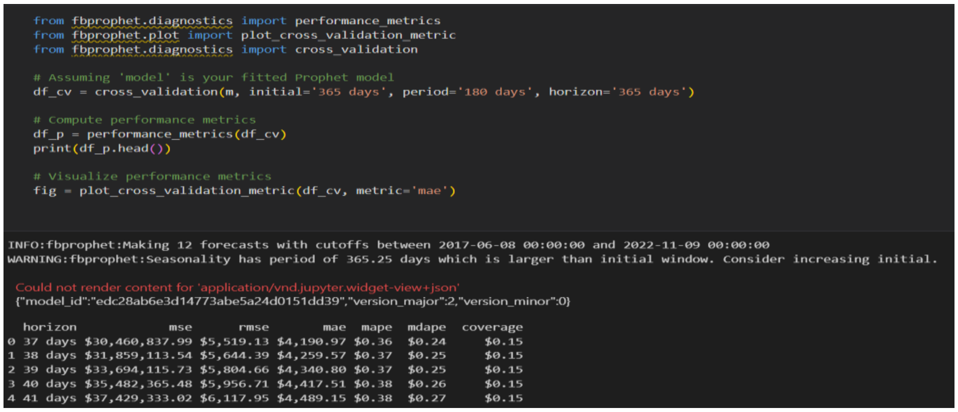
Model Development Phase Template

Date	03 October 2024
Team ID	LTVIP2024TMID24963
Project Title	Time Series Analysis For Bitcoin Price Prediction Using Prophet
Maximum Marks	4 Marks

Initial Model Training Code, Model Validation and Evaluation Report

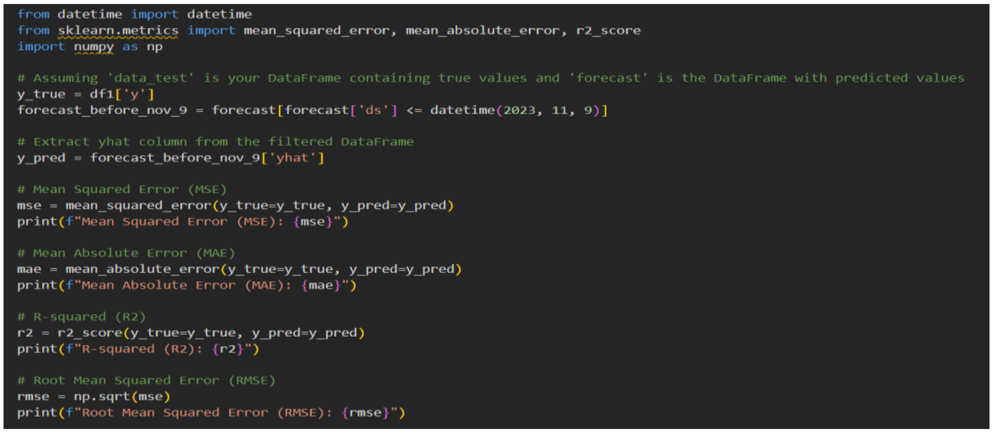
The initial model training code will be showcased in the future through a screenshot. The model validation and evaluation report will include classification reports, accuracy, and confusion matrices for multiple models, presented through respective screenshots.

Initial Model Training Code:



```
    from fbprophet.diagnostics import performance_metrics
    from fbprophet.plot import plot_cross_validation_metric
    from fbprophet.diagnostics import cross_validation

    # Assuming 'model' is your fitted Prophet model
    df_cv = cross_validation(m, initial='365 days', period='180 days', horizon='365 days')

    # Compute performance metrics
    df_p = performance_metrics(df_cv)
    print(df_p.head())

    # Visualize performance metrics
    fig = plot_cross_validation_metric(df_cv, metric='mae')
```

```
INFO:fbprophet:Making 12 forecasts with cutoffs between 2017-06-08 00:00:00 and 2022-11-09 00:00:00
WARNING:fbprophet:Seasonality has period of 365.25 days which is larger than initial window. Consider increasing initial.

 Could not render content for 'application/vnd.jupyter.widget-view+json'
 {"model_id":"edc28ab6e3d14773abe5a24d0151dd39","version_major":2,"version_minor":0}

  horizon              mse      rmse        mae  mape  mdape  coverage
0 37 days $30,460,837.99 $5,519.13 $4,190.97 $0.36  $0.24     $0.15
1 38 days $31,859,113.54 $5,644.39 $4,259.57 $0.37  $0.25     $0.15
2 39 days $33,694,115.73 $5,804.66 $4,340.80 $0.37  $0.25     $0.15
3 40 days $35,482,365.48 $5,956.71 $4,417.51 $0.38  $0.26     $0.15
4 41 days $37,429,333.02 $6,117.95 $4,489.15 $0.38  $0.27     $0.15
```



```
from datetime import datetime
from sklearn.metrics import mean_squared_error, mean_absolute_error, r2_score
import numpy as np

# Assuming 'data_test' is your DataFrame containing true values and 'forecast' is the DataFrame with predicted values
y_true = df1['y']
forecast_before_nov_9 = forecast[forecast['ds'] <= datetime(2023, 11, 9)]

# Extract yhat column from the filtered DataFrame
y_pred = forecast_before_nov_9['yhat']

# Mean Squared Error (MSE)
mse = mean_squared_error(y_true=y_true, y_pred=y_pred)
print(f"Mean Squared Error (MSE): {mse}")

# Mean Absolute Error (MAE)
mae = mean_absolute_error(y_true=y_true, y_pred=y_pred)
print(f"Mean Absolute Error (MAE): {mae}")

# R-squared (R2)
r2 = r2_score(y_true=y_true, y_pred=y_pred)
print(f"R-squared (R2): {r2}")

# Root Mean Squared Error (RMSE)
rmse = np.sqrt(mse)
print(f"Root Mean Squared Error (RMSE): {rmse}")
```

Model Validation and Evaluation Report:

Model	Mean Squared Error (MSE):	Mean Absolute Error (MAE	**Root Mean Square Error (RMSE)**
Prophet	**8006516.28151861**	**1897.6986625308068**	**2829.578816982946**