Link to dataset: https://archive.ics.uci.edu/dataset/2/adult

1. Background information (1-2 paragraphs)

   My project focuses on using demographic data such as race, sex, education, and occupation to predict whether someone makes over 50K/year(high income) or less than 50K/year(low income). This is a classification problem. Most of the 14 characteristics of the data are pretty self-explanatory, the only confusing one being fnlwgt which represents the proportion of a certain demographic in the population.

   The data was imported directly to colab using ucimlrepo. I also imported many python libraries that aided in looking and cleaning the data, such as pandas, libraries that created various machine learning models, and libraries that could measure various metrics. I first looked at the data by using nunique() to see how the missing data was encoded and to see if there were any duplicate data. I fixed the data in income to only have two options and replaced the missing data with np.nan.

2. Problem statement (1-2 sentences)

   Demographic data can be used to predict whether someone will eventually earn more or less than 50K. People can enter in their information and figure out what they must do to have their projected income be over 50K.

3. Methods (1 or more paragraphs, feel free to include different models you tried but emphasize on the model(s) you choose to report the results on)

   I started by making a neural network as this is a classification model. I first converted all of my string variables into integer values using a label encoder. I then split my data into training and testing. I made a multi-layer-perceptron classifier with 3 hidden layers and 5 neurons in each layer, set it to a random state of 1, and put it to 100,000 max iterations. I then fit the model to the training data. I then put the xtest data in the model and compared the predicted values to ytest. I printed the accuracy score and then plotted a confusion matrix using metrics and it showed the true positives, false positives, true negatives, and false negatives.

   I was unsatisfied with the results of my MLP, so I used a K-Nearest-Neighbors Classifier to classify the data. I set my n_neighbors to 13 and I changed the weighting from uniform to instead rely on distance. I then used a similar procedure to the model above and found the accuracy score and confusion matrix.

4. Results and Discussion (One or more paragraphs, include any results).

   This dataset inherently had a lot of bias in it because most of the data was of low income earners. 37155 data points were low income and 11687 data points were high income. Because of this split in the data, the machine learning models had relatively high accuracies just by favoring low income earners. Roughly 76% of the data was low income, so by automatically assuming the data point was low income, the machine learning model could have a 76% accuracy. The MLP model had an accuracy of 77%, but it was clearly biased because the false negative rate was so high at 96%. The K-nearest neighbors model had a slightly better accuracy score at 80%, but the false negative rate was much better at 69%. However, that is still quite high, so in the future I would make sure to find more data points that were high income in order to mitigate effects of the bias in the machine learning model.

5. Any outside resources that you use (cite your sources!)
   https://scikit-learn.org/dev/modules/generated/sklearn.neighbors.KNeighborsClassifier.html
   https://www.split.io/glossary/false-positive-rate/#:~:text=False%20positive%20rate%20(FPR)%20is,falsely%20rejecting%20the%20null%20hypothesis.
   https://scikit-learn.org/dev/modules/generated/sklearn.neural_network.MLPClassifier.html
   https://www.geeksforgeeks.org/how-to-convert-categorical-string-data-into-numeric-in-python/