

Model Optimization and Tuning Phase Template

Date	20 Nov 2024
Team ID	739720
Project Title	Time Series for Bitcoin Price Prediction using Prophet
Maximum Marks	10 Marks

Model Optimization and Tuning Phase

The Model Optimization and Tuning Phase involves refining neural network models for peak performance. It includes optimized model code, fine-tuning hyperparameters, comparing performance metrics, and justifying the final model selection for enhanced predictive accuracy and efficiency.

Hyperparameter Tuning Documentation (8 Marks):

Model	Tuned Hyperparameters
-------	-----------------------

<p>Model 1</p> <p>Prophet</p>	<p>Prophet is an open-source forecasting tool developed by Facebook, designed for time series data that has strong seasonal effects and missing values. It is robust to outliers and flexible in capturing non-linear trends.</p> <pre> # Sort the data by date to ensure chronological order df1 = df1.sort_values('ds') # Visualize the Bitcoin price time series fig = go.Figure() fig.add_trace(go.Scatter(x=df1['ds'], y=df1['y'], mode='lines', name='Bitcoin Open Price')) fig.update_layout(title='Bitcoin Open Price Time Series', xaxis=dict(rangeselector=dict(buttons=list([dict(count=1, label='1m', step='month', stepmode='backward'), dict(count=6, label='6m', step='month', stepmode='backward'), dict(count=1, label='YTD', step='year', stepmode='todate'), dict(count=1, label='1y', step='year', stepmode='backward'), dict(step='all')])), rangeslider=dict(visible=True), type='date')) fig.show() # Initialize and fit the Prophet model for full dataset prediction model = Prophet(seasonality_mode='multiplicative') model.fit(df1) # Create future dates and generate predictions for the next 365 days future = model.make_future_dataframe(periods=365) forecast = model.predict(future) # Display the last few predictions print("Last 5 Forecasted Values:") print(forecast[['ds', 'yhat', 'yhat_lower', 'yhat_upper']].tail()) </pre>
-------------------------------	---

Final Model Selection Justification (2 Marks):

Final Model	Reasoning
-------------	-----------

Model 1 Prophet

Prophet is an open-source forecasting tool developed by Facebook, designed for time series data that has strong seasonal effects and missing values. It is robust to outliers and flexible in capturing non-linear trends.

```
# Sort the data by date to ensure chronological order
df1 = df1.sort_values('ds')

# Visualize the Bitcoin price time series
fig = go.Figure()
fig.add_trace(go.Scatter(x=df1['ds'], y=df1['y'], mode='lines', name='Bitcoin Open Price'))
fig.update_layout(
    title='Bitcoin Open Price Time Series',
    xaxis=dict(
        rangeselector=dict(
            buttons=list([
                dict(count=1, label='1m', step='month', stepmode='backward'),
                dict(count=6, label='6m', step='month', stepmode='backward'),
                dict(count=1, label='YTD', step='year', stepmode='todate'),
                dict(count=1, label='1y', step='year', stepmode='backward'),
                dict(step='all')
            ])
        ),
        rangeslider=dict(visible=True),
        type='date'
    )
)
fig.show()

# Initialize and fit the Prophet model for full dataset prediction
model = Prophet(seasonality_mode='multiplicative')
model.fit(df1)

# Create future dates and generate predictions for the next 365 days
future = model.make_future_dataframe(periods=365)
forecast = model.predict(future)

# Display the last few predictions
print("Last 5 Forecasted Values:")
print(forecast[['ds', 'yhat', 'yhat_lower', 'yhat_upper']].tail())
```