**Progress Report**

**Title: Predicting Cryptocurrency Prices Using Historical Data**

Date: 09/06/2024

Prepared by: Afif Salman

**Part 1: Problem Statement and Objectives**

1. **Problem Statement**

Cryptocurrency markets are known for their high volatility and unpredictability. Investors and traders seek reliable models to predict future prices based on historical data to make informed decisions and optimize their trading strategies. This project aims to develop a regression model to predict daily closing prices of various cryptocurrencies using historical data.

1. **Objectives**

- **Specific**: Develop a regression model to predict daily closing prices of various cryptocurrencies.

- **Measurable**: Use RMSE, MAE, and R-squared metrics to evaluate model performance.

- **Achievable**: Utilize available historical data and advanced machine learning techniques.

- **Relevant**: Help investors optimize trading strategies for better financial outcomes.

- **Time-bound**: Complete the project within two months with specific milestones.

1. **Questions**

**1. How does the high volatility of cryptocurrency markets impact the accuracy and reliability of predictive models based on historical data?**

Kristoufek, L. (2015). What Are the Main Drivers of the Bitcoin Price? Evidence from Wavelet Coherence Analysis. PLoS ONE, 10(4), e0123923.

**2. What alternative data sources or supplementary analysis techniques can investors and traders use to enhance the predictive power of historical data in cryptocurrency price forecasting?**

Chen, T., Lin, C., & Chen, Y. (2020). Forecasting Cryptocurrency Prices with Deep Learning under Data Dynamics. IEEE Access, 8, 212173-212184.

**3. How can investors and traders use strategies to mitigate the risks of relying on historical data to predict cryptocurrency prices?**

Tashman, L. J. (2000). Out-of-sample tests of forecasting accuracy: an analysis and review. International Journal of Forecasting, 16(4), 437-450.

1. **Milestones**

1. Data Acquisition: Gather historical price data for multiple cryptocurrencies.

2. Exploratory Data Analysis (EDA): Understand data characteristics, identify patterns, and pre-process the data.

3. Modelling: Develop and evaluate regression models and fine-tune the best model.

4. Implementation and Testing: Deploy the model and assess its performance on unseen data.

5. Reporting and Presentation: Summarize findings and present the results.

**Part 2: Data Acquisition and Exploratory Data Analysis (EDA)**

1. **Data Acquisition**

- Data Source: The datasets were sourced from Kaggle, a reputable platform for obtaining historical cryptocurrency price data.

- Datasets: The project uses historical data from 24 different cryptocurrencies, including Bitcoin, Ethereum, Litecoin, and others.

- Date of Acquisition: 25/05/2024

1. **Exploratory Data Analysis (EDA)**

Steps Performed:

1. Loading and Combining Datasets:

- Datasets were loaded individually and combined into a single DataFrame for comprehensive analysis.

2. Data Types and Summary Statistics:

- Identified data types of each column.

- Generated summary statistics to understand the central tendency and dispersion of the data.

3. Missing Data:

- Checked for missing values.

- Visualized missing data using heatmaps.

- Handled missing data by either dropping rows with missing values or using forward fill methods.

4. Distribution Analysis:

- Examined the distributions of numerical features through histograms.

- Identified outliers using box plots and Z-score analysis.

1. **Findings:**

- Data Types: Most features are numerical, except for the date column which is in datetime format.

- Missing Data: Some datasets had missing values which were addressed appropriately.

- Outliers: Identified outliers in the data that could potentially skew the model's performance.

1. **Justification:**

EDA is a critical step to understand the underlying structure and patterns in the data. It helps in identifying potential issues such as missing values and outliers that need to be addressed before modeling. The insights gained from EDA guide the feature engineering and model selection processes.

**Part 3: Data Preprocessing, Modeling, and Evaluation**

1. **Data Preprocessing**

Steps Performed:

1. Date Conversion: Converted the 'Date' column to datetime format.

2. Feature Engineering: Created additional time-related features such as Year, Month, Day, and Day Of Week.

3. Feature Selection: Selected relevant features like Open, High, Low, and Volume along with the engineered time features.

4. Data Splitting: Split the data into training and testing sets (80% train, 20% test).

5. Scaling: Standardized the features to have a mean of 0 and standard deviation of 1 using `StandardScaler`.

1. **Modeling**

Models Developed:

1. Linear Regression:

- Simple and interpretable model.

- Trained on scaled training data.

- Evaluated using RMSE, MAE, and R-squared metrics.

2. Decision Tree Regression:

- Captures non-linear relationships in the data.

- Trained on scaled training data.

- Evaluated using the same metrics as Linear Regression.

3. Hyperparameter Tuning:

- Conducted GridSearchCV for Decision Tree Regressor to identify the best combination of hyperparameters.

4. Evaluation

Metrics Used:

- Root Mean Squared Error (RMSE): Measures the average magnitude of the errors between predicted and actual values.

- Mean Absolute Error (MAE): Measures the average magnitude of the errors without considering their direction.

- R-squared: Indicates the proportion of variance in the dependent variable predictable from the independent variables.

1. **Results:**

\*Linear Regression:

- RMSE: 126.07913914581827

- MAE: 18.73351670385705

- R-squared: 0.9993730096647094

\*Decision Tree Regression:

- RMSE: 130.68801924561987

- MAE: 15.119636803941761

- R-squared: 0.9993263319847792

1. **Best Model:**

- The best-performing model was the Decision Tree Regressor after hyperparameter tuning:

- RMSE: 147.20861317741108

- MAE: 22.966464492802356

- R-squared: 0.999145246634563

Justification:

Preprocessing ensures that the data fed into the models is clean and standardized, which is crucial for reliable predictions. Training multiple models allows for comparison and selection of the best-performing one. Hyperparameter tuning optimizes model performance by finding the best parameters.

1. **Visualization**

Explanation:

- X-axis: Represents the time index of the test set observations.

- Y-axis: Represents the cryptocurrency prices (actual vs predicted).

- The plot helps in visually assessing the model's performance by comparing the actual prices with the predicted ones.

**Topics for 1:1 Discussion**

1. Instead of using all types of cryptocurrency, just pick one type only.

2. Refer to time series reference & determine whether the data is in stationery or not.

**Weekly Progress Report - Part 4: Final Documentation**

Date: 23/06/2024

Prepared by: Afif Salman

**Goal:**

The objective of this project is to develop a model to predict Bitcoin prices using ARIMA (AutoRegressive Integrated Moving Average) methodology. The goal is to make reasonably accurate predictions by performing a thorough time series analysis.

**Data Source:**

The dataset used for this analysis contains historical Bitcoin prices, including the opening price, closing price, highest price, lowest price, and trading volume for each day. The data was sourced from [specific source/website].

**Key Metrics:**

- Mean Absolute Error (MAE)

- Mean Squared Error (MSE)

- Root Mean Squared Error (RMSE)

**Findings:**

- The ARIMA model with parameters (1,1,1) was found to be the best fit for our data.

- The model's forecast closely followed the actual Bitcoin price movements, with reasonably low MAE, MSE, and RMSE values.

- The model showed limitations during periods of extreme volatility, indicating the need for further refinement or hybrid models to improve prediction accuracy.

**Risks/Limitations/Assumptions:**

- Assumes historical patterns will repeat, which might not hold true in highly volatile markets like cryptocurrency.

- Model performance may degrade with unforeseen market shocks.

- The dataset might not capture all influencing factors of Bitcoin price movements.

**Summary of Statistical Analysis:**

**Implementation:**

- The ARIMA model was implemented using Python's `statsmodels` library.

- Time series data was preprocessed by checking for stationarity and applying differencing where necessary.

- ACF (Autocorrelation Function) and PACF (Partial Autocorrelation Function) plots were used to identify potential parameters for the ARIMA model.

**Evaluation:**

- The model was evaluated using the MAE, MSE, and RMSE metrics.

- The data was split into training and test sets to validate the model's performance on unseen data.

**Inference:**

- The ARIMA model captured the overall trend and seasonal patterns of the Bitcoin prices effectively.

- However, it struggled with accurately predicting sharp price changes, indicating potential areas for improvement.

**Detailed Documentation and Explanation:**

**1. Data Loading and Inspection:**

- Loaded the Bitcoin dataset and converted the 'Date' column to datetime format.

- Inspected the data types and basic statistics to understand the dataset structure.

**2. Exploratory Data Analysis (EDA):**

- Visualized the closing price over time to identify trends and patterns.

- Conducted summary statistics to understand the distribution of the data.

**3. Check for Stationarity:**

- Used the Augmented Dickey-Fuller (ADF) test to check for stationarity.

- Applied differencing to make the series stationary if required.

**4. ARIMA Modeling:**

- Plotted ACF and PACF to determine the parameters (p, d, q) for the ARIMA model.

- Fitted the ARIMA model and performed diagnostic checks on residuals.

- Forecasted future Bitcoin prices and plotted the results.

**5. Model Performance Evaluation:**

- Split the data into training and test sets.

- Fitted the ARIMA model on the training data and forecasted the test set.

- Calculated and reported MAE, MSE, and RMSE to assess model accuracy.

**Next Steps:**

- Final review and refinement of the documentation.

- Ensure the reproducibility and clarity of the notebook.

- Publish the GitHub repository and ensure all links and resources are correctly referenced.

- Share the Medium blog post link in the README.

**Summary:**

This week focused on completing the final documentation for the project. The key tasks included summarizing the analysis, documenting the process & organizing the repository. The project is nearing completion with the final review and publication pending.

**Weekly Progress Report - Part 5: Final Presentation & Demo**

Date: 30/06/2024

Prepared by: Afif Salman

**Objective: Predicting Bitcoin Prices to Assist Traders and Investors**

**Key Achievements:**

**1. Data Acquisition and Preparation:**

* Acquired Bitcoin price data from the Kaggle dataset provided by Sudalairajkumar.
* Cleaned and preprocessed the data, ensuring it is formatted correctly for analysis.

**2. Model Development:**

* Implemented an ARIMA (AutoRegressive Integrated Moving Average) model for Bitcoin price forecasting.
* Conducted thorough analysis including time series visualization, stationarity checks, and model selection using auto\_arima.
* Evaluated model performance using Mean Squared Error (MSE) and other metrics.

**3. Presentation Preparation:**

* Created a slide deck tailored for a 7 minute presentation to non-technical audiences plus 3 minutes Q&A session.
* Slides include clear explanations of each step in the modeling process, emphasizing why specific models and methods were chosen.
* Visual aids such as graphs and plots are designed to illustrate key concepts effectively.

**4. References and Citations:**

* Compiled references slide listing relevant sources including seminal works on time series analysis and Bitcoin technology.
* Ensured all references are formatted correctly in APA style for academic integrity.

**Next Steps:**

* Finalize slide content and practice delivery to ensure clarity and coherence.
* Review model outputs and results to prepare for potential questions during the Q&A session.
* Incorporate feedback from peers or mentors to polish presentation delivery and content.

**Conclusion:**

This progress report outlines the comprehensive preparation undertaken for the final presentation on Bitcoin price forecasting. The focus remains on delivering valuable insights to stakeholders while showcasing rigorous analysis and methodology.

**Limitations and Future Works for the Bitcoin Price Forecasting Model**

**Limitations:**

**1. High MSE Value:**

- The Mean Squared Error (MSE) of 437,327,606.99 suggests that there is significant variance between the predicted and actual Bitcoin prices, indicating room for improvement in model accuracy.

**2. Volatility of Bitcoin Prices:**

- Bitcoin prices are highly volatile and influenced by numerous external factors such as market sentiment, regulatory news, and macroeconomic events, which are not captured by the ARIMA model.

**3. Assumption of Linearity:**

- The ARIMA model assumes linear relationships within the time series data. Non-linear patterns or interactions might not be well-captured by this approach, potentially leading to less accurate forecasts.

**4. Stationarity Requirement:**

- The ARIMA model requires the data to be stationary, which necessitates differencing and transformation. This process might oversimplify complex dynamics in the price series.

**5. Lack of Exogenous Variables:**

- The model does not incorporate external predictors or exogenous variables that could influence Bitcoin prices, such as trading volume, social media sentiment, or economic indicators.

**6. Short-Term Forecasting:**

- ARIMA models are typically better suited for short-term forecasting. Their effectiveness might diminish over longer forecast horizons due to compounding errors and unaccounted variability.

**Future Works:**

**1. Incorporate Exogenous Variables:**

- Enhance the model by incorporating exogenous variables (e.g., trading volume, regulatory news, macroeconomic indicators) to better capture the factors influencing Bitcoin prices.

**2. Explore Advanced Models:**

- Experiment with advanced machine learning models such as Long Short-Term Memory (LSTM) networks, Gated Recurrent Units (GRUs), and other deep learning architectures that can capture non-linear relationships and complex patterns in the data.

**3. Hybrid Models:**

- Develop hybrid models that combine ARIMA with other techniques like neural networks to leverage the strengths of both statistical and machine learning approaches.

**4. Sentiment Analysis:**

- Integrate sentiment analysis from social media platforms, news articles, and forums to gauge market sentiment and its impact on Bitcoin prices.

**5. Model Ensemble:**

- Utilize ensemble methods that aggregate predictions from multiple models to improve forecast accuracy and robustness.

**6. Continuous Model Evaluation:**

- Implement a continuous evaluation and update mechanism for the model to adapt to new data and changing market conditions, ensuring sustained accuracy and relevance.

**7. Expand to Other Cryptocurrencies:**

- Apply similar modeling techniques to forecast prices of other cryptocurrencies, potentially developing a comprehensive predictive framework for the entire crypto market.