**Model Development Phase Template**

|  |  |
| --- | --- |
| Date | 15 JULY 2024 |
| Team ID | 739826 |
| Project Title | View count visionary:A data driven approach to forecasting youtube videos views |
| Maximum Marks | 4 Marks |

**Initial Model Training Code, Model Validation and Evaluation Report**

Data Loading: The code starts by loading the YouTube video views dataset from a CSV file.

Data Preprocessing: It includes steps such as cleaning the data, handling missing values, and performing feature engineering to extract relevant predictors for views forecasting.

Model Training: Utilizes a Random Forest Regressor to train the model on the preprocessed data.

**Initial Model Training Code:**

# Import necessary libraries

import pandas as pd

import numpy as np

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

from sklearn.ensemble import RandomForestRegressor

from sklearn.metrics import mean\_squared\_error

# Load dataset

data = pd.read\_csv('youtube\_views\_data.csv')

# Data preprocessing

# (Include code for data cleaning, feature engineering, etc.)

# Split data into training and test sets

X = data.drop(['views'], axis=1)

y = data['views']

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

# Initialize and train the model

model = RandomForestRegressor(n\_estimators=100, random\_state=42)

model.fit(X\_train, y\_train)

# Model evaluation

predictions = model.predict(X\_test)

mse = mean\_squared\_error(y\_test, predictions)

print(f'Mean Squared Error: {mse}')

# Save the trained model

# (Code to save the model for future predictions)

**Model Validation and Evaluation Report:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Model** | **Classification Report** | **Accuracy** | **Confusion Matrix** |
| Decision Tree |  | 0.72 |  |
| Random Forest |  | 0.77 |  |