Census Analysis

A Tech Talent South Production

Powered by Conor and Owen

Table of Contents

1. Overview

What? Who? Why?

Hello!

We are Conor and Owen!

Aspiring Data Scientists hoping to bring some unique insights using historical Census Data

Objective

- Provide high-level analysis of the Census data set
- Predict if an individual's income is more or less than $50k/year

Data Overview

- Extracted from 1994 census bureau database
- 15 original data columns such as age, education, marital.status etc.
    - Transformed based on preliminary analysis (see Data Cleaning and Data Engineering slides)

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 32561 entries, 0 to 32560
Data columns (total 15 columns):
 #   Column          Non-Null Count  Dtype 
---  ------          --------------  ----- 
 0   age             32561 non-null  int64 
 1   workclass       32561 non-null  object
 2   fnlwgt          32561 non-null  int64 
 3   education       32561 non-null  object
 4   education.num   32561 non-null  int64 
 5   marital.status  32561 non-null  object
 6   occupation      32561 non-null  object
 7   relationship    32561 non-null  object
 8   race            32561 non-null  object
 9   sex             32561 non-null  object
 10  capital.gain    32561 non-null  int64 
 11  capital.loss    32561 non-null  int64 
 12  hours.per.week  32561 non-null  int64 
 13  native.country  32561 non-null  object
 14  income          32561 non-null  object
dtypes: int64(6), object(9)
memory usage: 3.7+ MB
```

	age	workclass	fnlwgt	education	education.num	marital.status	occupation	relationship	race	sex	capital.gain	capital.loss	hours.per.week	native.country	income
0	90	?	77053	HS-grad	9	Widowed	?	Not-in-family	White	Female	0	4356	40	United-States	<=50K
1	82	Private	132870	HS-grad	9	Widowed	Exec-managerial	Not-in-family	White	Female	0	4356	18	United-States	<=50K
2	66	?	186061	Some-college	10	Widowed	?	Unmarried	Black	Female	0	4356	40	United-States	<=50K
3	54	Private	140359	7th-8th	4	Divorced	Machine-op-inspct	Unmarried	White	Female	0	3900	40	United-States	<=50K
4	41	Private	264663	Some-college	10	Separated	Prof-specialty	Own-child	White	Female	0	3900	40	United-States	<=50K

2. Preliminary Analysis

Data Cleaning and Visualization

Data Cleaning

- "?" in 'workclass', 'occupation', and 'native.country' columns
  - Change to null values, then fill them with the mode (most commonly appearing value) for that column

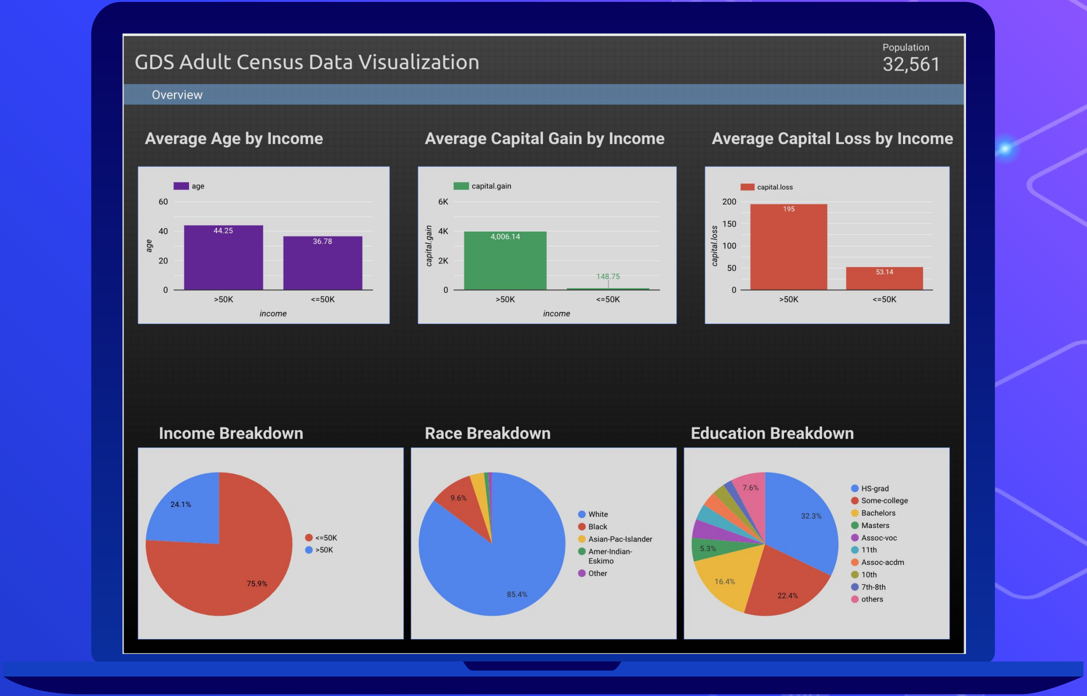
```
data[data == '?'] = np.nan
for column in ['workclass','occupation','native.country']:
    data[column].fillna(data[column].mode()[0], inplace=True)
```

- Income is a categorical variable
  - '<=50K' or '>50K'
- As the target variable, it needs to be numerical

```
#Replace the categorical variables with numerical variables
data['income'] = data['income'].replace({'<=50K':0, '>50K' :1})
```

Data Visualization

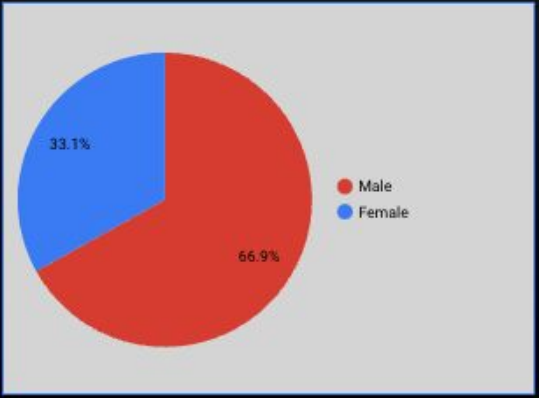
GDS Overview



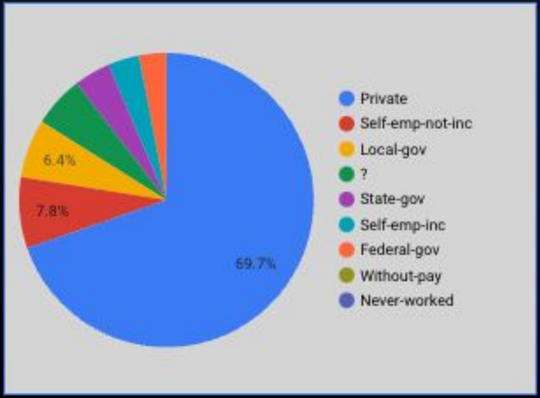
https://datastudio.google.com/reporting/3a9c195c-3a28-4a99-bda7-6acbfa4a068f

GDS Zoom In: Demographics

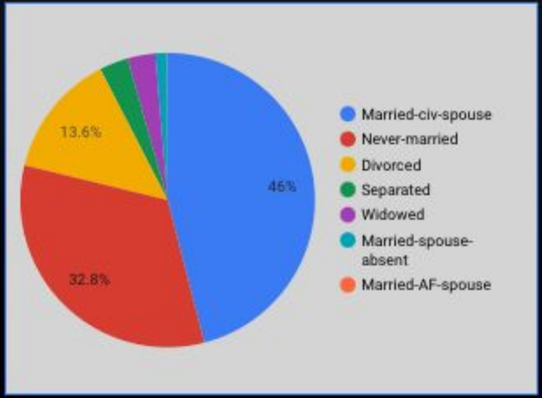
Sex Breakdown



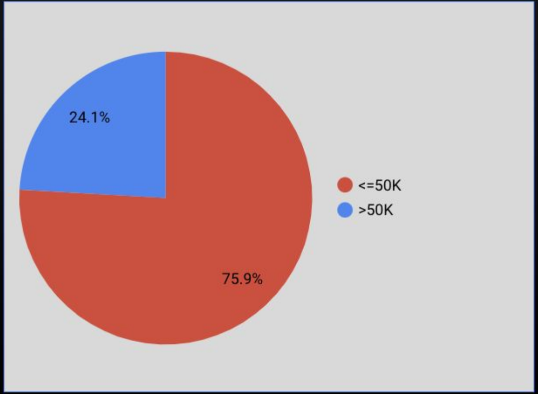
Employment Breakdown



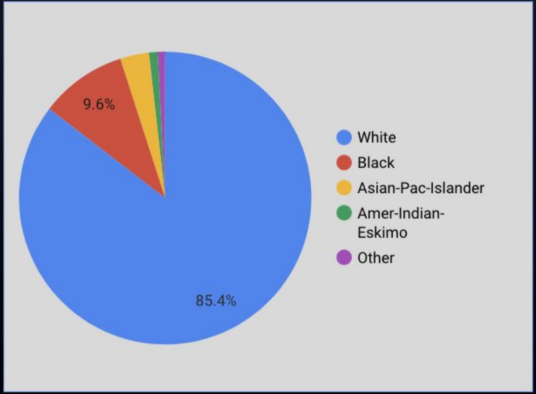
Marital Status



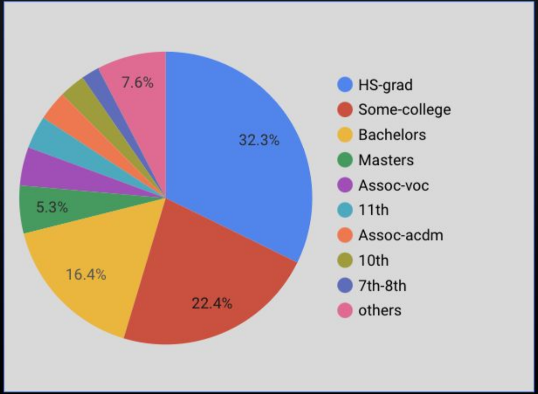
Income Breakdown



Race Breakdown

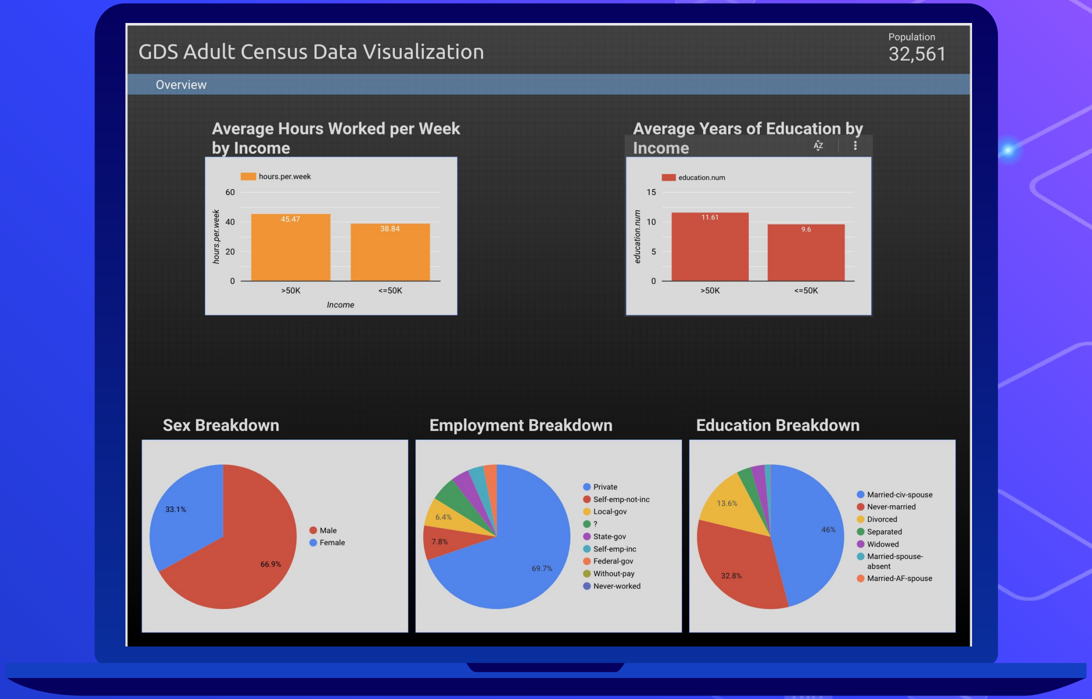


Education Breakdown



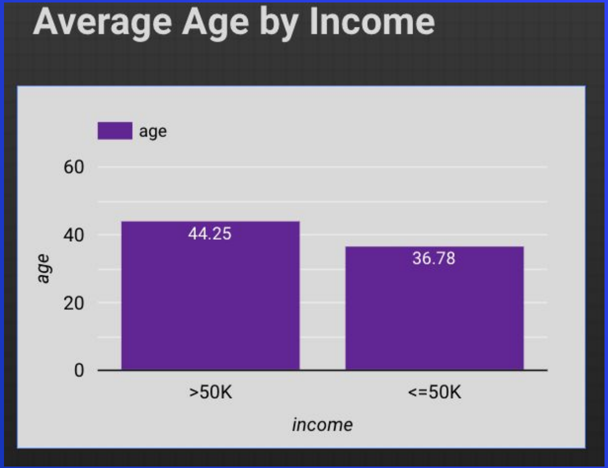
Data Visualization

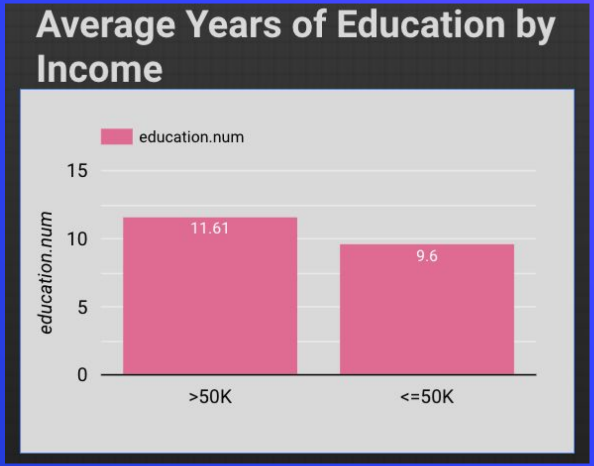
GDS
Overview 2

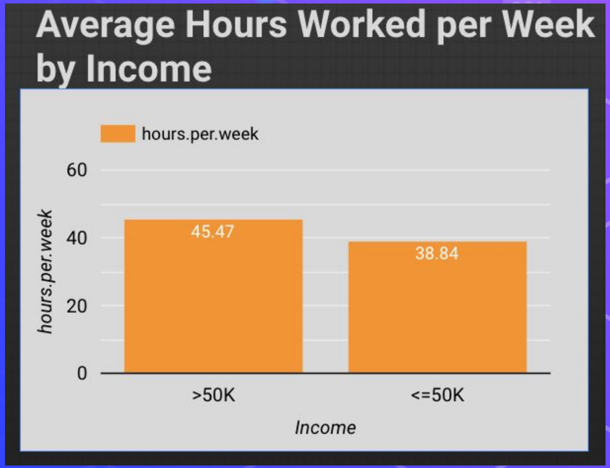


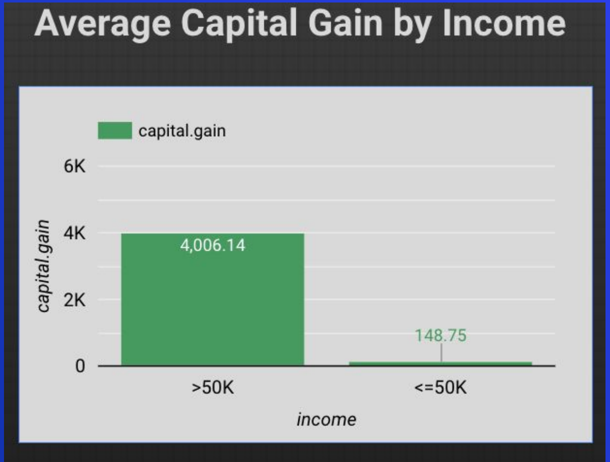
https://datastudio.google.com/reporting/3a9c195c-3a28-4a99-bda7-6acbfa4a068f

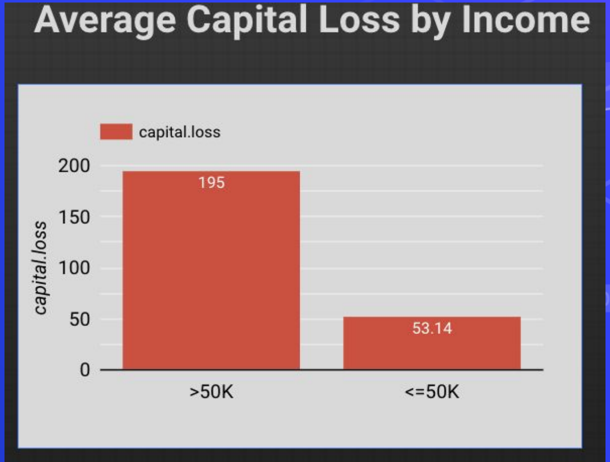
GDS Zoom In: Income







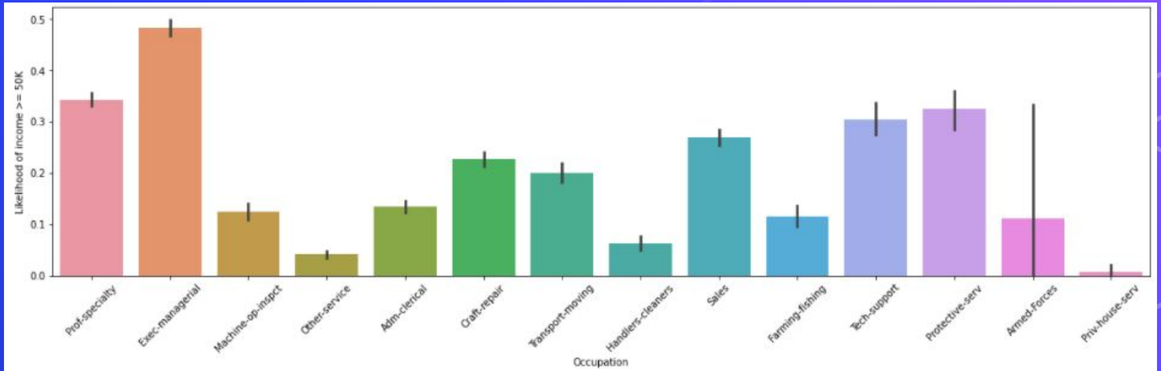


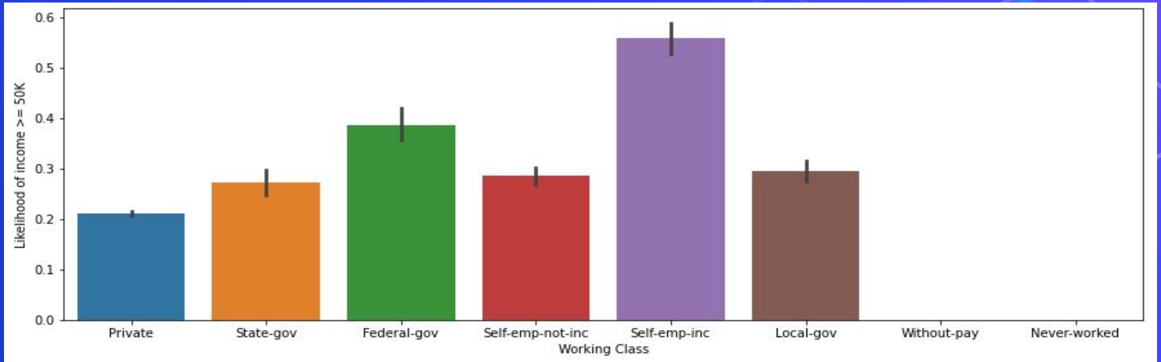


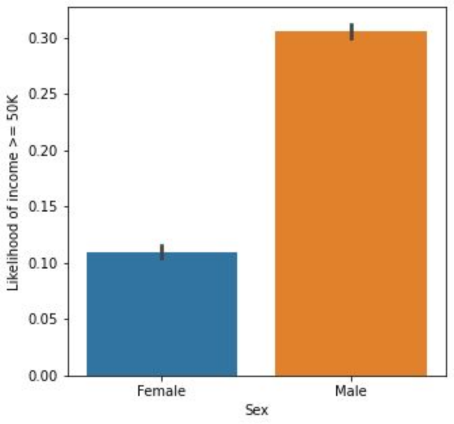
Visuals - Categorical part 1

- Occupation vs Workclass

- Sex

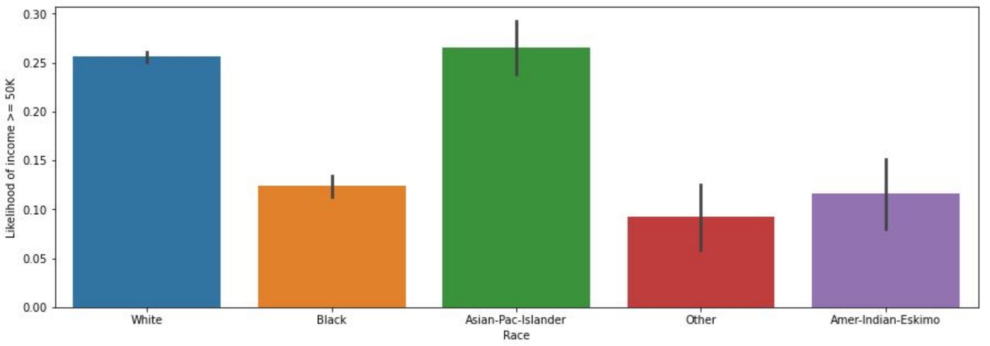




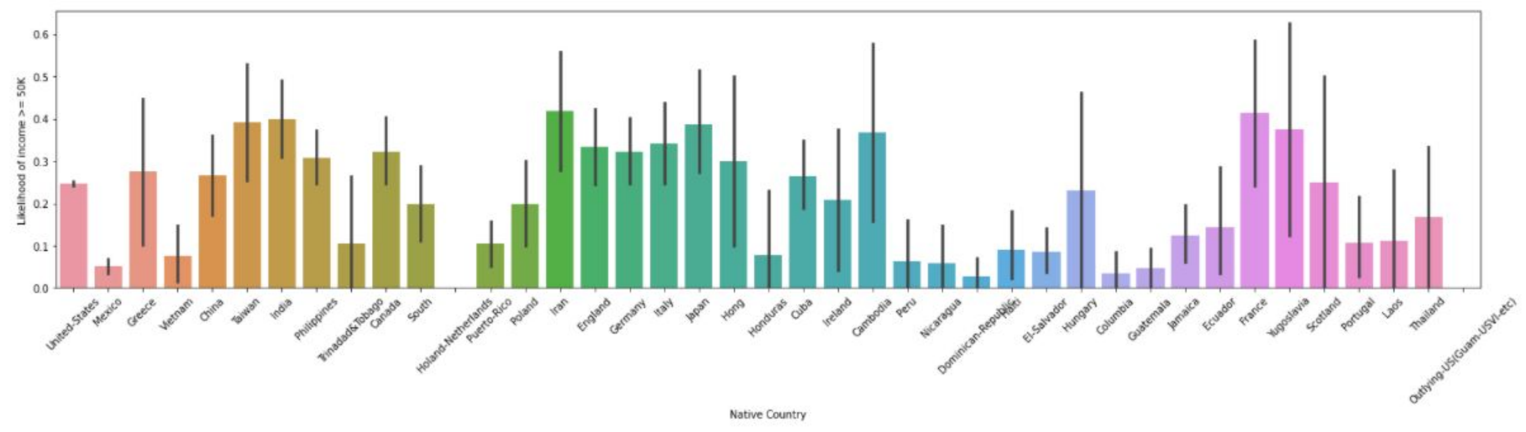


Visuals - Categorical part 2

- Race

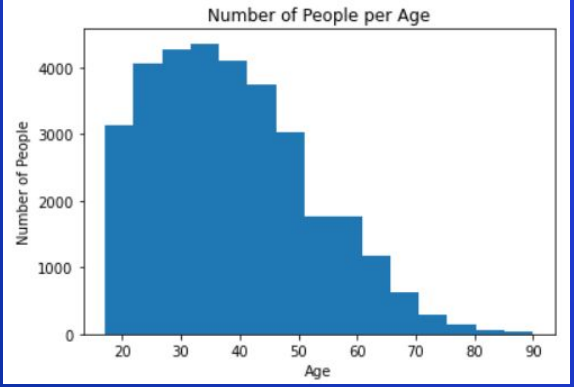


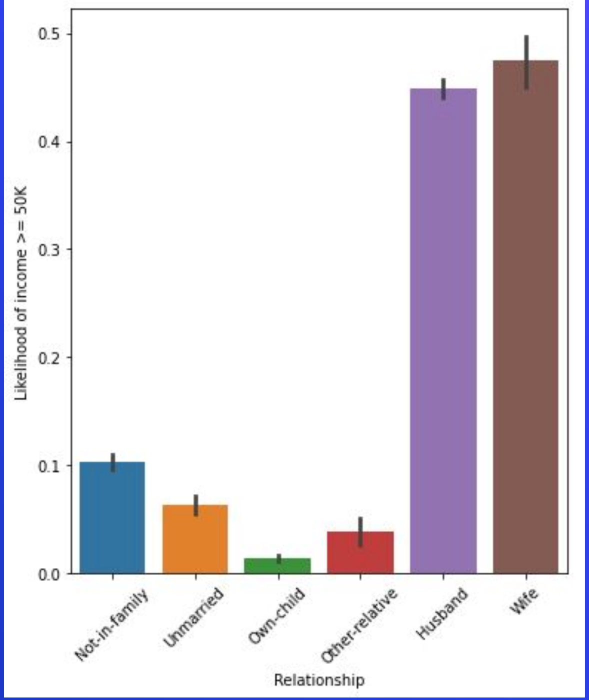
- Native Country

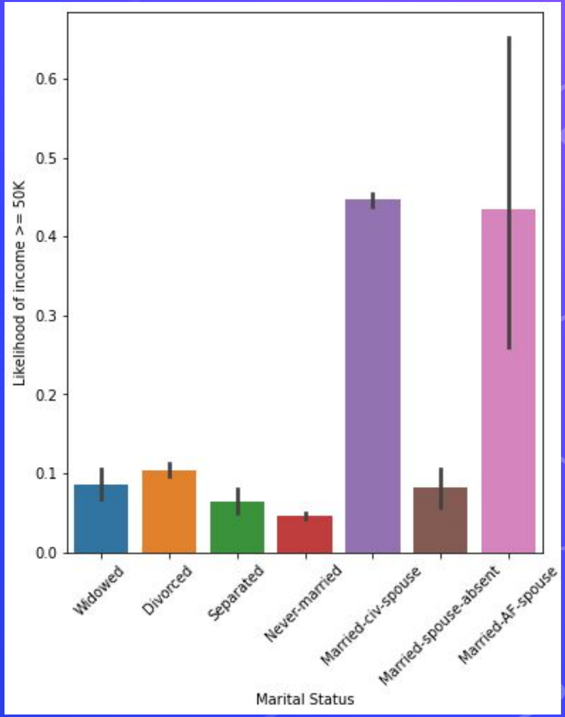


Visuals - Categorical part 2

- Relationship vs Marital Status

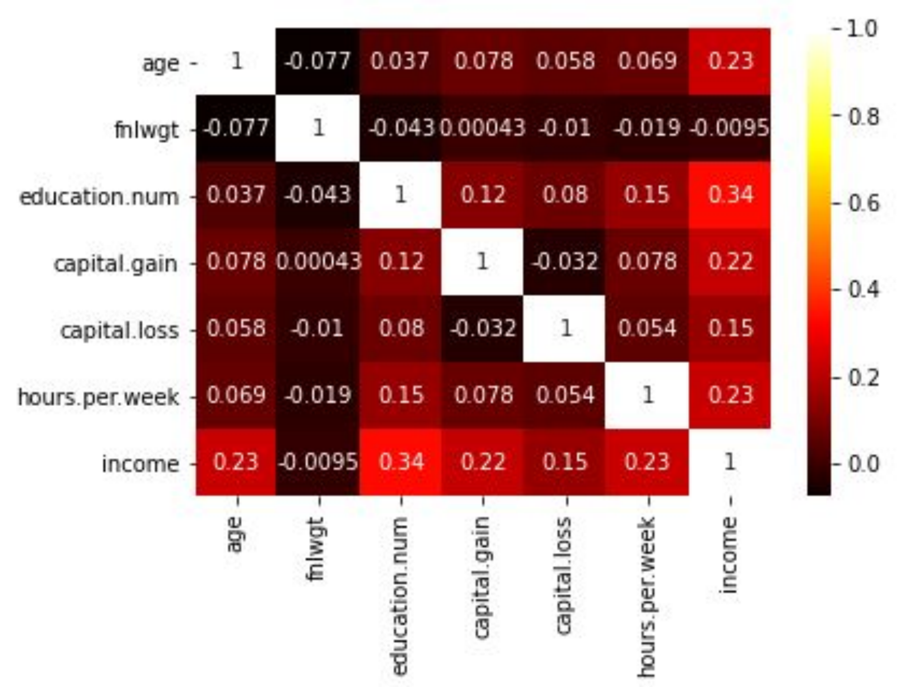






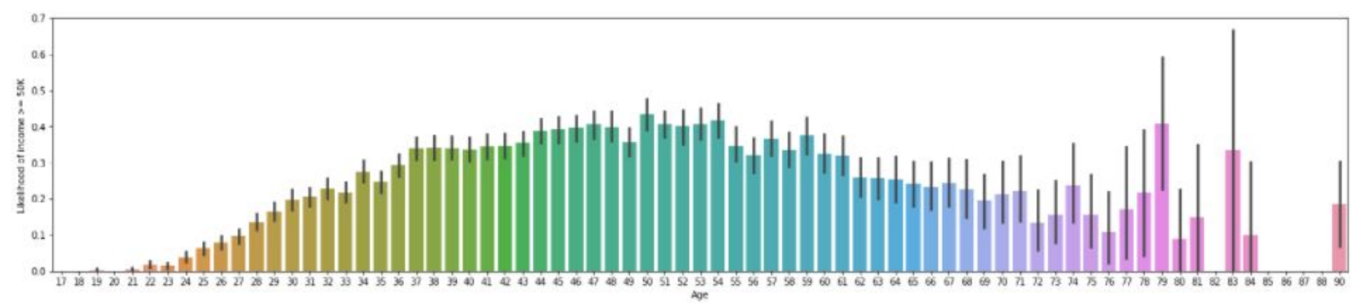
Visuals - Numerical part 1

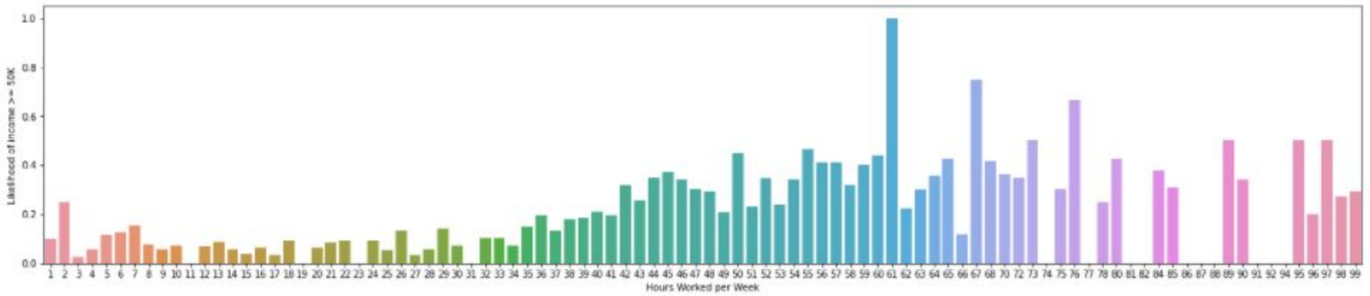
- Education.num strong correlation (.34) with income
- Age, Capital.gain and Hours/week all decent indicators

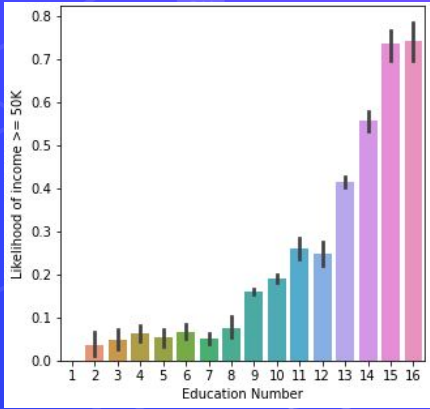


Visuals - Numerical part 2

- Age, Hours/Week, Education Number







3.
Data Engineering

Data Engineering

- Group ‘relationship’, ‘race’, ‘sex’, and ‘occupation’ by observed changes (from visualizations) and assign binary groups (0 or 1) to them
- Drop ‘native.country’, ‘workclass’, ‘marital.status’, ‘education’
  - Redundant and uninformative columns

```
#Use .replace to set 0s and 1s
data['relationship'] = data['relationship'].replace(['Husband','Wife'], 1)\
                                           .replace(['Not-in-family','Unmarried','Own-child','Other-relative'], 0)
data['race'] = data['race'].replace(['White','Asian-Pac-Islander'], 1)\
                           .replace(['Black','Other','Amer-Indian-Eskimo'], 0)
data['sex'] = data['sex'].replace('Male', 1)\
                         .replace('Female', 0)
data['occupation'] = data['occupation'].replace(['Exec-managerial', 'Prof-specialty','Adm-clerical', 'Sales',\
                                                 'Tech-support'], 1)\
                                       .replace(['Machine-op-inspct', 'Other-service', 'Craft-repair', 'Transport-moving',\
                                                 'Handlers-cleaners', 'Farming-fishing', 'Protective-serv', 'Armed-Forces',\
                                                 'Priv-house-serv'], 0)
data.drop(labels = ['native.country','workclass','marital.status','education'], axis = 1, inplace = True)
```

4.

ML Modeling

Data Cleaning and Visualization

“ All models are wrong, but some are useful.

- George E. P. Box

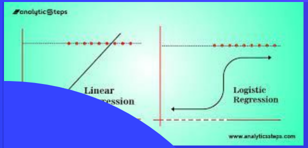
Model Choices

NAIVE BAYES

- Requires predictors be independent
- Works well with binary classification data and many data points
- Fast to employ

$$P(A|B) = \frac{P(A,B)}{P(B)}$$

N



L

LOGISTIC REGRESSION

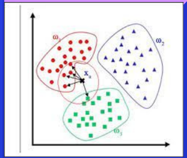
- The dependent variable is binary, multinomial, or ordinal (most often binary)
- No multicollinearity in the model (independence tenet)

- Can handle both numerical and categorical data.
- High Variance in the data can create totally unique "trees" (results)

DECISION TREE

D

K



- Used for both classification and regression problems
- Groups by "neighbors"

K-NEAREST NEIGHBOR

Applying Models

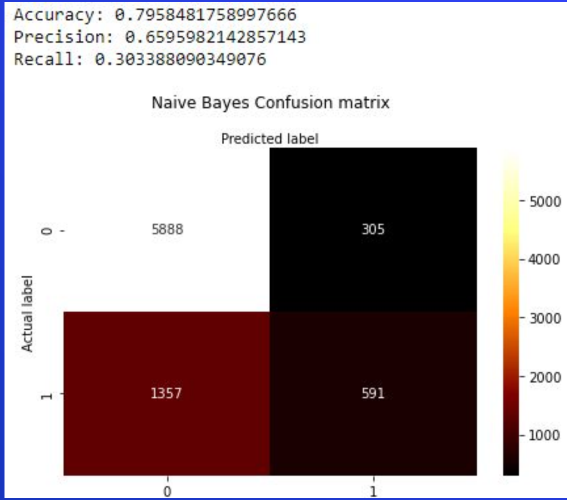
- Using Naive Bayes, Logistic Regression, K-Nearest Neighbors and Decision Trees
- Test_size set to 25%
- Seed set to 0

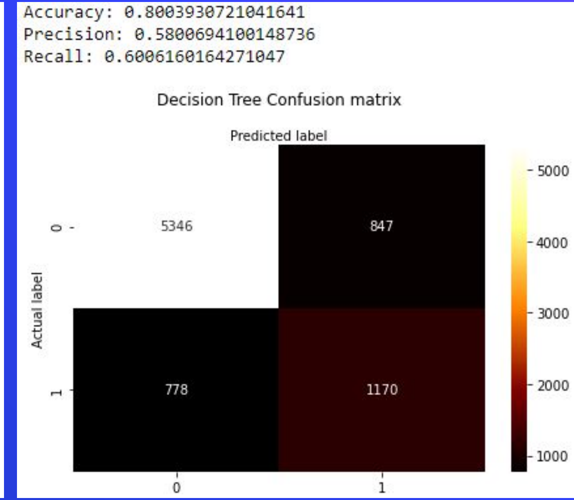
```
#Set features
X = data[['occupation', 'relationship', 'race', 'sex', 'age', 'fnlwgt', 'capital.gain', 'capital.loss', 'hours.per.week']]
#Set target Variable
y = data['income']

#Set test/train split
X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.25,random_state=0)

#select the model
nb = GaussianNB()
#fit the model
nb.fit(X_train,y_train)
#predict based on the model
y_pred=nb.predict(X_test)
```

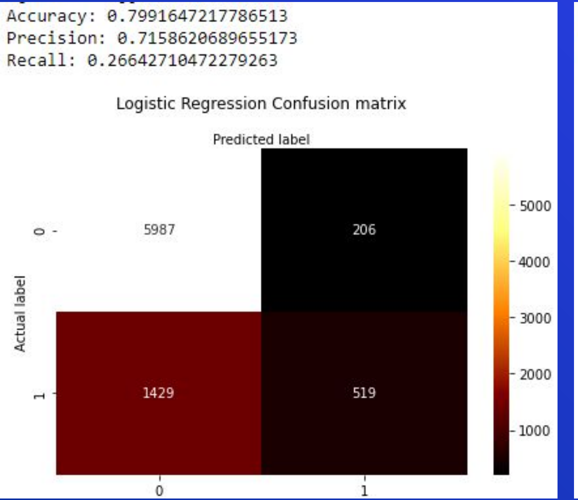
How to measure a model

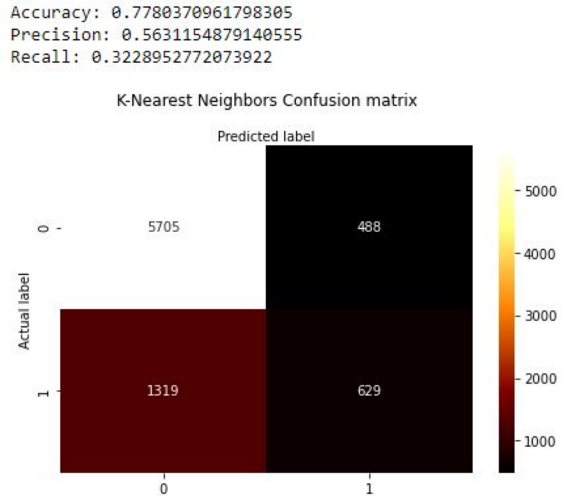
- TP = True Positive, TN = True Negative, FP = False Positive, FN = False Negative
- **Accuracy**
  - (TP + TN) / (TP + TN + FP + FN)
  - Compares accurate predictions vs total # of predictions
  - Describes overall accuracy of the model
  - Sample = Entire dataset
- **Precision**
  - (TP) / (TP + FP)
  - Compares accurate positive predictions vs all positive predictions
  - Describes a model's accuracy when only considering positive predictions made
  - Sample = All positives predicted
- **Recall**
  - (TP) / (TP + FN)
  - Compares accurate positive predictions vs the true total of positive values
  - Describes model's accuracy to predict positives in relation to the entire dataset
  - Sample = All positives in the dataset

5. Results









Comparing Models

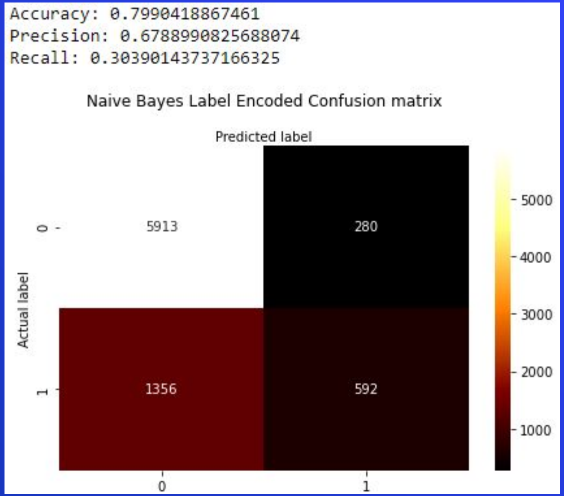
- Using our handmade, binary numeric variables

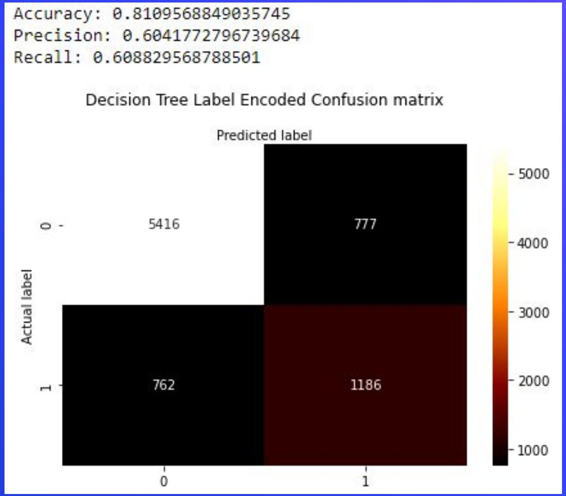
Model	Accuracy	Precision	Recall
Naive Bayes	79.58%	65.95%	30.33%
Decision Tree	**80.03%**	58.00%	**60.06%**
Logistic Regression	79.91%	**71.58%**	26.64%
KNN	77.80%	56.31%	32.28%

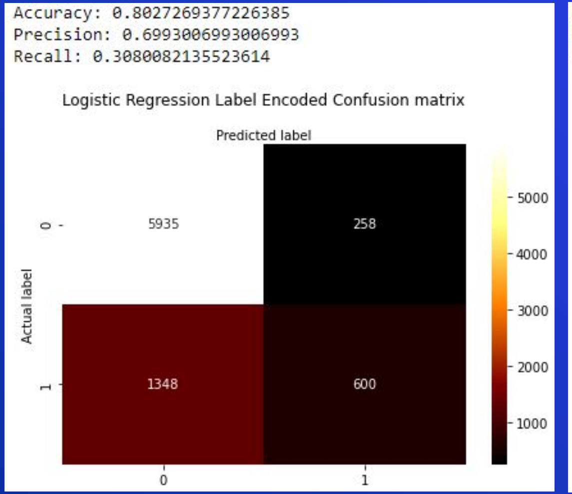
Applying sklearn preprocessing

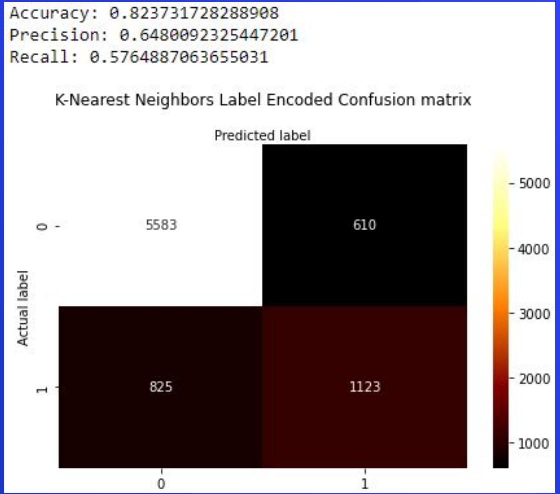
- Easy way of turning categorical data into numerical, and standardizing it.

```
#create a list of our categorical columns for our for loop to iterate over
cats = ['workclass','education','marital.status','occupation','relationship','race','sex','native.country']
#set our sklearn LabelEncoder to a varialbe
label_encoder = LabelEncoder()
#For each column from our list, fit the LabelEncoder and then transform the column as such
for column in cats:
    label_encoder.fit(data2[column])
    data2[column] = label_encoder.transform(data2[column])
#set our sklearn StandardScaler to a variable (this is like standardizing with z-scores, applied to all our columns)
scaler = StandardScaler()
#set our train/test split
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size = 0.25, random_state = 0)
#apply the scaler to the columns in test and train
x_train = pd.DataFrame(scaler.fit_transform(x_train), columns = x.columns)
x_test = pd.DataFrame(scaler.transform(x_test), columns = x.columns)
```









Label Encoding

- Using sklearn’s LabelEncoder and StandardScaler functions for categorical variable transformation.
- Applying the same models to this new dataset

Model	Accuracy	Precision	Recall
Naive Bayes	79.90% ↑	67.88% ↑	30.39% ↑
Decision Tree	81.09% ↑↑	60.41% ↑	**60.88%** ↑
Logistic Regression	80.27% ↑	**69.93%** ↓	30.80% ↑
KNN	**82.37%** ↑↑↑	64.80% ↑↑↑	57.64% ↑↑↑

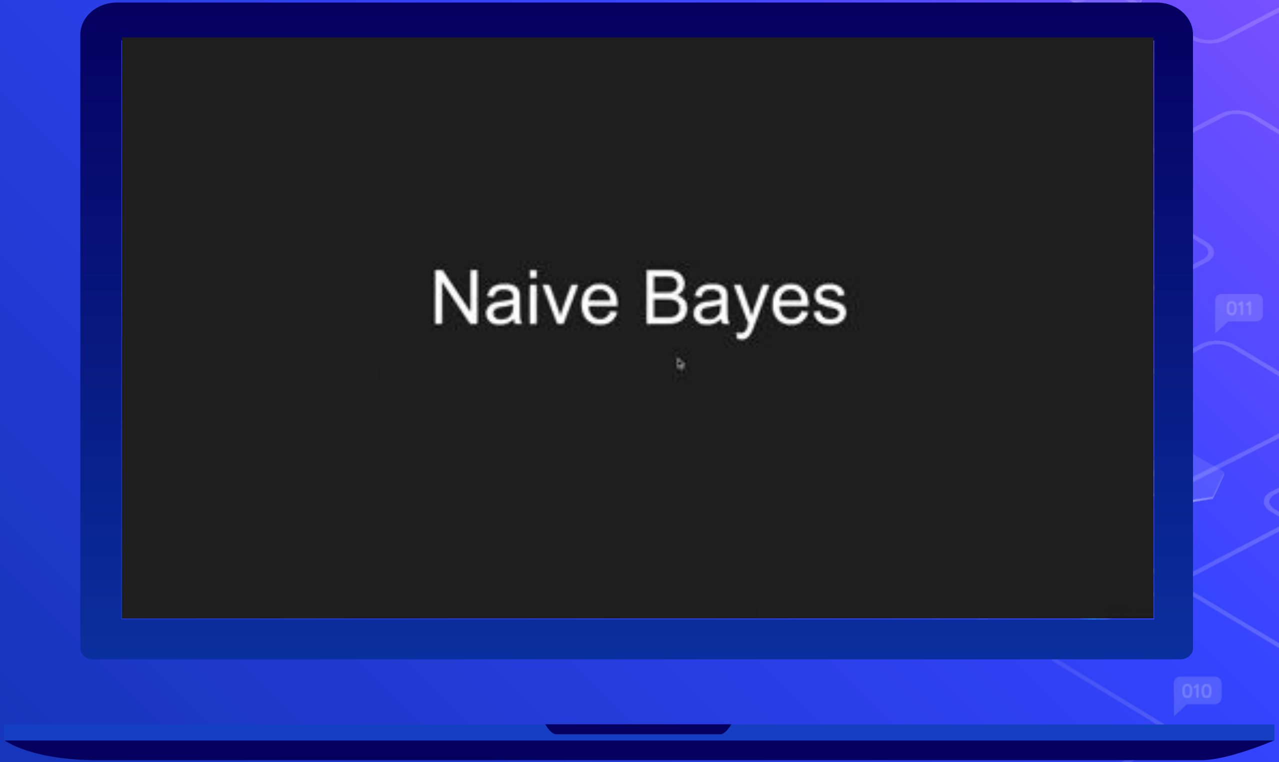
No single statistic tells the whole story.

Changing the Seed

- Seed originally set to 0
  - Iterating through seeds 0-9 for each model

```
#Set test/train split
X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.25,random_state=0)
```

Naive Bayes



https://media.giphy.com/media/EmF3ZMJ078JlZxfQq6/giphy.gif

Logistic Regression

https://media.giphy.com/media/bzl4qktSFgdJ8WMzRs/giphy.gif

Decision Tree

https://media.giphy.com/media/ZVpzkB8yWMxjm33SOn/giphy.gif

K-Nearest Neighbors

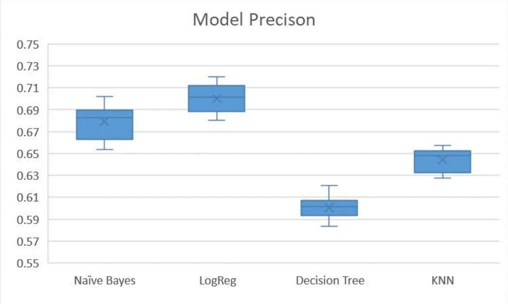
https://media.giphy.com/media/v2qOZXMdNxsB3SWqps/giphy.gif

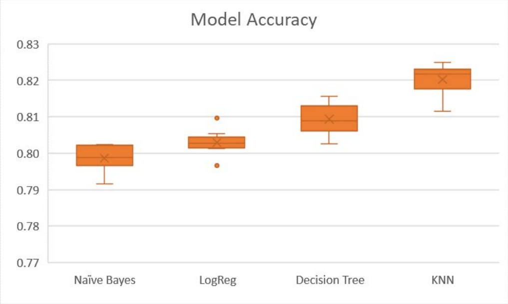
Results

- Most Accurate - KNN
- Most Precise - Logistic Regression
- Highest Recall - Decision Tree

Summary

- Since there is neither a high cost associated with False Negatives nor False Positives, the best model to use to predict an individual’s Income using the 1994 Census data is KNN.
- We are able to predict with over **80%** accuracy whether or not someone will make **greater than or less than $50k**
- Capital Gain, Age, and Hours Worked per Week were the **strongest predictors of Income**. All were **positively** correlated with **>$50k Income** (i.e. as capital.gain/age/hours.worked increased so did likelihood of earning greater than $50k





Appendix

Credits

Special thanks to all the people who made and released these awesome resources for free:

- Presentation template by SlidesCarnival designed by Jimena Catalina https://www.slidescarnival.com/aliena-free-presentation-template/4597#preview
- Photographs by Unsplash
- This data was extracted from the 1994 Census bureau database by Ronny Kohavi and Barry Becker (Data Mining and Visualization, Silicon Graphics
- Ron Kohavi, "Scaling Up the Accuracy of Naive-Bayes Classifiers: a Decision-Tree Hybrid", Proceedings of the Second International Conference on Knowledge Discovery and Data Mining, 1996. (PDF)
- Kaggle inspiration: https://www.kaggle.com/uciml/adult-census-income
- George Box image: https://en.wikipedia.org/wiki/File:GeorgeEPBox_(cropped).jpg

SlidesCarnival icons are editable shapes.

This means that you can:

- Resize them without losing quality.
- Change fill color and opacity.
- Change line color, width and style.

Isn’t that nice? :)

Examples:

Find more icons at
slidescarnival.com/extra-free-resources-icons-and-maps

Thank you!

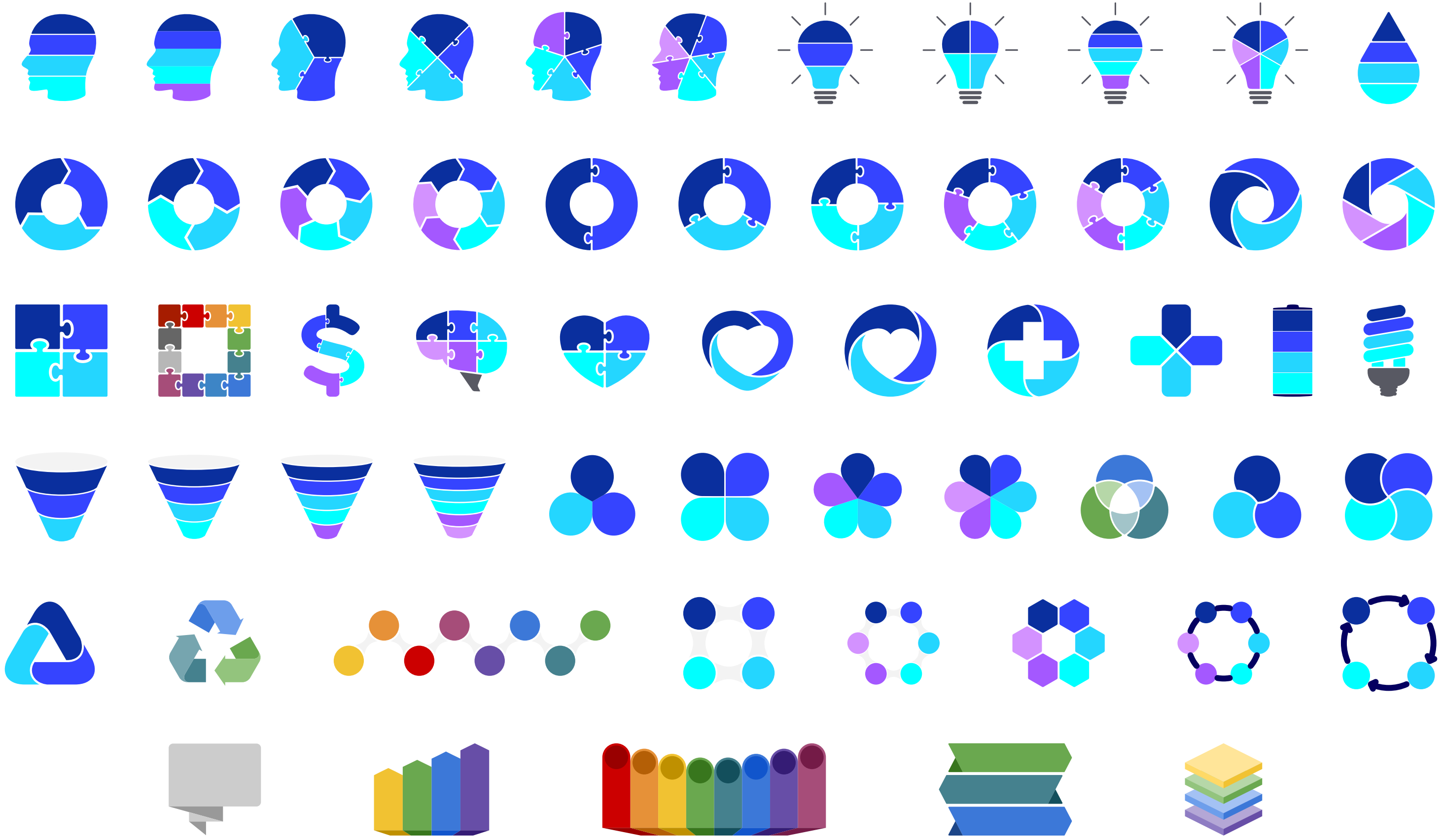
Any questions?

You can find us at:

owinters58@gmail.com

conoranderson2@gmail.com

Diagrams and infographics



You can also use any emoji as an icon!
And of course it resizes without losing quality.

How? Follow Google instructions
https://twitter.com/googledocs/status/730087240156643328

✋👆👉👍👤👦👧👨👩👪💃🏃
💑🖤😂😉😋😒😭👶😸🐟🍒🍔💣
📌📖🔨🎃🎈🎨🏈🏰🌏🔌🔑 and many more...