Systems Documentation Report

Arizona State University, CSE578 - Team 8
Mohamed Mohamed - mmoham69@asu.edu
Pankaj Kumar Singh - psing109@asu.edu
Parth Bhatt - prbhatt@asu.edu
Christopher Azzara - cazzara@asu.edu
Nihar Parida - nparida@asu.edu

Roles and Responsibilities

Name	Responsibilities
Mohamed Mohamed	Developed user stories and visualization for relationship and work class.
Pankaj Kumar Singh	Developed user stories and visualization for .Reviewed system documentation report and executive report.
Parth Bhatt	Generated visualization and user stories for Ethnicity and Native Countries.
Christopher Azzara	Developed user stories and analysis for Age, Education Level, Education Years.
Nihar Parida	Developed executive report and reviewed system documentation report.

Project Goals

The project goals are as follows.

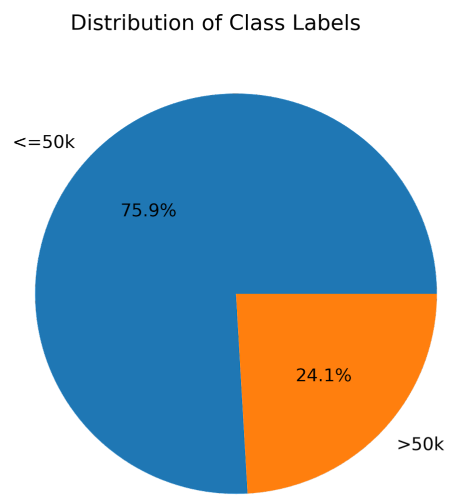
1. Develop user stories and visualizations using different data attributes/features available in the data.
2. Analyze the data and produce insights which could be used to make decisions to include or exclude specific attributes for marketing profiles for enrollment advertisement.
3. Analyze data by grouping them under two income categories i.e. income<=50 K & income >50K -- group comparison analysis.

Tools and Techniques

1. Identify dimensions and metrics.
2. Categorize into univariate and multivariate dimension groupings.
3. Plot various charts through Pie Chart, Histogram, Scatter Plot, Bar Charts and Geo visualization using python(3.6 and above), matplotlib and plotly express libraries in Jupyter notebook.
4. Identify relationships and discover patterns to explore influential attributes to achieve project goals.
5. Derive conclusion and Identify the most significant group for Campaign.

Assumptions

The analysis is done under the assumption that the data were collected in a random, independent manner from the 1994 US Census. The examples in the dataset were older than 16 years and reported more than 0 hours worked per week.

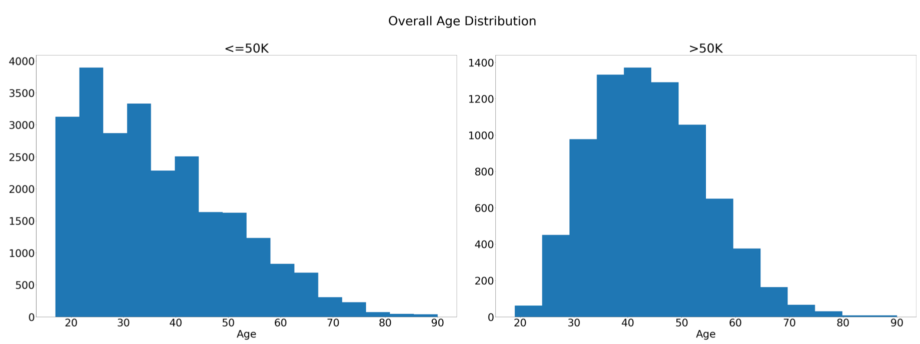


This visualization is important to include because it gives a frame of reference for comparing the features which follow. It is shown that the lower income class label comprises more than 75% of the dataset and so the distribution of labels is not equal. This makes intuitive sense as generally it's expected that there are more lower income members of the population than higher income members.

User Stories and Visualizations

User Story - Age

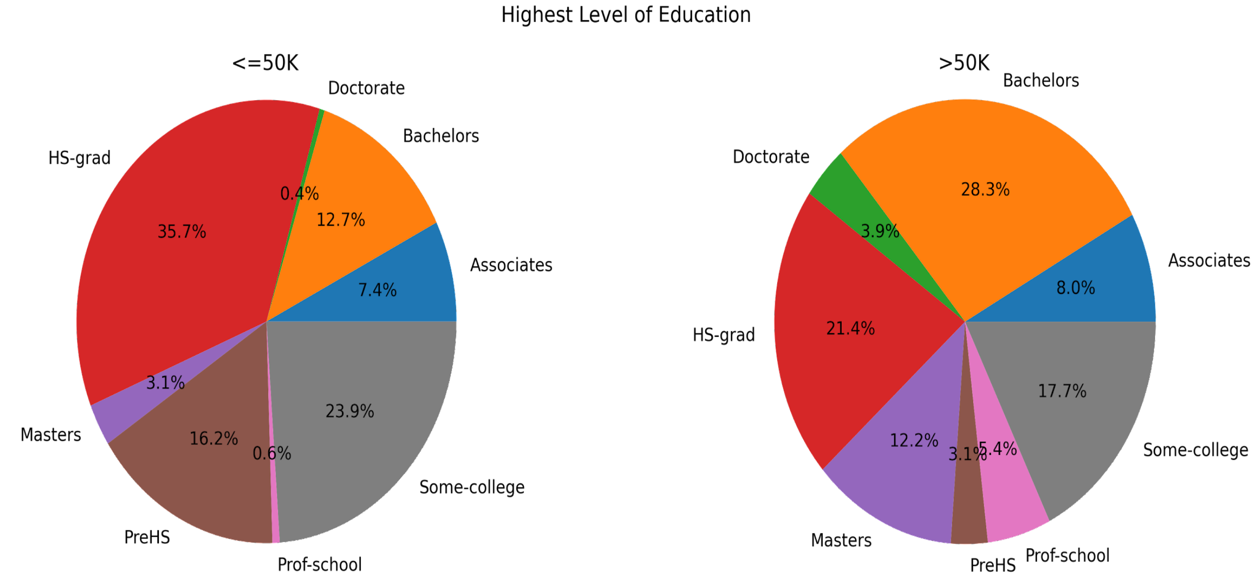
To explore the age variable, we compared the distribution of ages in each class label. It was found that the lower income had a right tail distribution with most of the examples having a median age of 34 whereas the higher income class had a median age of 44.



It can be seen from the images above that the two class labels have clearly different distributions for the age feature.

User Story - Education Level

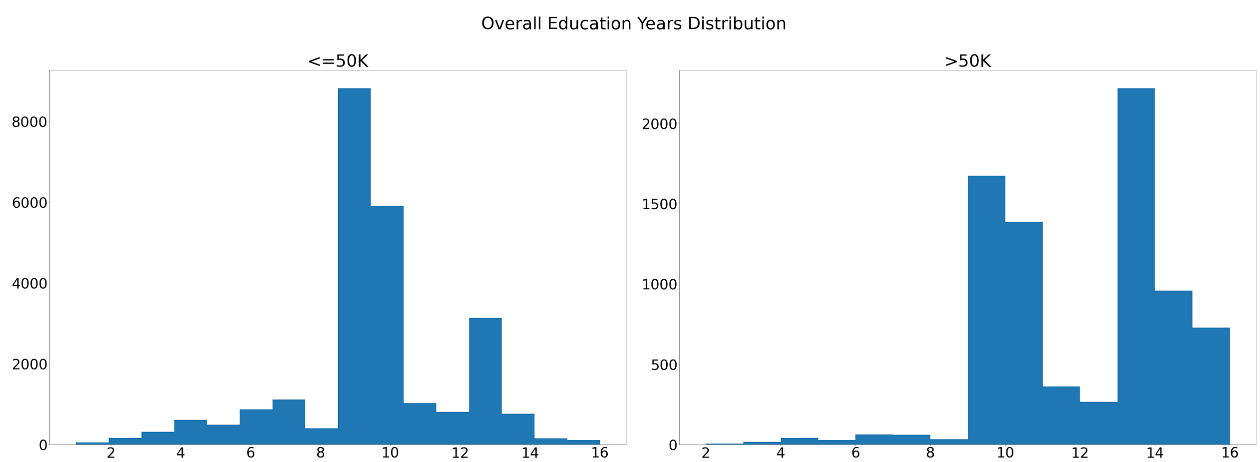
For this story, we looked at the highest degree of education held by each of the examples in the high income and low income class label.
It must be noted that we combined the labels 9th/10th/11th/12th/1st-4th/5th-6th/7th-8th/9th/Preschool as “PreHS” and Assoc-acdm/ Assoc-voc as Associates, this was to reduce the amount of education level labels as well as aggregate the values.



From this visualization it can be seen that there was a much larger proportion of people with college education among the higher income class label. Whereas the percentage of people without a high school education is larger in the lower income class.

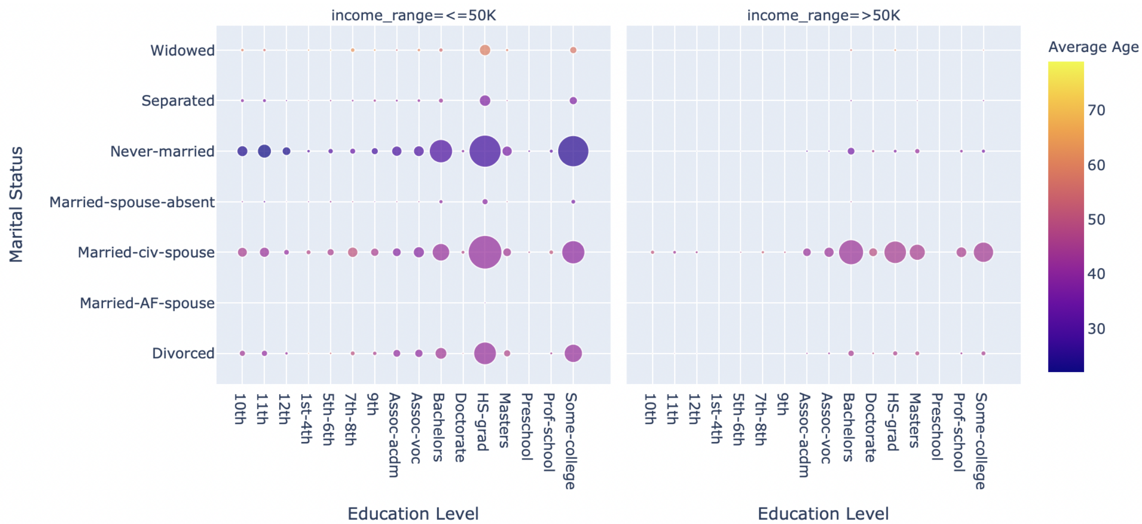
User Story - Education Years

This story again tries to contrast the amount of years of education that both class labels had. Again it became apparent from visualizing the distribution of education years that people from the higher income label were more likely to have a higher number of years of education. The median number of years of education for the lower income class was 9 while for the higher income class the median was 12.



User Story - Marital Status

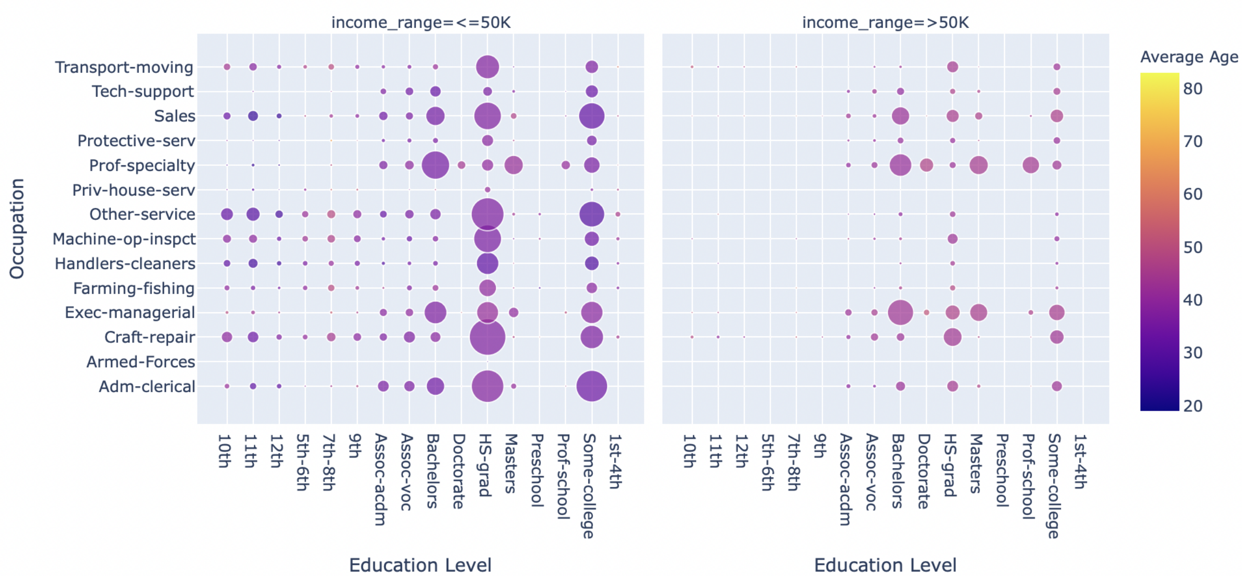
This story explores the correlation between marital status and education level along with the average age of the individuals.With the help of scattered points,its color(representing age) and size(representing the number of individuals), it suggests which marital status and education level to be considered for a marketing campaign.



The visualization clearly shows that for income level <=50K, the number of people(1000-3200) with “HS-grade” or “Some college” and marital status “Never Married” & “Married” is significant whereas in the income level >50K, the number of “Married” individuals(1000-1500) with “HS-grade” or “Some college” is significant.

User Story - Occupation

This story explores the correlation between occupation type and education level along with the average age of the individuals.With the help of scattered points,its color(representing age) and size(representing the number of individuals), it suggests which occupation type and education level to be considered for a marketing campaign.

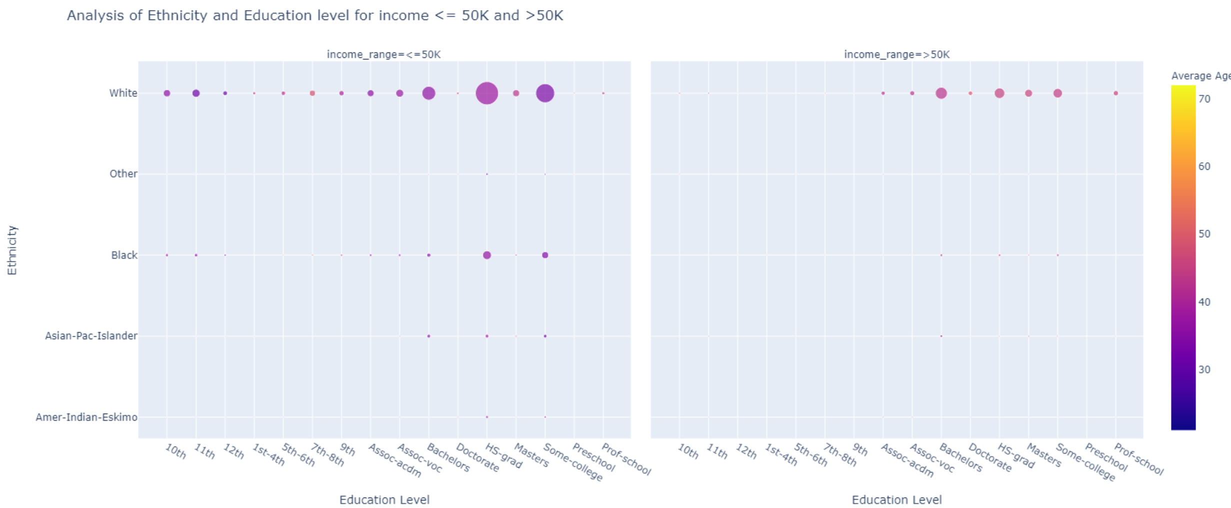


The visualization suggests that for income level <=50K, having education level “HS-Grade” & “Some-college” the count(200-1500) is significant across different occupations whereas For income >50K,having education level ”HS-Grade” and “Some-college” the count(100-300) is not significant across occupation types.

User Story - Ethnicity

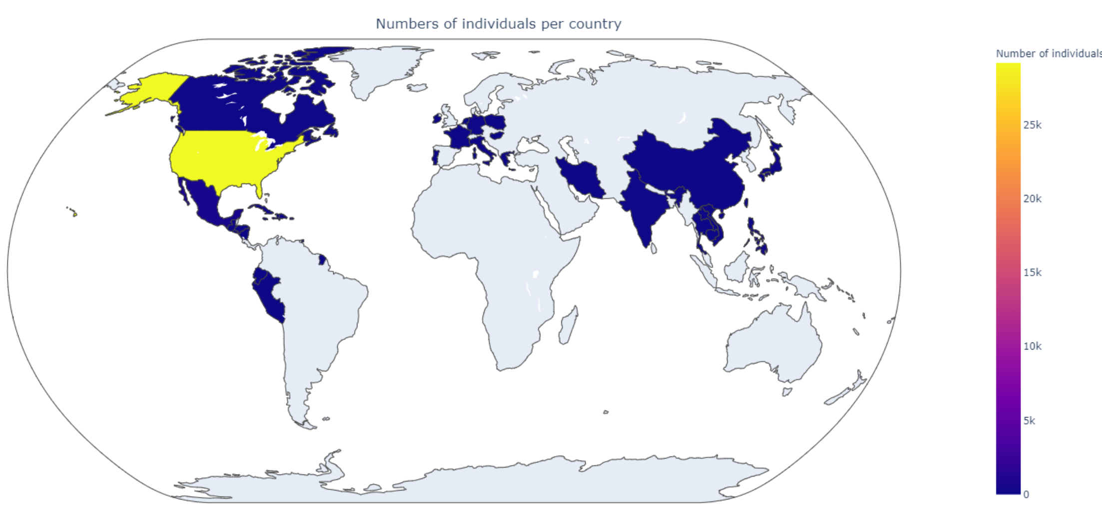
From visualization it is clear that the count of White is higher than the rest in both income groups <=50K and >50K having education level either “HS-grade”, “Some-college”,and “Bachelors”.The color of the sidebar provides age information to further assist the analysis from age perspective.

Targeting white ethnic individuals who have education level “HS-grad” or “Some College” with income less than 50K are more likely to go for enrollment as the average age is in mid 30s.

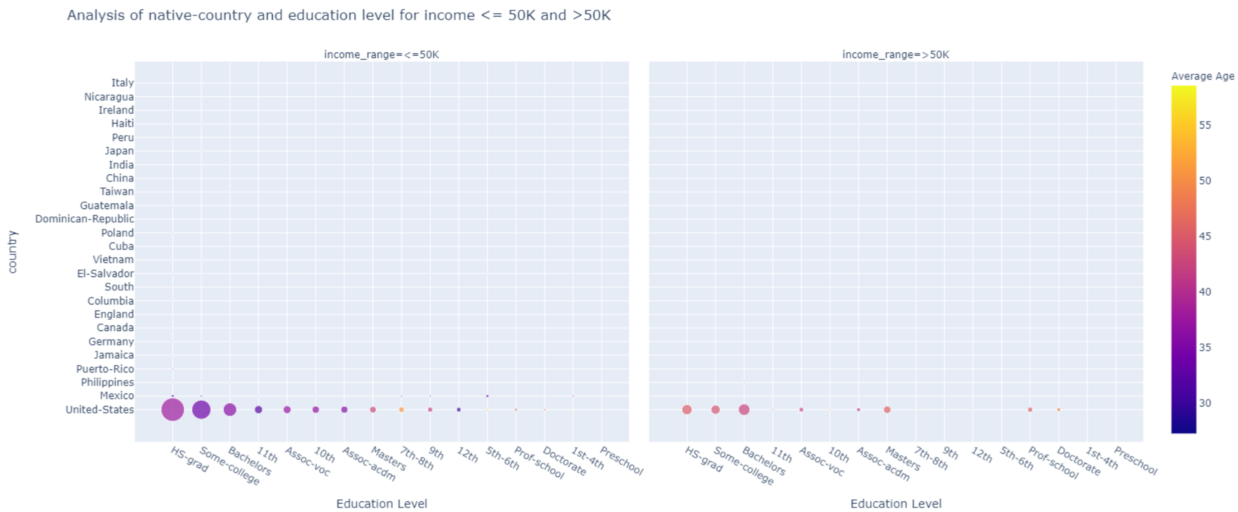


User Story - Native Country

From visualization it is clear that the United States has the highest number of individuals provided by this dataset”.The color of the sidebar provides the number of individuals to clearly assist the analysis.

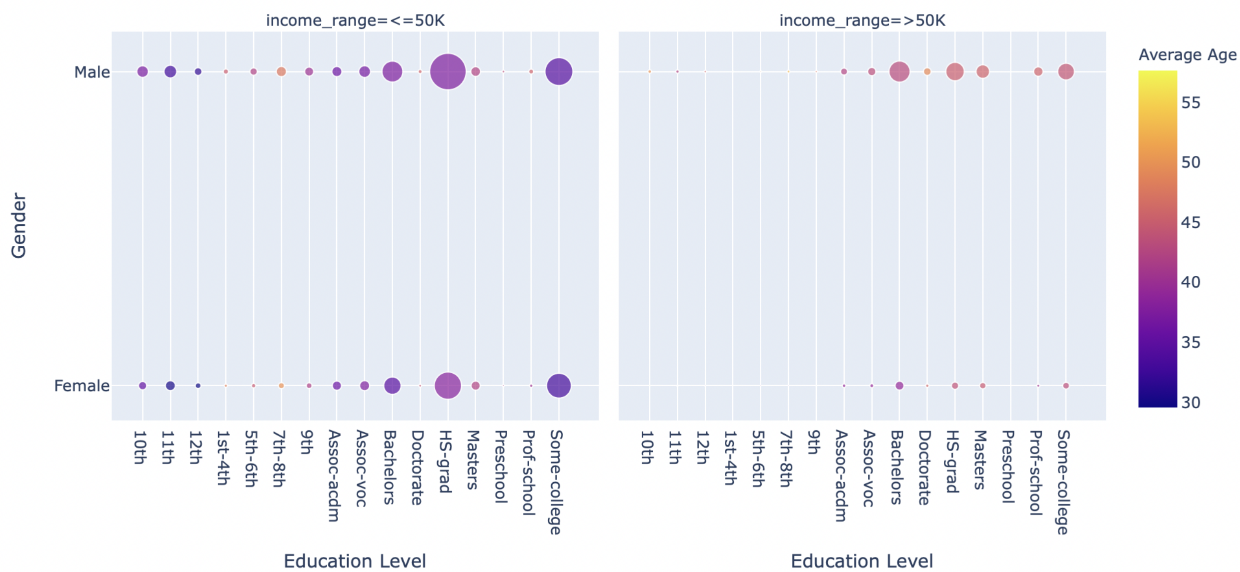


From visualization it is clear that the count of individuals,who have United States as their native country, is higher than the rest in both income groups <=50K and >50K having education level either “HS-grade”, “Some-college”,and “Bachelors”.The average age of this group mentioned in the previous sentence is mid 30s for income level <=50K and due to that reason this group of people will be more likely to pursue education for their betterment. The color of the sidebar provides age information to further assist the analysis from an age perspective.



User Story - Gender

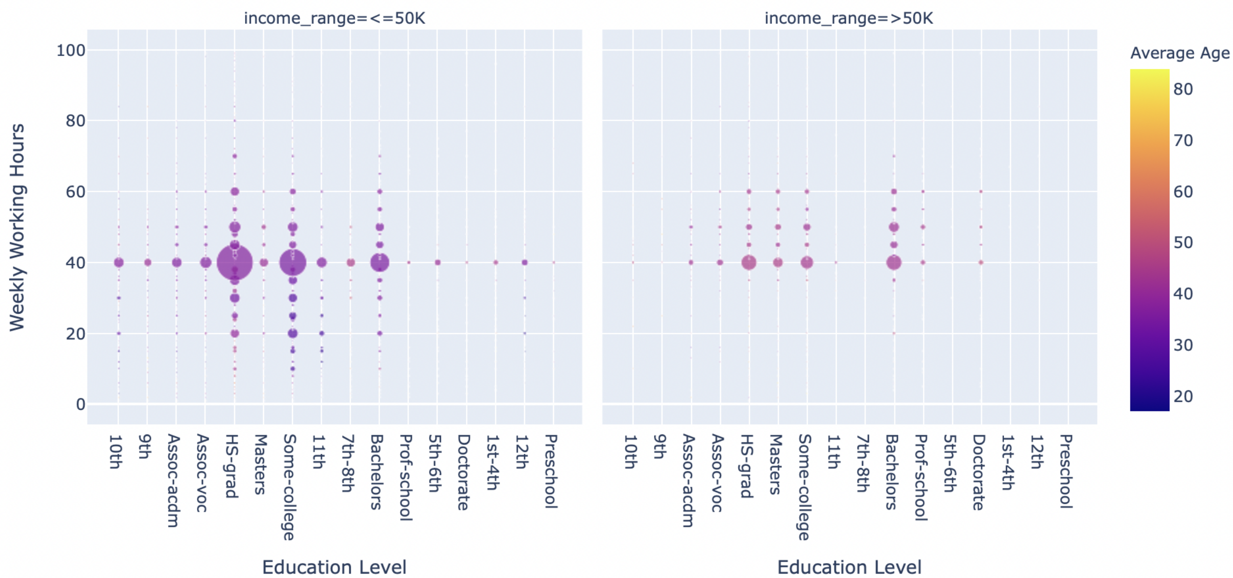
This story explores the correlation between gender and education level along with the average age of the individuals.With the help of scattered points,its color(representing age) and size(representing the number of individuals), it suggests which gender type and education level to be considered for a marketing campaign.



From visualization it is clear that for income level <=50K, males(5662 & 3295) and females(3164 & 3295) with education level “HS-grade” or “Some-college” respectively, should be considered for the campaign. For income level >50K, males with education level “HS-grade”(1449) or “Some-college”(1190) are the right candidates for the campaign, though their age is in 40 which needs to be factored in as well.

User Story - Hours Per Week

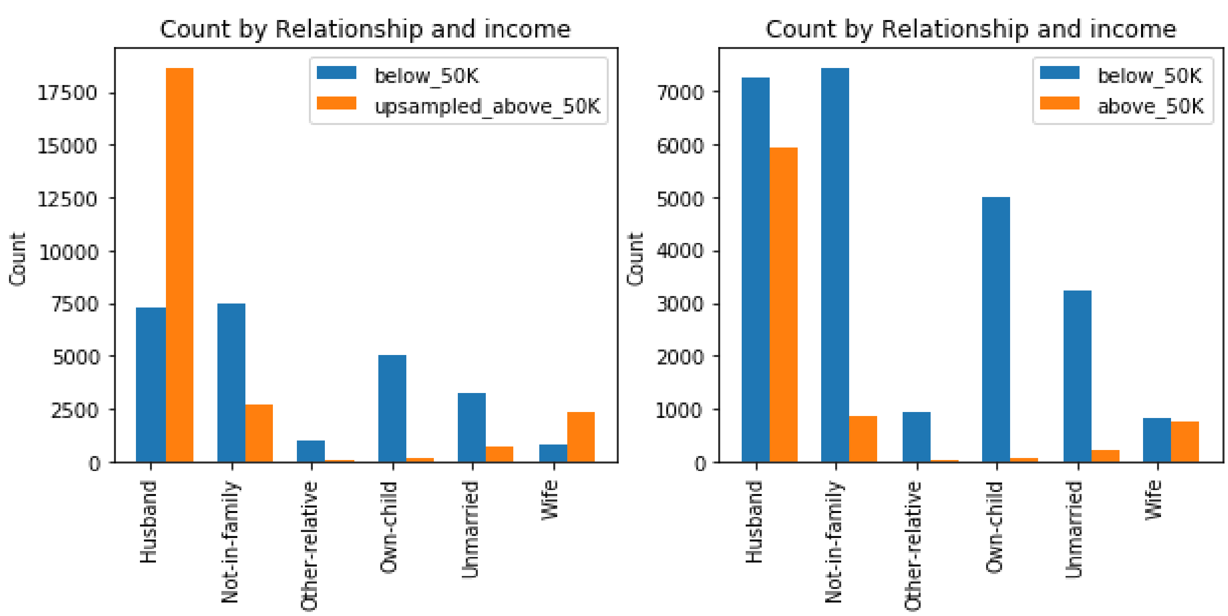
This story explores the correlation between hours worked per week and education level along with the average age of the individuals.With the help of scattered points,its color(representing age) and size(representing the number of individuals), it suggests which weekly working hours group and education level to be considered for a marketing campaign.



From visualization it is clear that for income level <=50K, that the population with High school and some college level education working 40 hours per week are good in numbers(between 100-5000).For income level >50K, population with High school and some college level education working between 40 to 60 hours per week are good in numbers(between 100-1000).

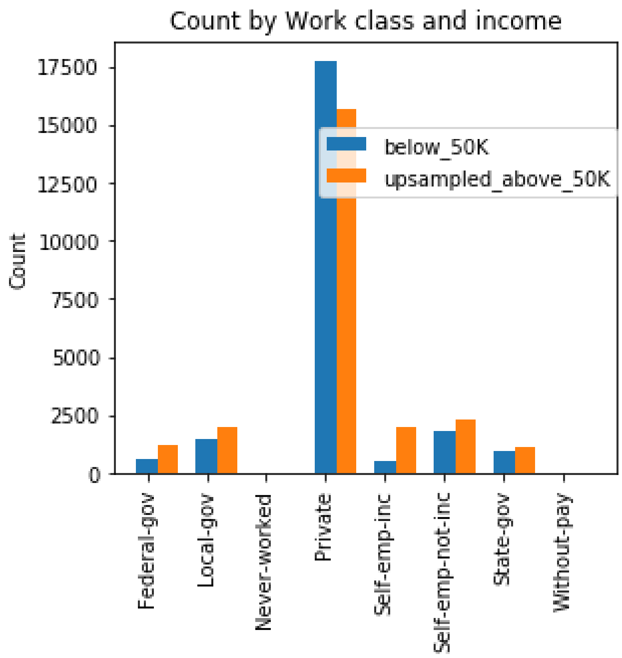
User Story - Relationship

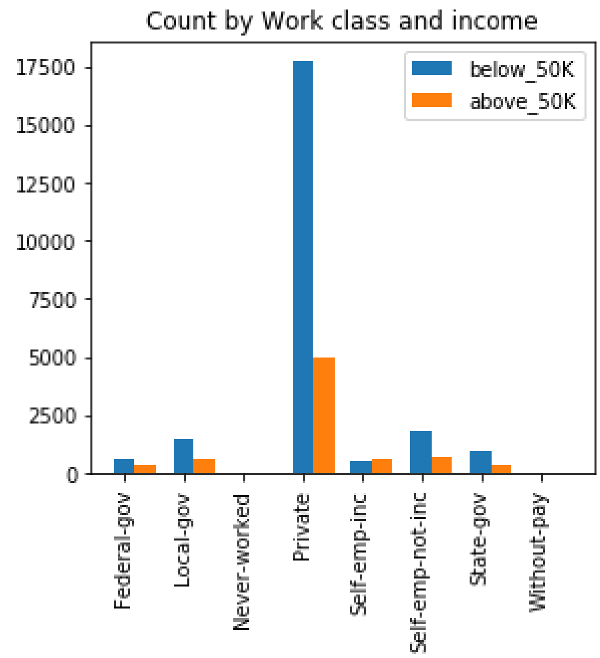
Our classes have different distributions as shown in the graph below. Most of the people who make more than 50K are “Husband”, therefore we can ignore that group. We can focus on people who are “not-in-family”, “other-relative”, “own-child”, and “unmarried”. Also, “wife” has low purity, so we can ignore them as well.
Performing data augmentation makes it clear that “Husband” relationship is dominant in people who make more than 50K. Data was augmented by adding data instances to “below 50K” while keeping the same distribution.



User Story - Work Class

Both classes have similar distribution. Therefore, it doesn't seem like this attribute will help us differentiate between people who make above or below 50K.
Performing data augmentation makes it clear that both classes have a very similar distribution.





Open Questions

Team came across below open questions during the project.

1. Team first explored all the 14 attributes and then went through a brainstorming session to decide the final data attributes to be analyzed based on feature importance.
2. Team discussed and came up with the most important features to be captured in the executive report.

Out of Scope

We found that below could be the next steps to extend this work in future.

1. Make the code as a packaged solution supported with a user interface.
2. Provide users the option to select specific attribute(s) and trigger the visualization.
3. Provide an option to send email and share the visualization link with other users.
4. Build a classification model to predict the income label of new examples.

Appendix

This section describes the python code used in creating different visualizations for the project.

Python Libraries

```
# Python version 3.6 and above
import pandas as pd
import matplotlib.pyplot as plt
# this library was first installed and then imported for use
import plotly.express as px
```

Data Cleaning

```
df = pd.read_csv(filepath+'adult.data')
# replace spaces from columns
df = df.replace({"^\s*|\s*$":""}, regex=True)
# filter records for income <= 50K
df_below_50 = df[df['income_range']=='<=50K']
# filter records for income > 50 K
df_above_50 = df[df['income_range']=='>50K']
```

Visualization Code

- Age Analysis

```
fig, axes = plt.subplots(nrows=1, ncols=2, tight_layout=True, figsize=(60,22))
fig.suptitle("Overall Age Distribution")
axes[0].hist(low_income['age'], bins='sturges')
axes[0].set_xlabel("Age")
axes[0].set_title("<=50K")
axes[1].hist(high_income['age'], bins='sturges')
axes[1].set_xlabel("Age")
axes[1].set_title(">50K")
plt.show()
```

- Education Years Analysis

```
fig, axes = plt.subplots(nrows=1, ncols=2, tight_layout=True, figsize=(60,22))
fig.suptitle("Overall Education Years Distribution")
axes[0].hist(low_income['education_years'], bins='sturges')
axes[0].set_title("<=50K")
axes[1].hist(high_income['education_years'], bins='sturges')
axes[1].set_title(">50K")
plt.show()
```

- Education Level Analysis

```
"""
Relabel 9th/10th/11th/12th/1st-4th/5th-6th/7th-8th/9th/Preschool as PreHS
"""
PREHS_LABELS = ["9th","10th","11th","12th","1st-4th","5th-6th","7th-8th","Preschool"]
ASSOCIATE_LABELS = ["Assoc-acdm", "Assoc-voc"]
def degree_label(x):
    if x in PREHS_LABELS:
        return "PreHS"
    elif x in ASSOCIATE_LABELS:
        return "Associates"
    return x
df["degree"] = df["degree"].transform(degree_label)

labels=list(degree_df.index)
fig, axes = plt.subplots(nrows=1, ncols=2, tight_layout=True, figsize=(60,22))
fig.suptitle("Highest Level of Education")
axes[0].pie(li_degree_counts, labels=labels, autopct='%1.1f%%')
axes[0].set_title("<=50K")
axes[0].axis("equal")
axes[1].pie(hi_degree_counts, labels=labels, autopct='%1.1f%%')
axes[1].set_title(">50K")
axes[1].axis("equal")
plt.show()
```

- Marital Status Analysis

```
#create dataframe with average age and count of individuals across different marital statuses and education levels

grouped_multiple_above50 = df_above_50.groupby(['marital-status', 'education']).agg({'age': ['mean', 'count']})

grouped_multiple_above50.columns = ['age_mean', 'count']

grouped_multiple_above50 = grouped_multiple_above50.reset_index()

# add label for > 50K income

grouped_multiple_above50['income_range'] = ">50K"

grouped_multiple_below50 = df_below_50.groupby(['marital-status', 'education']).agg({'age': ['mean', 'count']})

grouped_multiple_below50.columns = ['age_mean', 'count']

grouped_multiple_below50 = grouped_multiple_below50.reset_index()

# add label for <50K income

grouped_multiple_below50['income_range'] = "<=50K"
```

concat below and above 50K income group in a single dataframe for scatter plot

grouped_multiple_above_and_below50 = pd.concat([grouped_multiple_below50,grouped_multiple_above50],ignore_index=True)

fig = px.scatter(grouped_multiple_above_and_below50,y="marital-status",x='education', color='age_mean', size='count',title = 'Analysis of marital status and education level for income <= 50K and >50K',labels={'marital-status':'Marital Status','education':'Education Level','age_mean':'Average Age'},facet_col = "income_range")

fig.show()

- Occupation Analysis

```
#create dataframe with average age and count of individuals across different
occupations and education levels
grouped_multiple_above50_occup = df_above_50.groupby(['occupation',
'education']).agg({'age': ['mean', 'count']})
grouped_multiple_above50_occup.columns = ['age_mean', 'count']
grouped_multiple_above50_occup = grouped_multiple_above50_occup.reset_index()
grouped_multiple_above50_occup =
grouped_multiple_above50_occup[grouped_multiple_above50_occup['occupation']!='?']
# add label for > 50K income
grouped_multiple_above50_occup['income_range'] = ">50K"
grouped_multiple_below50_occup = df_below_50.groupby(['occupation',
'education']).agg({'age': ['mean', 'count']})
grouped_multiple_below50_occup.columns = ['age_mean', 'count']
grouped_multiple_below50_occup = grouped_multiple_below50_occup.reset_index()
grouped_multiple_below50_occup =
grouped_multiple_below50_occup[grouped_multiple_below50_occup['occupation']!='?']
# add label for <50K income
grouped_multiple_below50_occup['income_range'] = "<=50K"
# concat below and above 50K income group in a single dataframe for scatter plot
grouped_multiple_above_and_below50_occup =
pd.concat([grouped_multiple_below50_occup,
grouped_multiple_above50_occup ],ignore_index=True)
# scatter plot for occupation and education level for income <=50K and > 50K
fig = px.scatter(grouped_multiple_above_and_below50_occup, y="occupation",
x='education',color='age_mean',size='count',
        title = 'Analysis of occupation and education level for income <= 50K and >50K',
        labels={'occupation':'Occupation','education':'Education Level',
'age_mean':'Average Age'},facet_col = "income_range")
fig.show()
```

- Gender Analysis

```
#create dataframe for average age and count of individuals across different genders and
education levels
grouped_multiple_above50_gender = df_above_50.groupby(['sex',
'education']).agg({'age': ['mean', 'count']})
grouped_multiple_above50_gender.columns = ['age_mean', 'count']
```

```
grouped_multiple_above50_gender = grouped_multiple_above50_gender.reset_index()
# add label for > 50K income
grouped_multiple_above50_gender['income_range'] = ">50K"
grouped_multiple_below50_gender = df_below_50.groupby(['sex',
'education']).agg({'age': ['mean', 'count']})
grouped_multiple_below50_gender.columns = ['age_mean', 'count']
grouped_multiple_below50_gender = grouped_multiple_below50_gender.reset_index()
# add label for <= 50K income
grouped_multiple_below50_gender['income_range'] = "<=50K"
# concat below and above 50K income group in a single dataframe for scatter plot
grouped_multiple_above_and_below50_gender =
pd.concat([grouped_multiple_below50_gender,
grouped_multiple_above50_gender],ignore_index=True)
# scatter plot for occupation and education level for income <=50K and > 50K
fig = px.scatter(grouped_multiple_above_and_below50_gender,y="sex",x='education',
        color='age_mean',
        size='count',
        title = 'Analysis of occupation and education level for income <= 50K and >50K',
        labels={'sex':'Gender','education':'Education Level','age_mean':'Average Age'},
        facet_col = "income_range"
       )
fig.show()
```

- Hours Per Week Analysis

```
#create dataframe for average age and count of individuals across different weekly
working hours and education levels
grouped_multiple_above50_hrs = df_above_50.groupby(['hours-per-week',
'education']).agg({'age': ['mean', 'count']})
grouped_multiple_above50_hrs.columns = ['age_mean', 'count']
grouped_multiple_above50_hrs = grouped_multiple_above50_hrs.reset_index()
# add label for > 50K income
grouped_multiple_above50_hrs['income_range'] = ">50K"
grouped_multiple_below50_hrs = df_below_50.groupby(['hours-per-week',
'education']).agg({'age': ['mean', 'count']})
grouped_multiple_below50_hrs.columns = ['age_mean', 'count']
grouped_multiple_below50_hrs = grouped_multiple_below50_hrs.reset_index()
# add label for <= 50K income
grouped_multiple_below50_hrs['income_range'] = "<=50K"
# concat below and above 50K income group in a single dataframe for scatter plot
grouped_multiple_above_and_below50_hrs =
pd.concat([grouped_multiple_below50_hrs,grouped_multiple_above50_hrs],ignore_inde
x=True)
# scatter plot for weekly working hours and education level for income > 50K
fig =
px.scatter(grouped_multiple_above_and_below50_hrs,y="hours-per-week",x='education'
,color='age_mean',size='count',
 title = 'Analysis of hours per week and education level for income <= 50K and >50K',
```

```
labels={'hours-per-week':'Weekly Working Hours','education':'Education Level','age_mean':'Average Age'},facet_col = "income_range")
fig.show()
```

- Ethnicity/Race analysis

```
#create dataframe to analysis ethnicity,education level,and age for income range(<=50k)
ethnicity_education_50korbelow = df_below_50.groupby(['race', 'education']).agg({'age': ['mean', 'count']})
ethnicity_education_50korbelow.columns = ['age_mean', 'count']
ethnicity_education_50korbelow = ethnicity_education_50korbelow.reset_index()
ethnicity_education_50korbelow['income_range'] = "<=50K"
#create dataframe to analysis ethnicity,education level,and age for income range(>50k)
ethnicity_education_above50k = df_above_50.groupby(['race', 'education']).agg({'age': ['mean', 'count']})
ethnicity_education_above50k.columns = ['age_mean', 'count']
ethnicity_education_above50k = ethnicity_education_above50k.reset_index()
ethnicity_education_above50k['income_range'] = ">50K"
# concat two dataframe for analyzing using scatter plot
ethnicity_education_above_and_below50 =
pd.concat([ethnicity_education_50korbelow,ethnicity_education_above50k],ignore_index
=True)
# use the plotpy express function scatter to analyze ethnicity, education level,and age
plot = px.scatter(ethnicity_education_above_and_below50, y="race", x='education', color='age_mean', size='count', title='Analysis of Ethnicity and Education level for income <= 50K and >50K', labels={'race': 'Ethnicity', 'education': 'Education Level', 'age_mean': 'Average Age'}, facet_col="income_range")
plot.show()
```

- Native-Country analysis

```
# create dataframe to calculate number of individuals and age-mean in each country
nativecountry_df= df.groupby(['native-country']).agg({'age': ['mean', 'count']})
nativecountry_df.columns = ['age_mean', 'count']
nativecountry_df = nativecountry_df.reset_index()
# plot the choropleth map to analyze number of individuals per country
fig = px.choropleth(nativecountry_df, locations='native-country', locationmode='country names', color='count', projection="natural earth",labels={'count':'Number of individuals'})
fig.update_layout(
  title={
     'text': "Numbers of individuals per country",
     'y':0.95,
     'x':0.5,
     'xanchor': 'center',
     'yanchor': 'top'})
fig.show()
```

```
#create dataframe to analysis nativecountry,education level,and age for income range(<=50k)
nativecountry_education_50korbelow = df_below_50.groupby(['native-country',
'education']).agg({'age': ['mean', 'count']})
nativecountry_education_50korbelow.columns = ['age_mean', 'count']
nativecountry_education_50korbelow =
nativecountry_education_50korbelow.reset_index()
nativecountry_education_50korbelow['income_range'] = "<=50K"
#create dataframe to analysis nativecountry,education level,and age for income range(>50k)
nativecountry_education_above50k = df_above_50.groupby(['native-country',
'education']).agg({'age': ['mean', 'count']})
nativecountry_education_above50k.columns = ['age_mean', 'count']
nativecountry_education_above50k = nativecountry_education_above50k.reset_index()
nativecountry_education_above50k['income_range'] = ">50K"
# concat two dataframe for analyzing using scatter plot
nativecountry_education_above_and_below50 =
pd.concat([nativecountry_education_50korbelow, nativecountry_education_above50k],
ignore_index=True)
# sort and limit the count to understand the data easily and remove less useable data
nativecountry_education_above_and_below50 =
nativecountry_education_above_and_below50.sort_values(by=['count'],ascending=False
)
nativecountry_education_above_and_below50 =
nativecountry_education_above_and_below50[nativecountry_education_above_and_bel
ow50['count'] >= 10]
# remove the native country row which has ‘?’ as the data
index_names =
nativecountry_education_above_and_below50[nativecountry_education_above_and_bel
ow50['native-country'] == '?'].index
nativecountry_education_above_and_below50.drop(index_names, inplace=True)
# use the plotpy express function scatter to analyze nativecountry, education level,and age
plot = px.scatter(nativecountry_education_above_and_below50, y='native-country',
x='education', color='age_mean', size='count', title='Analysis of native-country and
education level for income <= 50K and >50K', labels={'native-country': 'country',
'education': 'Education Level', 'age_mean': 'Average Age'}, facet_col="income_range")
plot.show()
```

- Relationship analysis

```
df = pd.read_csv('adult.data', header=None)
df.reset_index(inplace=True)
df = df.rename(columns = {'index':'id'})
df = df.rename(columns={0: "age", 1: "workClass", 2:"fnlwgt", 3:"education",
4:"education-num", 5:"marital-status", 6:"occupation", 7:"relationship", 8:"race", 9:"sex",
10:"capital-gain", 11:"capital-loss", 12:"hours-per-week", 13:"native-country",
14:"income"})
df = df.replace({"^\s*|\s*$":""}, regex=True)
df_below_50K = df[df['income'].isin(["<=50K"])]
df_above_50K = df[df['income'].isin([">50K"])]
```

```
data_above_50K = df_above_50K.groupby('relationship')['id'].nunique()
data_below_50K = df_below_50K.groupby('relationship')['id'].nunique()
multiplicationFactor = data_below_50K.sum()/data_above_50K.sum()
upsampled_data_above_50K = data_above_50K.multiply(multiplicationFactor)
labels = ['Husband', 'Not-in-family', 'Other-relative', 'Own-child', 'Unmarried', 'Wife']
width = 0.35  # the width of the bars
data_above_50K = data_above_50K.tolist()
data_below_50K = data_below_50K.tolist()
x1 = list(range(0,6))
x2 = list(range(0,6))
for i in range(len(x1)):
    x1[i] = x1[i] - (width/2)
for i in range(len(x2)):
    x2[i] = x2[i] + (width/2)
plt.figure(figsize=(10,4))
plt.subplot(1, 2, 1)
plt.bar(x1, data_below_50K, width, label='below_50K')
plt.bar(x2, upsampled_data_above_50K, width, label='upsampled_above_50K')
plt.ylabel('Count')
plt.title('Count by Relationship and income')
plt.xticks(x1, rotation=90, labels=labels)
plt.legend()
plt.subplot(1, 2, 2)
plt.bar(x1, data_below_50K, width, label='below_50K')
plt.bar(x2, data_above_50K, width, label='above_50K')
plt.xticks(x1, rotation=90, labels=labels)
plt.ylabel('Count')
plt.title('Count by Relationship and income')
plt.legend()
```

- Work class analysis

```
df = pd.read_csv('adult.data', header=None)
df.reset_index(inplace=True)
df = df.rename(columns = {'index':'id'})
df = df.rename(columns={0: "age", 1: "workClass", 2:"fnlwgt", 3:"education",
4:"education-num", 5:"marital-status", 6:"occupation", 7:"relationship", 8:"race", 9:"sex",
10:"capital-gain", 11:"capital-loss", 12:"hours-per-week", 13:"native-country",
14:"income"})
df = df.replace({"^\s*|\s*$":""}, regex=True)
df_below_50K = df[df['income'].isin(["<=50K"])]
df_above_50K = df[df['income'].isin([">50K"])]
data_above_50K = df_above_50K.groupby('workClass')['id'].nunique()
data_below_50K = df_below_50K.groupby('workClass')['id'].nunique()
multiplicationFactor = data_below_50K.sum()/data_above_50K.sum()
upsampled_data_above_50K = data_above_50K.multiply(multiplicationFactor)
width = 0.35  # the width of the bars
data_above_50K = data_above_50K.tolist()
```

```
data_below_50K = data_below_50K.tolist()
data_above_50K.insert(3,0)
upsampled_data_above_50K = upsampled_data_above_50K.tolist()
upsampled_data_above_50K.insert(3,0)
data_above_50K.insert(8,0)
upsampled_data_above_50K.insert(8,0)
x1 = list(range(0,8))
x2 = list(range(0,8))
for i in range(len(x1)):
    x1[i] = x1[i] - (width/2)
for i in range(len(x2)):
    x2[i] = x2[i] + (width/2)
data_above_50K.pop(0)
data_below_50K.pop(0)
labels = ['Federal-gov', 'Local-gov', 'Never-worked', 'Private', 'Self-emp-inc',
'Self-emp-not-inc', 'State-gov', 'Without-pay']
upsampled_data_above_50K.pop(0)
plt.figure(figsize=(10,4))
plt.subplot(1, 2, 1)
plt.bar(x1, data_below_50K, width, label='below_50K')
plt.bar(x2, upsampled_data_above_50K, width, label='upsampled_above_50K')
plt.ylabel('Count')
plt.title('Count by Work class and income')
plt.xticks(x1, rotation=90, labels=labels)
plt.legend(bbox_to_anchor=(1.0, 0.8), loc=1, borderaxespad=0.)
plt.subplot(1, 2, 2)
plt.bar(x1, data_below_50K, width, label='below_50K')
plt.bar(x2, data_above_50K, width, label='above_50K')
plt.xticks(x1, rotation=90, labels=labels)
#plt.ylabel('Count')
plt.title('Count by Work class and income')
plt.legend()
```