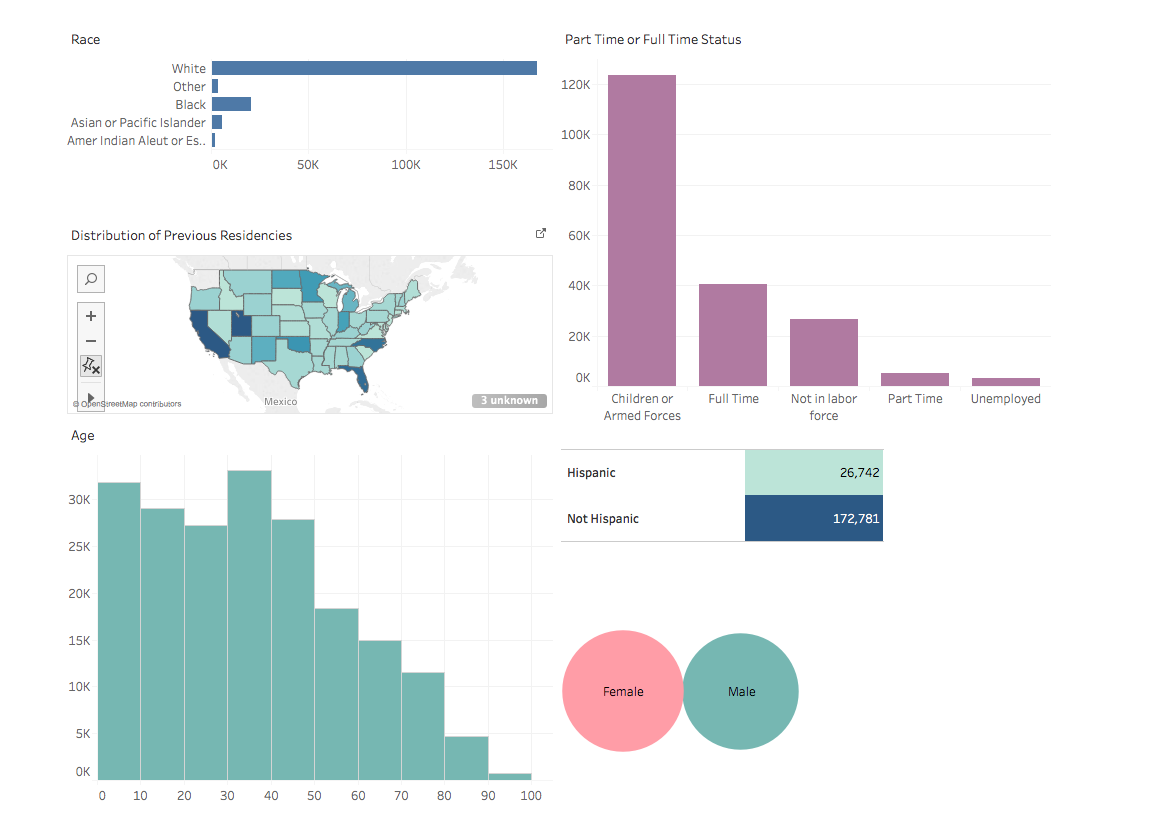
**Dataiku Technical Exercise: Data Scientist Position**

**Leigha Jarett**

**Step 1: Investigating and Visualizing Data**

Before beginning analysis, I wanted to get a feel for the available data. Therefore, I examined the distribution of the data, the maximum and minimum values, and looked to see how much information was missing.

Based on this investigation, along with some ‘common sense’, I was able to determine which variables should be used in the prediction of income status for the test set. I included some descriptive figures, which were created in Tableau, below. You can see the full details of the investigation in the attached Python notebook.



52.9%

47.1%

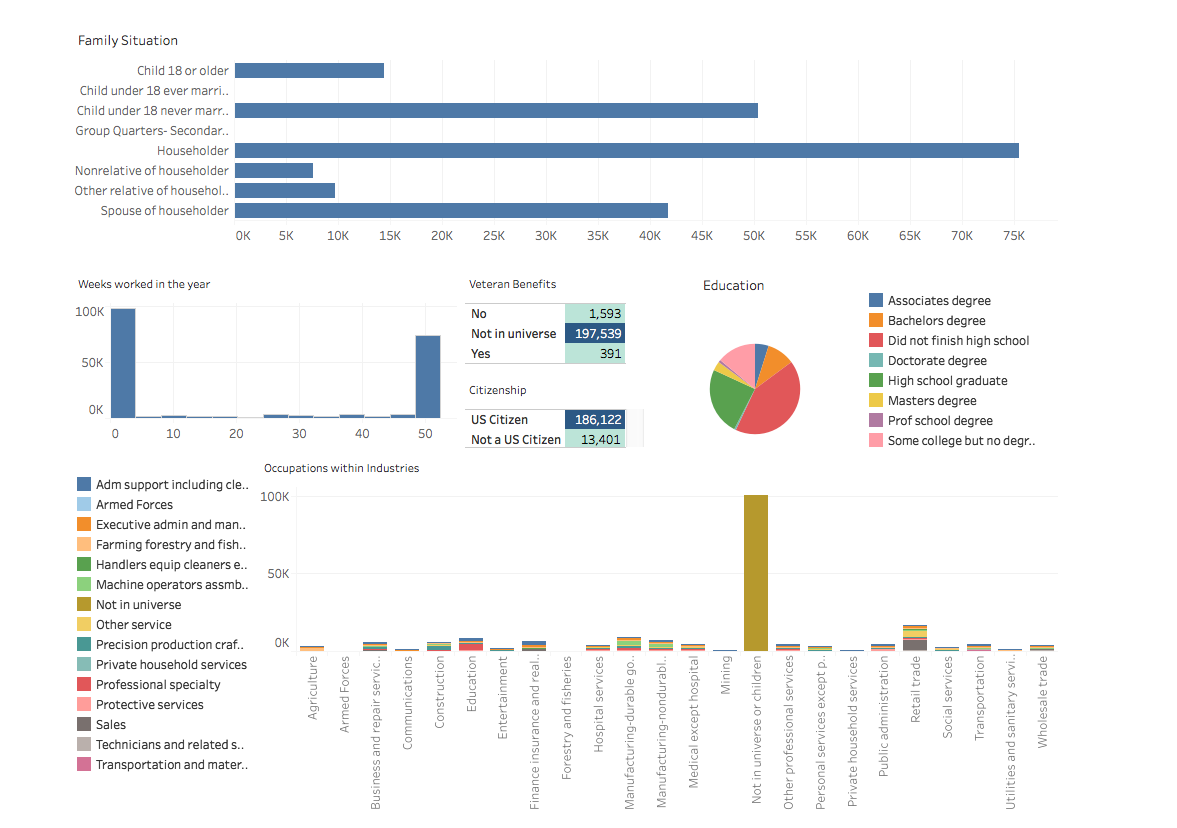
average: 34.5

min = 0

max = 90

Age (years)

average = 23

**Step 2: Input Data**

Max: 90

Min: 0

Mean: 34.5

47.9%

52.1%

--% Missing

Below, I briefly discuss the changes I made to the input data. Red variables were not included, variables not listed were kept in the analysis as is, or changed from character to number.

* Age: grouped into categories (0-16), (16-25), (26-35), (36-45), (46-55), (56-65), (65+) to correct for skewedness
* Work Class: felt that data was not useful for predicting income
* Education: grouped into categories: did not graduate high school, graduate from high school, some college, associated degree, bachelors degree, masters degree, doctoral degree, professional degree
* Wage: mostly zeros, even for those who work, probably because they aren’t paid hourly
* Currently in school: grouped into categories: in school or not in school
* Marital status: grouped into categories: not married or married
* Major Industry and Occupation codes: this information is already covered by the previous industry and occupation codes
* Hispanic: grouped into categories: not Hispanic or Hispanic
* Member of Labor Union: felt that data was not useful for predicting income
* Reason for Unemployment: unemployment information covered in other variables
* Full or part time status: grouped into full time, part time, or other
* Captial gains, capital losses, and dividends from stocks: mostly none
* Tax filer status: information contained in other variables
* Region and state: large amount of missing data
* Detailed household and family set: information already covered in household summary
* Migration codes: relevant citizenship information is covered in citizenship column
* Live in house 1 year ago: not relevant
* Number of persons worked for employer: not relevant
* Family members under 18: covered in household summary
* Country of birth Mother/Father/Self: relevant citizenship information is covered in citizenship column
* Citizenship: make categories citizen or not citizen
* Veteran admin: covered in veteran status column
* Weeks worked in the year: divided into worked more than half the year or less than half the year

**Other Comments:**

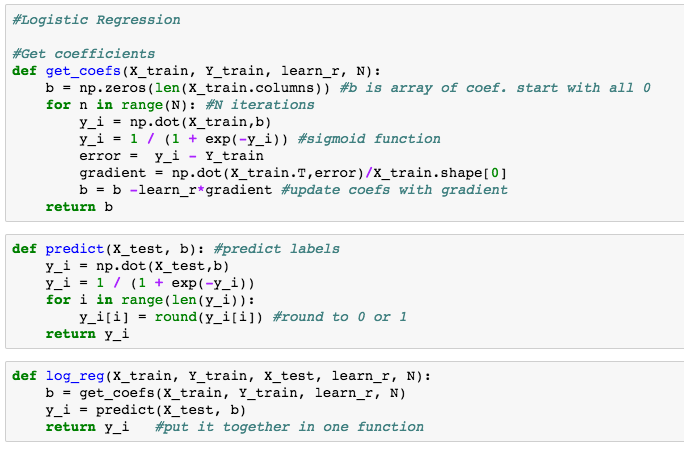
For a more in-depth analysis I would want to individually examine co-linearity of variables by examining the Pearson’s Correlation Coefficient or the Spearman Correlation Coefficient. This would allow us to get rid of variables that strongly correlate with each other.

**Step 3: Cross Validation and Logistic Regression using Stochastic Gradient Descent**

I decided to implement a Logistic Regression classifier to label test data as either above $50,000 income, or below $50,000 income (1 or 0). I used Logistic Regression, with Stochastic Gradient Descent for coefficient optimization, because it is a simple and effective tool for implementing binary classification.

I included a snapshot of my Python code on the next page.

I then performed 3-fold cross validation with the logistic regression classifier, which resulted in an average accuracy of 92%



**Step 4: Using Logistic Regression Classifier on Test Data**

I then applied my trained Logistic Regression model to the testing data, in order to predict which individuals would have an income of above or below $50,000. This resulted in an accuracy of 94%. However, I decided to further investigate the number of true positives, true negatives, false positives and false negatives. I found that while the model gave a high number of TP, there was also a high number of FN, meaning that most of the testing data had income levels below $50,000, so by classifying almost all of the data as below $50,000 is achieved a high accuracy. Therefore, the model might not perform well on data that has more incomes above $50,000. When working on specific projects, it is important to consider if it’s more important to limit false positives or false negatives. This will greatly affect the way we design the model.

From examining the weights of the model, we can see which variables are most significant in classifying new data. The most important variable was if the individual worked full time or part time. The next was if they owned their own business or not. The next was age, and next was race.

**Step 5: Using SKlearn’s decision tree**

In an effort to try and reduce the number of false negatives, I used the sklearn package to fit a decision tree classifier with the training data, and then predict the labels of the testing data.

This new model resulted in an accuracy of 93%. It had a lower number of false negatives, but also had a higher number of false positives then the regression model.

**Final thoughts:** The most difficult part of this exercise, for me, was figuring out which classifiers to use. I began with logistic regression, because it is simple and effective for binary classification. I then tried out a decision tree in an effort to improve the number of false negatives. With more time, I would have liked to implement other classifiers such as KNN.