**PROJECT REPORT TITLE**

**Bitcoin Future Cast**

**A**

**Project Report**

**Submitted**

**in the partial fulfillment**

**for the award of degree of**

**Master of Computer Applications**

**(2023-2025)**

***Undertaken at***

**Solitaire Infosys Pvt. Ltd.  
E-261, Industrial Area, Phase 8B,  
Sector 74, Mohali, Punjab – 160055**

***Submitted by***

**Priyanka**

***Under Supervision of***

***Internal Supervisor(s) External Supervisor***

***Arun Bansal Name***

***Designation Designation***

***to***

***Department of Computer Science***

***Punjabi University, Patiala - 147 002***

***May, 2025***

***Letter Head of Organization***

**Dated: 23-04-2025**

***CERTIFICATE***

It is certified that the project entitled **“Bitcoin Future Cast**” is submitted in partial fulfillment of the requirement for the degree of **Master of Computer Applications (MCA)** in the department of Computer Science, Punjabi University, Patiala. This work has been done by **Priyanka** a bonafide student of the department, in **Solitaire Infosys Pvt. Ltd.** organization from **January 2025** to **April 2025** under my supervision.

**Project Supervisor**

**Letter Head of the Department of Computer Science**

**Dated: 23-04-2025**

***CANDIDATE DECLARATION***

This is to certify that the project entitled **“Bitcoin Future Cast**” is my own work, carried out in **Solitaire Infosys Pvt. Ltd**. organization from **January 2025** to **April 2025**, under the external guidance of \_\_\_\_\_\_\_\_\_\_\_ and internal supervision of \_\_\_\_\_\_\_\_\_\_.

Priyanka

***CERTIFICATE***

It is certified that the project entitled **“Bitcoin Future Cast**” is submitted in partial fulfillment of the requirement for the degree of Master of Computer Applications in the Department of Computer Science, Punjabi University, Patiala. This work has been done by **Priyanka**, a bonafide student of the Department.

This work is fit for the consideration of award of the said degree to her/him.

Internal Supervisor(s)

Traning In-charge



# TABLE OF CONTENTS

|  |  |  |
| --- | --- | --- |
| **SR.NO** | **CONTENTS** | **PAGE NO.** |
| 1 | CHAPTER 1: INTRODUCTION TO THE ORGANIZATION  1.1 Organization Overview  1.1.1 Mission and Vision  1.1.2 Core Values  1.1.3 Company History  1.1.4 Client Companies | 1-3 |
| 2 | Introduction:  2.1 Purpose  2.2 Intended Audience  2.3 Product Scope  2.4 Project Features  2.5 **System Requirements & Feasibility** | 4-9 |
| 3 | System Development:  3.1 Project Management  3.2 System Analysis and Design  3.2.1 Detailed DFDs and Structure Diagrams  3.3 Data Structure | 10-15 |
| 4 | 4.1Coding  4.2 Snapshots of System |  |
|  |  |  |
| 5 | 5 Testing  5.1 Test Plan  5.2 SQL Queries Testing | 16-25 |
| 6 | Project Legacy:  6.1 Current Status  6.2 Problems Faced  6.3 Limitations  6.4 Future Enhancements  6.5 Conclusion | 26-27 |
| 7 |  |  |

# 1. INTRODUCTION TO THE ORGANIZATION

# 1.1 Organization Overview

### Solitaire Infosys Pvt. Ltd. delivers cost-effective and futuristic IT solutions. We develop robust applications tailored to our clients’ needs, ensuring complete satisfaction and improved business outcomes. Our expert developers have extensive experience in online tests, security management, ERP systems, and more.

### About Us: Established in Mohali in 2011, our mission is to be the best IT service provider worldwide. We continuously upgrade our skills to deliver high-quality applications that drive ROI.

### Our Expertise: We assist enterprises and startups in building customer-centric digital products for mobile and web. Our services include UX & UI Design, Mobile Applications, Web Development, Enterprise Applications, Digital Marketing, and Support & Maintenance.

### Corporate Offices: Mohali, India: C-110, Industrial Area, Phase-VII, Mohali | Phone: +91-987-665-6700 | Email: info@slinfy.com Patiala, India: SCO 8-9, FF, Factory Area, Near Hotel Flyover, Patiala (Pb) 147001 | Phone: +91-9915749931 | Email: slinfypta@gmail.com1.1.1 Mission and Vision

****

**1.1.2 Core Values**

Integrity: Upholding ethical practices and transparency.

Innovation: Continuously exploring new ideas and technologies.

Customer Focus: Putting client needs at the forefront.

Excellence: Striving for high quality in every project.

Collaboration: Fostering teamwork and shared success.

**1.1.3 Company History**

Founded in 2011 in Mohali, Solitaire Infosys has grown from a small startup into a global IT service provider. Our journey includes successful projects in ERP, security management, and web development.

**1.1.4 Client Companies**

Solitaire Infosys has collaborated with numerous esteemed clients, delivering exceptional IT solutions tailored to their needs.

****

# *Bitcoin Cast Future Logo*

# 

# 

# INTRODUCTION

## 2.1 PURPOSE

## The purpose of this project is to develop a Bitcoin price prediction and visualization system using historical data and machine learning models. It aims to help users, researchers, and investors understand Bitcoin market trends and forecast future prices. The project integrates data analysis, machine learning, and visualization tools for a comprehensive analytics solution

## 2.2 INTENDED AUDIENCE

This project is intended for:

* **Students and Data Science Enthusiasts** seeking to learn predictive modeling and data visualization.
* **Trainers and Faculty Members** for academic evaluation.
* **Internship Mentors and Evaluators** at Solitaire Infosys.
* **Business Analysts and Investors** interested in Bitcoin price trends and predictions.

## 2.3 PRODUCT SCOPE

The scope of this product includes:

* Importing and cleaning historical Bitcoin price data.
* Performing exploratory data analysis (EDA).
* Applying machine learning models (e.g., Linear Regression, Random Forest) for prediction.
* Storing predicted data in a SQL Server database.
* Building an interactive Power BI dashboard for visualization.
* Providing accuracy reports and performance insights.

## 2.4 PROJECT FEATURES

 Automated Data Cleaning and Preprocessing

 Prediction of Future Bitcoin Prices

 Visualization with Matplotlib, Seaborn, and Power BI

 Storage and Retrieval from SQL Server

 Easy-to-understand Graphical Reports

 Export of Predicted Results in CSV Format

## ****2.5 System Requirements & Feasibility****

### ****2.5.1 Problem Statement****

The core problem addressed by this project is the **prediction of future Bitcoin prices** and **analysis of historical price trends** using a data-driven approach. Given Bitcoin's high volatility and its sensitivity to external factors such as regulatory changes, market sentiment, and global news, traditional forecasting methods fall short. Hence, there is a need for a **robust and adaptive solution** that leverages deep learning and time-series forecasting.

### ****2.5.2 Functions to be Provided****

The proposed system will offer the following functionalities:

* **Data Loading and Preprocessing:**  
  Load historical Bitcoin price data from CSV files and perform preprocessing tasks including handling missing values, detecting and removing outliers, feature engineering (e.g., moving averages), and scaling.
* **Model Training and Prediction:**  
  Implement and train an **LSTM (Long Short-Term Memory)** model on historical data to learn patterns and dependencies. Generate future Bitcoin price predictions and assess performance using metrics like **RMSE**.
* **Visualization and Analysis:**  
  Provide graphical representations of historical data, predictions, and price volatility using charts, histograms, and correlation matrices. Support deeper insights into trends and model behavior.
* **Data Export:**  
  Export predicted price data and other analytics results in user-friendly formats such as **CSV**, facilitating further use in external tools (e.g., Power BI).

### ****2.5.3 Processing Environment (Hardware/Software)****

* **Hardware Requirements:**
  + Processor: Intel Core i5 or higher
  + RAM: Minimum 8 GB
  + Storage: Minimum 256 GB
* **Primary Processing Environment:**
  + **Google Colab** (utilizing free GPU access for model training)
* **Software Requirements:**
  + Python (with libraries: pandas, NumPy, matplotlib, seaborn, scikit-learn, TensorFlow/Keras)
  + Microsoft SQL Server (for structured data storage and query execution)
  + Power BI Desktop (for creating dashboards and reports)
  + Visual Studio Code / Jupyter Notebook
  + Operating System: Windows 10 or later

### ****2.5.4 Solution Strategy****

The system leverages a time-series forecasting approach using LSTM neural networks. The solution pipeline involves:

1. Loading and preprocessing Bitcoin historical data.
2. Creating additional predictive features (e.g., lag values, technical indicators).
3. Training an LSTM model and evaluating its performance.
4. Visualizing outcomes and insights for end-users. This is an **iterative process** that allows for refinement and improvement over time based on model performance and market changes.

### ****2.5.5 Acceptance Criteria****

* **Model Accuracy:**  
  Targeting a **Root Mean Square Error (RMSE)** below a pre-defined threshold (e.g., 500 USD).
* **Prediction Horizon:**  
  Forecast Bitcoin prices for the **next 15 days** based on the most recent available data.
* **Data Visualization:**  
  Graphs and charts should be **clear, informative, and interactive** where applicable.
* **System Usability:**  
  The system should be **user-friendly**, with easy data loading, training, and interpretation of results without requiring in-depth technical expertise.

## ****2.5.6 Feasibility****

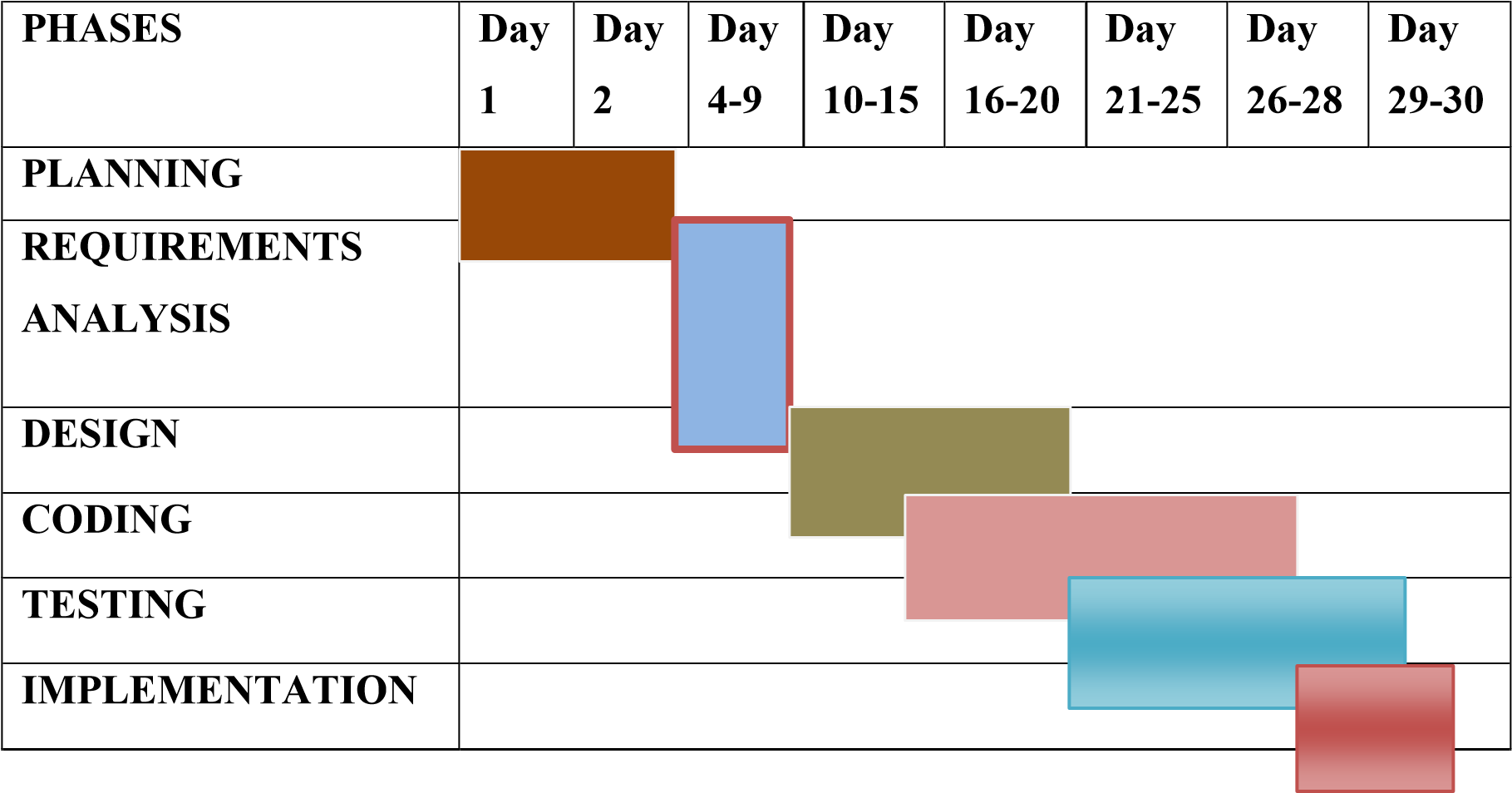
* **Technical Feasibility:**  
  The project utilizes **widely supported platforms** like Python, SQL Server, and Power BI. The availability of cloud resources (Google Colab) makes it feasible to train complex models efficiently.
* **Economic Feasibility:**  
  It primarily uses **open-source tools** and **freely available resources**, significantly reducing costs. No expensive software licenses or hardware upgrades are necessary.
* **Operational Feasibility:**  
  The solution is **simple to operate**, deployable on local systems or Google Colab, and requires only moderate computing resources. The system supports users in model training, prediction generation, and visualization seamlessly.

## ****2.5.7 System Requirements****

* **Input:**  
  Bitcoin historical price dataset in CSV format (daily data from 2018 to 2025).
* **Processing:**  
  Data cleaning, feature engineering, LSTM-based time series modeling, prediction, and accuracy analysis.
* **Output:**
  + CSV file containing predicted prices.
  + Visual charts representing trends and model evaluation.
  + SQL database tables with structured output.
  + Power BI dashboard for intuitive business insights.

**3. SYSTEM DEVELOPMENT**

# 3.1 Project Management: Gantt Chart



**3.1 Project Management**

Effective project management is essential for the successful completion of this data science and analytics project. The process followed includes planning, scheduling, monitoring, and documentation.

### ****Project Objectives****

* To analyze historical Bitcoin prices.
* To develop a machine learning-based predictive model.
* To visualize the trends and predictions using Power BI.
* To store and manage data via SQL Server.

### ****Project Phases****

1. **Requirement Gathering**
   * Discussed goals, datasets, tools, and expectations.
2. **Design**
   * Designed DFDs, database schema, and visual layout.
3. **Development**
   * Python scripts for data processing and model training.
4. **Testing**
   * Unit testing for functions, and performance testing of models.
5. **Implementation**
   * Deployed database and linked with Power BI for dashboards.
6. **Documentation**
   * Prepared user guide, technical documentation, and project report.

### ****Tools Used****

* **Project Planning**: Trello, Google Sheets
* **Version Control**: GitHub (for code backup)
* **Documentation**: MS Word, PDF

**3.2 SYSTEM ANALYSIS AND DESIGN**

## ****3.2.1 Detailed DFDs and Structure Diagrams****

## This represents the system as a single process and shows its interaction with external entities.

**External Entities:**

* **User**: Provides the dataset and views predictions/visualizations.
* **CSV File System**: Source of historical Bitcoin price data.
* **SQL Server Database**: Stores cleaned data and prediction results.

**Main Process:**

* **Bitcoin Price Prediction System**

**Data Flows:**

* **Input**: CSV dataset (historical data)
* **Output**: Predictions, visualizations, exported CSV, and database entries

### ****3.2.2 Level 1 DFD****

Breaks the main process into sub-processes:

**1.0 Data Loading & Preprocessing**

* Load CSV, clean data (nulls/outliers), normalize features, add technical indicators.

**2.0 Model Training**

* Train LSTM model using time-series features.

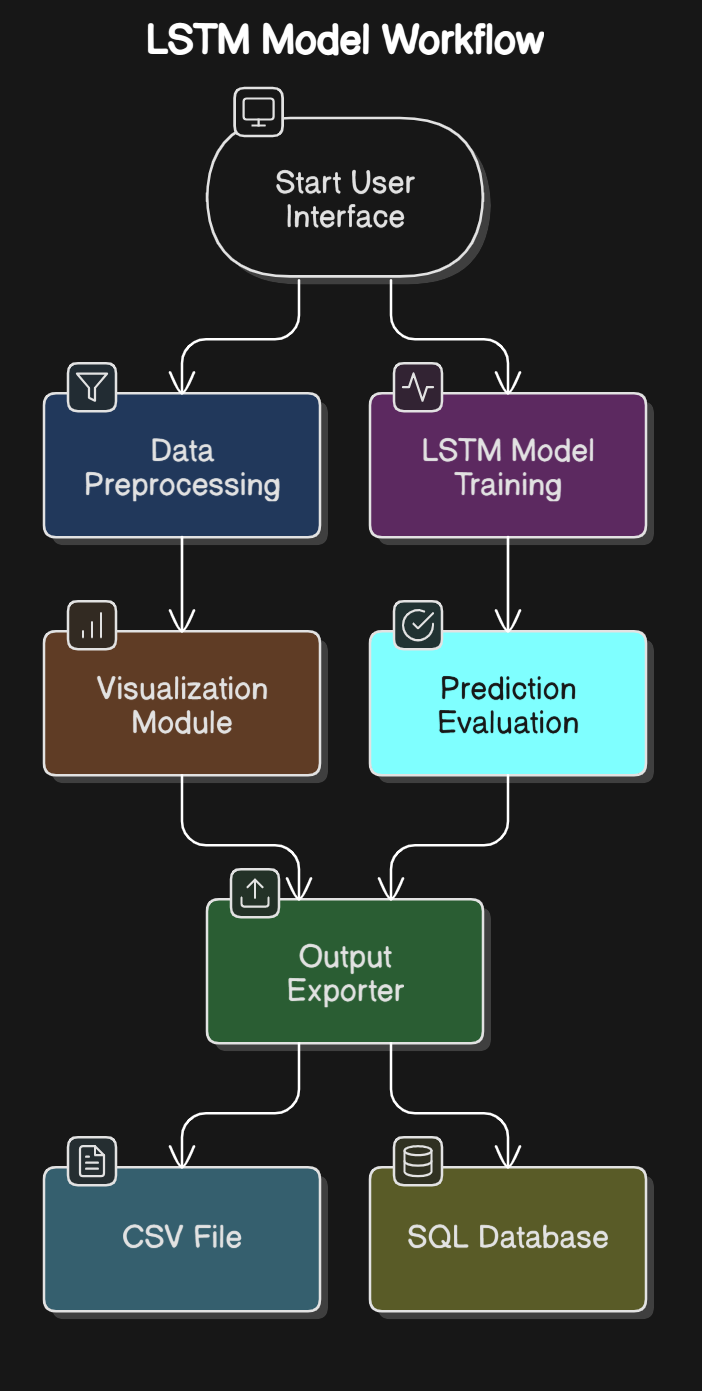
**3.0 Prediction & Evaluation**

* Generate future predictions, calculate performance metrics (RMSE, MAE).

**4.0 Visualization**

* Plot historical and predicted prices, model evaluation metrics, and volatility.

**5.0 Exporting**

* Export results to a CSV file and insert into SQL Server database

### ****1. Pandas DataFrame****

Used to store and manipulate historical and processed Bitcoin price data.

import pandas as pd

df = pd.read\_csv('btc\_1d\_data\_2018\_to\_2025.csv')

**Important Columns and Data Types:**

* Date: date time
* Open, High, Low, Close, Volume: float

### ****2. Numpy Arrays****

Used for reshaping and feeding sequential data into the LSTM model.

import numpy as np

X\_train = np.array([...])

y\_train = np.array([...])

### ****3. Dictionary****

Used to store model performance metrics and configuration parameters.

metrics = {

'RMSE': 154.23,

'MAE': 120.56

}

### ****4. SQL Table Rows****

Each row represents a record of actual and predicted Bitcoin closing prices.

(Date DATE, Actual\_Close FLOAT, Predicted\_Close FLOAT)

## ****Database and File Specifications****

### ****1. CSV Files****

**Input File**: btc\_1d\_data\_2018\_to\_2025.csv

* **Format**: CSV
* **Columns**: Date, Open, High, Low, Close, Volume

**Output File**: predicted\_btc\_prices.csv

* **Format**: CSV
* **Columns**: Date, Actual\_Close, Predicted\_Close

### ****2. SQL Server Database****

**Database Name**: BitcoinDB  
**Table Name**: BitcoinPrices

CREATE TABLE BitcoinPrices (

Date DATE PRIMARY KEY,

Actual\_Close FLOAT,

Predicted\_Close FLOAT

);

**Operations:**

* INSERT INTO BitcoinPrices ... (to store predicted data)
* SELECT \* FROM BitcoinPrices ... (to fetch data for dashboards)

### ****3. Power BI Integration****

* **Source**: SQL Server Table or Exported CSV

### ****System Architecture****

### ****1. Start Data Source****

* **Purpose**: Begin the process by identifying and loading the source of the data.
* **Component**:
  + CSV File with Historical Bitcoin Data
* **Details**:  
  Historical Bitcoin prices (date-wise) are stored in a CSV file format which includes attributes like Date, Open, High, Low, Close, Volume, etc.

### 🧹 ****2. Data Preprocessing Layer****

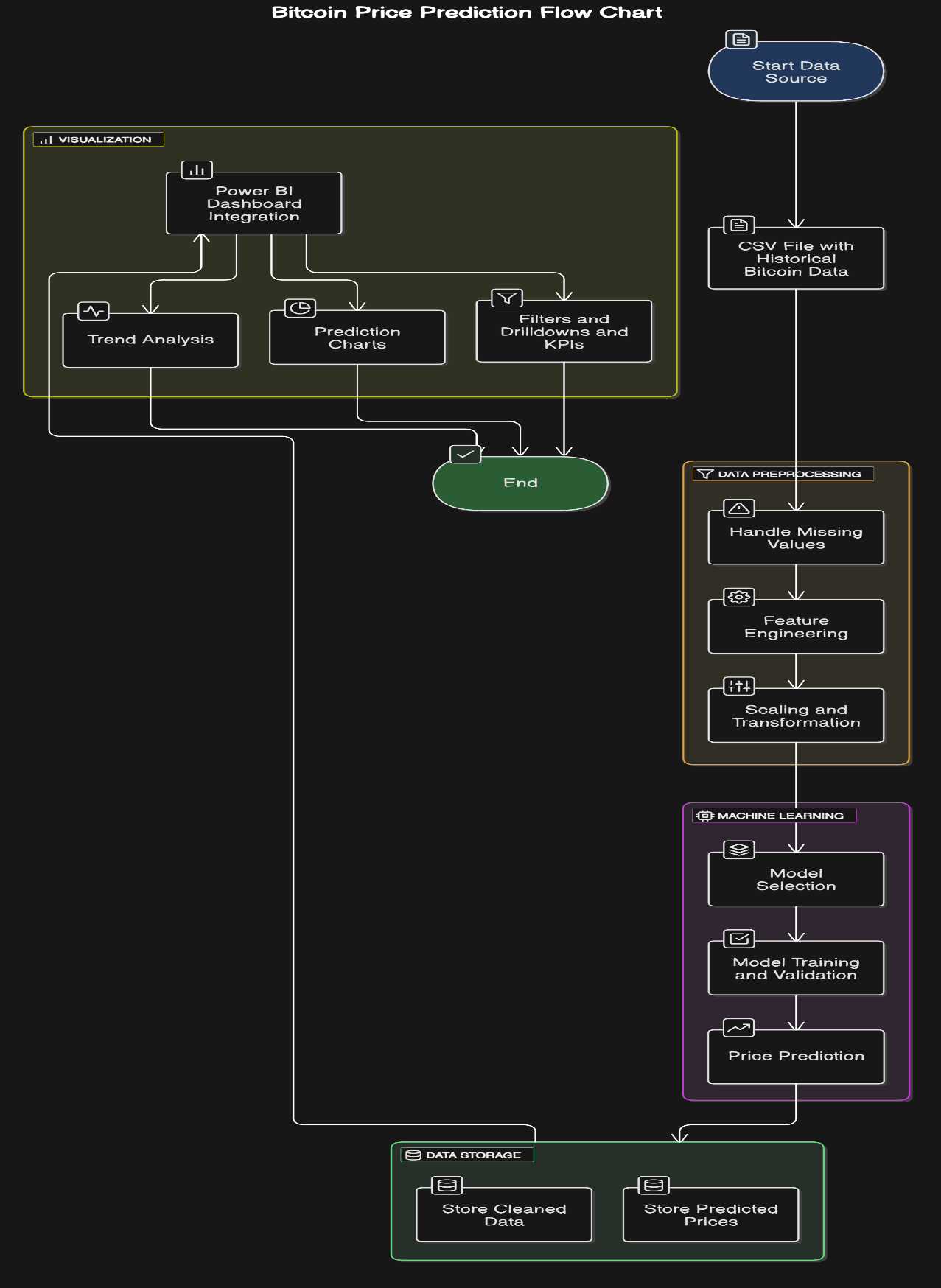
* **Goal**: Prepare raw data for model training by making it clean and consistent.
* **Components**:
* **Handle Missing Values**: Detect and fill or remove missing/NULL values.

**Feature Engineering**: Create new features (e.g., Moving Average, Day Difference) to improve prediction accuracy.

* + **Scaling and Transformation**: Normalize or standardize data for uniformity and improved model performance.

### ****3. Machine Learning Layer****

* **Objective**: Use predictive modeling to forecast future Bitcoin prices.
* **Steps**:
  + **Model Selection**: Choose algorithms like Linear Regression or Random Forest based on the nature of the data.
  + **Model Training and Validation**: Train the model using historical data, validate it using testing data, and tune hyperparameters.
  + **Price Prediction**: Once the model is trained, generate predictions for upcoming days/weeks
* **Model Training and Validation**: Train the model using historical data, validate it using testing data, and tune hyperparameters.
  + **Price Prediction**: Once the model is trained, generate predictions for upcoming days/weeks.



### ****4. Data Storage Layer****

* **Tool**: SQL Server Database
* **Purpose**: Structured storage of both cleaned and predicted data for further analysis.
* **Tables/Entities**:
  + Store Cleaned Data: Save the output of preprocessing.
  + Store Predicted Prices: Save forecasted results from ML models for future visualization.

### ****5. Visualization Layer****

* **Tool**: Power BI Dashboard
* **Goal**: Present insights and predictions to end users in a visually interactive manner.
* **Visual Elements**:
  + **Trend Analysis**: Shows historical price movement trends.
  + **Prediction Charts**: Line/Bar charts showing forecasted vs actual prices.
  + **Filters and Drilldowns and KPIs**: Allow users to filter by date, time range, and display metrics like average price, predicted increase/drop, etc.

### ****3. Machine Learning Layer****

* **Objective**: Use predictive modeling to forecast future Bitcoin prices.
* **Steps**:
  + **Model Selection**: Choose algorithms like Linear Regression or Random Forest based on the nature of the data.
  + **Model Training and Validation**: Train the model using historical data, validate it using testing data, and tune hyperparameters.
  + **Price Prediction**: Once the model is trained, generate predictions for upcoming days/weeks.

### 3.3 Data Structures

The system environment for this project includes the tools, technologies, and configurations required to perform end-to-end development, from data collection to predictive modeling and visualization. Each component of the environment plays a critical role in achieving the project objectives.

Programming Language: Python

 Python is a powerful, easy-to-use, and versatile programming language widely used in the data science and machine learning community. It offers robust libraries and tools for data analysis, model building, and visualization.

 **Version Used:** Python 3.8 or above

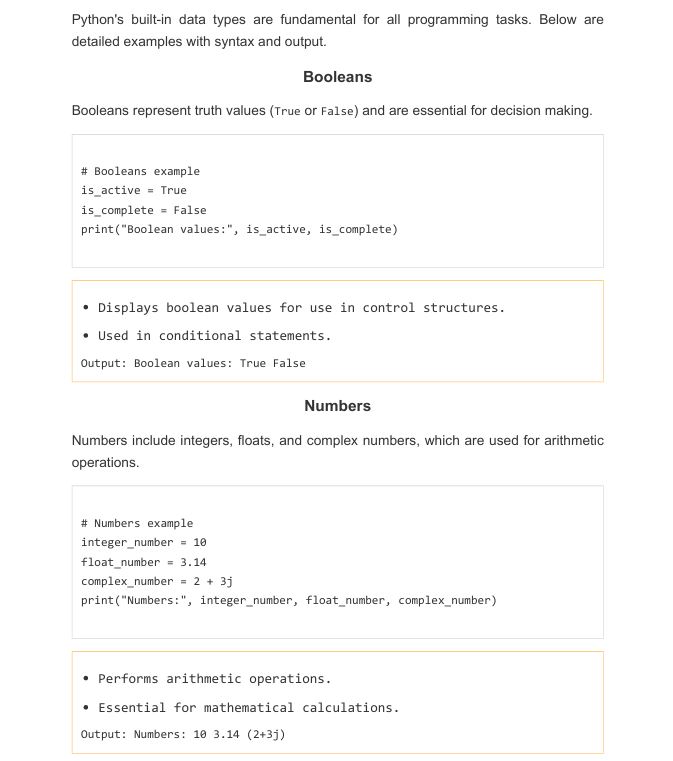
 **Key Libraries:**

* pandas: For data preprocessing, cleaning, and analysis
* numpy: For numerical and array operations
* matplotlib and seaborn: For visualizing trends and correlations
* scikit-learn: For implementing machine learning models like Linear Regression and Random Forest
* pyodbc: For connecting to the SQL Server database



### TECHNICAL DETAILS

### Data Types and Basic Operations



### TECHNICAL DETAILS

### Data Types and Basic Operations

### 

**2.2 Overview of Tools & Libraries**

Python has a rich ecosystem of libraries essential for data science and analytics.

### ****Python Libraries Overview****

* Key libraries: **NumPy**, **Pandas**, **Matplotlib**, **Seaborn**
* Used for numerical computing, data handling, and visualization

### ****NumPy for Numerical Computing****

* Supports multi-dimensional arrays
* Provides fast mathematical operations

### ****Pandas for Data Handling****

* Provides DataFrame for structured data
* Supports data manipulation and analysis

### ****Matplotlib & Seaborn for Data Visualization****

* **Matplotlib**: Basic plots like line graphs, bar charts
* **Seaborn**: Advanced and stylish statistical plots

## Database and File Specifications

## SQL Server Management Studio (SSMS)

### ✅ What is SSMS?

**SQL Server Management Studio (SSMS)** is a **free integrated environment** by Microsoft used to manage SQL Server databases. It provides a **graphical user interface (GUI)** for:

* Writing and executing SQL queries
* Managing database objects (tables, views, stored procedures)
* Monitoring performance
* Connecting to local or remote SQL Servers

### 🖥️ Features of SSMS

### ****1. Database and Table Specifications****

**Database Name:** BitcoinDB  
**Table Name:** BitcoinPrices

#### 📌 ****Table Structure:****

| **Column Name** | **Data Type** | **Description** |
| --- | --- | --- |
| Date | DATE | The primary key representing the daily timestamp. |
| Actual\_Close | FLOAT | The actual closing price of Bitcoin on the given date. |
| Predicted\_Close | FLOAT | The predicted closing price generated by the model. |

#### ✅ ****Design Rationale:****

* **Naming Convention:** The names BitcoinDB and BitcoinPrices clearly reflect the contents and purpose of the database and table, following best practices.
* **Data Types:**
  + DATE: Efficient for time-series data analysis and ensures proper sorting/filtering.
  + FLOAT: Ideal for continuous numeric values like prices, offering a good balance between precision and storage efficiency.
* **Primary Key:**
  + The Date field ensures each record is unique and chronologically ordered, which is essential for time series forecasting.
* **Constraints:**
  + All price columns must **not be null**, ensuring data integrity and enabling accurate model evaluations.

#### ⚙️ ****Platform:****

* **SQL Server** is used for:
  + Scalable data storage
  + Efficient querying and indexing
  + Integration with Power BI for business intelligence

### ****2. File Specifications****

#### 📥 ****Input File:**** btc\_1d\_data\_2018\_to\_2025.csv

* **Contents:** Historical Bitcoin price data including:
  + Date, Open, High, Low, Close, Volume
* **Purpose:** Used for:
  + Data cleaning and preprocessing
  + Feature engineering
  + Training machine learning and deep learning models

#### 📤 ****Output File:**** predicted\_btc\_prices.csv

* **Contents:** Results generated by the model, including:
  + Date, Actual Close, Predicted Close
* **Purpose:**
  + Evaluation of model accuracy
  + Insertion into SQL Server
  + Used for visual analysis and reporting in Power BI



## 2.3.8 Visual Studio

## ✅ What is Visual Studio?

**Visual Studio** is an **Integrated Development Environment (IDE)** developed by Microsoft. It is used by developers to **write, debug, test, and deploy code** across different programming languages like Python.

It supports both **Windows desktop** and **web development**, and is widely used for **.NET applications**, **data science**, **machine learning**, and **SQL Server integration**.

### 🖥️ Key Features of Visual Studio

| **Feature** | **Description** |
| --- | --- |
| 💻 **Code Editor** | Smart editor with IntelliSense (code suggestions), syntax highlighting, and error checking |
| 🧪 **Debugger** | Helps find and fix errors with breakpoints and step-by-step execution |
| 🧰 **Extensions** | Supports many plugins for Python, Jupyter Notebook, GitHub, Azure, etc. |
| 📊 **Data Tools** | Easily connect to SQL Server, run queries, and integrate databases |
| 🧠 **AI Assistance** | Features like GitHub Copilot (optional) for smart code suggestions |
| 🔄 **Version Control** | Built-in Git and GitHub support for collaboration and version tracking |

## 

## Power BI

### ✅ What is Power BI?

### **Power BI** is a **business analytics tool** developed by Microsoft. It helps you **analyze data**, **visualize insights**, and **share interactive dashboards and reports**.

### It connects to a wide range of **data sources** (like Excel, SQL Server, Python, and even online services like Google Analytics), and turns raw data into **meaningful visualizations** and **data stories**.

### ****. Power BI Integration****

#### 🔗 ****Data Connection:****

* Use **Import Mode** to connect Power BI with the SQL Server BitcoinDB.
* Load the BitcoinPrices table for building dashboards.

#### 🧩 ****Data Modeling:****

* Optionally, create a **Date Dimension Table** for better time-series filtering and trend analysis.
* Establish relationships between tables if additional metadata or logs are used.

#### 📊 ****Visualizations:****

* Create charts to show:
  + Actual vs Predicted Closing Prices (Line Chart)
  + Error Metrics like MAE, RMSE (Card or KPI Visual)
  + Daily or Monthly Price Trends (Area or Column Charts)
  + RSI (Relative Strength Index) or Technical Indicators

#### 🎛️ ****Interactive Dashboards:****

* Add **filters, slicers, and drill-down** options to make dashboards interactive and user-friendly.
* Examples:
  + Date range slicer
  + Monthly trend drill-down
  + Toggle between actual and predicted views

### 📊 Key Features of Power BI

| Feature | Description |
| --- | --- |
| 📈 **Data Visualization** | Create stunning charts, graphs and dashboards |
| 🔌 **Data Connectivity** | Connects to databases (SQL Server, Excel, etc.) |
| ⚙️ **Power Query** | Clean, transform, and shape your data using drag-and-drop interface |
| 📁 **Data Models** | Combine multiple data sources into a single model |
| 🚀 **Publishing** | Share your reports with others via Power BI Service |
| 📲 **Cross-platform** | Available on desktop, web, and mobile |

#### 4 Coding

#### 4.1 Data Loading and Preprocessing

#### 

#### 

#### 

#### #4.2 Model Training & Evaluation

#### import pandas as pd

#### import numpy as np

#### from sklearn.model\_selection import train\_test\_split

#### # Load cleaned data

df = pd.read\_csv(**'**Final\_Prediction.csv**‘**)

#### # ... (Feature engineering, data splitting, model building - see previous responses for LSTM code) ...

#### # Train the model

#### model.fit(X\_train, y\_train, epochs=100, batch\_size=32)

#### # Evaluate the model (consider metrics like RMSE, MAE)

**4.3 Prediction & Visualization**

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

from tensorflow import load\_model

(Prepare data for prediction, apply inverse scaling to get actual prices) ...

Create visualizations (e.g., line charts, scatter plots)

plt.plot(df['Date'], df['Actual\_Close'], label='Actual')

plt.plot(df['Date'], df['Predicted\_Close'], label='Predicted')

plt. legend()

plt. show()

# Save predictions to CSV (Milestone: Visual graphs, CSV ready)

df[['Date', 'Close', 'Predicted\_Close']].to\_csv('predicted\_btc\_prices.csv', index=False)

**4.4 SQL + Power BI Integration**

import pyodbc # Or other suitable SQL library

# Establish SQL connection

conn = pyodbc.connect(...)

# Create table if it doesn't exist (adapt for SQL Server syntax)

cursor = conn.cursor()

cursor.execute("""

CREATE TABLE IF NOT EXISTS BitcoinPrices (

Date DATE PRIMARY KEY,

Actual\_Close FLOAT NOT NULL,

Predicted\_Close FLOAT NOT NULL

);

""")conn.commit()

**4.5 In Power BI Desktop:**

# 1. Get Data -> SQL Server

# 2. Connect to your database

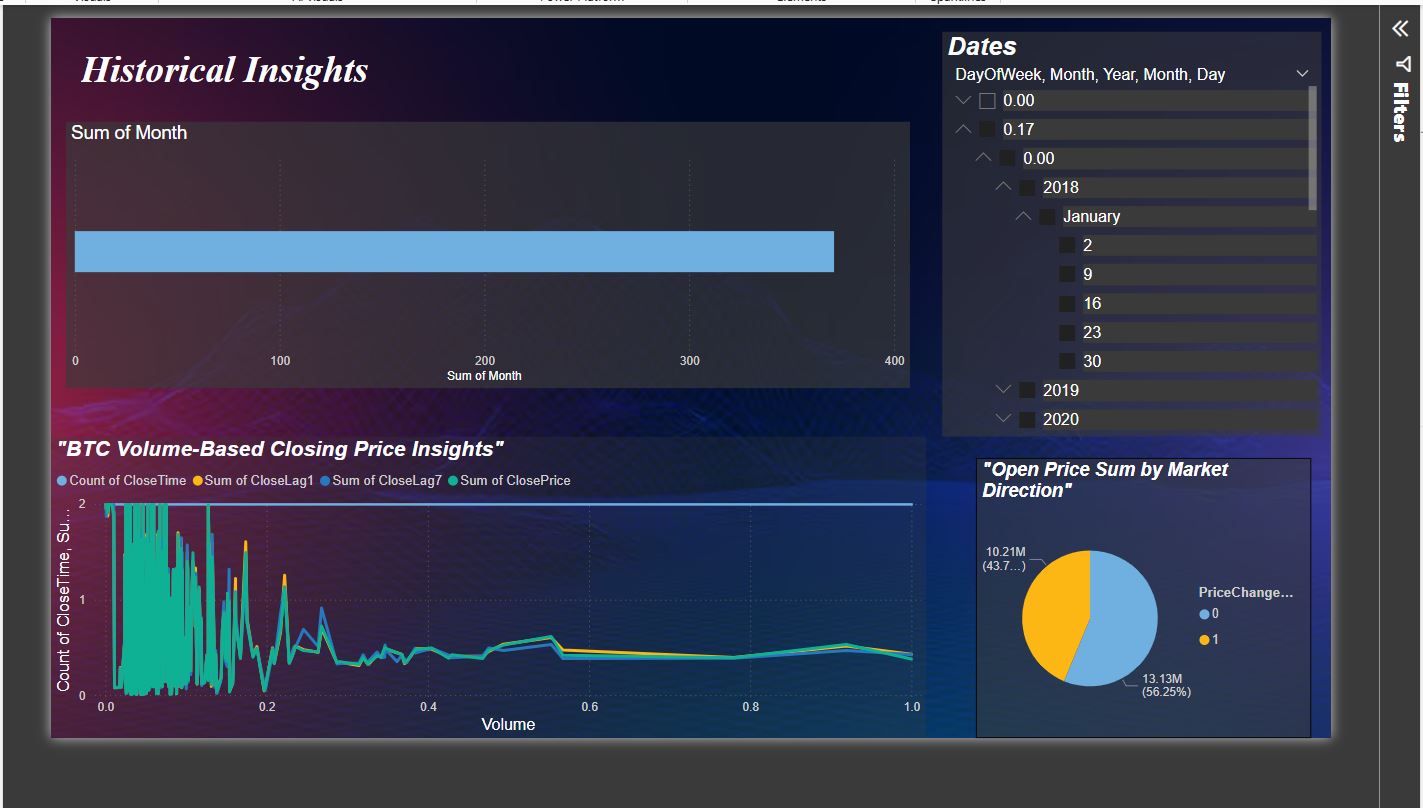
# 3. Select the BitcoinPrices table

# 4. Build visualizations and dashboard

# 5. Publish to Power BI service ( Power BI dashboard live)

#### 4.2 SNAPSHOTS OF SYSTEM





# 

# 

# 

# 5 TESTING

## ****5.1 Test Plan****

### ****1. Objectives****

* Ensure the **accuracy, reliability, and performance** of the Bitcoin price prediction system, including machine learning models and Power BI dashboard.
* Detect and mitigate **defects and integration issues** across components.
* Validate that the **entire system meets functional and non-functional requirements**, including ease of use, data integrity, and predictive accuracy.

### ****2. Testing Approach****

* **Unit Testing**
  + Validate each Python function/module (e.g., null handling, scaling, prediction generation).
* **Integration Testing**
  + Test the full pipeline: from CSV → SQL Server → Power BI.
  + Validate model output aligns with inserted data.
  + Check if dashboard reflects updates after SQL insert.
* **System Testing**
  + Simulate end-user workflows.
  + Validate usability, performance under load (Power BI visuals), and data consistency.
* **Regression Testing**
  + Re-run previous test cases after any changes in model or code to ensure existing functionality is unaffected.

### ****4. Testing Environment****

| **Component** | **Tool/Platform** |
| --- | --- |
| Model Development | Visual Studio , Google Colab (Python) |
| Database Storage | SQL Server Management Studio |
| Visualization Tool | Power BI Desktop |
| Data Format | CSV (for loading and output) |

### ****5. Test Data****

* **Primary Source:** btc\_1d\_data\_2018\_to\_2025.csv (historical data)
* **Prediction Output:** final\_prediction.csv.csv (actual + predicted prices)
* **Test Dataset:** A separate segment from recent data (e.g., January–April 2025), unseen during training, used to evaluate generalization.

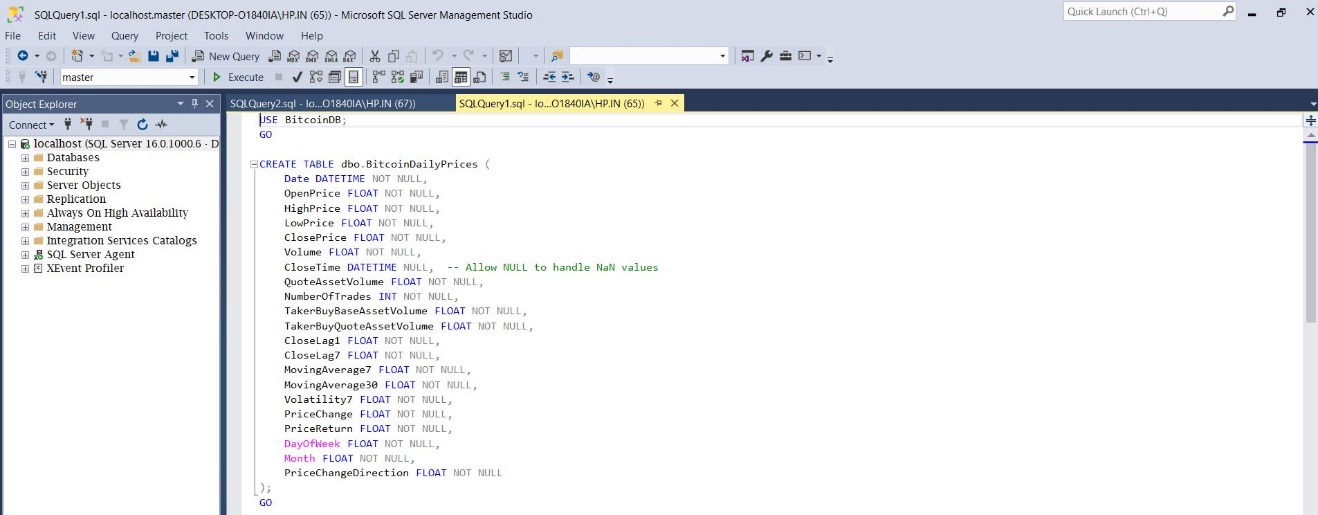
### ****6. Success Criteria****

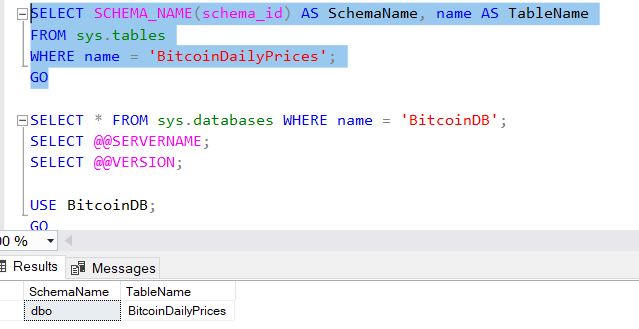
| **Component** | **Success Criteria** |
| --- | --- |
| Data Insertion to SQL | All rows inserted with correct formatting, no NULLs or mismatches |
| Power BI Visualizations | Charts reflect real-time SQL data, correct filters, interactive elements work |
| User Experience | Users can explore trends and comparisons intuitively; no broken interactions |
| Automation & Refresh | Power BI dashboard updates seamlessly with refreshed SQL data |

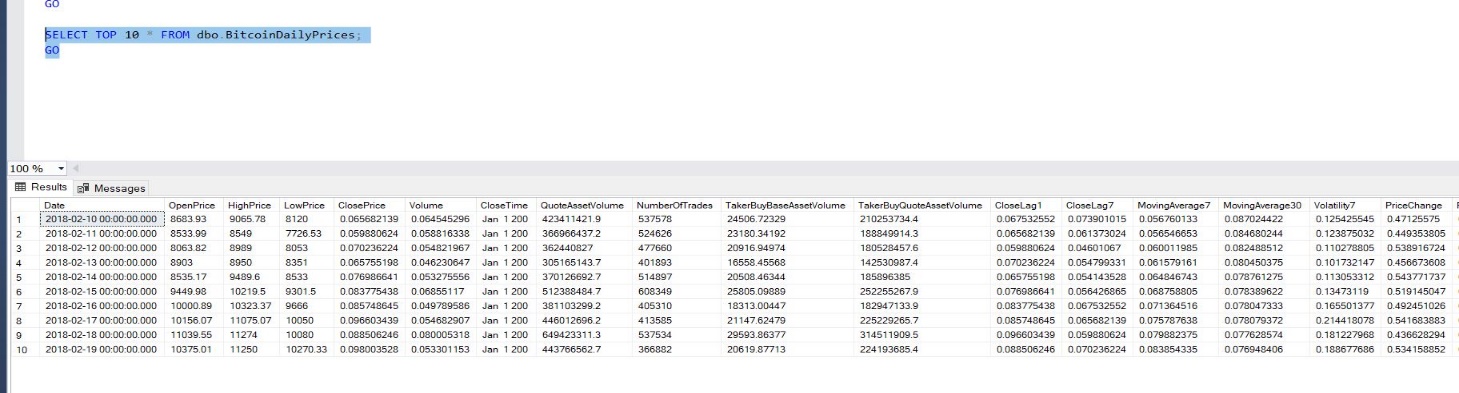
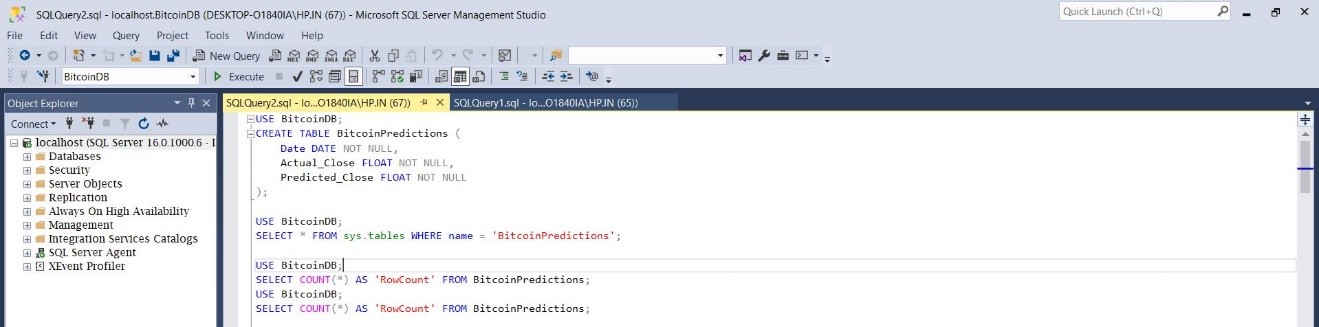
### ****7. Risks and Mitigations****

| **Risk** | **Mitigation Strategy** |
| --- | --- |
| Format mismatch during SQL import | Preprocess and validate date and float formatting in Python |
| Model overfitting or underperformance | Use validation metrics, cross-validation, regularization |
| Power BI refresh delay | Set up scheduled refresh and monitor refresh logs |
| Data leakage or inconsistent predictions | Strict separation of train/test data and tracking data lineage |

* **5.2 SQL Queries Testing**
* **BitcoinPrices table**



* **Schema Query**
* **Bitcoin Prediction table**

****

# Prediction data query

# 

# 6 PROJECT LEGACY

### 6.1 CURRENT STATUS:

### The Bitcoin Price Prediction project is currently **successfully completed** with the following components in place:

* Cleaned and preprocessed raw Bitcoin price data using **pandas**
* Performed **Exploratory Data Analysis (EDA)** using matplotlib and seaborn
* Built and evaluated **machine learning models** like **Linear Regression** and **Random Forest** to predict future prices
* Exported predicted results into a **CSV file**
* Created a **SQL Server database (BitcoinDB)** and inserted both historical and predicted data
* Visualized the results using **Power BI**, including trend lines, comparison charts, and daily price dashboards

## 6.2 PROBLEMS FACED

* While working with the original CSV file, **some missing and inconsistent data** required **preprocessing** to ensure accurate and clean input for the model.
* Initially, **model accuracy was lower** due to basic feature selection and lack of scaling, but these issues were identified and resolved through **iterative tuning and analysis**.
* Setting up the **Python-SQL Server connection** involved some delays due to driver compatibility, which was eventually overcome by configuring the correct versions and settings.
* The **time series nature of the Bitcoin dataset** presented challenges during data splitting, which were addressed using time-aware techniques to maintain the data’s sequence.
* **Formatting and customizing Power BI visuals** for time-based data required extra effort, but it helped produce clear and interactive visualizations once resolved.

**6.3 LIMITATION:**

* **Simplified focus**: Concentrating on daily data provides a clear picture of long-term trends without the complexity of minute-by-minute fluctuations.
* **Room for growth**: Using basic models opens opportunities to implement advanced techniques, like deep learning, for improved predictions.
* **Foundation for experimentation**: The lack of real-time data allows for a simpler model, perfect for future experiments and integrating live feeds.
* **Actionable insights**: Daily data still provides valuable insights, especially for long-term investors.
* **Wider accessibility**: The model’s simplicity makes it more accessible to users without advanced resources or expertise.

**6.4 FUTURE ENHANCEMENTS:**

* Integrate **sentiment analysis** from Twitter or news headlines
* Add **live API integration** (e.g. Binance API) for real-time predictions
* Include **additional features** like trading volume, RSI, and moving averages
* Publish interactive dashboards online using **Power BI Cloud Service**
* Add user authentication and role-based dashboards if turned into a web app

**6.5 CONCLUSION:**

This project demonstrates the **end-to-end application of data science** — from data collection and cleaning, to modeling, storing in a database, and finally visualizing insights with interactive dashboards.  
It shows how **machine learning** and **visual tools** like Power BI can be used to make informed predictions about **financial markets**.  
While the model provides meaningful insights, further improvements can enhance accuracy and real-world usability.  
This project lays the **foundation for a scalable and intelligent cryptocurrency price forecasting system**.