Final Project (Individual)

Start Assignment

Due Dec 13 by 3am **Points** 100 **Submitting** a file upload **File Types** html

Who's a user?: Building and Deploying a Machine Learning App in Python to Predict LinkedIn Users

Sometimes it seems like the whole world uses LinkedIn. Of course, while it is a popular social networking site and useful for marketing purposes, not everyone uses it. In this project, you are working with the marketing analytics team of your organization and have been tasked with evaluating options for promoting the business on different mediums. Your CEO would like you to analyze data on social media habits among the US public and build a model that takes predicts social media usage--in this case whether someone uses LinkedIn--as a function of individual attributes and demographics. The goal is for the marketing team to use what you create to examine options and platforms to target for marketing campaigns and potential segments of customers. The application must be interactive, publicly hosted, and use machine learning to produce predictions in real-time.

The project consists of two parts: (1) Processing and cleaning the data, analyzing it, and building a supervised classification model; and (2) deploying the model to **Streamlit (https://streamlit.io/)** (on **Streamlit cloud (https://share.streamlit.io/)** via GitHub) so that it is publicly available and the URL can be shared with your colleagues.

Data: **social_media_usage.csv (https://georgetown.instructure.com/courses/177216/files/11635281?wrap=1) (https://georgetown.instructure.com/courses/177216/files/11635281/download?download_frd=1)**

Data dictionary for features and target to use in project: **social_media_usage_README.txt (https://georgetown.instructure.com/courses/177216/files/11635331?wrap=1) (https://georgetown.instructure.com/courses/177216/files/11635331/download?download_frd=1)**

Part 1 (80%): Building a classification model to predict LinkedIn users

1. Read in the data, call the dataframe "s" and check the dimensions of the dataframe
2. Define a function called clean_sm that takes one input, x, and uses `np.where` to check whether x is equal to 1. If it is, make the value of x = 1, otherwise make it 0. Return x. Create a toy dataframe with

three rows and two columns and test your function to make sure it works as expected

3. Create a new dataframe called "ss". The new dataframe should contain a target column called sm_li which should be a binary variable (that takes the value of 1 if it is 1 and 0 otherwise (use clean_sm to create this) which indicates whether or not the individual uses LinkedIn, and the following features: income (ordered numeric from 1 to 9, above 9 considered missing), education (ordered numeric from 1 to 8, above 8 considered missing), parent (binary), married (binary), female (binary), and age (numeric, above 98 considered missing). Drop any missing values. Perform exploratory analysis to examine how the features are related to the target.
4. Create a target vector (y) and feature set (X)
5. Split the data into training and test sets. Hold out 20% of the data for testing. Explain what each new object contains and how it is used in machine learning
6. Instantiate a logistic regression model and set class_weight to balanced. Fit the model with the training data.
7. Evaluate the model using the testing data. What is the model accuracy for the model? Use the model to make predictions and then generate a confusion matrix from the model. Interpret the confusion matrix and explain what each number means.
8. Create the confusion matrix as a dataframe and add informative column names and index names that indicate what each quadrant represents
9. Aside from accuracy, there are three other metrics used to evaluate model performance: precision, recall, and F1 score. Use the results in the confusion matrix to calculate each of these metrics by hand. Discuss each metric and give an actual example of when it might be the preferred metric of evaluation. After calculating the metrics by hand, create a classification_report using sklearn and check to ensure your metrics match those of the classification_report.
10. Use the model to make predictions. For instance, what is the probability that a high income (e.g. income=8), with a high level of education (e.g. 7), non-parent who is married female and 42 years old uses LinkedIn? How does the probability change if another person is 82 years old, but otherwise the same?

Part 2 (20%): Deploying the model on Streamlit

You will now use the model you have developed in part 1 to make live predictions, given a set of inputs set by users of your application. You will use your code and build an application in streamlit. To do so, you must move your code into a .py script file, create a virtual environment with required packages, create a git repository locally, create a git repository remotely on GitHub, push your local repo (.py script, requirements.txt file with packages, and dataset) to a GitHub repo, and host it on Streamlit cloud using your GitHub repo. The app should take user input for the features included in the model. The app should return (1) whether the person would be classified as a LinkedIn user or not and (2) the probability that the person uses LinkedIn.

Two required submissions:

(1) A notebook with all code and answers to Part 1 submitted in HTML form (Part 1)

(2) A public URL where we can find and use your app (Part 2)

In addition to the core questions above, the project grade will be based on the following standard:

- Organization: 20%
- Flow of content: 20%
- Use of visuals/data visualization 20%
- Content and data clarity: 20%
- Depth of content engagement: 20%

Note: The data used come from Pew and, while publicly available, are to be used for educational purposes only.

Final Project Rubric		
Criteria	**Ratings**	**Pts**
Oragnization		20 pts
Flow of content		20 pts
Use of visuals/data visualization		20 pts
Content and data clarity		20 pts
Depth of Content Engagement		20 pts
		Total Points: 100