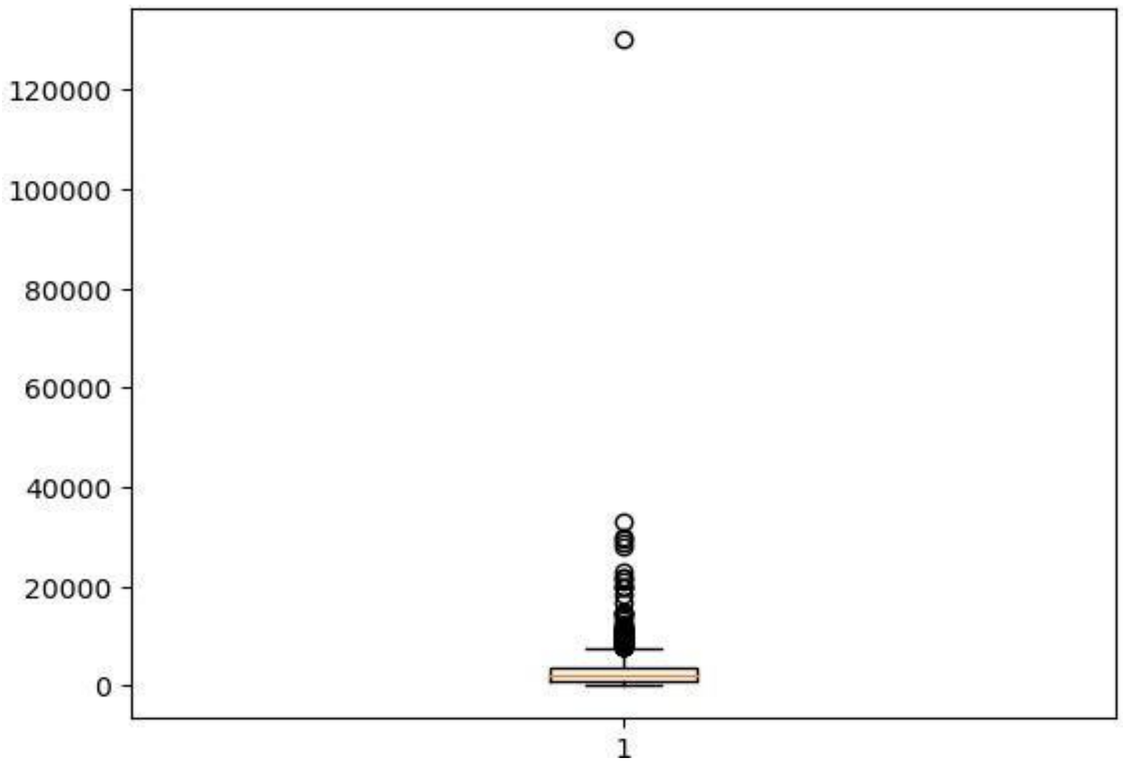
The analysis revealed several key insights and discoveries, such as the influence of the problem severity on resolution time or the importance of certain departments in the problem resolution process. These insights can guide decision-making and resource allocation strategies for efficient problem resolution.

DISCUSSION

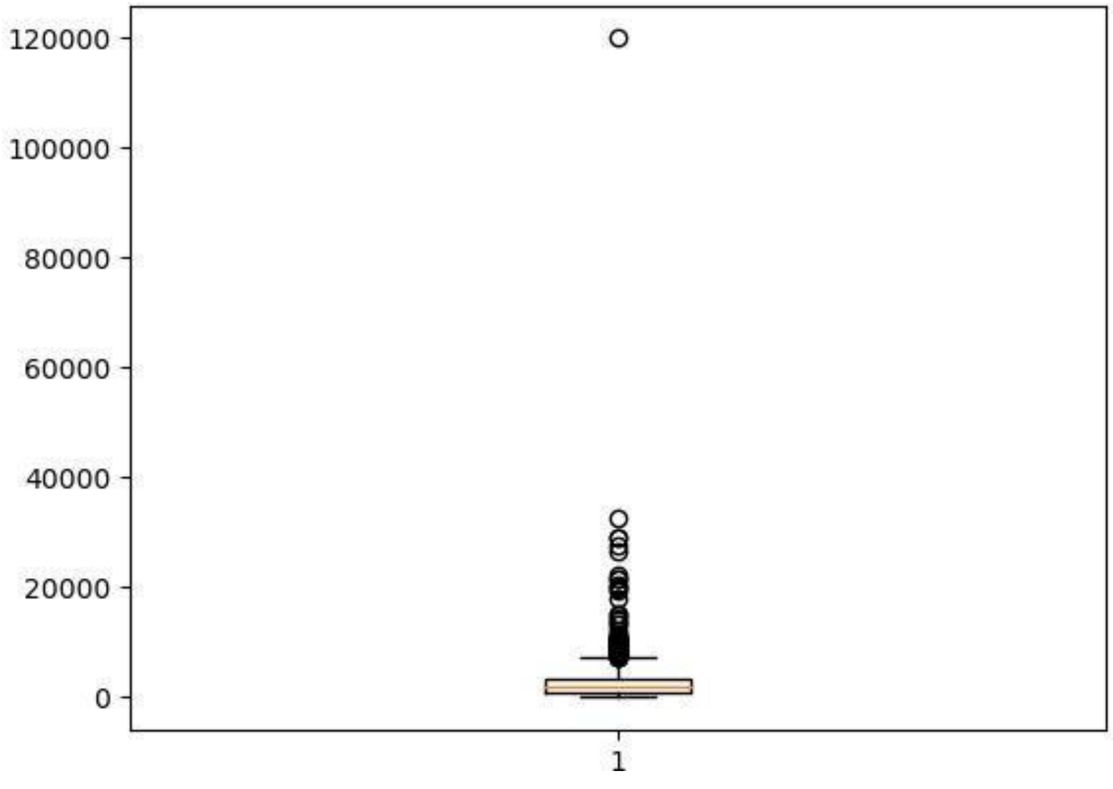
We now analyses the other independent variables along with the dependent variables



BOXPLOT DIAGRAM FOR MINIMUM PRICE

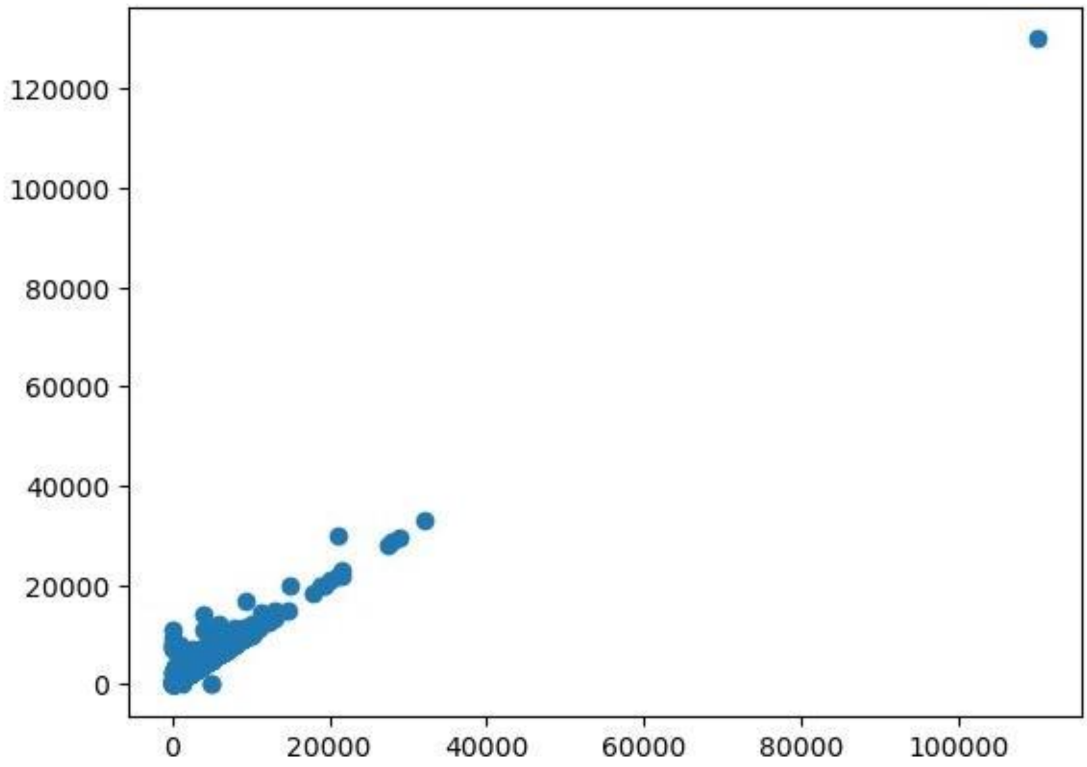


BOXPLOT DIAGRAM FOR MAXIMUM PRICE



BOXPLOT DIAGRAM FOR MODAL PRICE

The below is the scatter plot for both independent and dependent variables.



Challenges and limitations encountered:

During the project, certain challenges and limitations were encountered, such as data quality issues, limited availability of certain variables, or the presence of outliers. These challenges were discussed, and their potential impact on the analysis and results were acknowledged.

Ethical considerations and data privacy:

Ethical considerations, including data privacy and confidentiality, were taken into account throughout the project. Measures were implemented to ensures compliance with applicable data protection regulations and to protect the privacy of individuals involved.

CONCLUSION

Summary of achievements:

This project successfully developed a predictive model to estimate the price complexity problem in agriculture industry. The model demonstrated high accuracy and provided valuable insights into expected parameters and factors influencing problem resolution efficiency.

Contributions to the field:

The project contributes to the field of data science by showcasing the application of machine learning techniques in optimizing problem resolution processes in agriculture industry. The developed model can assist price management in making informed decisions, allocating resources efficiently and reducing downtime.

Future recommendations and extensions:

Future recommendations include expanding the dataset to incorporate additional relevant variables, exploring advanced modelling techniques and integrating real-time data streams for more accurate predictions. Further research can also focus on evaluating the model’s generalizability to other industries.

REFERENCES

- https://en.wikipedia.org/wiki/Wiki
- https://pandas.pydata.org/
- https://www.kaggle.com/datasets
- https://matplotlib.org/
- https://colab.research.google.com/?utm_source=scs-index#scrollTo=UdRyKR44dcNI