

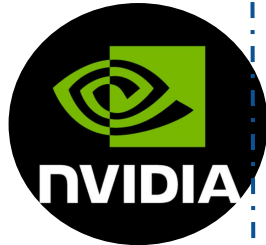
Market cap: \$50B



Market cap: \$500B



Market cap: \$3T



ML
by PATIL



ML
by PATIL

Descriptive_Statistics

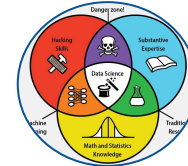
Pandurang Patil Sindhe
IIT Roorkee (Engineering)
BITS Pilani (Data Science)



IIT Roorkee
(Engineering)



BITS Pilani
(Data Science)



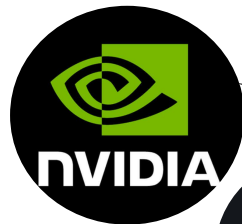
Pandurang Patil Sindhe

LinkedIn

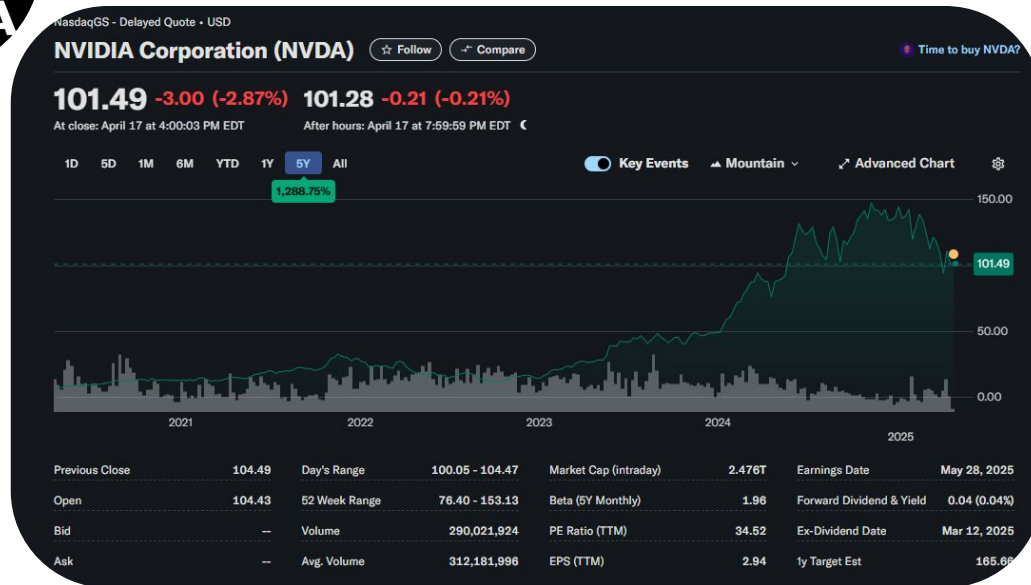
Business >> Concepts >> Data >> STL >> Stationarity >> Transformation >> Modeling >> Evaluation >> Deployment

ML by PATIL

Business Problem



- Provide **actionable insights** through time series forecasting.
- Evaluate **ARIMA's performance** for financial forecasting.
- **Predict** NVIDIA stock prices for better investment decisions.



Business

Concepts

Data

STL

Setup

STL - Stationarity

Modeling

Evaluation

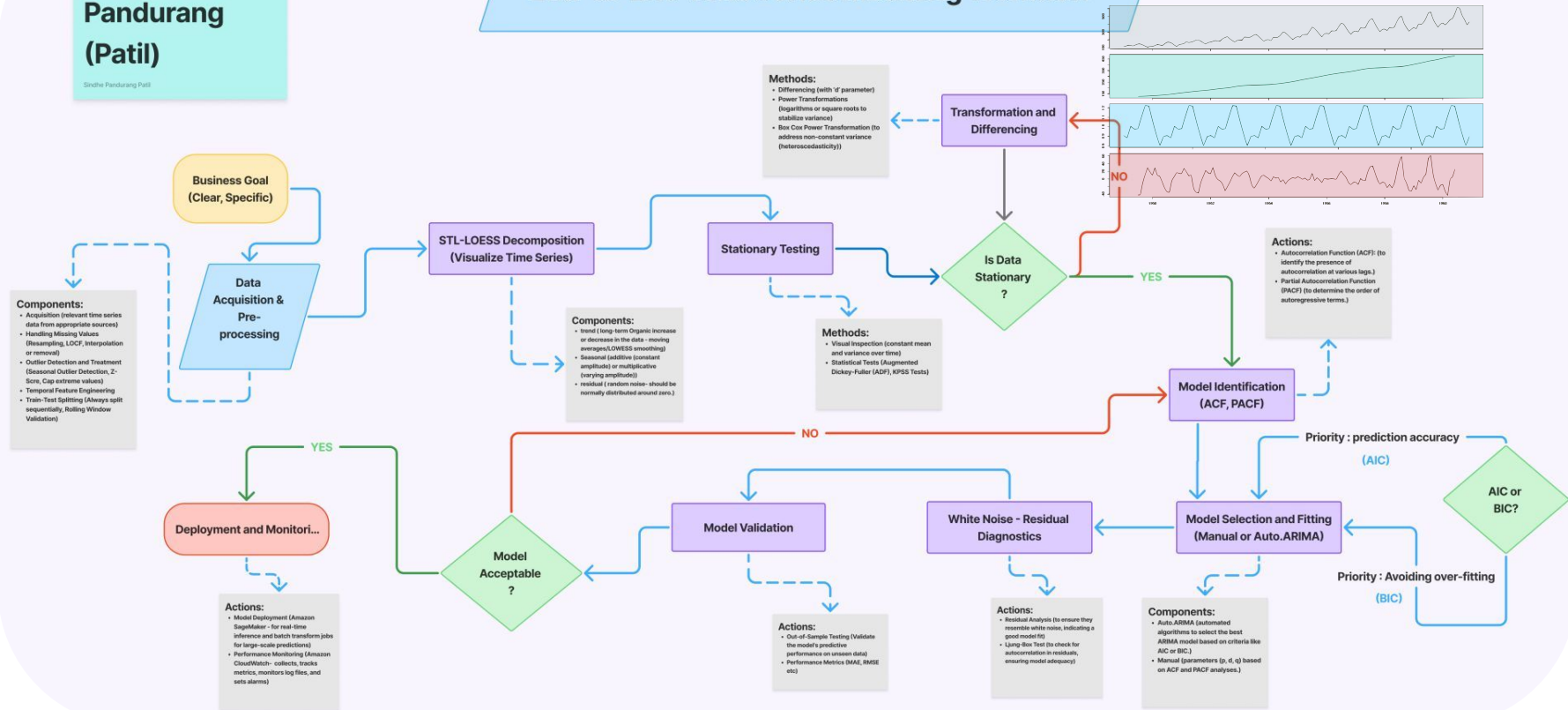
Deployment

Project Flow

Pandurang
(Patil)

Sindhe Pandurang Patil

End-to-End Time Series Modeling Workflow



Business

Concepts

Data

STL

Stationarity

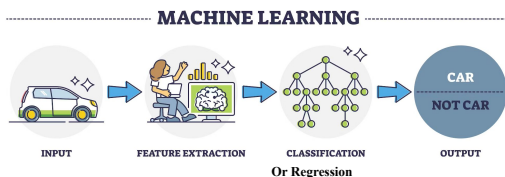
Transformation

Modeling

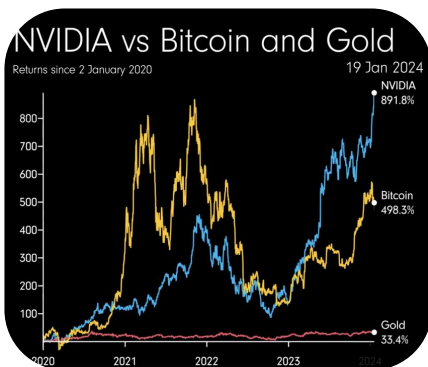
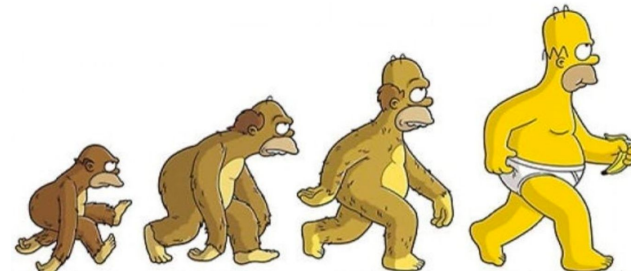
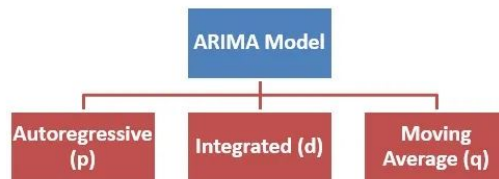
Evaluation

Deployment

ML Models vs Time Series Models




V/S



Aspect	Time Series Models	ML Models
Focus	Trend & seasonality	Complex, non-linear patterns
Data	Univariate, needs stationarity	Multivariate, flexible inputs
Interpretability	High, transparent	Low, often a black box
Seasonality/Trend	Built-in (e.g., SARIMA)	Needs manual feature engineering
Forecast Horizon	Best for short-term	Suits long-term, complex cases
Speed	Fast, simple tuning	Slower, more tuning needed

Business



 Concepts


Data



STL



Stationarity



Transformation



Modeling



Evaluation

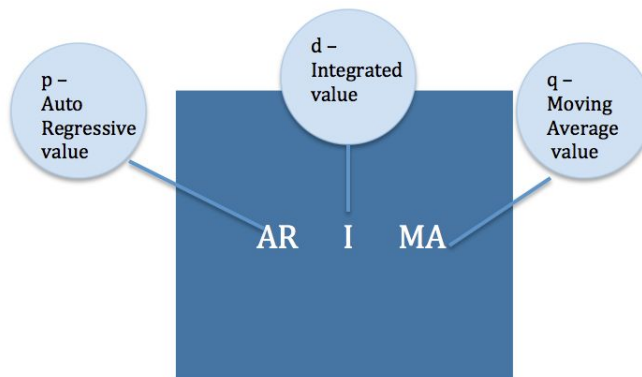


Deployment

ARIMA / SARIMA Modeling

ARIMA (AutoRegressive Integrated Moving Average) is a statistical model used for forecasting time series data by combining:

- **AR (AutoRegression):** Uses past values to predict future ones.
- **I (Integrated):** Makes the data stationary by differencing.
- **MA (Moving Average):** Models the error of past predictions.



$$y'_t = \underbrace{c}_{\text{intercept}} + \underbrace{\phi_1 y'_{t-1} + \dots + \phi_p y'_{t-p}}_{\text{lagged values}} + \underbrace{\theta_1 \varepsilon_{t-1} + \dots + \theta_q \varepsilon_{t-q} + \varepsilon_t}_{\text{lagged errors}}$$

differenced time series

ARIMA / SARIMA Assumptions

- **Stationarity:** The time series should have a constant mean and variance over time (or can be made stationary through differencing).
- **Linearity:** The relationship in the data is linear and can be captured with AR and MA terms.
- **No autocorrelated residuals:** Residuals (errors) should behave like white noise.
- **Normality of residuals:** Residuals should be normally distributed for reliable confidence intervals (especially in inference).



**KPSS
TEST**

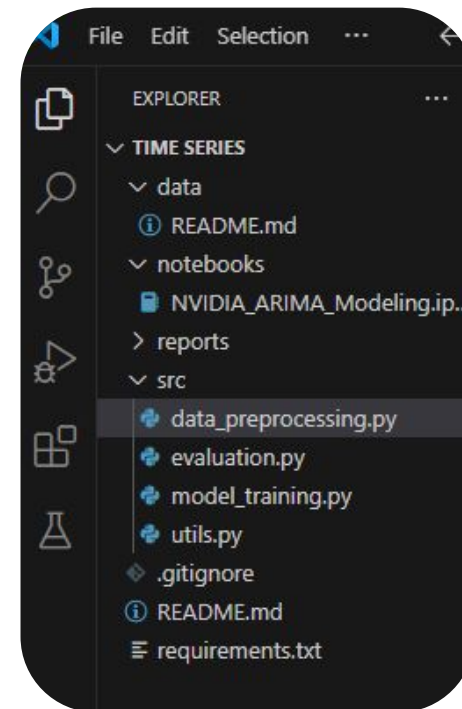
**Augmented Dickey-Fuller Test
(ADF)**



Ljung-Box Test

Setting up Environment

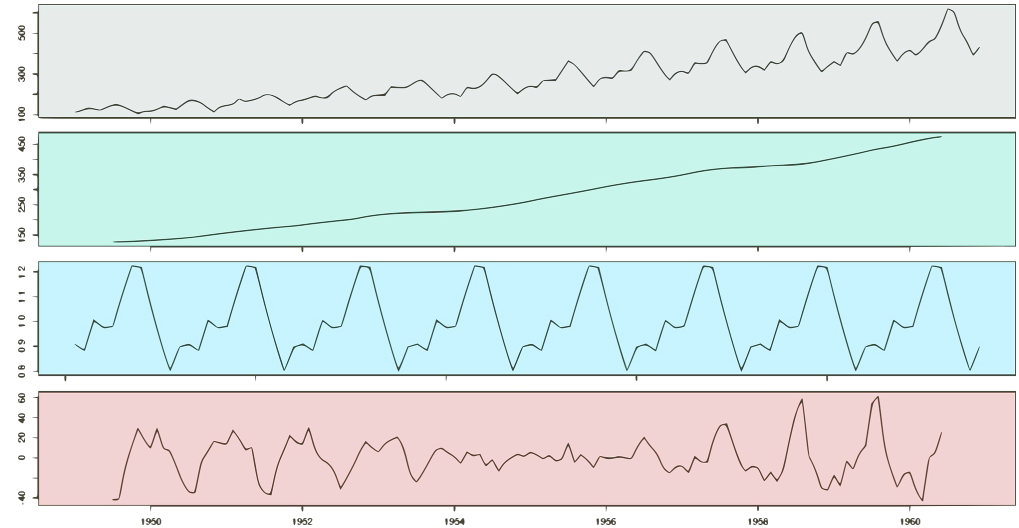
- **numpy** - Fast numerical computations and array handling.
- **pandas** - Data manipulation and time series handling.
- **matplotlib** - Basic plotting and visualizations.
- **statsmodels** - Statistical modeling and ARIMA implementation.
- **pmdarima** - Auto ARIMA and model selection made easy.
- **yfinance** - Fetch historical stock/financial data from Yahoo Finance.
- **seaborn** - Enhanced statistical data visualizations.



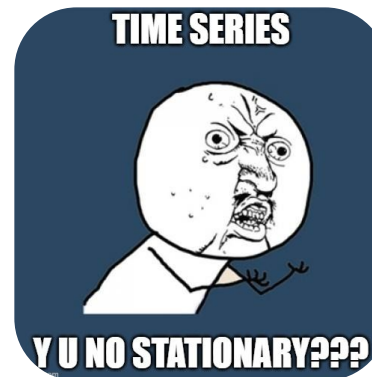
STL - LOESS Decomposition

```
stl = STL(data['Price'], period=365, seasonal=13, trend=91, robust=True)
```

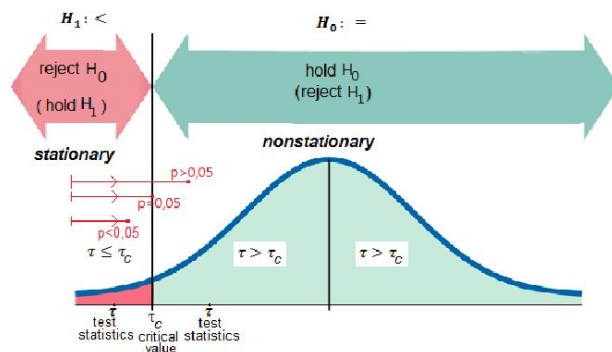
- **period=365** - Defines the seasonal cycle (yearly seasonality).
- **seasonal=13** - Controls the smoothing of the seasonal component.
- **trend=91** - Controls the smoothing of the trend component.
- **robust=True** - Makes decomposition more resistant to outliers.



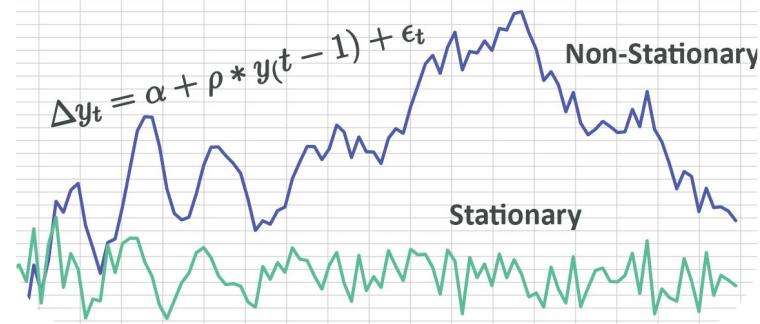
Stationarity Test



```
from statsmodels.tsa.stattools import adfuller
```



Augmented Dickey-Fuller Test (ADF)



Data Transformation

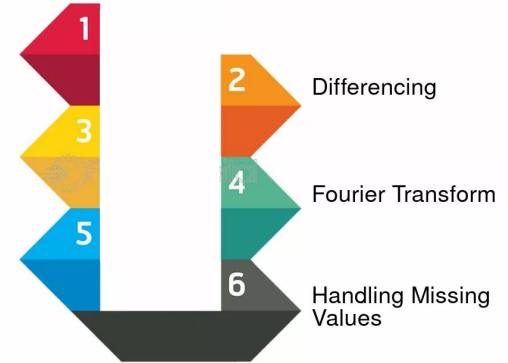
```
data['LogPrice'] = np.log(data['Price'])  
data['LogDiff'] = data['LogPrice'].diff()  
data.dropna(inplace=True)
```



Smoothing
Techniques

Lag Features

Scaling and
Normalization



→ ADF Testing Second time : Check Stationarity Again

Business



Concepts



Data



STL



Stationarity



Transformation



Modeling



Evaluation

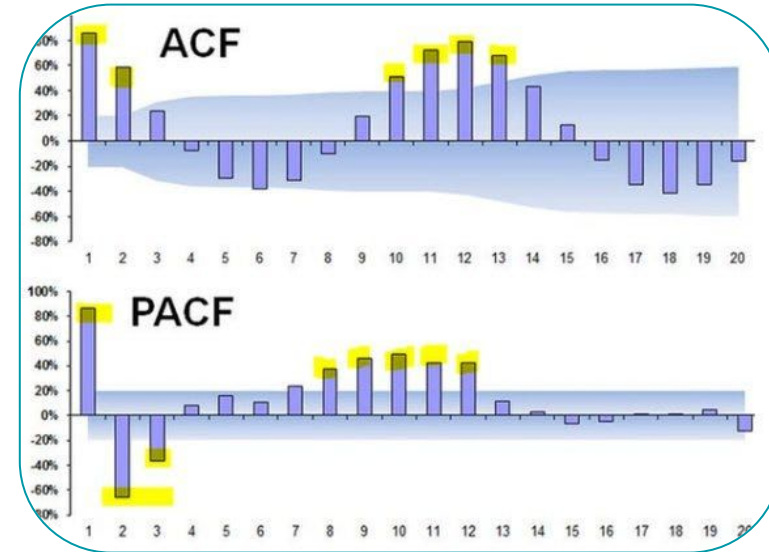


Deployment

ACF / PACF Plots

```
from statsmodels.graphics.tsaplots import plot_acf, plot_pacf
```

- **ACF** helps detect the **MA (q)** part – shows how current value is related to past residuals.
- **PACF** helps detect the **AR (p)** part – shows direct relationship with past values.
- Helps in **model selection** by identifying significant lags.



Auto.Arima (ARIMA / SARIMA)

```
from pmdarima import auto_arima

model1 = auto_arima(data['LogPrice'], start_p=1, start_q=1,
                    max_p=5, max_q=5, seasonal=False,
                    d=1, trace=True, error_action='ignore',
                    suppress_warnings=True, stepwise=True)
```

- **start_p / start_q**: Starting point for AR (p) & MA (q) terms.
- **max_p / max_q**: Max values to search for optimal p & q.
- **d=1**: Degree of differencing to make data stationary.
- **seasonal=False**: Disables seasonal component.
- **stepwise=True**: Fast model search using stepwise algorithm.
- **trace=True**: Prints progress during model fitting.



Manual Fitting with for Loop (multiple ARIMAs)

- **Model Comparison:** Lets you compare different (p,d,q) combinations for best accuracy.
- **Fine-tuning:** Gives more control over model behavior for better forecasting.
- **Insights:** Helps understand how different AR/MA terms affect the data fit.

Model

ARIMA (0,1,0)

ARIMA (1,1,0)

ARIMA (0,1,1)

ARIMA (0,2,0)

Manual Fitting with for Loop (multiple ARIMAs)

```
# Multiple_ARIMAs  
params = [  
    (2, 1, 2),  
    (2, 1, 0),  
    (0, 1, 0),  
    (0, 1, 1),  
    (0, 1, 2),  
    (2, 1, 1),  
    (1, 1, 0),  
    (1, 1, 1)  
]
```



→ ACF/PACF (p, q parameters)

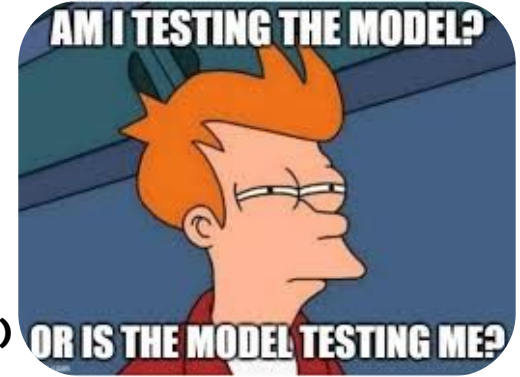
```
from statsmodels.graphics.tsaplots import plot_acf
```

→ Residual Significance (Ljung Box)

```
from statsmodels.stats.diagnostic import acorr_ljungbox
```

→ Mean Square & Absolute errors

```
from sklearn.metrics import mean_squared_error, mean_absolute_error
```



Prediction for NVIDIA Stock

Fund
Manager

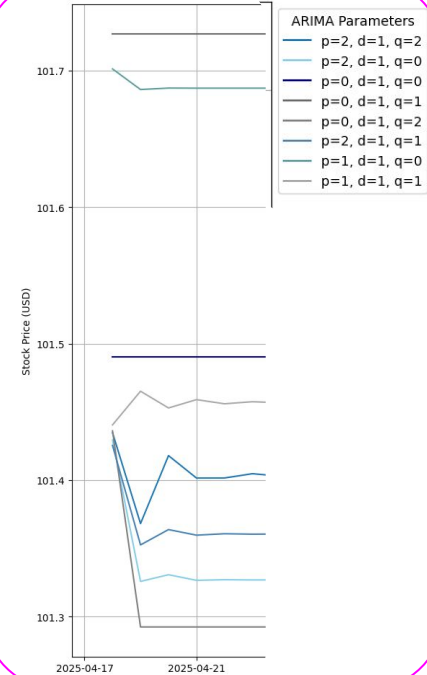
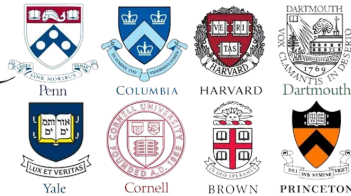


\$500
Millions

+0.2%

\$1 M

10 Ivy



Business



Concepts



Data



STL



Stationarity



Transformation



Modeling



Evaluation

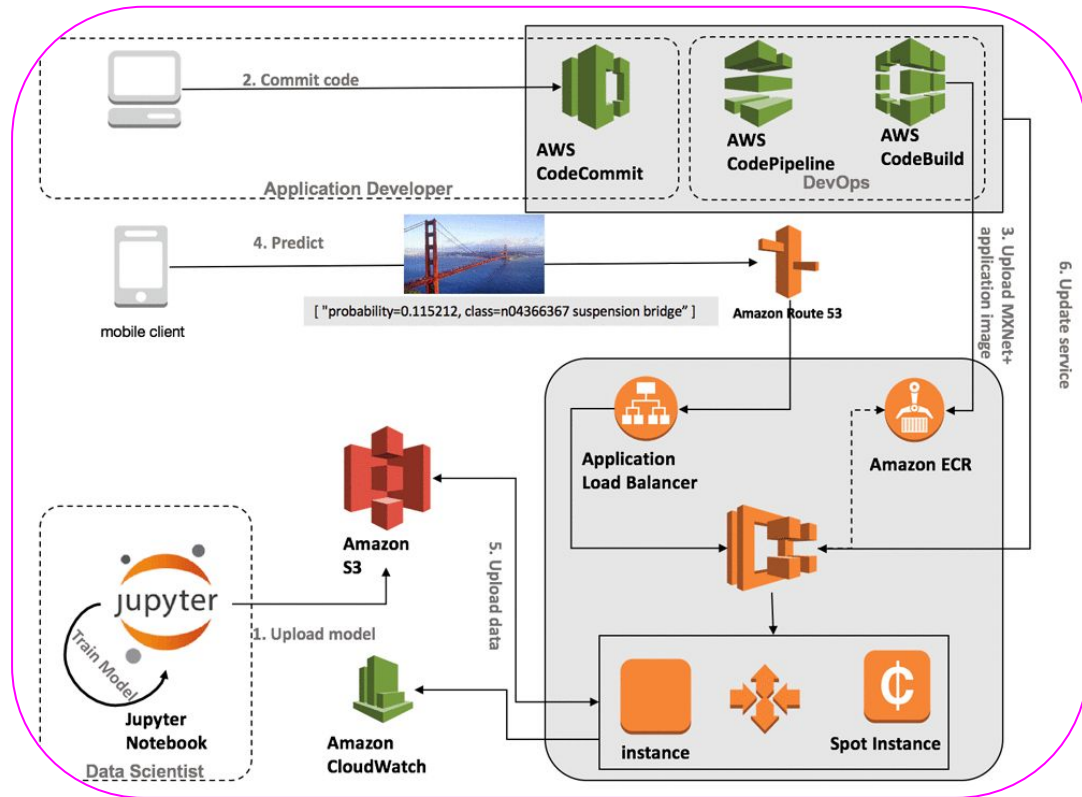


Deployment



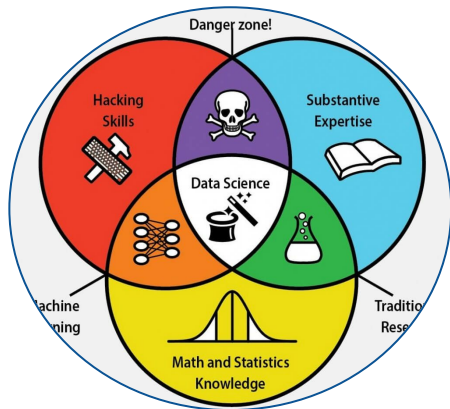
Model Deployment

- **EC2** - Run ARIMA stock prediction model
- **S3** - Store data, models, outputs
- **SageMaker** - Train/deploy ARIMA
- **Lambda** - Serverless prediction function
- **API Gateway** - Create REST API for model
- **CloudWatch** - Monitor logs & metrics
- **IAM** - Manage permissions securely



Outro & What's Next

Topics



Comments



Feedback



Descriptive_Statistics
Pandurang Patil Sindhe
IIT Roorkee (Engineering)
BITS Pilani (Data Science)



IIT Roorkee
(Engineering)



BITS Pilani
(Data Science)



Pandurang Patil Sindhe

LinkedIn

Business



Concepts



Data



STL



Stationarity



Transformation



Modeling



Evaluation



Deployment

ML by PATIL