Week 4 – Flask Deployment Assignment Report

Problem Statement

Customer engagement to online advertisements determines the effectiveness of the advertising strategies that an organization decides to implement. The strategies are affected by the following main factors: advertisement content & characteristics of the target market.

The following data analysis & prediction model aims to determine customer engagement based on the frequency of clicking on the running online advertisement.

In this scenario, the analysis is focused on the target market that reacted (or did not) to the online advertisement.

The key features for consideration:

1. Age
2. Daily Internet Use – amount of time spent online in minutes
3. Daily Site Access – amount of time spent on the website
4. Area Income – average income in the customer's area.

Given the provided dataset, is it possible to determine the survivors based on the key features?

Data Acquisition & Preparation

Data obtained from:

https://www.kaggle.com/imakash3011/customer-personality-analysis/download

Data Intake Report

Name: Deployment on Flask – Advertisement Engagement Predictor

Report date: 25

th October 2021

Internship Batch: <Enter your batch code from Canvas course>

Version: 1.0

Data intake by: Teddy Waweru

Data intake reviewer:

Data storage location:

Tabular data details:

Total number of observations	1000
Total number of files	1
Total number of features	10
Base format of the file	.csv
Size of the data	212KB

Note: Replicate same table with file name if you have more than one file.

Proposed Approach:

- Mention approach of dedup validation (identification)
  Verified unique rows in the dataset.
- Mention your assumptions (if you assume any other thing for data quality analysis)
  The advertisement content was assumed to be neutral for all engagements. Ie. all candidates in the dataset would react in a similar way to the advertisements that were displayed, & there was no particular bias to or against the advertisement.

Data Analysis

The features in the dataset included:

- Daily Time Spent on Site –
- Age
- Area Income – Average Income Level around the customer's area
- Daily Internet Usage
- Ad Topic Line
- City
- Male – Refers to Gender
- Country
- Timestamp – Timestamp of Ad Display
- Clicked on Ad – *Feature to be predicted.*

The following features were utilized in developing the prediction model:

Age, Daily Internet Usage, Daily Time Spent on Site, Area Income

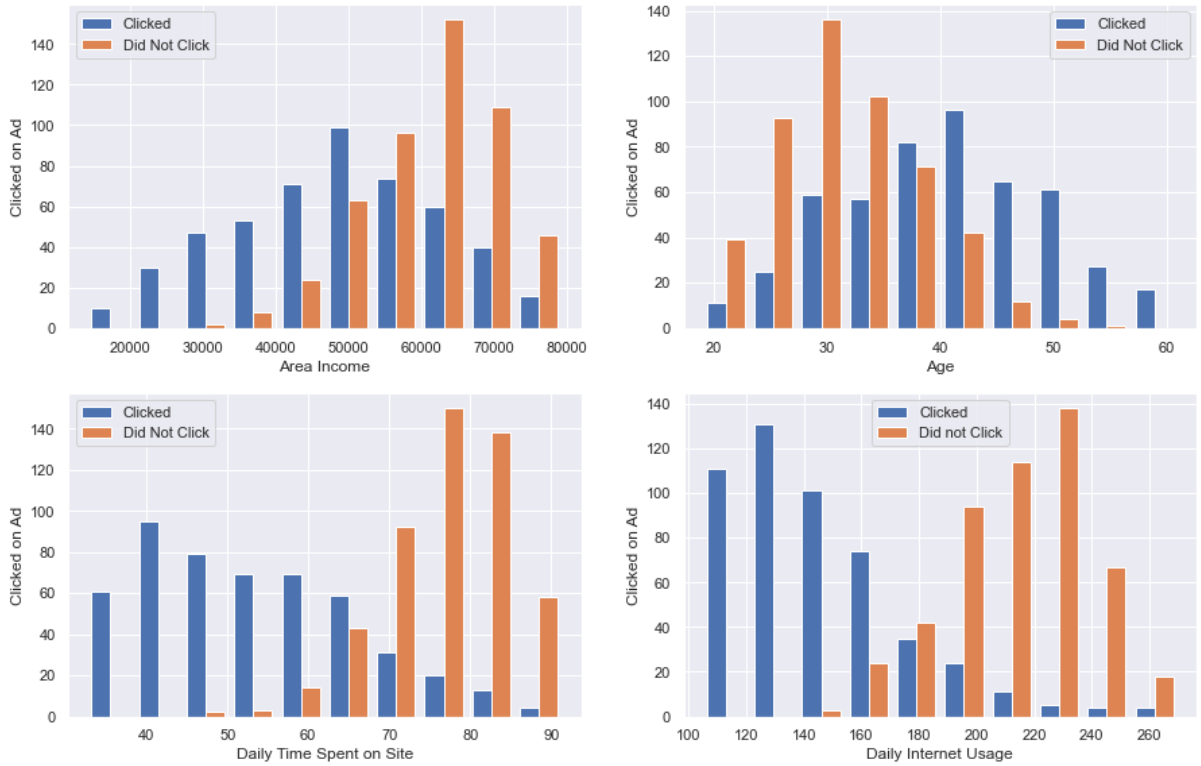

Figure 1: Graphs of Feature to be Predicted against the Feature model against.

The following features were dropped from the modelling dataset:

- Ad Topic Line – Feature held 1000 unique string values ie. each data row held a unique value, & would not have yielded any improvement to the prediction model.
- City – Similar to above: unique string values equal to the entire dataset.
- Country – Held 239 unique values, & would complicate the model further & was automatically dropped.
- Male – (Gender). Dataset was equally split between Male & Female options, showing no usability. Grouping the other features based on Gender showed an equal split as well.

Model creation

The model utilized was the RandomForestTreeClassifier, which would develop leaf nodes dependent on the features selected to generate the model. Based on analysis, it was clear that the features that were selected affected the predicted value in varying ways. The RandomForestTreeClassifier would **develop conclusions** whilst considering the effects of each of these features.

Model training

The data was split automatically by utilizing *scikit-learn.model_selection train_test_split()* module.

Model evaluation

By utilizing *scikit-learn.metrics mean_absolute_error*, the error range for the model was calculated as **0.04**, which would suffice for the application to be developed.

Application Development

Application setup was a basic app.py referencing HTML files, & carrying out the predictions based on the features collected from a HTML form.

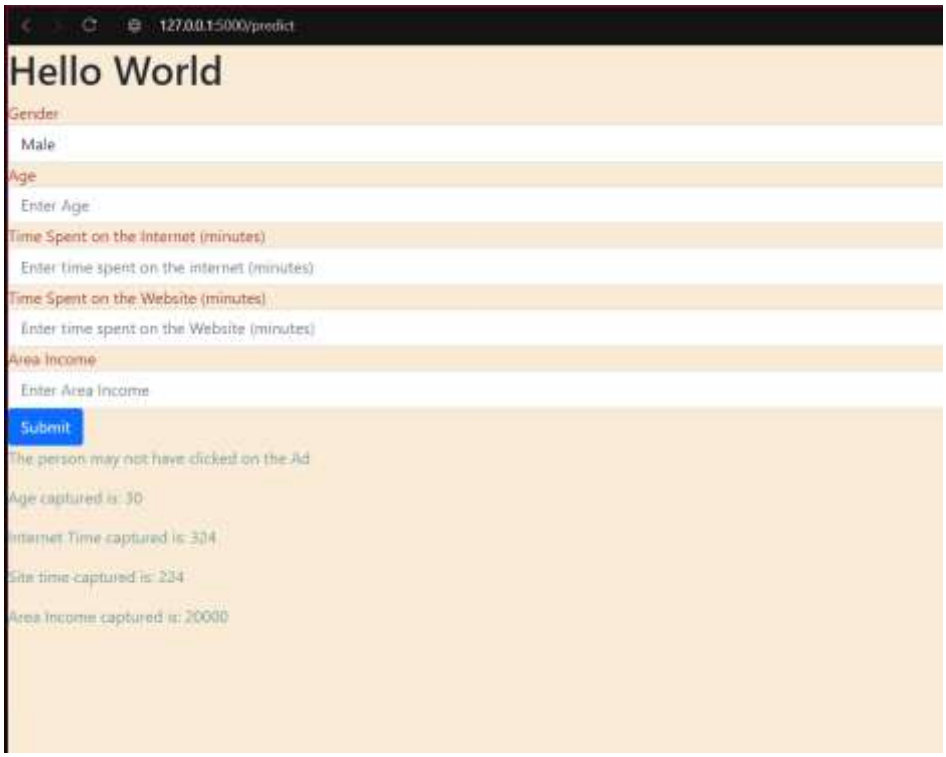

Figure 2: HTML index page, with a form that submits a POST call to the FLASK APP (development mode).

Results & Conclusion

The application runs successfully on the development platform & integrates well with the developed prediction model.

Github Link to Code:

https://github.com/teddywaweru/DataGlacier/tree/main/Week4_Flask_Deployment_Advertising