Income Predictions

Daniel Adamson
Em Greene
Geoff Hoehn
Jason Naper

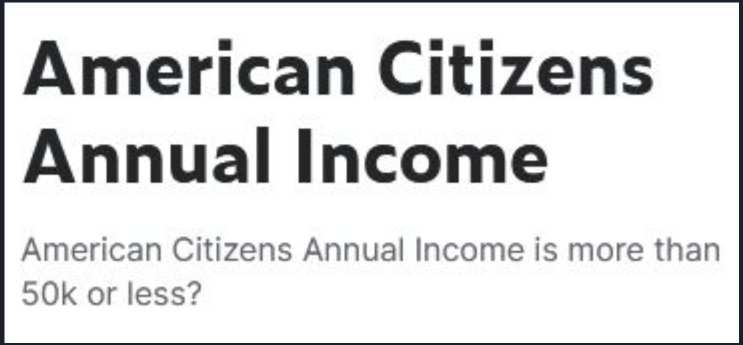
Data Details

-Gathered original dataset from Kaggle

- approx 23000 entries

- 15 columns

- we used this data to train a Neural Network model to see if we could predict income



American Citizens Annual Income

American Citizens Annual Income is more than 50k or less?

Data Cleaning

-Used Spark SQL to process the data and drop the columns

- we dropped capital gain, capital loss, fnlwgt, and education_num

- Also Used Pandas to code ?'s as NA's and drop them

Encode & Standardize

- Used oneHotEncoder to encode the strings into columns with numbers
- Used StandardScaler to scale the data to be uniform

NN Model

We used a keras Neural Network model with:

- 3 layers with
  - 25,12, and 6 nodes for each layer
- Each layer was run with relu, and the output layer is sigmoid
- 50 epochs

Optimization attempts:

- Attempted changing epochs
- Attempted changing number of layers
- Attempted changing number of nodes

The Webpage: Front-End and Back-End

- Users can interact with the neural network through a web application, made with HTML and JavaScript.
- User inputs are recorded for each feature the neural network uses to make its predictions.
- Inputs are recorded as a JSON object, then accessed using Flask.
- In our Python code, the JSON object is converted to a DataFrame and provided to the neural network, which is saved as a .PKL file.
- The network makes its predictions, jsonifies the prediction, and returns it to the web application.
- The JavaScript code interprets the prediction to indicate income falls in the >50K or <=50K categories, then prints the prediction.

Let’s See it