Marketing strategies for UVW Executives using Data Visualization & Machine Learning Techniques

CSE 578 Data Visualization

System Documentation Report

Group: Sparky Visualizers

Team Members

Arunkumar Selvaraj

Raghuveer Gowda H

Ravi Teja Upmaka

Shubham Pathak

1. Roles and Responsibilities

1.1. Arunkumar Selvaraj

- Dataset analysis
- Plotting multiple visualizations for all features against salary label as <=50K or >50 to understand which features have more discriminating power.
- Plotting features against one another to determine redundant features (Features like education number and education have high correlation so they were eliminated).
- Preprocessing operations like removal of unnecessary columns, normalizing the data and converting the categorical data into numerical data.
- Trained and evaluated accuracies for SVM, Linear SVM.
- Visualize the decision tree for further analysis and obtaining the decision rules.
- Visualize the decision rules and form an analysis for executives.

1.2. Raghuveer Gowda H

- Dataset analysis
- Plotting multiple visualizations for all features against salary label as <=50K or >50 to understand which features have more discriminating power.
- Plotting features against one another to determine redundant features (Features like education number and education have high correlation so they were eliminated).
- Preprocessing operations like removal of unnecessary columns, normalizing the data and converting the categorical data into numerical data.
- Trained and evaluated accuracies for SVM, Linear SVM.
- Visualize the decision tree for further analysis and obtaining the decision rules.
- Visualize the decision rules and form an analysis for executives.

1.3. Ravi Teja Upmaka

- Selection of decision tree-based classifier by analyzing pros and cons against SVM and similar techniques.
- Build decision tree model on the training dataset.
- Select the stopping criteria for decision tree by evaluating criteria like entropy and Gini index and decide the maximum depth levels for tree.
- Evaluate the accuracy of the model in terms of f1 score, precision and recall by dividing the training data set into 80:20 ratio for training and testing, and for separate testing data set provided as well.
- Visualize the decision tree for further analysis and obtaining the decision rules.
- Visualize the decision rules and form an analysis for executives.

1.4. Shubham Pathak

- Selection of decision tree-based classifier by analyzing pros and cons against SVM and similar techniques.
- Build decision tree model on the training dataset.

- Select the stopping criteria for decision tree by evaluating criteria like entropy and Gini index and decide the maximum depth levels for tree.
- Evaluate the accuracy of the model in terms of f1 score, precision and recall by dividing the training data set into 80:20 ratio for training and testing, and for separate testing data set provided as well.
- Visualize the decision tree for further analysis and obtaining the decision rules.
- Visualize the decision rules and form an analysis for executives.

2. Team Goals and Business objective

- Analysis of dataset with the help of visualizations
- Building a model to predict whether a person has income <=50K or >50K.
- Visualizing the model and criteria for targeting the audiences for enrollment.
- Forming a marketing analysis and advertising policies based on visualizations for right audiences to increase the enrollment.

3. Assumptions

- The input salary dataset is up to date and updated frequently to match the current market trends.
- The dataset must have the 'salary' feature.
- The input dataset contains salary details of the people who is in the same region as the UVW university.
- The dataset contains only finite number of features and data objects.
- The dataset contains only valid and finite values for each of the features. Should there be an invalid or infinite value, it should be denoted as '?' which can be removed during the preprocessing step.

4. User Stories

- US 1: Data Preprocessing
  - Task 1: Remove invalid data (question mark).
  - Task 2: Adding labels (target feature) to data objects based on the income, to do binary classification.

- US 2: Feature Selection
  - Task 1: Plot features against the added target label to validate feature's association with the target label (salary).

- Task 2: Based on the visualizations created in the previous task, select the most discriminating features.
- Task 3: Modify the dataset to remove the features that is unrelated to the target feature.

- US 3: Training Model
  - Task 1: Due to the unequal distribution of the labelled data, upsampling techniques provided in Scikit library were performed.
  - Task 2: Train linear SVM and Decision Tree classifier to classify the unlabeled data (<=50K or >50K).

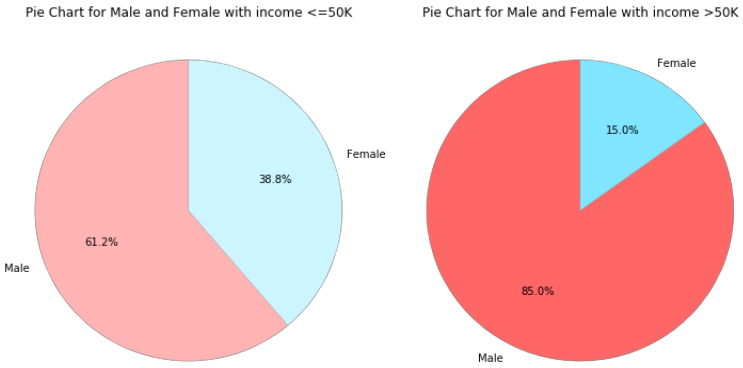
- US 4: Evaluation of models
  - Task 1: Tested the Decision Tree classifier for different depths (depth of the tree) and evaluated them based on the entropy values.
  - Task 2: Calculate F1 score, Precision and Recall values to evaluate the accuracy of the trained model.

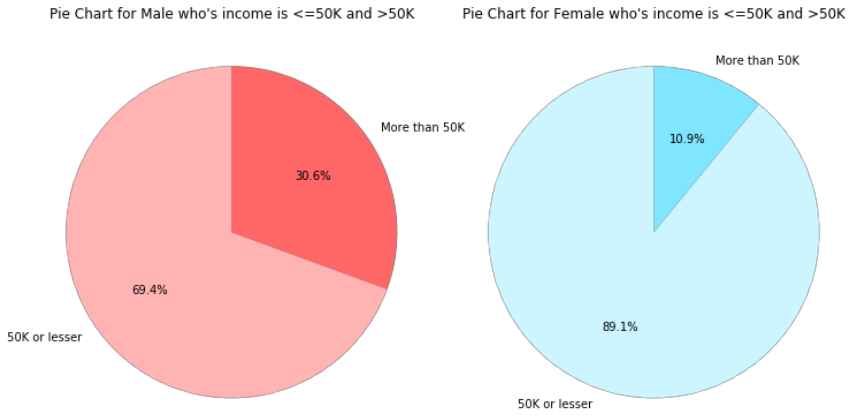
- US 5: Marketing analysis for UVW executives
  - Task 1: Visualize the decision tree classifier to get the decision rules.
  - Task 2: Plot the obtained decision rules using various visualization techniques including bar charts, pie charts, scatter plots, etc.,
  - Task 3: Using the visualizations created in the previous step, understand the feature characteristics of the corresponding data segment.
  - Task 4: Use the derived feature characteristics to create appropriate marketing strategies.

5. Visualizations

5.1. Pre-processing

5.1.1. Gender





Explanation

We have plotted pie charts to compare the ratio of male and female employees who earn 50K or lesser and also, between male and female who earn more than 50K. This can used to understand how the 'sex' of a person can be used to estimate the association with his/her salary and marketed.

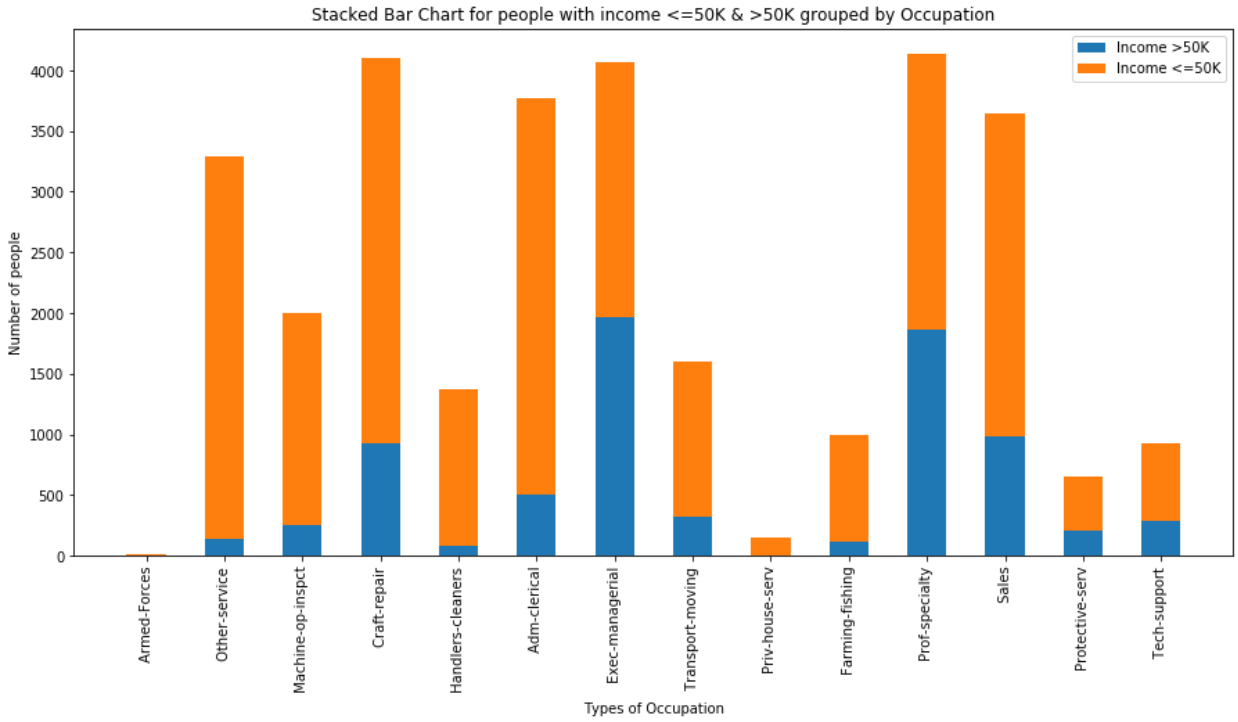
For a very similar reason, we have plotted pie charts to compare the ratio of males who earn 50K or lesser and earn more than 50K, and females to compare the same ratio. This can be used to understand given the 'sex' of a person, how probable is it for him/her to earn the <=50K and >50K, and can be chosen to market appropriately.

Code

```
import numpy as np

less_income_sex = adult_df[adult_df.income.str.contains("<=50K")]['sex']
high_income_sex = adult_df[adult_df.income.str.contains(">50K")]['sex']
less_income_list = less_income_sex.tolist()
high_income_list = high_income_sex.tolist()

male_less_income = less_income_list.count(' Male')
female_less_income = less_income_list.count(' Female')
male_high_income = high_income_list.count(' Male')
female_high_income = high_income_list.count(' Female')
total_male = male_less_income + male_high_income
total_female = female_less_income + female_high_income

female_high_income_color = '#80e5ff'
female_less_income_color = '#ccf5ff'
male_high_income_color = '#ff6666'
male_less_income_color = '#ffb3b3'
```

```
fig = plt.figure(figsize = (17, 10))
labels = 'Male', 'Female'

less_income_data = [male_less_income, female_less_income]
ax1 = fig.add_axes([0, 0, .5, .5], aspect = 1)
ax1.pie(less_income_data, colors = [male_less_income_color, female_less_income_color], labels = labels,
autopct = '%1.1f%%', startangle = 90)

high_income_data = [male_high_income, female_high_income]
ax2 = fig.add_axes([0.3, .0, .5, .5], aspect = 1)
ax2.pie(high_income_data, colors = [male_high_income_color, female_high_income_color], labels =
labels, autopct = '%1.1f%%', startangle = 90)

ax1.set_title('Pie Chart for Male and Female with income <=50K')
ax2.set_title('Pie Chart for Male and Female with income >50K')
plt.show()

fig2 = plt.figure(figsize = (17, 10))
labels2 = '50K or lesser', 'More than 50K'

male_data = [male_less_income, male_high_income]
ax1 = fig2.add_axes([0, 0, .5, .5], aspect = 1)
ax1.pie(male_data, colors = [male_less_income_color, male_high_income_color], labels = labels2, autopct
= '%1.1f%%', startangle = 90)

female_data = [female_less_income, female_high_income]
ax2 = fig2.add_axes([0.3, .0, .5, .5], aspect = 1)
ax2.pie(female_data, colors = [female_less_income_color, female_high_income_color], labels = labels2,
autopct = '%1.1f%%', startangle = 90)

ax1.set_title('Pie Chart for Male who\'s income is <=50K and >50K')
ax2.set_title('Pie Chart for Female who\'s income is <=50K and >50K')

plt.show()
```

5.1.2. Occupation



Explanation

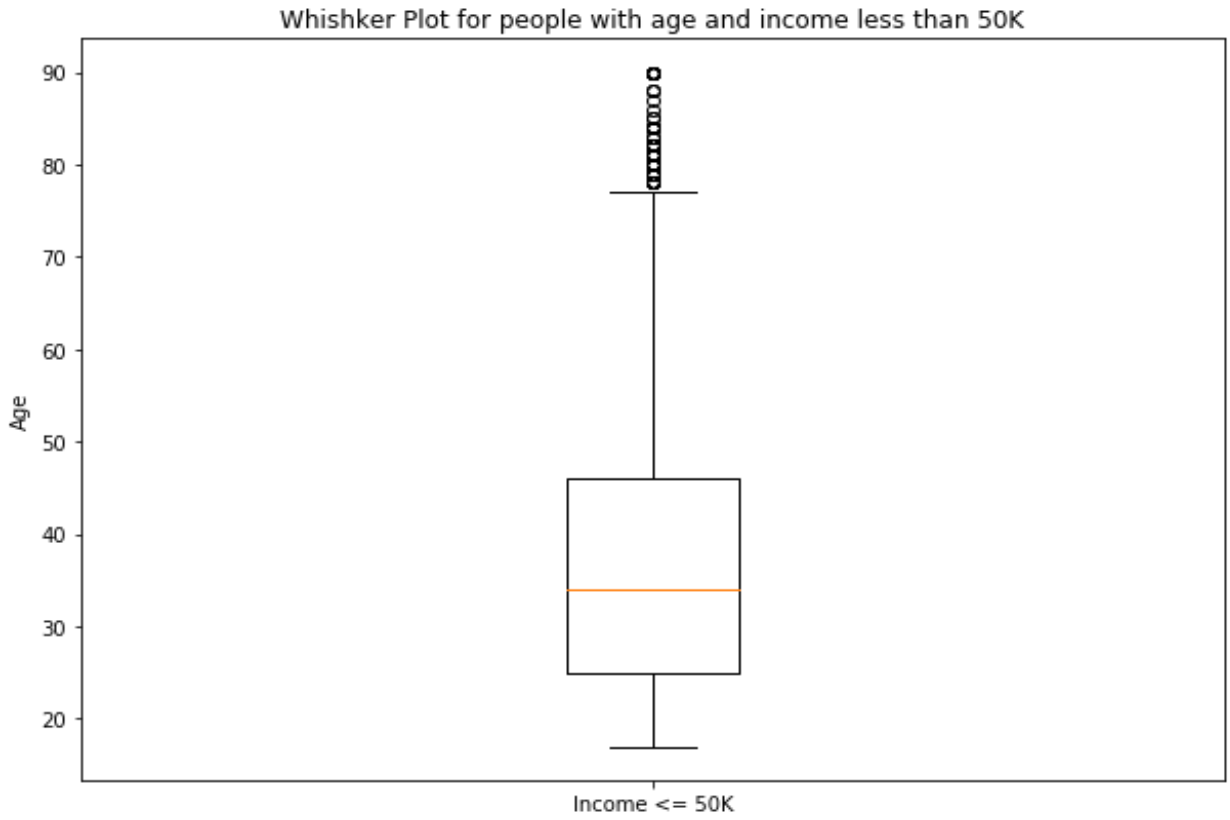
This is one of the important visualizations in our project which explains the importance and significance of the feature, 'Occupation'. The above stacked bar chart visualization proves that the occupation feature is directly associated with the salary. It also tells that the occupation type of any given user can be directly used to estimate his earnings and so, used to market in the right way for best results.

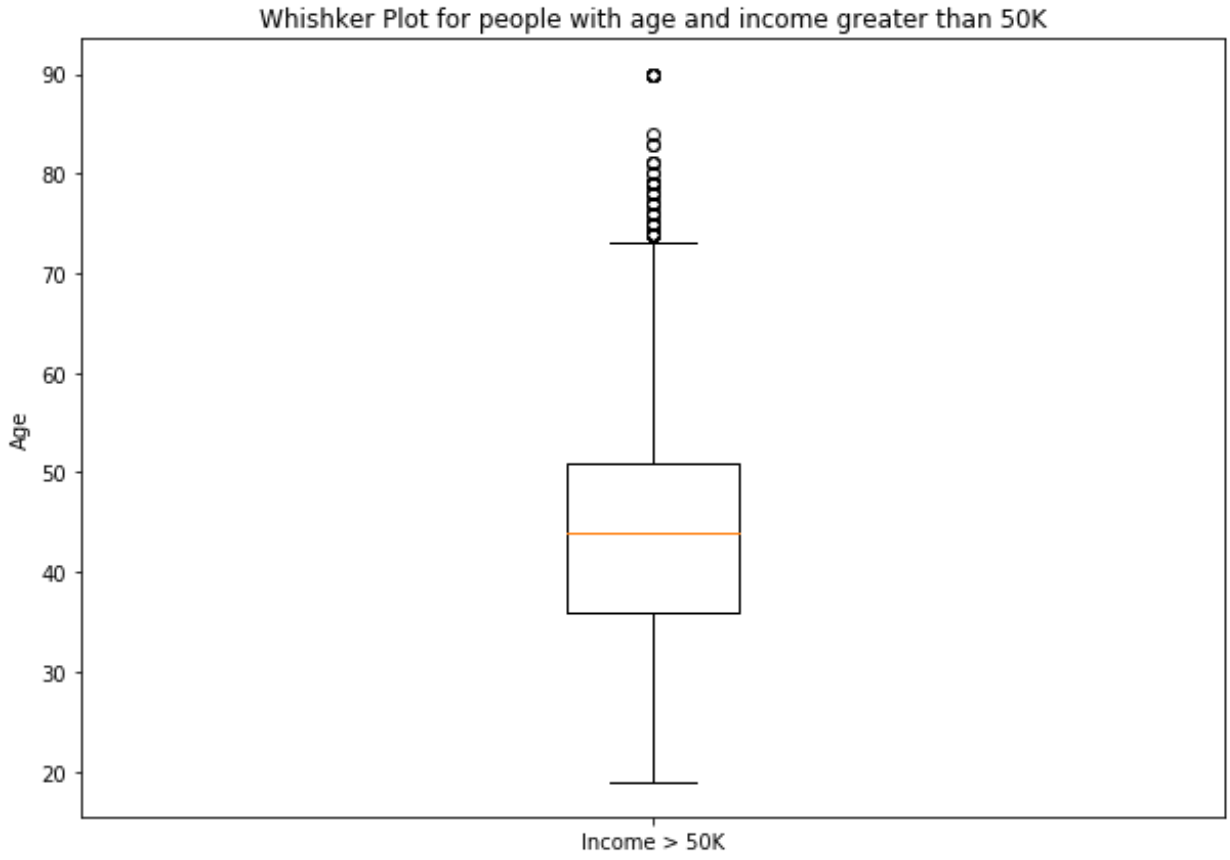
Code

```
occupation = adult_df['occupation']
occupation_list = occupation.tolist()
occupation_set = set(occupation_list);
occupation_set.remove(' ?')

less_income_occupation = adult_df[adult_df.income.str.contains("<=50K")]['occupation']
high_income_occupation = adult_df[adult_df.income.str.contains(">50K")]['occupation']
less_income_occupation_list = less_income_occupation.tolist()
high_income_occupation_list = high_income_occupation.tolist()

less_income_list = []
high_income_list = []
for occupation in occupation_set:
  less_income_count = less_income_occupation_list.count(occupation)
```

```
  less_income_list.append(less_income_count)
  high_income_count = high_income_occupation_list.count(occupation)
  high_income_list.append(high_income_count)

N = len(occupation_set)
index = np.arange(N)
width = 0.5
plt.figure(figsize=(15,7))
p1 = plt.bar(index, high_income_list, width)
p2 = plt.bar(index, less_income_list, width, bottom = high_income_list)
-plt.ylabel('Number of people')
plt.xlabel('Types of Occupation')
plt.title('Stacked Bar Chart for people with income <=50K & >50K grouped by Occupation')
plt.xticks(index, occupation_set)
plt.xticks(rotation=90)
plt.legend((p1[0], p2[0]), ('Income >50K', 'Income <=50K'))
plt.show()
```

5.1.3. Age





Explanation

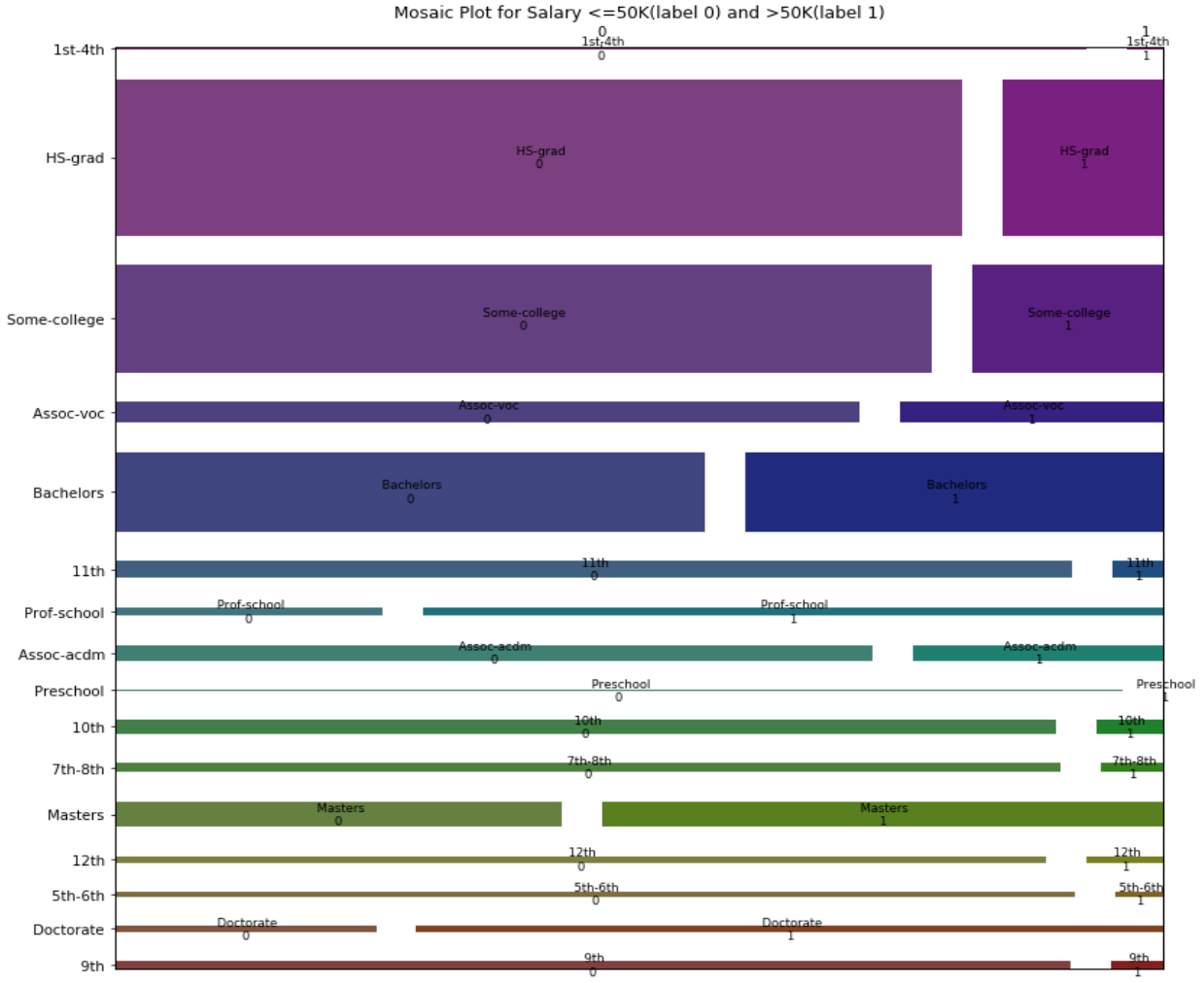
If we look at the whisker plots for age for both the labels, income >50K and income <=50K, we can see that the Q2 quartile or the medians for both the plots are fairly separated. So, age can be one of the attribute that can be used for marketing.

Code

```
import pandas as pd
import matplotlib.pyplot as plt

adult_df = pd.read_csv('adult.csv')
income_df = adult_df[adult_df.income.str.contains("<=50K")]['age']
age_list = income_df.tolist()
plt.figure(figsize=(10,7))
plt.boxplot(age_list, labels=['Income <= 50K'])
plt.ylabel('Age')
plt.title('Whisker Plot for people with age and income less than 50K')
plt.show()

gt_income_df = adult_df[adult_df.income.str.contains(">50K")]['age']
gt_age_list = gt_income_df.tolist()
```

plt.figure(figsize=(10,7))
plt.boxplot(gt_age_list, labels=['Income > 50K'])
plt.ylabel('Age')
plt.title('Whishker Plot for people with age and income greater than 50K')
plt.show()

5.1.4. Education



Explanation

The above visualization represents the mosaic plot for education vs income. For each education level or degree, we can clearly observe the difference in proportion of people earning income <=50K and >50K. This is one of the important attributes that can be used for further grouping of the people for marketing.

Code

```
import statsmodels.graphics.mosaicplot as mosaicplot

data = {}
filtered_df = adult_df[['education', 'income']]
education_set = set(adult_df['education'].tolist())
print(education_set)
for row in education_set:
  temp_df = filtered_df[filtered_df.education.str.contains(row)]
  data[(row, 0)] = temp_df[temp_df.income.str.contains("<=50K")].shape[0];
  data[(row, 1)] = temp_df[temp_df.income.str.contains(">50K")].shape[0];
with plt.rc_context():
  plt.rc("figure", figsize=(13,13))
  mosaicplot.mosaic(data, gap=0.06, title='Mosaic Plot for Salary <=50K(label 0) and >50K(label 1)',
statistic = False, horizontal = False)
plt.show()
```

5.1.5. Result

Using the above visualizations, we have dropped few features including *education* (As we have *education-num* feature which gives the education level of the candidate, we can easily map the *education-num* to highest level of education), and also dropped the feature *fnlwgt* (the given dataset does not have clear information regarding what the feature *fnlwgt* corresponds to!).

Similarly, we have plotted the visualizations for the features and found that *age*, *occupation* and *gender* have a good discriminating power to build a model for the given dataset.

5.2. Training the classifier model

5.2.1. Machine Learning Algorithm used - Decision Trees

5.2.2. Why Decision Trees?

Decision trees repetitively divide the working data space into sub-regions by identifying decision boundaries or lines. This helps us in identifying the regions in vector space. This can be done by tracing the decision rules from the classifier which are based on the features that we have selected earlier. So while making a marketing policy for UVW college, we can target the specific set of population with the help of these identified sets obtained with respect to features like age, capital-gain, education etc. in our data space.

Therefore, an important question arises that when does the continuous division of data space end, there will be two cases when this would happen i.e. when pure classes are obtained (often leads to overfitting)

and certain conditions are met. For the second part, we have concepts like impurity then based on it the concept of entropy, information gain and Gini index.

5.2.3. Impurity and Entropy

Impurity is present in our classification model when the traces of one class label are present in other. Whereas, entropy is defined as the degree of randomness or measure of impurity. And is given as:

$$H = -\sum p(x) \log p(x)$$

We have used entropy as a criterion for building the model and corresponding entropy values can be observed in the visualization obtained post classification at every split of the node. 0 entropy value means pure classes whereas 1 means equal number of samples from either class are present. However, we can't solely rely on entropy as a criterion since the entropy values might provide a deceptive image of classification, so considering the number of samples for which it is being calculated is also important. These factors can be observed in the visualization.

5.2.4. Code for preprocessing and decision tree classification

```
import pandas as pd
from sklearn.utils import resample
from sklearn import tree
from sklearn.preprocessing import StandardScaler
from sklearn import svm
import sklearn.metrics as sm
import numpy as np
from sklearn import tree
from dtreeviz.trees import *

# method for assigning the labels to each data point; '1' for candidates having income more than 50K and '0' for people having income <= 50K
def get_label(row):
  if "<=50K" in row.income:
    return 0
  return 1

# Reading the csv file using pandas and filling out the empty cell values with zero's
adult_df = pd.read_csv("C:/Users/ravit/Project/adult.csv")
adult_df = adult_df.fillna(0)

adult_df['label'] = adult_df.apply(lambda row: get_label(row), axis=1)
adult_df.drop(columns=['native-country','income', 'fnlwgt', 'education'],inplace=True)

# Factorizing the columns of data frame as all of them have to be in integer format to use the scikit library decision tree classifier
adult_df['workclass'], val1 = pd.factorize(adult_df['workclass'])
```

```
adult_df['marital-status'], val2 = pd.factorize(adult_df['marital-status'])
adult_df['occupation'], val3 = pd.factorize(adult_df['occupation'])
adult_df['relationship'], val4 = pd.factorize(adult_df['relationship'])
adult_df['race'], val5 = pd.factorize(adult_df['race'])
adult_df['sex'], val6 = pd.factorize(adult_df['sex'])

adult_df.to_csv("factorized_file.csv", sep=',', encoding='utf-8')
label0_df = adult_df[adult_df.label == 0]
label1_df = adult_df[adult_df.label == 1]

#Upsampling the data as the given training data has an unequal distribution labelled data
df_minority_upsampled = resample(label1_df, replace=True, n_samples = label0_df.shape[0],
random_state=123)

result_df = pd.concat([label0_df, df_minority_upsampled])

msk = np.random.rand(len(result_df)) < 0.8
train_df = result_df[msk]
test_df = result_df[~msk]
df_upsampled = train_df.copy()

y = df_upsampled.label
X = df_upsampled.drop('label', axis=1)
feature_list = list(result_df)
print(feature_list)

y_test=test_df.label
x_test=test_df.drop('label', axis=1)

# model = svm.SVC(kernel='linear', class_weight='balanced', C=1, gamma=1)
model= tree.DecisionTreeClassifier(criterion='entropy', max_depth=5, random_state=0)
# scaler = StandardScaler()

# X_std = scaler.fit_transform(X)
# x_test_std = scaler.fit_transform(x_test)

model.fit(X, y)
predicted= model.predict(x_test)

print('training score',model.score(X, y))
print('testing score',sm.accuracy_score(y_test,predicted))
print('f1 score',sm.f1_score(y_test,predicted))
print('recall',sm.recall_score(y_test,predicted))
print('precision',sm.precision_score(y_test,predicted))

tree.export_graphviz(model, out_file='tree.dot', feature_names=feature_list[:len(feature_list)-1])
Source.from_file('tree.dot')
```

viz = dtreeviz(model, X, y, target_name='variety', feature_names=feature_list[:len(feature_list)-1], class_names=["0", "1"]) # need class_names for classifier)

viz.view()

5.3. Evaluation of Trained models

5.3.1. Performance Measures

For the decision tree classifier, in order to avoid the problem of overfitting the model, we have pruned the decision tree at a mid-level depth of 5. This ensures that the decision tree does not create too many chunks or clusters of the data space which makes the task of executives and marketing people difficult when it comes to targeting audiences. We want to ensure that larger and appropriate audience is being targeted by UVW College so after testing with max depth levels ranging from 4 to 10 and by keeping no limit on the depth levels (by overfitting model) we have observed that good decision rules can be obtained for values in range of 5 to 9.

The accuracies were calculated in terms of precision, recall and finally in terms of f1 score below,

Precision: It is the number of True Positives divided by the number of True Positives and False Positives.
Recall: It is the number of True Positives divided by the number of True Positives and the number of False Negatives.
F1 Score: It is computed as *2*((precision * recall) / (precision + recall))*. It is also called the F Score or the F Measure. Put another way, the F1 score conveys the balance between the precision and the recall.

The accuracy values that we have obtained by restricting the max depth of the decision tree to 5 are as follows,

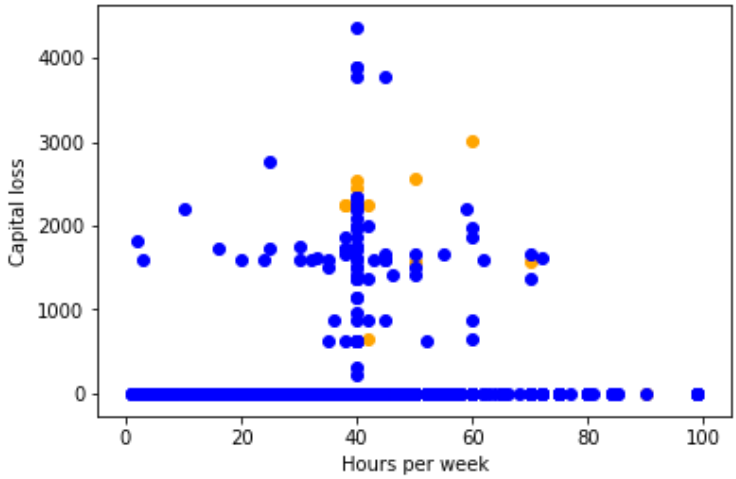
F1 score: 0.8094291394490202
Recall: 0.8687258687258688
Precision: 0.7577100319035803

Increasing the number of levels results in an improved f1 score at the cost of more chunks of distinct regions in data space and might result in overfitting the model.

The visualizations for the decision tree classifier can be found at
https://drive.google.com/open?id=135Lyo03S8iRPU3fjNQ-eGbC3XY61xXci &
https://drive.google.com/open?id=10NvwViKIXO94_oDMGcRjGnz3uGF5UVrO

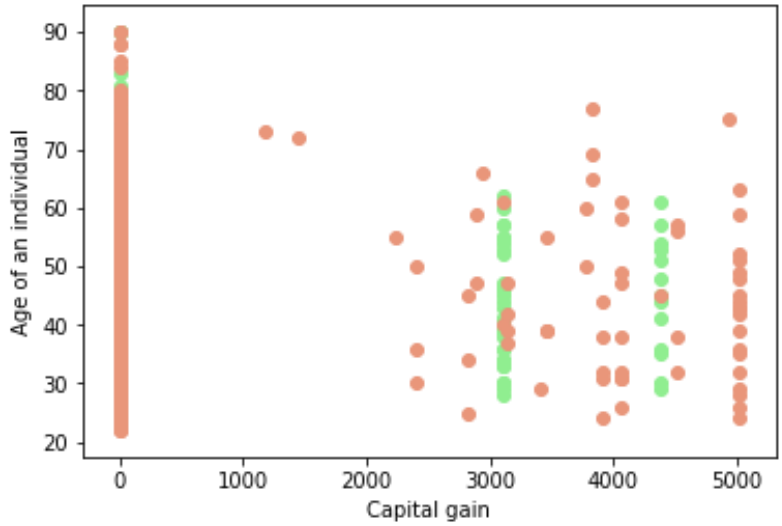
5.4. Post Classification Analysis

5.4.1. Analysis 1



Explanation: As we can see for age group >= 29 years having a capital loss, with constraints like marital status as unmarried and other constraints, the majority of this population works in the range of 20-60 hours and have an income < 50K as suggested by the blue dots in the scatter plots. So UVW College can target this segment of population by offering degree programs above 'Assoc-acdm' and provide flexibility with college hours. Also for the segment with capital loss > 1000 and that works less than 40 hours, tuition fee waivers in form of research roles can be provided which will aid in the research growth of UVW and the crowd that needs to cover the capital loss will benefit from it.

Analysis 2



Explanation: We can see we can separate the reduced data segment using decision boundaries as shown in the figure. We can see that the majority of the population <= 50K salary did not have capital gain for this

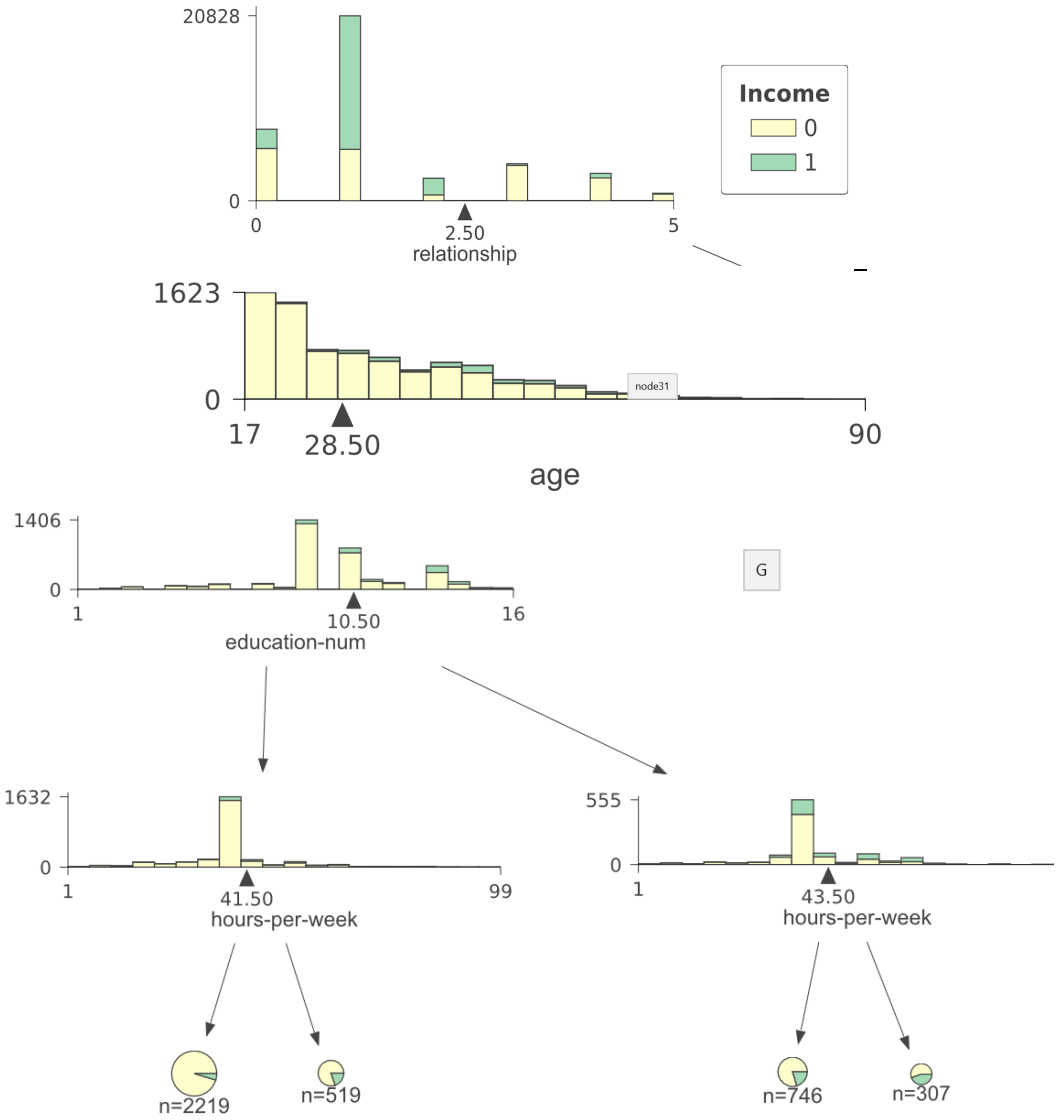
particular split in the node (i.e. segment with constraints like, relationship status in ['Husband', 'Wife'] and education-num >= 13 and capital-gain <= 5095.0). In order to target this audience some form of financial aid like scholarships can be utilized to increase the enrollment from this section.

Analysis 3

A sample decision rule focused from stacked bar charts for tree visualization:

For the following decision rule, relationship status in [Unmarried, Own-child, Other-relative] and age >= 29 years and education-level <= 'Assoc-acdm'.

We have the following visualization with 0 indicating income <= 50K and 1 indicating > 50K:



Explanation: The decision rule vividly displays that for the constraints with average age 28.5 and education number 10.5 the majority of the population working for 40-45 hours gets a salary <50k. In order to promote the enrollment program for this segment of the population that works for 40 hours, flexible programs or professional degree programs can be introduced by UVW executives that provide evening classes or online course degrees.

6. Questions

- Which classifier should be selected and why?
    - Decision Tree Classifier was used because it is possible to extract the rules from the generated model which can be used for identification of right audiences.

- Is there any need for considering the hidden features or using feature reduction technique?
    - Hidden features can be considered for increasing the accuracy, but it will be difficult to map them to a data segment because of its difficulty to understand their physical significance.
    - So, PCA was not considered, instead we visualized the features to identify the redundant and useful ones.

- Which stopping criteria for the decision tree should be selected?
    - Entropy and Information gain are two most widely used criteria. Entropy gives the number of samples in each node and indicates degree of impurity compared to information gain which in this particular case can be used to identify the enrollment audience. So, entropy was selected as a stopping criterion.

7. Future Scope

- Ensemble learning technique can be used to obtain better classification based on the weighted accuracy of the multiple classifiers that's implemented.
- The geographical details can be incorporated that gives a weight to the data points based on their location and proximity to the given university.
- Principal Component Analysis can also be done to uncover the hidden features of the dataset and the same can be used for better analysis of the data.

The entire project work can be found at https://drive.google.com/open?id=1SuYL-4lr56K4loi62kCzuc4Ar8YPdR9s