

## Model Development Phase Template

Date	15 JULY 2024
Team ID	740680
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:

Mod el	Classification Report	Acc urac y	Confusion Matrix
Deci sion Tree	<pre>import numpy as np  # Assuming preds is your predicted values from # Example preds: preds = np.array([1.2, 2.5, 3.7, 4.1, 5.9])  # Convert preds to np.int16 preds = preds.astype(np.int16)  # Reshape and convert to list preds_list = preds.reshape(1, -1)[0].tolist()  print(preds_list)</pre>	0.72	<div>[1, 2, 3, 4]</div>

Random Forest	<pre>import joblib from sklearn.preprocessing import StandardScaler  # Initialize the scaler object scaler = StandardScaler()  # ... (Code to fit the scaler using your data, e.g.)  joblib.dump(scaler, "sc.pkl") # Now you can save it</pre>	0.77	<div data-bbox="1141 336 1356 407">['sc.pkl']</div>
---------------	--	------	---