Random Forest Classification of Income Evaluation

Randall Beeman
Nayeong Kong, Mentor
Ramish Karki, Editor

Keywords

data science, data collection, data analysis, Random Forest methodology, data presentation

1 Abstract

We analyzed a dataset of income evaluation with a combination of python and a data science technique known as a random forest, where we build connec-tions between data points to understand which data values are most significant. Through this process we find that self-employed people have a higher incidence of making more than $50,000 when compared with private industry or govern-ment employees.

2 Introduction

Data Science is a domain of computer science that utilizes mathematics, statistics, analytics, machine learning and more to provide support to decision making and future planning. In this endeavour data science uses varying data sources, algorithms, and techniques to provide these insights. While data science has become much more known over the last few years, it dates back to 1962 with a Mathematician named John W. Tukey who is first credited expressing the value that can come from analyzing data sources ([2]). It would be a number of years until computing power caught up with Tukey's ideas.

Moving forward to the early 2000s, data science began to emerge as a discipline in its own right and in 2005 the National Science Board advocated data science become a career path to allow for individuals become experts in the field ([4]). Over the intervening years, data science has become increasingly important to both large companies looking to maximize the value of their data and small organizations looking to have the largest impact. Data science has continued to grow and data scientists are increasingly making impacts in various domains such as technology, economics, health care, education, and sports, to name a few.

A project in data science has three major steps. The first is data collection, this is either finding a data set that has already been collected or collecting new data related to the topic that is being studied. The second major step, is analysing the data. This can take multiple forms depending on the methods

that are being implemented. We will be focusing on a method called a Random Forest here. However, there are many other techniques that each have their own advantages and considerations depending on the type and shape of data being analyzed. The last large step is communicating findings t hat resulted from steps one and two. This step should not be understated and is just as important, or perhaps more important, than the previous steps since this is where the conclusions are relayed to other stakeholders that need to trust the results to allow decisions to be made from the analysis.

3 Random Forest

A Random Forest classification method is a machine learning algorithm that is based on the use of multiple decision trees ([1]). Decision trees, on their own, can have bias or overfit data and decrease the validity of conclusions ([3]). This brings us to our first definitions:

Definition 3 .1. *A d ecision t ree i s a n on p arametric s upervised l earning algorithm used for classification a nd r egression t asks. I t c onsists o f t wo t ypes of elements, nodes and branches. A node is a feature of the data while a branch connects to another feature of the data.*

Definition 3 .2. *A random f orest is an algorithm that combines a large number of decision trees, where each tree casts a vote for the most relevant input nodes.*

On the other hand, ensemble learning methods are generated from a set of classifiers s uch a s d ecision t rees a nd g enerate t heir p redictions f rom multiple independent runs of the classifiers. R andom f orest m odeling i s a method that utilizes bagging and feature randomness to generate an uncorrelated set of decision trees, or a random forest.

4 Main Body

We reviewed a dataset on income classification ([5]). T his c ontained 32561 entries and has information regarding age, work-class, education, marital status, income, hours worked, and more. To evaluate the data we used a random-forest machine learning algorithm. The python code used to evaluate this data set came from the same source as the data ([5]).

In reviewing this data we see that self employed people have the highest percentage of people with above $50,000 incomes while privately employed has the lowest percentage of people with above $50,000 incomes. Private industry workers had 21.87% of workers making above $50,000, while government workers had 30.82% making above $50,000 and self employed had 36.81% making above $50,000.

The method of evaluating the data was done using Python to generate the random forest. Before working with the data, we want to make sure we import some of the relevant modules we will need to work with. The `numpy` module

allows us to do mathematics more efficiently, `pandas` allows us access to data analysis tools, `os` provides tools for interacting with the operating system, and `seaborn` along with `matplotlib` provide data visualization tools. The relevant code for importing them is here:

```
import numpy as np
import pandas as pd
import os
import seaborn as sns
import matplotlib.pyplot as plt
```

We also used parts of the `scikit` module for various data science tools, these provide us with efficient access to set up a random forest predictor for our data as well as methods to analyze the predictive power of the random forest.

```
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import RobustScaler
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score
from sklearn.metrics import confusion_matrix
from sklearn.metrics import classification_report
```

Using `matplotlib`, we generated Figure 1 which shows the frequency of each type of work class in the data.

```
f, ax = plt.subplots(figsize=(10, 6))
ax = df.workclass.value_counts().plot(kind="bar", color="green")
ax.set_title("Frequency distribution of workclass variable")
ax.set_xticklabels(df.workclass.value_counts().index, rotation=30)
plt.show()
```

From that, we wanted to visualize how workclass and income related to each other. We used `matplotlib` to generate Figure 2, which shows the income differentiation for workclass. The data was classified for above $50,000 and less than or equal to $50,000, so no finer distinction was possible. Below we see the python code to generate this plot, in it we set the x-axis to be the workclass variable and the categorize by the income.

```
f, ax = plt.subplots(figsize=(12, 8))
ax = sns.countplot(x="workclass", hue="income", data=df, palette="
                                    Set1")
ax.set_title("Frequency distribution of workclass variable wrt
                                    income")
ax.legend(loc='upper right')
plt.show()
```

From the data we also needed to split the dataset into a training set and a test set. The training set is to run our random forest on and compare the predictions of the random forest on our test set. In this process we are also making sure to differentiate between numerical and categorical data entries.

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size
                                    = 0.3, random_state = 0)
categorical = [col for col in X_train.columns if X_train[col].
                                   dtypes == 'O']
numerical = [col for col in X_train.columns if X_train[col].dtypes
                                   != 'O']
for col in categorical:
    if X_train[col].isnull().mean()>0:
        print(col, (X_train[col].isnull().mean()))
for df2 in [X_train, X_test]:
    df2['workclass'].fillna(X_train['workclass'].mode()[0], inplace
                                    =True)
    df2['occupation'].fillna(X_train['occupation'].mode()[0],
                                    inplace=True)
    df2['native_country'].fillna(X_train['native_country'].mode()[0
                                    ], inplace=True)
```

Following splitting the test and train data sets out from the original data set we are then able to use `scikit` RandomForestClassifier to generate predictive models, we did this with 10 estimators and then 100 estimators to see the difference in the effectiveness between the number of estimators.

```
rfc = RandomForestClassifier(n_estimators=10, random_state=0)
rfc.fit(X_train, y_train)
y_pred = rfc.predict(X_test)
print('Model accuracy score with 10 decision-trees :{0:0.4f}'.
                                   format(accuracy_score(y_test,
                                   y_pred)))
rfc_100 = RandomForestClassifier(n_estimators=100, random_state=0)
rfc_100.fit(X_train, y_train)
y_pred_100 = rfc_100.predict(X_test)
print('Model accuracy score with 100 decision-trees :{0:0.4f}'.
                                   format(accuracy_score(y_test,
                                   y_pred_100)))
```

5 Data Presentation

In using the random forest we see that the class of employment is a strong indicator of income above $50,000. The self employed group contained 3657 samples, which included individuals from two workclass categories. Self-emp-not-inc, which are non-incorporated businesses such as sole-proprietorships and Self-emp-inc, which are incorporated busnesses such as S-corps. The government work class group which contained 4351 samples. This is a combination of the classifications State-gov, employees in state government, Federal-gov, employees of the federal government, and Local-gov, employees in local government. Private workers were the largest sample at 22696. Further, there were 21 samples that either worked without pay or had never worked.

The precision from the random forest for predicting above $50,000 was 0.89 and below $50,000 was 0.73. These precision values were found using `classification_report` from the `sklearn` module, and are calculated by tak-

ing total positives divided by the sum of total positives and false positives. The random forest was generated with both 10 and 100 decision trees, when using 10 decision trees the accuracy score was 0.8446 and with 100 decision trees it had an accuracy of 0.8521. This was done with `accuracy_score` from the `scikit` module in Python.

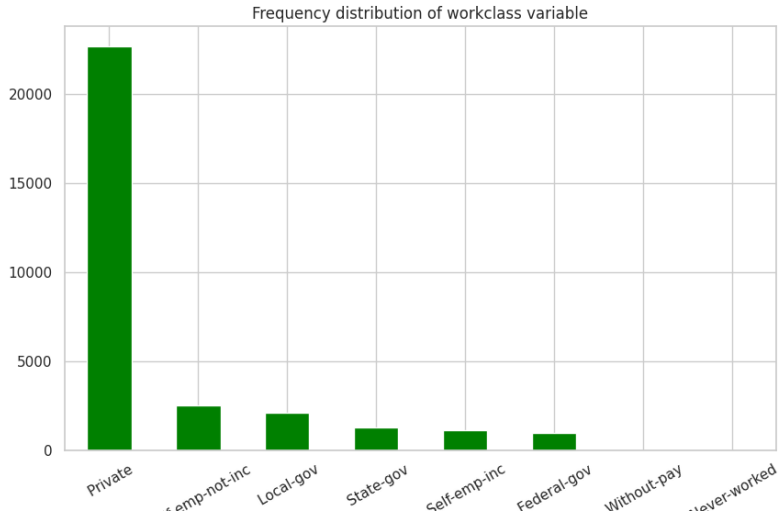

Figure 1: Frequency of work class

6 Conclusion

Random Forest Classification can be an effective tool to draw out high validity conclusions from data when used with the limitations in mind, and is much more robust than a single decision tree. One of the limitations of a Random Forest, since it is aggregating from a portion of the data set, it cannot extrapolate outside of the given data ([1]). Thus, all conclusions from a random forest will be within the upper and lower bounds of the data. A review of the dataset prior to applying the random forest classifier is needed to see if it is appropriate to use.

This dataset has a lot of information that can be drawn from it, one of them is that higher incidence of making above $50,000 is associated with being self-employed as opposed to working in government or in private industry. We saw that private industry had 21.87% of workers making above $50,000, while government had 30.82% making above $50,000 and self employed had 36.81% making above $50,000. These percentages were found by totaling the number of working in each grouping making more than $50,000 and dividing it by the total entries in the dataset. One of the limitations of these findings are that

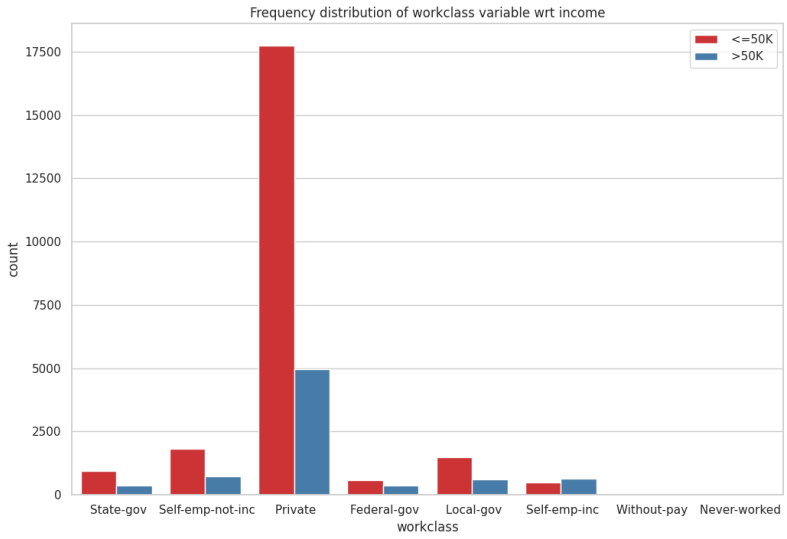

Figure 2: Work class with respect to income

the data set had a significantly larger portion of samples from private industry and relatively low sample from both government and self-employed, this could have skewed the results. A further survey or access to a different data set with similar data would be useful in supporting or refuting these results.

Bibliography

[1] Leo Breiman. In: *Machine Learning* 45.1 (2001), pp. 5–32. DOI: 10.1023/a:1010933404324.

[2] David R. Brillinger. "John W. Tukey: His life and professional contributions". In: *The Annals of Statistics* 30.6 (2002). DOI: 10.1214/aos/1043351246.

[3] Lior Rokach and Oded Maimon. "Decision trees". In: *Data Mining and Knowledge Discovery Handbook* (2005), pp. 165–192. DOI: 10.1007/0-387-25465-x_9.

[4] UW Data Science Team. *A modern history of data science*. July 2017. URL: https://datasciencedegree.wisconsin.edu/blog/history-of-data-science/.

[5] Prashant Banerjee. *Random forest classifier + feature importance*. Dec. 2019. URL: https://www.kaggle.com/code/prashant111/random-forest-classifier-feature-importance.