ViewCount Visionary: A Data-Driven Approach to Forecasting YouTube Video Views

1 INTRODUCTION

1.1 Overview

In today's world, social media has become a crucial part of our lives. One such platform that has gained immense popularity over the years is youtube. It is a platform where content creators can share their videos with a global audience, and advertisers can promote their products and services through video ads. Advertisers pay content creators based on the number of adviews and clocks their ads receive In recent years, advertisers have been looking for ways to estimate the number of adviews based on other matrics like comments, likes, dislikes, duration, year and category of the video. This is where machine learning comes in. by training various regression models on the available data, we can predict the number of adviiews that an advertisement is likely to receive.

Catering to the problem stated above, we have developed a model which can predict the number of adviews given the specified inputs. This model can be used by advertisement agencies on deciding whether or not give a particular content creator on youtube to gove the contract for a particular ad or not based on the number of ad-views predicted by our model.

1.2 Purpose

The purpose of this project is to develop and evaluate various regression models to accurately predict the number of adviews for Youtube advertisements based on the engagement metrics. Additionally, the project aims to deploy the best performing model using the Flask framework. Allowing advertisers and content creators to input relevant metrics and obtain an estimated number of ad-views. This can provide valuable insights to stakeholders, helping them to make informed decisions and improve their marketing strategies.

2 LITERATURE SURVEY

2.1 Existing problem

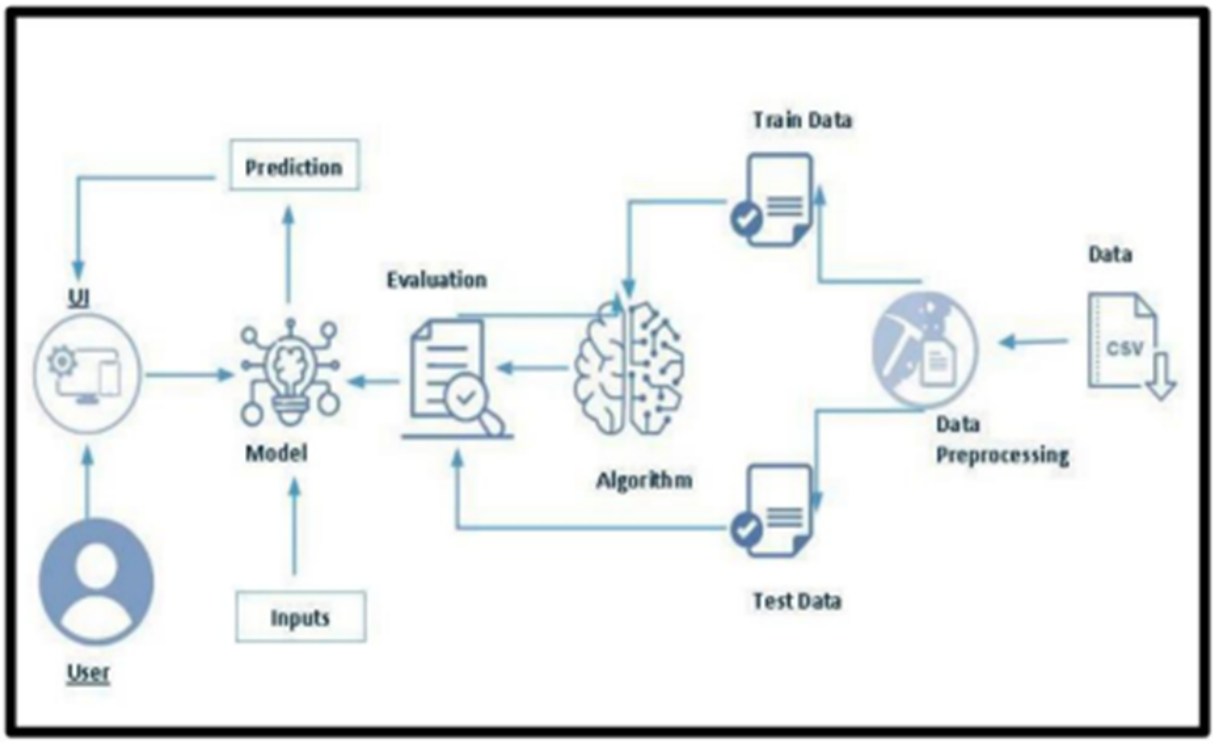
1. Time-Series Analysis**.**
2. ARIMA (Auto Regressive Integrated Moving Average)**.**
3. Machine Learning Models**.**
4. Regression Models**.**
5. Random Forests and Decision Trees**.**
6. Neural Networks.
7. Content-Based Features.

2.2 Proposed solution

The aim of this project is to develop, evaluate and deploy the best performing regression model to predict the number of adviiews for Youtube advertisements based on ther engagement metrics using Flask Framework.

3 THEORITICAL ANALYSIS

3.1 Block diagram



3.2 Hardware / Software designing

Hardware Requirements:

1. **CPU**: A modern multi-core CPU is essential for data preprocessing, feature engineering, and model training. T
2. **RAM**: Sufficient RAM is crucial. Depending on your data size.
3. **Storage**: Adequate storage space is required for storing your dataset and any intermediate files. High-speed SSDs are recommended for improved data read/write performance.

Software Requirements:

1. **Operating System**: Your choice of operating system may depend on personal preference,we done this using windows 8.
2. **Python**: Python is the most commonly used programming language for data-driven forecasting

projects. Install Python and manage packages using a package manager like Visual Studio and Collab.

3. **Front End:** Html, CSS, Bootstrap.
4. **Back End:** Flask, Python.

4 EXPERIMENTAL INVESTIGATIONS

When working on a data-driven approach to forecasting YouTube video views, thorough analysis and investigation play a crucial role in understanding the data, identifying patterns, and refining the forecasting model.

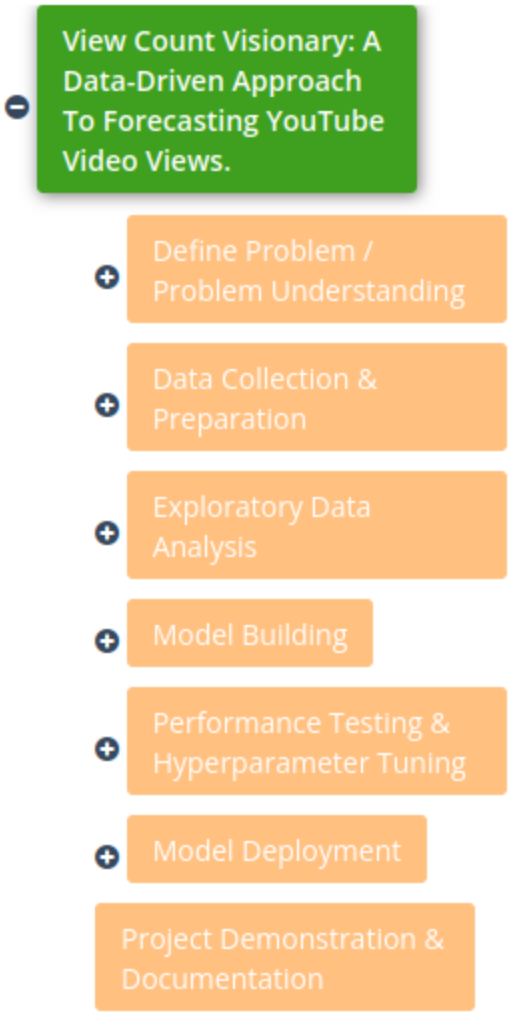
1. Data Exploration, Descriptive Statistics**,** Data Visualization.
2. Time-Series Analysis**,** Feature Importance**,** Lag Analysis**.**
3. Seasonal Patterns**,** User Behavior**,** Model Residuals**,**
4. Hyperparameter Tuning **,**Cross-Validation**,** Error Metrics**.**
5. Anomaly Detection**,** Feature Engineering Iteration**.**
6. Model Interpretability**,** Predictive Maintenance**.**
7. Documentation and Reporting**.**

-

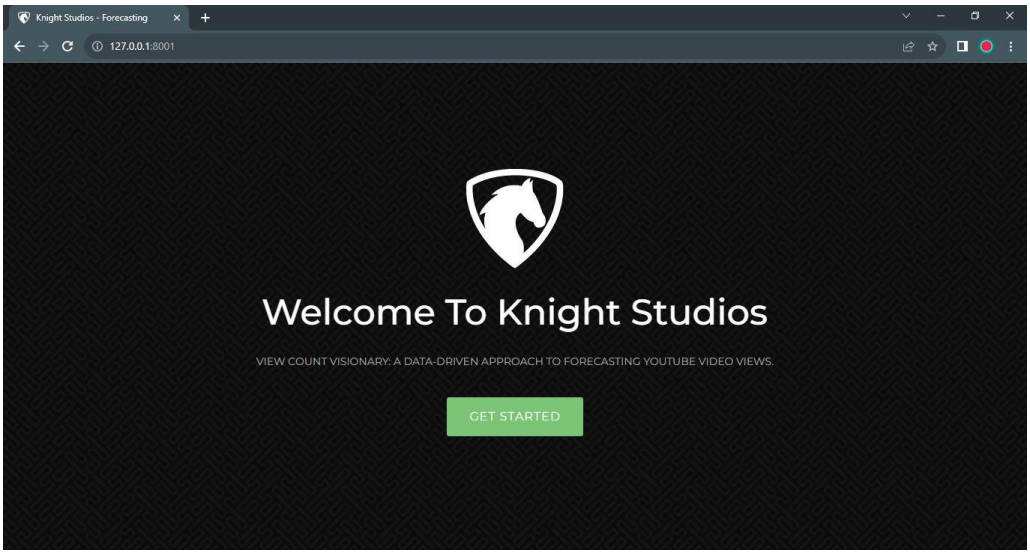
-

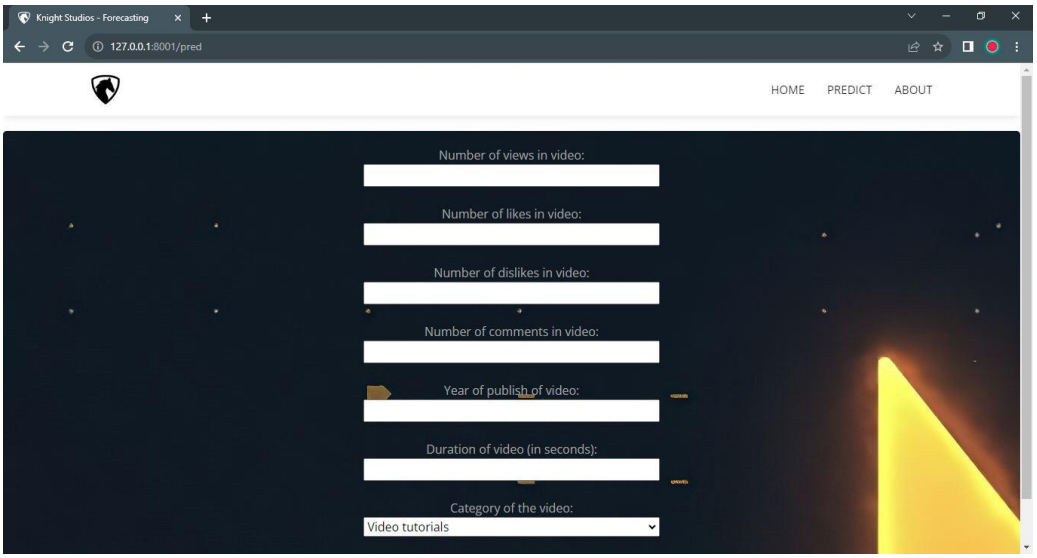
-

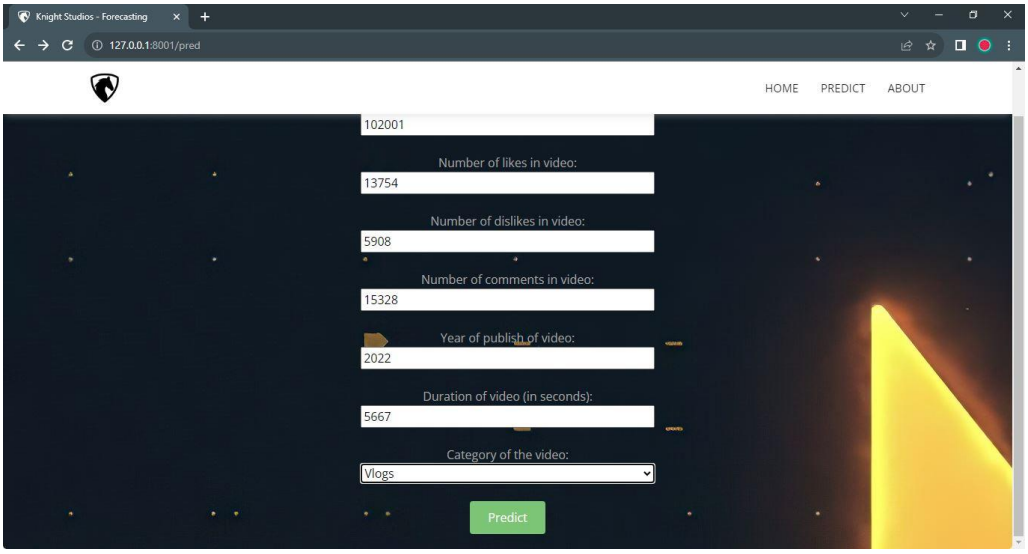
5 FLOWCHART

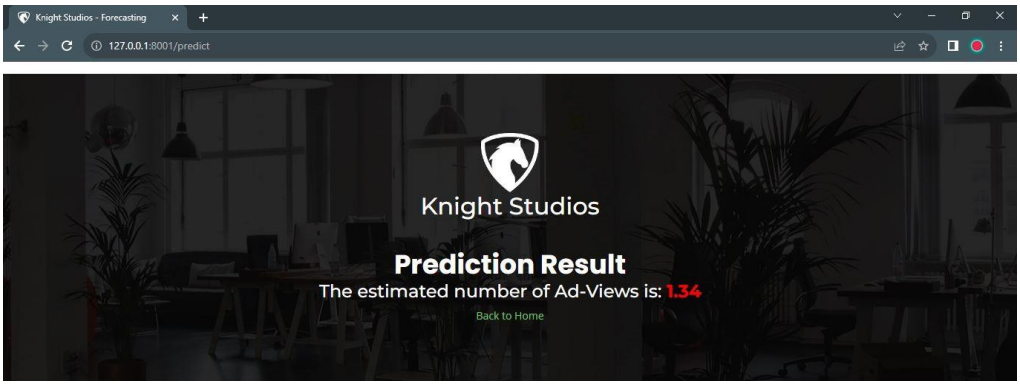

View Count Visionary: A Data-Driven Approach To Forecasting YouTube Video Views.
Define Problem / Problem Understanding
Data Collection & Preparation
Exploratory Data Analysis
Model Building
Performance Testing & Hyperparameter Tuning
Model Deployment
Project Demonstration & Documentation

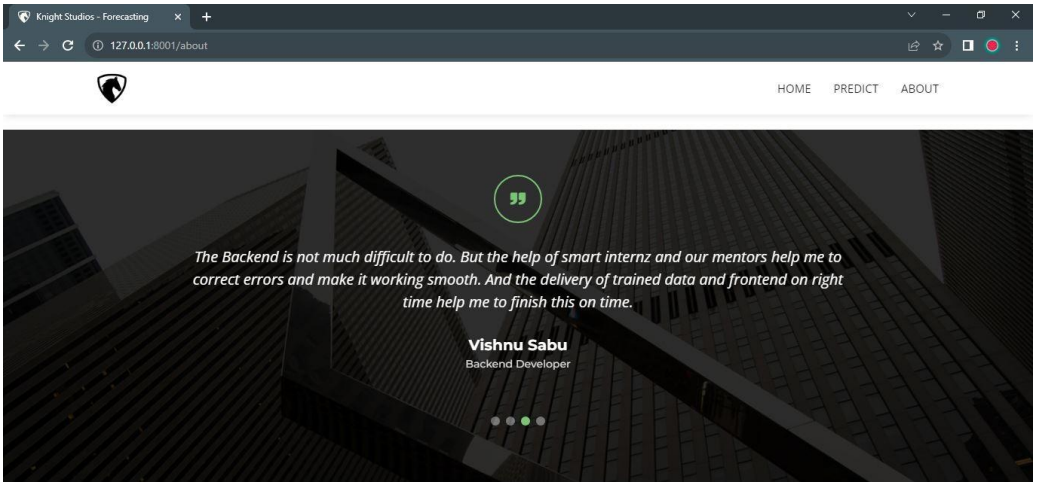
6 RESULT

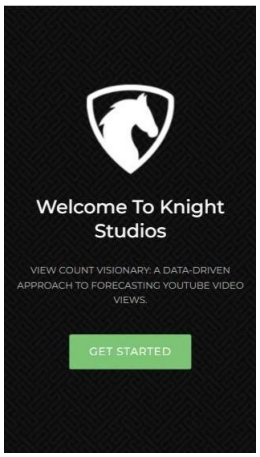

Knight Studios - Forecasting
127.0.0.1:8001
Welcome To Knight Studios
VIEW COUNT VISIONARY: A DATA-DRIVEN APPROACH TO FORECASTING YOUTUBE VIDEO VIEWS.
GET STARTED


Knight Studios - Forecasting
127.0.0.1:8001/pred
HOME
PREDICT
ABOUT
Number of views in video:
Number of likes in video:
Number of dislikes in video:
Number of comments in video:
Year of publish of video:
Duration of video (in seconds):
Category of the video:
Video tutorials


Knight Studios - Forecasting
127.0.0.1:8001/pred
HOME
PREDICT
ABOUT
102001
Number of likes in video:
13754
Number of dislikes in video:
5908
Number of comments in video:
15328
Year of publish of video:
2022
Duration of video (in seconds):
5667
Category of the video:
Vlogs
Predict

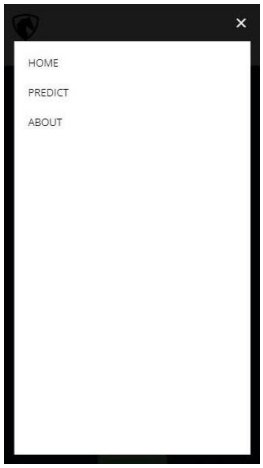

Knight Studios - Forecasting
127.0.0.1:8001/predict
Knight Studios
Prediction Result
The estimated number of Ad-Views is: 1.34
Back to Home

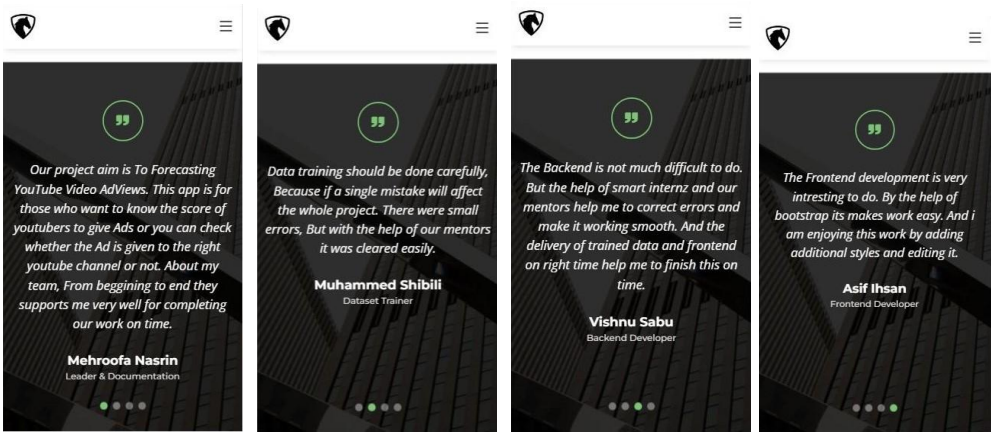

Knight Studios - Forecasting
127.0.0.1:8001/about
HOME
PREDICT
ABOUT
The Backend is not much difficult to do. But the help of smart internz and our mentors help me to correct errors and make it working smooth. And the delivery of trained data and frontend on right time help me to finish this on time.
Vishnu Sabu
Backend Developer


Welcome To Knight Studios
VIEW COUNT VISIONARY: A DATA-DRIVEN APPROACH TO FORECASTING YOUTUBE VIDEO VIEWS.
GET STARTED


23144323
Number of likes in video:
2342233
Number of dislikes in video:
2344
Number of comments in video:
234333
Year of publish of video:
2023
Duration of video (in seconds):
10343
Category of the video:
Challenge Videos
Predict


Knight Studios
Prediction Result
The estimated number of Ad-Views is: 13.87
Back to Home


HOME
PREDICT
ABOUT



7 ADVANTAGES & DISADVANTAGES

7.1 Advantages

1. Improved Decision-Making, Accurate Predictions, Resource Allocation.
2. Content Strategy Optimization**,** Effective Marketing**,** Scalability.
3. Better Timing**,** Enhanced Monetization, Competitive Advantage, Educational Value.
4. Adaptation to Trends, Engagement Improvement, Content Diversification.
5. Effective Audience Growth Strategies, Data-Backed Sponsorship, Evolving Insights.

2. **Disadvantages**

1. Data Quality Issues**,** Data Privacy Concerns**,** Data Volume and Storage**.**
2. Data Acquisition Costs**,** Data Bias**,** Overfitting**.**

8 APPLICATIONS

1. **Content Creation Strategy:** Video Ideas, Optimal Video Length.
2. **Content Release Timing:** Publishing Schedule, Event Planning.
3. **Promotion and Marketing:** Paid Promotion, Social Media Sharing.
4. **Audience Engagement:** Viewer Retention, Interactive Elements.
5. **Monetization:** Ad Revenue Estimation, Sponsorship Opportunities.
6. **Resource Allocation:** Production Resources, Content Quality.
7. **Competitive Analysis:** Benchmarking, Trend Analysis.

8. **Long-Term Strategy:** Audience Growth, Content Diversification.
9. **A/B Testing:** Experimentation.
10. **Feedback Loop:** Continuous Improvement.

9 CONCLUSION

In conclusion, the project on "A Data-Driven Approach to Forecasting YouTube Video Views" has yielded valuable insights and findings that can significantly enhance content creators' decision-making processes and content strategies on the YouTube platform. Through a thorough analysis of historical data and the implementation of predictive models.In essence, the project provides content creators, marketers, and channel managers with a powerful toolkit for optimizing their YouTube channel's performance. By harnessing the predictive capabilities of data analysis and machine learning, creators can stay ahead of the curve, deliver more engaging content, and achieve greater success in the ever-evolving landscape of online video content.

10 FUTURE SCOPE

The future scope for "A Data-Driven Approach to Forecasting YouTube Video Views" is promising, as advancements in data analytics, machine learning, and the online video industry continue to evolve. As technology and the online video landscape continue to evolve, there will be ongoing opportunities to enhance the accuracy and utility of forecasting models, providing content creators with valuable insights to optimize their content strategies and engage with their audience effectively.

11 APPENDIX

Source Code:

base.html

```
<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta content="width=device-width, initial-scale=1.0" name="viewport">

<title>Knight Studios - Forecasting</title>

<meta content="" name="description">
```

```
<meta content="" name="keywords">

<link href="{{ url_for('static', filename= 'img/hero-logo.png') }}" rel="icon">

<link href="{{ url_for('static', filename= 'img/hero-logo.png')  }}" rel="apple-touch-
icon"><linkhref="https://fonts.googleapis.com/css?family=Open+Sans:300,300i,400,400i,600,600i,700,7
00i|Montserrat:300,300i,400,400i,500,500i,600,600i,700,700i|Poppins:300,300i,400,400i,500,500i,600,60
0i,700,700i" rel="stylesheet">

<link href="{{ url_for('static', filename= 'vendor/aos/aos.css') }}" rel="stylesheet">

<link href="{{ url_for('static', filename= 'vendor/bootstrap/css/bootstrap.min.css') }}" rel="stylesheet">

<link href="{{ url_for('static', filename= 'vendor/bootstrap-icons/bootstrap-icons.css') }}"
rel="stylesheet">

<link href="{{ url_for('static', filename= 'vendor/boxicons/css/boxicons.min.css') }}" rel="stylesheet">

<link href="{{ url_for('static', filename= 'vendor/glightbox/css/glightbox.min.css') }}" rel="stylesheet">

<link href="{{ url_for('static', filename= 'vendor/swiper/swiper-bundle.min.css') }}" rel="stylesheet">

<link rel="stylesheet" href="{{ url_for('static', filename= 'css/style.css') }}">

</head>

<body>

{%block content%}

{%endblock%}

<script src="{{ url_for('static', filename= 'vendor/aos/aos.js') }}"></script>

<script src="{{ url_for('static', filename= 'vendor/bootstrap/js/bootstrap.bundle.min.js') }}"></script>

<script src="{{ url_for('static', filename= 'vendor/glightbox/js/glightbox.min.js') }}"></script>

<script src="{{ url_for('static', filename= 'vendor/isotope-layout/isotope.pkgd.min.js') }}"></script>

<script src="{{ url_for('static', filename= 'vendor/swiper/swiper-bundle.min.js') }}"></script>

<script src="{{ url_for('static', filename= 'vendor/php-email-form/validate.js') }}"></script>

<script src="{{ url_for('static', filename= 'js/main.js') }}"></script>

</body>
```

```
</html>
```

nav.html

```
{% extends 'base.html'%}

{%block content%}

<header id="header" class="d-flex align-items-center">

<div class="container d-flex align-items-center justify-content-between">

<div class="logo">

<a href="{{ url_for('index')}}"><img src="{{ url_for('static', filename= 'img/logo.png') }}" alt="" class="img-fluid"></a>

</div>

<nav id="navbar" class="navbar">

<ul>

<li><a class="nav-link scrollto " href="{{ url_for('index')}}">Home</a></li>

<li><a class="nav-link scrollto" href="{{ url_for('pred')}}">Predict</a></li>

<li><a class="nav-link scrollto " href="{{ url_for('about')}}">About</a></li>

</ul>

<i class="bi bi-list mobile-nav-toggle"></i>

</nav

</div>

</header>

{%block main%}

{%endblock%}

{%endblock%}
```

ind.html

```
{% extends 'nav.html'%}

{%block main%}

<div>

<section id="contact" class="contact section-bg">

<div >

<form action="{{ url_for('predict')}}" method="post" class="pf" >

<label>Number of views in video:</label><br>

<input type="number" name="views" required><br><br>

<label>Number of likes in video:</label><br>

<input type="number" name="likes" required><br><br>

<label>Number of dislikes in video:</label><br>

<input type="number" name="dislikes" required><br><br>

<label>Number of comments in video:</label><br>

<input type="number" name="comments" required><br><br>

<label>Year of publish of video:</label><br>

<input type="number" name="year" required><br><br>

<label>Duration of video (in seconds):</label><br>

<input type="number" name="duration" required><br><br>

<label>Category of the video:</label><br>

<select name="category" required>

<option value="1">Video tutorials</option>

<option value="2">Educational videos</option>

<option value="3">Product reviews</option>
```

```
<option value="4">Challenge Videos</option>

<option value="5">Product Launch Videos</option>

<option value="6">Short Videos</option>

<option value="7">Vlogs</option>

<option value="8">Video Testimonials</option>

</select><br><br>

<input type="submit" value="Predict">

</form>

</div></div>

</section>

</div>

{%endblock%}
```

result.html

```
{% extends 'base.html'%}

{%block content%}

<p></p>

<footer id="footer">

<div class="footer-top">

<div class="container">

<div class="row justify-content-center">

<div class="col-lg-6">

<a href="#header" class="scrollto footer-logo"><img src="{{ url_for('static', filename= 'img/hero-logo.png')}}" alt=""></a>

<h2>Knight Studios</h2>

<h3>Prediction Result</h3>
```

```
<h4>The estimated number of Ad-Views is: <span style="color: #ff0000; font-weight: bold;">{{ prediction_text }}</span></h4>

</div></div>

<a href="{{ url_for('index')}}" class="pf">Back to Home</a>

</div></div>

</div></div>

</footer>

{%endblock%}
```

app.py

```
import numpy as np

from flask import Flask, jsonify, request, render_template

import joblib

import pickle

model = pickle.load(open('model.pkl',"rb"))

app = Flask(__name__)

@app.route('/')

def index():

return render_template('home.html')

@app.route('/about')

def about():

return render_template('about.html')

@app.route('/pred')

def pred():

return render_template('ind.html')

@app.route('/predict', methods=['GET','POST'])
```

```
def predict():
views = float(request.form['views'])
likes = float(request.form['likes'])
dislikes = float(request.form['dislikes'])
comments = float(request.form['comments'])
year = float(request.form['year'])
duration = float(request.form['duration'])
category = float(request.form['category'])
features = np.array([[views, likes, dislikes, comments, year, duration, category]])
prediction = model.predict(features)
output =round(prediction[0],2)
return render_template('result.html', prediction_text='{}'.format(output))
if __name__ == '__main__':
app.run(debug=True, port=8001)
```