CLASSIFYING INCOME FROM CENSUS DATA

By: Lauren Simms and Joshua Eli Swick

THE PROBLEM WE ARE TRYING TO SOLVE

We are attempting to predict the income bracket, over $50,000 per year or under $50,000 per year, for a person based on census data as accurately as possible (Binary Classification).

This model focuses on accuracy, as we want the model to accurately predict an individual's income category.

```
38, Private, 215646, HS-grad, 9, Divorced, Handlers-cleaners, Not-in-family, White, Male, 0, 0, 40, United-States, <=50K
53, Private, 234721, 11th, 7, Married-civ-spouse, Handlers-cleaners, Husband, Black, Male, 0, 0, 40, United-States, <=50K
28, Private, 338409, Bachelors, 13, Married-civ-spouse, Prof-specialty, Wife, Black, Female, 0, 0, 40, Cuba, <=50K
37, Private, 284582, Masters, 14, Married-civ-spouse, Exec-managerial, Wife, White, Female, 0, 0, 40, United-States, <=50K
49, Private, 160187, 9th, 5, Married-spouse-absent, Other-service, Not-in-family, Black, Female, 0, 0, 16, Jamaica, <=50K
52, Self-emp-not-inc, 209642, HS-grad, 9, Married-civ-spouse, Exec-managerial, Husband, White, Male, 0, 0, 45, United-States, >50K
31, Private, 45781, Masters, 14, Never-married, Prof-specialty, Not-in-family, White, Female, 14084, 0, 50, United-States, >50K
42, Private, 159449, Bachelors, 13, Married-civ-spouse, Exec-managerial, Husband, White, Male, 5178, 0, 40, United-States, >50K
37, Private, 280464, Some-college, 10, Married-civ-spouse, Exec-managerial, Husband, Black, Male, 0, 0, 80, United-States, >50K
30, State-gov, 141297, Bachelors, 13, Married-civ-spouse, Prof-specialty, Husband, Asian-Pac-Islander, Male, 0, 0, 40, India, >50K
```

STATE OF THE ART APPROACHES

We found one instance where another data scientist used this dataset for income classification. This model used the Python package ‘sklearn' and had an accuracy of 82.5%.

OUR APPROACHES

We are trying to solve a classification problem, with the potential for it to be a clustering problem.

The ‘ <=50k’ and ‘ >50k’ target features were converted to a 0 and 1 respectively in order for them to be represented numerically.

In [11]:
```
dataset['income'].unique()
```
Out[11]:
```
array([' <=50K', ' >50K'], dtype=object)
```
In [12]:
```
list=[]
for income in dataset['income']:
    if income == ' <=50K':
        list.append(0)
    if income == ' >50K':
        list.append(1)

dataset['income'] = list
```

OUR APPROACHES

Numeric features were normailzed and categorical features were encoded.

```
In [86]: training_data.head()
```

Out[86]:

	age	workclass	fnlwgt	education	education-num	marital-status	occupation	relationship	race	sex	capital-gain	capital-loss	hours-per-week	native-country	native-country
24645	0.150685	2	162298	1	0.533333	0	6	3	0	0	0.0	0.000000	0.397959	0	0
19112	0.739726	1	130436	13	0.066667	2	6	0	0	1	0.0	0.000000	0.275510	0	0
11592	0.260274	2	181721	12	0.333333	0	8	3	1	0	0.0	0.000000	0.602041	0	0
5755	0.534247	2	182460	1	0.533333	1	2	1	0	0	0.0	0.000000	0.397959	0	0
31139	0.136986	2	215504	0	0.800000	1	5	1	0	0	0.0	0.424242	0.551020	0	0

THE MODEL

Activation Functions: SoftMax and Relu

Loss Functions: Binary Cross Entropy

Metrics: Accuracy

In [103]:
```
model = keras.Sequential([
    keras.layers.Dense(16, input_shape=(15,), activation=tf.nn.relu),
    keras.layers.Dense(2, activation=tf.nn.softmax)
])
```

In [104]:
```
model.compile(
    optimizer='rmsprop',
    loss='binary_crossentropy',
    metrics=['accuracy']
)
```

In [106]:
```
model.fit(
    training_data,
    to_categorical(training_labels),
    epochs=3
)
```

RESULTS

Accuracy: 76.16%

```
Epoch 1/3
26048/26048 [==============================] - 1s 28us/sample - loss: 3.8223 - acc: 0.7616
Epoch 2/3
26048/26048 [==============================] - 1s 31us/sample - loss: 3.8223 - acc: 0.7616
Epoch 3/3
26048/26048 [==============================] - 1s 28us/sample - loss: 3.8223 - acc: 0.7616

Out[106]: <tensorflow.python.keras.callbacks.History at 0x7f8249420c88>
```

LESSONS LEARNED

Datasets that include numerical data, text labels, and inconsistent values need thoughtful preparation for TensorFlow to interpret the values properly.

Deep learning is still an ever-changing field and the tools and techniques are constantly evolving.