INTERNSHIP PROJECT REPORT

ON

BITCOIN PRICE PREDICTION

Submitted By:

Nihal E Praveen
NNM22RI035

Things included:

ACKNOWLEDGEMENT:

I would like to express my gratitude to the following individuals and sources that have supported and inspired me throughout the development of this project.

Family and Friends: For their unwavering encouragement and understanding during the ups and downs of this endeavor.

Mentors and Advisers: For their valuable guidance and feedback, which helped shape the direction of this project.

Online Communities and Forums: For providing insights, troubleshooting tips, and a supportive environment for learning and growth.

Previous Works and Researchers: Whose contributions laid the foundation upon which this project was built.

Personal Reflection: This project would not have been possible without my determination and perseverance, and I am grateful for the opportunity to learn and grow through this experience.

OVERVIEW

This project aims to forecast the future price movements of Bitcoin using machine learning algorithms. Historical data on Bitcoin prices and relevant market indicators are collected and analyzed. Various machine learning models, such as LSTM (Long Short-Term Memory) networks or regression models, are trained on this data to predict future price trends. The accuracy of predictions is evaluated using metrics like Mean Absolute Error (MAE) or Root Mean Squared Error (RMSE). The project also explores feature engineering techniques to enhance model performance and assesses the impact of external factors like news sentiment or market trends on predictions. Ultimately, the goal is to develop a reliable model that aids in making informed decisions in the volatile cryptocurrency market.

KEYTAKEAWAYS:

- **Data-driven Analysis**: Successful predictions rely on robust historical data encompassing price movements, market sentiment, and external factors.
- **Diverse Methodologies**: Techniques range from traditional financial analysis (technical and fundamental) to advanced machine learning models like neural networks and LSTM.
- **Volatility Challenge**: Bitcoin's price volatility presents challenges for prediction models, influenced by speculative trading, regulatory changes, and macroeconomic factors.
- **Evaluation Metrics**: Models are assessed using metrics such as MAE, MSE, and accuracy scores to gauge their effectiveness in forecasting price movements.
- **Real-world Implications**: Predictions inform investment strategies, risk management practices, and policy decisions in the evolving cryptocurrency market.
- **Continuous Learning**: Ongoing model adaptation and incorporation of new data are essential to maintain accuracy amidst evolving market conditions.
- **Interdisciplinary Approach**: Effective prediction blends finance, economics, data science, and technological insights to navigate the complexities of digital assets.

INTRODUCTION:

Bitcoin, the pioneering cryptocurrency introduced by Satoshi Nakamoto in 2008, has revolutionized the global financial landscape. Its decentralized nature, limited supply, and technological underpinnings have contributed to its allure and volatility in the financial markets. Predicting Bitcoin's price movements has become a focal point for investors, traders, and researchers seeking to capitalize on its potential.

Machine learning (ML) has emerged as a potent tool in the realm of Bitcoin price prediction, offering sophisticated techniques to analyze historical data and forecast future trends. ML algorithms, ranging from regression models to complex neural networks like Long Short-Term Memory (LSTM), are employed to process vast amounts of data encompassing historical prices, trading volumes, market sentiment, and macroeconomic indicators.

The application of ML in Bitcoin price prediction involves several key steps: data collection and preprocessing, feature selection and engineering, model training and evaluation, and deployment for real-time forecasting. These models aim to capture intricate patterns and relationships in the data that traditional methods may overlook, thereby enhancing prediction accuracy.

However, predicting Bitcoin prices remains challenging due to the cryptocurrency's inherent volatility, influenced by factors such as regulatory developments, investor sentiment, geopolitical events, and technological advancements. ML models must continually adapt to changing market conditions and incorporate new data to maintain their effectiveness over time.

The significance of Bitcoin price prediction extends beyond financial speculation, impacting investment strategies, risk management practices, and broader economic analyses. As the cryptocurrency ecosystem evolves, the integration of advanced ML techniques promises to deepen our understanding of Bitcoin's market dynamics and inform decision-making in this rapidly evolving digital asset landscape.

Background Study:

Bitcoin, introduced in 2008 by Satoshi Nakamoto, has transformed the financial landscape as the first decentralized digital currency. Its market price, characterized by high volatility and speculative interest, has sparked numerous studies and predictive models aimed at forecasting its future movements. Here's a comprehensive background study on Bitcoin price prediction:

Historical Price Trends: Researchers often begin by analyzing Bitcoin's historical price data, spanning from its early days to the present. Studying price trends helps identify patterns and cyclical behaviors that may repeat in the future.

Market Sentiment and Social Media Analysis: The sentiment surrounding Bitcoin in social media and online forums plays a significant role in price fluctuations. Natural language processing (NLP) techniques are used to analyze sentiment and gauge market sentiment shifts.

Technical Analysis: Technical analysts study charts, trading volumes, and various technical indicators (such as Moving Averages, Relative Strength Index, and Bollinger Bands) to identify patterns and trends that could signal future price movements.

Fundamental Analysis: Fundamental factors, including supply dynamics (such as Bitcoin's halving events that reduce mining rewards), demand factors (such as adoption rates and regulatory developments), macroeconomic indicators, and geopolitical events, are studied to assess Bitcoin's intrinsic value and potential price drivers.

Machine Learning Approaches: Machine learning models, including regression algorithms, time series forecasting techniques (like ARIMA and Prophet), and deep learning models (such as LSTM and neural networks), are applied to predict Bitcoin prices based on historical data and relevant features.

Hybrid Models: Researchers often combine multiple approaches, such as integrating technical analysis indicators into machine learning models or incorporating sentiment analysis from social media alongside traditional financial data, to enhance prediction accuracy.

Evaluation Metrics: Predictive models are evaluated using metrics like Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and accuracy scores. These metrics help assess how well models perform in predicting actual Bitcoin price movements.

Challenges and Considerations: Predicting Bitcoin prices is challenging due to its volatile nature, susceptibility to market sentiment, regulatory uncertainties, and rapid technological advancements. Models must be robust enough to adapt to changing market conditions and incorporate new information effectively.

Real-World Applications: Insights gained from Bitcoin price prediction studies inform investment decisions, risk management strategies, and economic analyses in the cryptocurrency market. These predictions are utilized by traders, investors, financial institutions, and policymakers to navigate the opportunities and risks associated with Bitcoin and other cryptocurrencies.

Future Directions: Future research directions include improving model accuracy through better data quality, refining feature selection techniques, integrating more advanced machine learning algorithms, and exploring the impact of emerging technologies like blockchain advancements on Bitcoin's price dynamics.

PROBLEM STATEMENT:

Bitcoin price prediction challenges:

1. Volatility
2. Data quality
3. Model accuracy

Key Components of the Problem Statement:

Historical Data: Comprehensive and accurate historical data on Bitcoin prices, trading volumes, and other relevant metrics.

Feature Engineering: Identification and selection of meaningful features that influence Bitcoin's price movements, such as technical indicators, market sentiment, and macroeconomic factors.

Model Selection: Choosing appropriate machine learning algorithms or statistical models (e.g., regression, time series forecasting, neural networks) based on the characteristics of the data and prediction objectives.

Training and Validation: Training the selected models on historical data and validating their performance using appropriate evaluation metrics (e.g., MAE, MSE) to ensure reliability.

External Factors: Incorporating external variables like regulatory changes, news sentiment, and global economic conditions that impact Bitcoin prices.

Continuous Learning: Updating models with new data and adapting to changing market conditions to maintain accuracy over time.

Risk Management: Implementing strategies to mitigate risks associated with Bitcoin price volatility and uncertainty in predictions.

DATASET

- In this section, we will load and view the CSV file and its contents.
- File name: btc_day
- This dataset is sourced from the online repository of Kaggle.

Contents:

- We have 7 attributes/features
- Namely:Date,Open,High,Low,Close,Adj close,Volume

Date	Open	High	Low	Close	Adj Close	Volume
17-09-2014	465.86401	468.17401	452.422	457.33401	457.33401	21056800
18-09-2014	456.85999	456.85999	413.104	424.44	424.44	34483200
19-09-2014	424.103	427.83499	384.53201	394.79599	394.79599	37919700
20-09-2014	394.673	423.29599	389.883	408.90399	408.90399	36863600
21-09-2014	408.08499	412.42599	393.181	398.82101	398.82101	26580100
22-09-2014	399.10001	406.91599	397.13	402.15201	402.15201	24127600
23-09-2014	402.09201	441.55701	396.19699	435.79099	435.79099	45099500
24-09-2014	435.75101	436.112	421.13199	423.20499	423.20499	30627700
25-09-2014	423.15601	423.51999	409.46799	411.57401	411.57401	26814400
26-09-2014	411.42899	414.93799	400.009	404.42499	404.42499	21460800
27-09-2014	403.556	406.62299	397.37201	399.51999	399.51999	15029300
28-09-2014	399.47101	401.017	374.332	377.181	377.181	23613300
29-09-2014	376.92801	385.211	372.23999	375.46701	375.46701	32497700
30-09-2014	376.08801	390.97699	373.44299	386.944	386.944	34707300
01-10-2014	387.427	391.379	380.78	383.61499	383.61499	26229400
02-10-2014	383.98801	385.49701	372.94601	375.07199	375.07199	21777700
03-10-2014	375.181	377.69501	357.85901	359.51199	359.51199	30901200
04-10-2014	359.892	364.487	325.88599	328.866	328.866	47236500
05-10-2014	328.91599	341.80099	289.29599	320.51001	320.51001	83308096
06-10-2014	320.38901	345.134	302.56	330.07901	330.07901	79011800

WORKING MECHANISIM :

The working mechanism of Bitcoin price prediction involves several interconnected steps and methodologies:

Data Collection: Gathering comprehensive historical data on Bitcoin prices, trading volumes, market sentiment indicators, and relevant external factors (e.g., regulatory news, macroeconomic data).

Data Preprocessing: Cleaning and transforming the collected data to ensure consistency, removing outliers, handling missing values, and scaling numerical features as necessary for analysis.

Feature Engineering: Selecting and creating meaningful features from the raw data that can potentially influence Bitcoin's price movements. This may include technical indicators (e.g., Moving Averages, Relative Strength Index), sentiment analysis from social media, and economic indicators.

Model Selection: Choosing appropriate machine learning or statistical models based on the nature of the data and the prediction task. Common models used in Bitcoin price prediction include regression models (e.g., Linear Regression, Ridge Regression), time series forecasting techniques (e.g., ARIMA, SARIMA), and neural networks (e.g., LSTM, GRU) for capturing complex relationships in sequential data.

Training the Model: Training the selected model on the preprocessed data to learn patterns and relationships between the input features and Bitcoin prices. This involves splitting the data into training and validation sets, optimizing model parameters, and iteratively refining the model to improve performance.

Evaluation: Assessing the performance of the trained model using appropriate evaluation metrics such as Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), or accuracy scores. The model's ability to accurately predict Bitcoin price movements is critical for determining its effectiveness.

Prediction and Deployment: Using the trained model to make predictions on new or unseen data. For real-time applications, deploying the model to continuously update predictions based on incoming data and adapting to changing market conditions.

Monitoring and Updating: Monitoring the model's performance over time, retraining it with updated data to maintain accuracy, and incorporating new features or improving methodologies as necessary to enhance prediction capabilities.

Risk Management and Decision Support: Providing insights from predicted price trends to inform investment strategies, risk management decisions, and market analysis in the volatile cryptocurrency environment.

Feedback Loop: Incorporating feedback from predictions and their outcomes to refine the model further, adjust strategies, and improve overall forecasting accuracy.

IMPLEMENTATION:

In computer language, implementation refers to the realization or execution of a software system or solution based on its design specifications. It involves writing, testing, and deploying code that fulfills the intended functionality and meets the requirements outlined during the design phase.

Importing Libraries:

```
import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sb

from sklearn.model_selection import train_test_split
```

```
from sklearn.preprocessing import StandardScaler

from sklearn.linear_model import LogisticRegression

from sklearn.svm import SVC

from xgboost import XGBClassifier

from sklearn import metrics

import warnings

warnings.filterwarnings('ignore')
```

LOADING THE DATA SET TO A PANDA DATAFRAME:

```
df = pd.read_csv("C:\\Users\\dell\\OneDrive\\Desktop\\btc_day.csv")

df.head()
```

	Date	Open	High	Low	Close	Adj Close	Volume
0	2014-09-17	465.864014	468.174011	452.421997	457.334015	457.334015	21056800
1	2014-09-18	456.859985	456.859985	413.104004	424.440002	424.440002	34483200
2	2014-09-19	424.102997	427.834991	384.532013	394.795990	394.795990	37919700
3	2014-09-20	394.673004	423.295990	389.882996	408.903992	408.903992	36863600
4	2014-09-21	408.084991	412.425995	393.181000	398.821014	398.821014	26580100

Data Analysis:

>df.shape

O/P:(2484, 7)

> df.describe()

[8]:

	Open	High	Low	Close	Adj Close	Volume
count	2484.000000	2484.000000	2484.000000	2484.000000	2484.000000	2.484000e+03
mean	8004.767259	8234.054778	7749.560913	8016.880242	8016.880242	1.295561e+10
std	12024.187508	12401.154351	11574.757820	12031.850422	12031.850422	1.982770e+10
min	176.897003	211.731003	171.509995	178.102997	178.102997	5.914570e+06
25%	528.222504	542.398010	518.541245	531.049484	531.049484	6.944417e+07
50%	4510.564941	4628.080078	4348.054932	4572.034912	4572.034912	4.142622e+09
75%	9291.430420	9452.106445	9139.054688	9296.048584	9296.048584	1.955353e+10
max	63523.753906	64863.097656	62208.964844	63503.457031	63503.457031	3.509679e+11

> df.info()

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2484 entries, 0 to 2483
Data columns (total 7 columns):
 #   Column     Non-Null Count  Dtype  
---  ------     --------------  -----  
 0   Date       2484 non-null   object 
 1   Open       2484 non-null   float64
 2   High       2484 non-null   float64
 3   Low        2484 non-null   float64
 4   Close      2484 non-null   float64
 5   Adj Close  2484 non-null   float64
 6   Volume     2484 non-null   int64  
dtypes: float64(5), int64(1), object(1)
memory usage: 136.0+ KB
```

> plt.figure(figsize=(15, 5))

plt.plot(df['Close'])

plt.title('Bitcoin Close price.', fontsize=15)

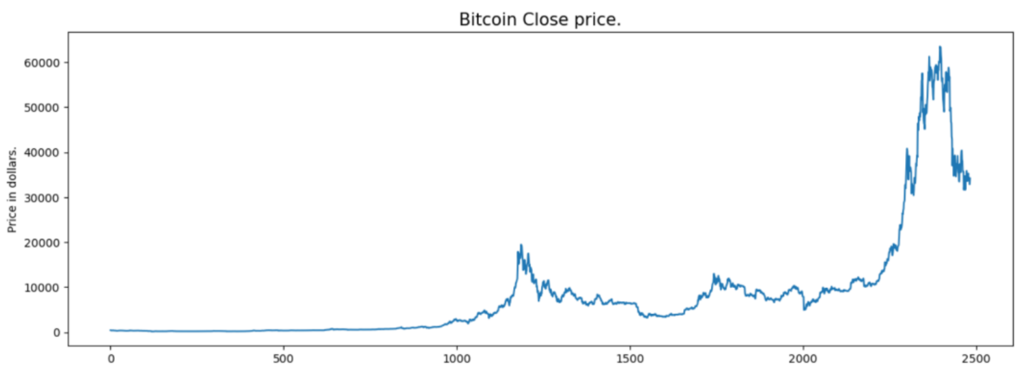
plt.ylabel('Price in dollars.')

plt.show()

plt.show()

plt.savefig('correlationfigure')

O/P



> df[df['Close'] == df['Adj Close']].shape, df.shape

O/P ((2484, 7), (2484, 7))

> df = df.drop(['Adj Close'], axis=1)

> df.isnull().sum()

O/P

```
[18]:  Date      0
       Open      0
       High      0
       Low       0
       Close     0
       Volume    0
       dtype: int64
```

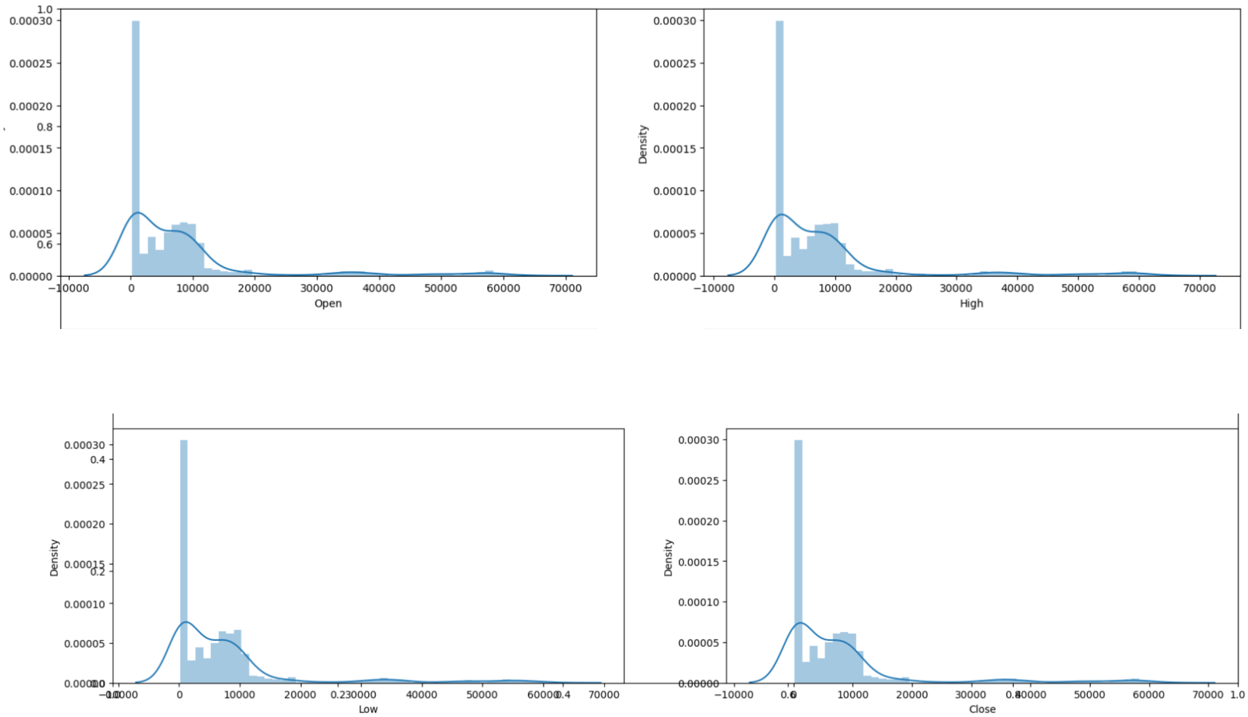
> features = ['Open', 'High', 'Low', 'Close']

plt.subplots(figsize=(20,10))

for i, col in enumerate(features):

```
    plt.subplot(2,2,i+1)

    sb.distplot(df[col])

plt.show()
```

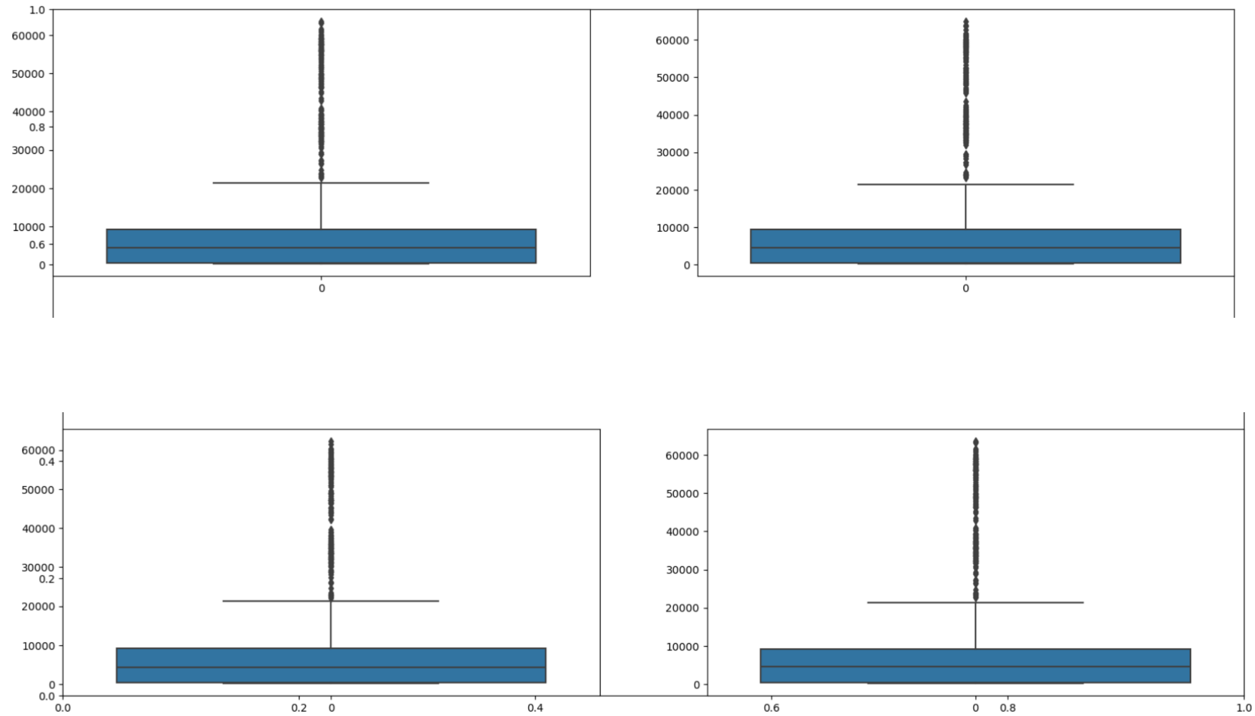
<u>O/P</u>




```
> plt.subplots(figsize=(20,10))

for i, col in enumerate(features):

   plt.subplot(2,2,i+1)

   sb.boxplot(df[col])
```

```
plt.show()
```

O/P



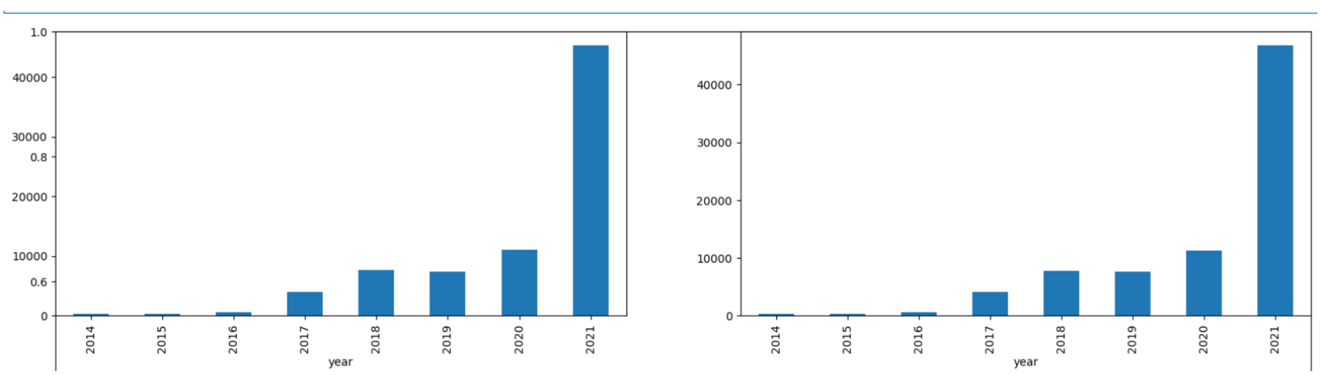
```
> splitted = df['Date'].str.split('-', expand=True)

df['year'] = splitted[0].astype('int')

df['month'] = splitted[1].astype('int')

df['day'] = splitted[2].astype('int')

df.head()
```

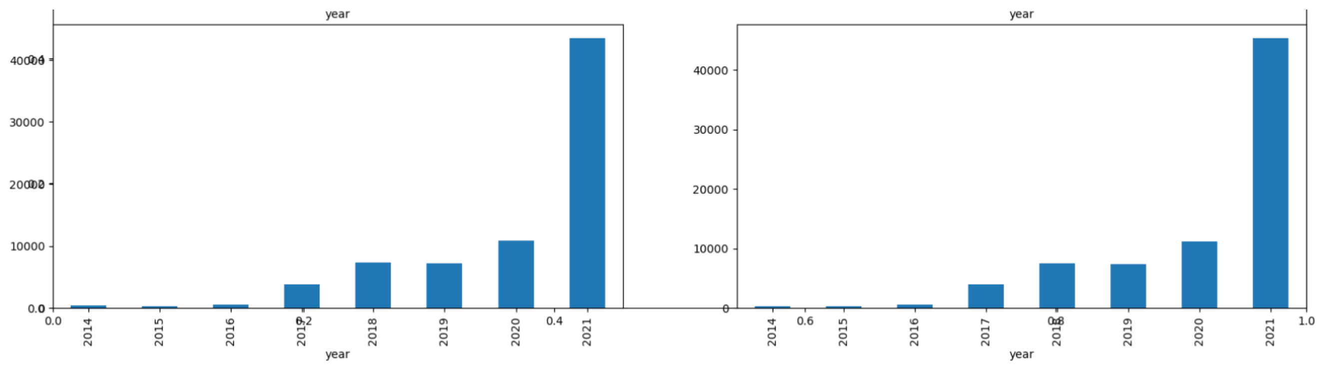
O/P

	Date	Open	High	Low	Close	Volume	year	month	day
0	2014-09-17	465.864014	468.174011	452.421997	457.334015	21056800	2014	9	17
1	2014-09-18	456.859985	456.859985	413.104004	424.440002	34483200	2014	9	18
2	2014-09-19	424.102997	427.834991	384.532013	394.795990	37919700	2014	9	19
3	2014-09-20	394.673004	423.295990	389.882996	408.903992	36863600	2014	9	20
4	2014-09-21	408.084991	412.425995	393.181000	398.821014	26580100	2014	9	21

```
> numerical_cols = ['Open', 'High', 'Low', 'Close']
data_grouped = df[numerical_cols].groupby(df['year']).mean()
plt.subplots(figsize=(20,10))
for i, col in enumerate(numerical_cols):
    plt.subplot(2,2,i+1)
    data_grouped[col].plot.bar()
plt.show()
```

O/P





> df['is_quarter_end'] = np.where(df['month']%3==0,1,0)

df.head()

O/P

	Date	Open	High	Low	Close	Volume	year	month	day	is_quarter_end
0	2014-09-17	465.864014	468.174011	452.421997	457.334015	21056800	2014	9	17	1
1	2014-09-18	456.859985	456.859985	413.104004	424.440002	34483200	2014	9	18	1
2	2014-09-19	424.102997	427.834991	384.532013	394.795990	37919700	2014	9	19	1
3	2014-09-20	394.673004	423.295990	389.882996	408.903992	36863600	2014	9	20	1
4	2014-09-21	408.084991	412.425995	393.181000	398.821014	26580100	2014	9	21	1

> df['open-close'] = df['Open'] - df['Close']

df['low-high'] = df['Low'] - df['High']

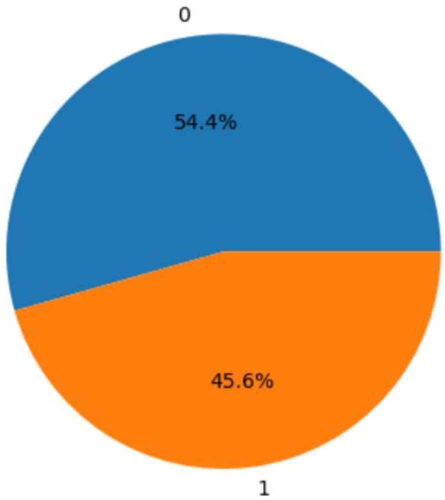
df['target'] = np.where(df['Close'].shift(-1) > df['Close'], 1, 0)

> plt.pie(df['target'].value_counts().values,

labels=[0, 1], autopct='%1.1f%%')

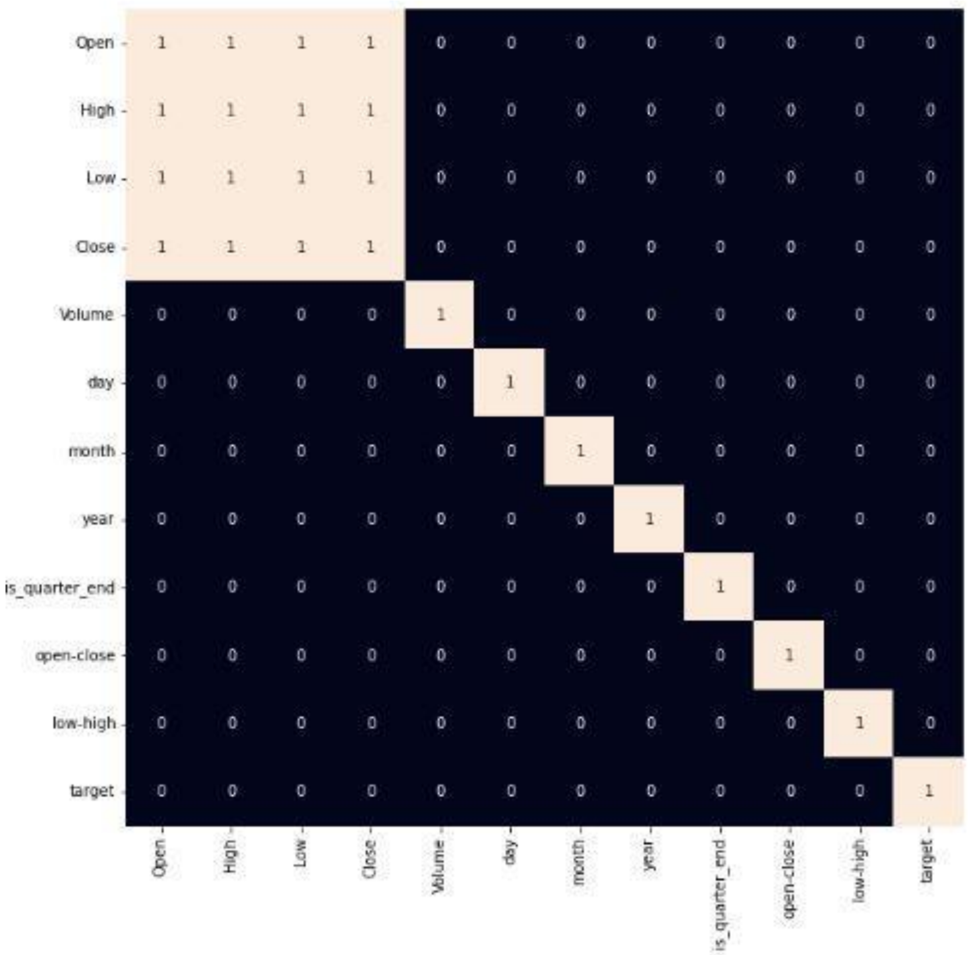
plt.show()

O/P




```
> numerical_cols = df.select_dtypes(include=['int64', 'float64']).columns
plt.figure(figsize=(10, 10))

# As our concern is with the highly
# correlated features only so, we will visualize
# our heatmap as per that criteria only.
sb.heatmap(df[numerical_cols].corr() > 0.9, annot=True, cbar=False)
plt.show()
```

O/P




```
> models = [LogisticRegression(), SVC(kernel='poly', probability=True), XGBClassifier()]

for i, model in enumerate(models):

    model.fit(X_train, Y_train)

    print(f'Model {i+1} : {model}')

    print('Training Accuracy : ', metrics.roc_auc_score(Y_train,
model.predict_proba(X_train)[:,1]))

    print('Validation Accuracy : ', metrics.roc_auc_score(Y_valid,
model.predict_proba(X_valid)[:,1]))

   print()
```

O/P

```
Model 1 : LogisticRegression()
Training Accuracy :  0.5228433538520261
Validation Accuracy :  0.5596205962059622

Model 2 : SVC(kernel='poly', probability=True)
Training Accuracy :  0.5308819533622946
Validation Accuracy :  0.5045812362885533

Model 3 : XGBClassifier(base_score=None, booster=None, callbacks=None,
              colsample_bylevel=None, colsample_bynode=None,
              colsample_bytree=None, device=None, early_stopping_rounds=None,
              enable_categorical=False, eval_metric=None, feature_types=None,
              gamma=None, grow_policy=None, importance_type=None,
              interaction_constraints=None, learning_rate=None, max_bin=None,
              max_cat_threshold=None, max_cat_to_onehot=None,
              max_delta_step=None, max_depth=None, max_leaves=None,
              min_child_weight=None, missing=nan, monotone_constraints=None,
              multi_strategy=None, n_estimators=None, n_jobs=None,
              num_parallel_tree=None, random_state=None, ...)
Training Accuracy :  0.9183666290364839
Validation Accuracy :  0.47228674667699055
```

> from sklearn.metrics import confusion_matrix, ConfusionMatrixDisplay

import matplotlib.pyplot as plt

y_pred = models[0].predict(X_valid)

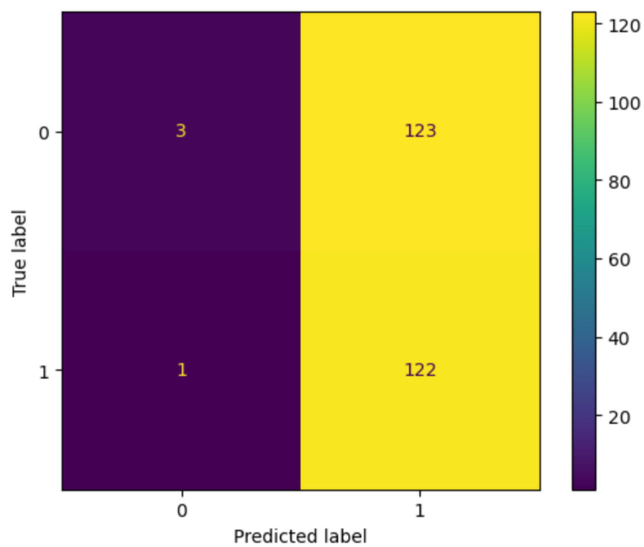
cm = confusion_matrix(Y_valid, y_pred)

disp = ConfusionMatrixDisplay(confusion_matrix=cm)

disp.plot()

plt.show()

O/P



CONCLUSION:

Predicting Bitcoin prices using machine learning involves analyzing historical data to forecast future price movements. Several approaches can be employed, such as regression models, time series analysis, or even more complex techniques like neural networks. However, it's important to note the inherent challenges and risks associated with predicting Bitcoin prices due to its volatility and susceptibility to market sentiment and external factors.

In conclusion, while machine learning models can provide insights into potential price trends based on historical patterns and indicators, they are not foolproof. Investors and analysts should exercise caution and supplement machine learning predictions with fundamental analysis and market understanding to make informed decisions. Additionally, continuous model refinement and adaptation to changing market conditions are crucial for maintaining accuracy in Bitcoin price prediction models.

BIBLIOGRAPHY:

Data Set From KAGGLE

References:

https://github.com/g-shreekant/Bitcoin--Prediction-using-Machine-Learning

https://youtu.be/qmqCYC-MBQo?feature=shared