Machine Learning Engineer Nanodegree

Capstone Project

Trung Kien Nguyen

January 25, 2018

I. Definition

Project Overview

After the tax reform recently, articles and analysis about tax are everywhere. Working as a tax analyst, I conduct data analysis over financial data every day for tax purposes, but I still find it is a pain to fill out my own individual tax return every year. You might think we can use Turbo tax or other tax preparation software to get them done, but were you ever curious about the amount you paid compared to others. Did you pay more than others or less? How's the tax payment distribution amount US like? There are also people who have income other than salary need to figure out how much estimated tax they should pay by the end of the year to avoid fines and penalties.

US has a very complicated tax system that not everyone has time to fully understand. Errors might happen when people type in or write down the wrong amount, and no one wants to be audited or get penalties from IRS.

There is no benchmark for people to estimate their tax, other than their previous year return. What should we do if this is the first year return? What should we do if our income structure changes completely? How to minimize the potential errors? IRS has the database of the tax information in US for the past several years. If we have a model that can predict roughly about how much tax we owe each year, we will be able to understand if we need to pay estimate tax or if there might be some potential errors.

Problem Statement

As discussed above, it is hard for each individual to understand how much tax they need to pay each year. People are vulnerable if they do not know if they need to pay an estimate tax or if they have paid the right amount. How to avoid these situations? The potential solution is to build up a model to estimate how much tax they should pay. In this way, there will be a benchmark for people to compare to so that they know if there are abnormal amount caused by errors or other factors.

Metrics

This is a regression problem, which means the normal accurate score cannot be a good measure. I want to measure how close my prediction is compared to the real value. There are many good measures, including ‘mean absolute error’. I chose ‘explaned_variance_score’ . It computes the explained variance regression score, which measures the proportion of the variance the dependent variable that is predictable from the independent

variable. The higher the score, the accurate the model will be. The maximum score is 1. To maximum the score, the algorithm will try to explain as much information as possible, and the final model will be more accurate. Here is he formula of this measure:

$$\texttt{explained\_variance}(y, \hat{y}) = 1 - \frac{Var\{y - \hat{y}\}}{Var\{y\}}$$

II. Analysis

Data Exploration

The dataset can be found in Kaggle or IRS. It is saved separately as csv files for analysis. The detail information about column name and explanation can be found in “field_definitions.csv” file. Detail information can also be found in IRS. The original dataset is not consistent each year, so I will only use the following columns for this analysis. I have manipulate them to make it easier to understand. The number in the original dataset is the total amount per zip-code.

- state - 2 letter state abbreviation
- agi_class
- num_of_returns - number of returns
- num_of_exemptions - number total exemptions
- num_of_dependents - number of total dependents
- num_of_itemized - total number of returns with itemized deduction

- agi - AGI
- total_salary - total amount of salaries & wages
- taxable_interest - total amount of taxable interest
- ordinary_dividend - total amount of ordinary dividend
- net_capital_gl - net amount of capital gain/loss
- total_tax - Total income tax amount
- prep - Number of returns using a Paid Preparer

AGI_Stub information shows below:

- 1 = $1 under $25,000
- 2 = $25,000 under $50,000
- 3 = $50,000 under $75,000
- 4 = $75,000 under $100,000
- 5 = $100,000 under $200,000
- 6 = $200,000 or more

	year	state	agi_class	num_of_returns	num_of_exemptions	num_of_dependents	num_of_itemized	agi
1605612	2013	VA	3	30.0	30.0	70.0	50.0	1735.0
793383	2008	IN	5	71.0	44.0	204.0	NaN	35.0
62385	2005	IL	4	55.0	140.0	41.0	3368.0	2692.0
265467	2006	CO	1	15.0	14.0	0.0	47.0	NaN
1174680	2011	KY	5	51.0	42.0	146.0	52.0	10491.0

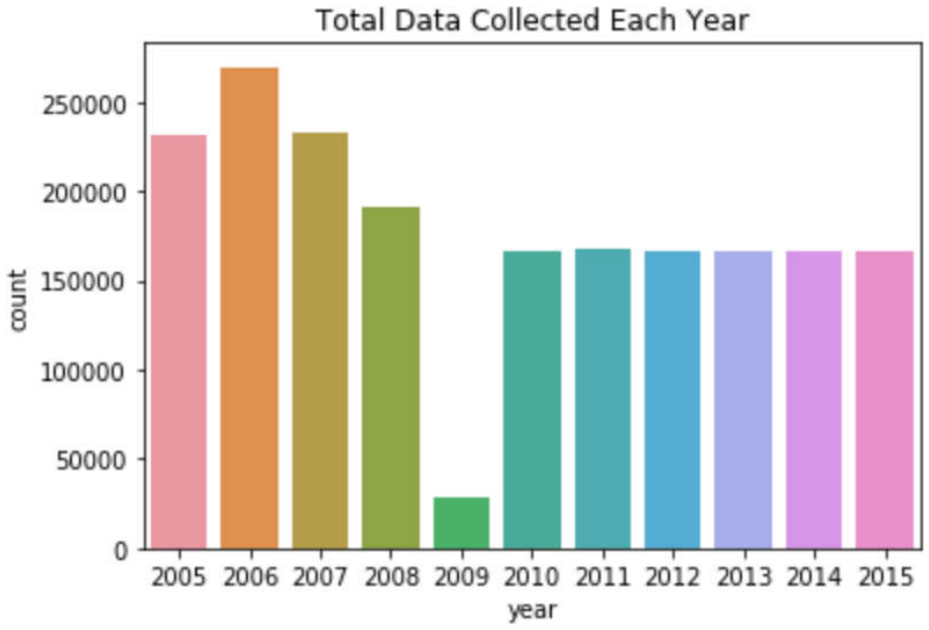
sample data

After concatenate the 10 year tax data, there are 1,953,802 rows in total with 16 total variables. Here are some descriptive statistics of the features:

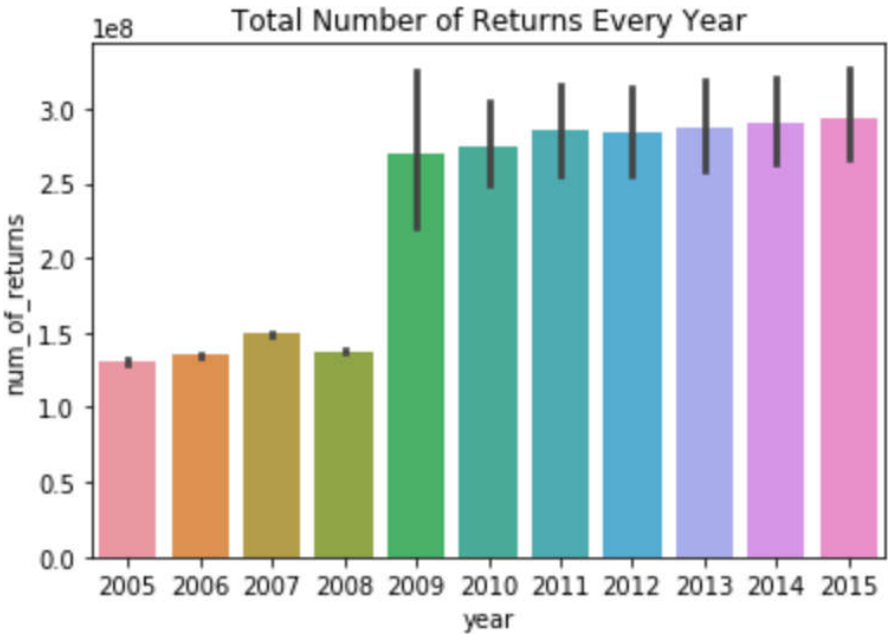
Stats	AGI	Total Salary	Total Tax
mean	4.414340e+06	7.192329e+06	2.601555e+05
std	4.605642e+07	4.617545e+07	6.510224e+06
min	–5.442273e+06	–8.545497e+06	–2.309389e+06
25%	1.444000e+03	4.790000e+02	3.000000e+01
50%	1.047600e+04	9.094500e+03	3.270000e+02
75%	7.353475e+04	1.436490e+05	3.312250e+03
max	1.516317e+10	1.067744e+10	3.138205e+09

We can see the data is widely spread, which will be explored more later through visualization.

Exploratory Visualization



Total Data Collections 2005–2015

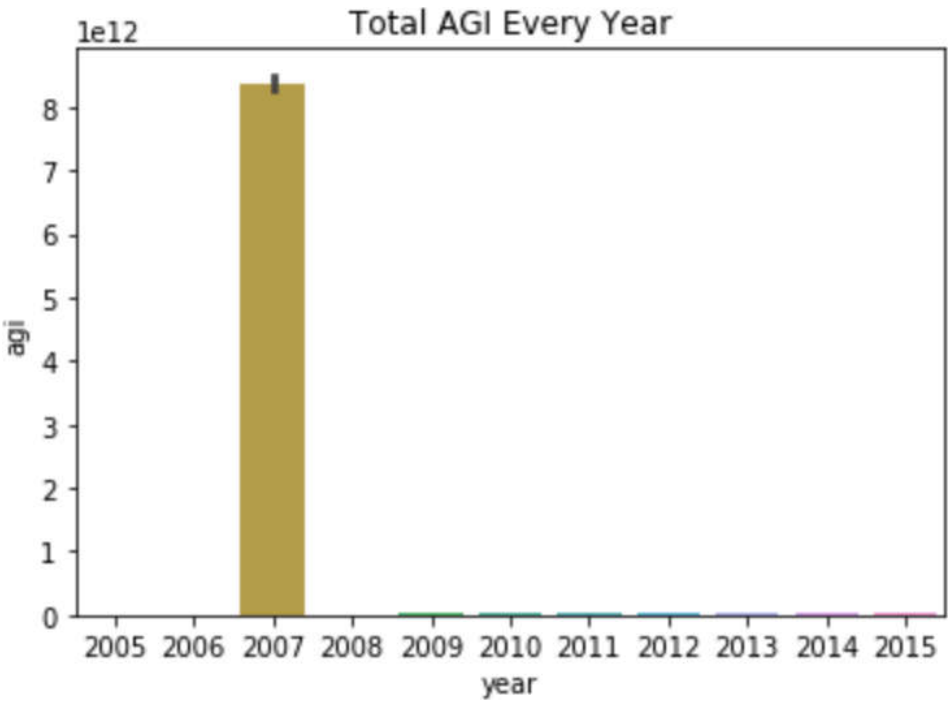


Total Number of Returns 2005–2015

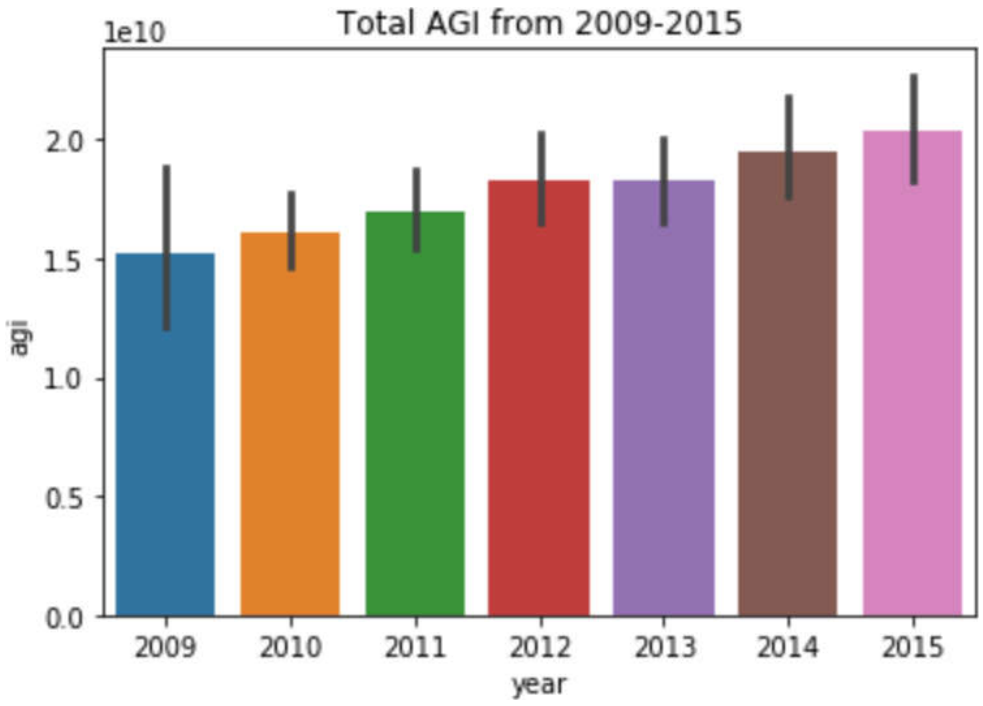
Although the total amount of returns seems right, the data collected 2009 is much less than the other years. After reading through the documents, I did

not find any reasons why it is significantly less than the other years. Since there is no way to fix it, the plot with average number will not be appropriate. 2009 will have higher average value since it has less data amount in total. The future plots will focus on total amount.

Put the 2009 issue aside, the total number of returns jumped almost twice from 2008 to 2009. Since there is no documents in IRS explained the possible reasons, I would assume this is because of the financial crisis. More people are filing returns to get tax refund or figuring out their situations because of the financial crisis. After 2009, the total amount increased slightly each year.

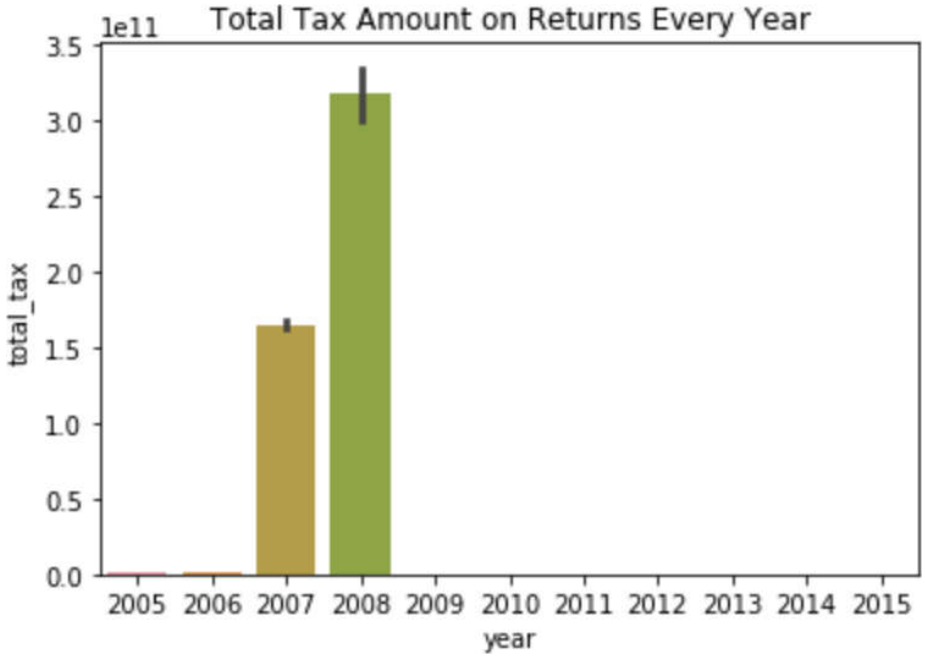


Total AGI 2005–2015



Total AGI 2009–2015

AGI, stands adjust gross income, is the base amount to calculate the tax amount for each return. It is the total income of a return with some expenses. It is supposed to be similar each year for the past 10 years. Other than the financial crisis in 2008–2011, there was no major events that can significantly influence the AGI of each return in US. However, the graph shows a very high amount in 2007. It is so high that we cannot even see the amount in the other years. This is very unreasonable. After closely looking at the data from 2009 to 2015, the amount is more consistent and more reasonable.

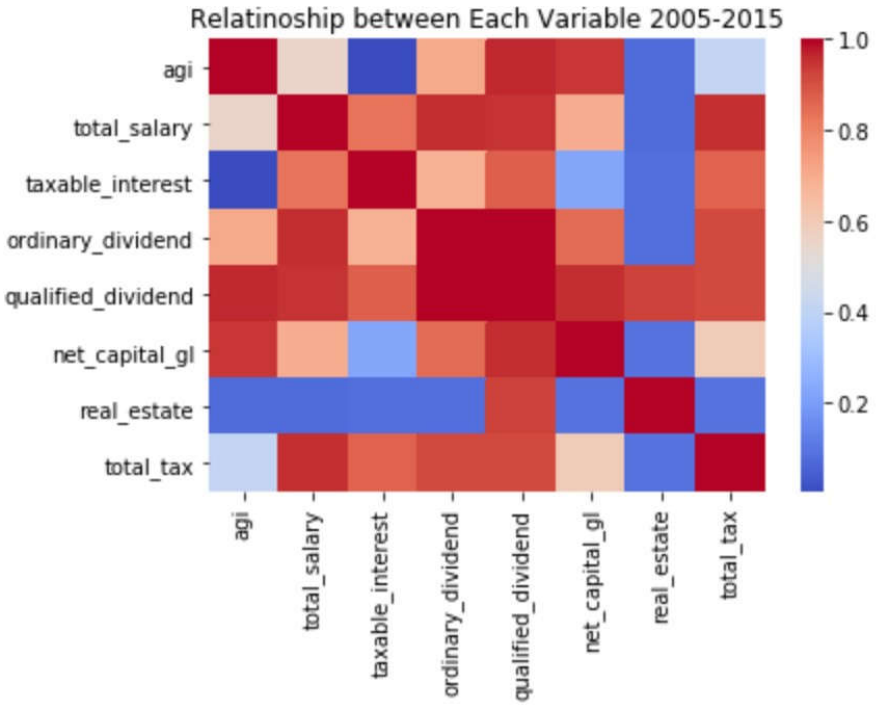


Total Tax 2005–2015

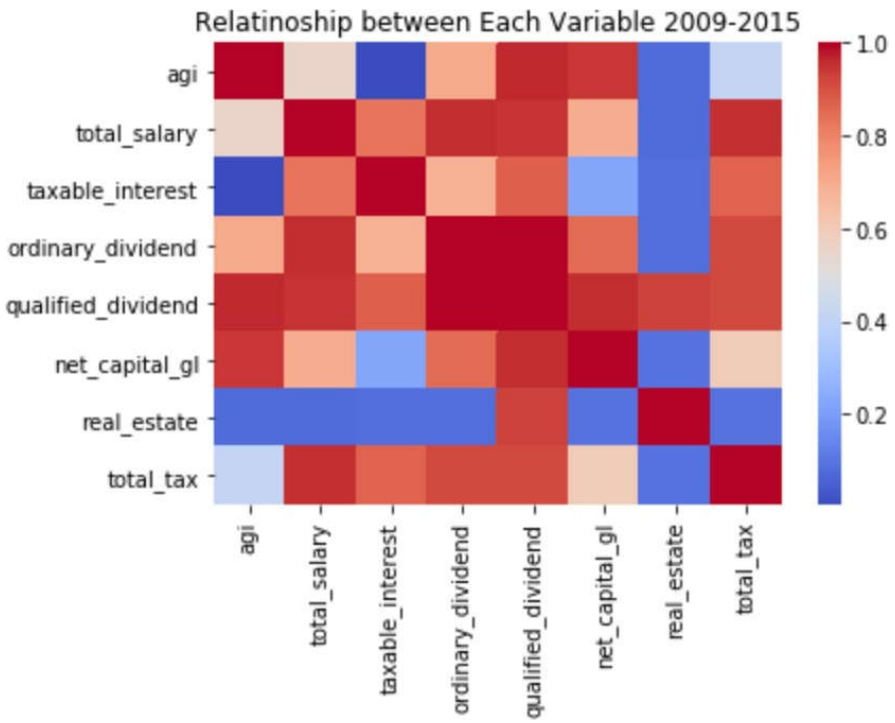
Total Tax 2009–2015

Similar to AGI, 2007 has much higher total tax amount, which is consistent with AGI. However, 2006 also has a very high total tax amount, which is inconsistent with AGI shown above. 2009–2015 amount is more reasonable.

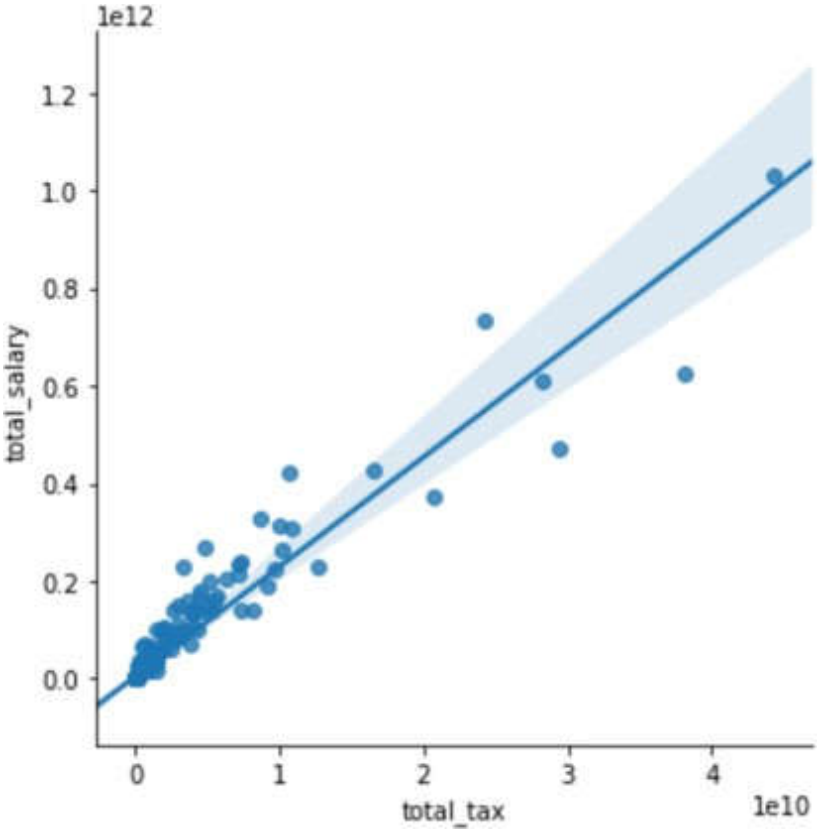
The above analysis shows some issues with original dataset. Data is not well collected in 2009. The total amount of data in 2009 is significantly less than that of the other years. 2005 to 2008 data is not consistent with the other year, they has unreasonable higher amount than the other years. Since the goal of the model is to predict the tax due in current year and in the future, the chaos in the 2005–2008 data could do more harm than good. I will use both the whole dataset and the 2009 (or 2010 since 2009 doesn't have enough data) to 2015 data to train the model separately to see which one is better.



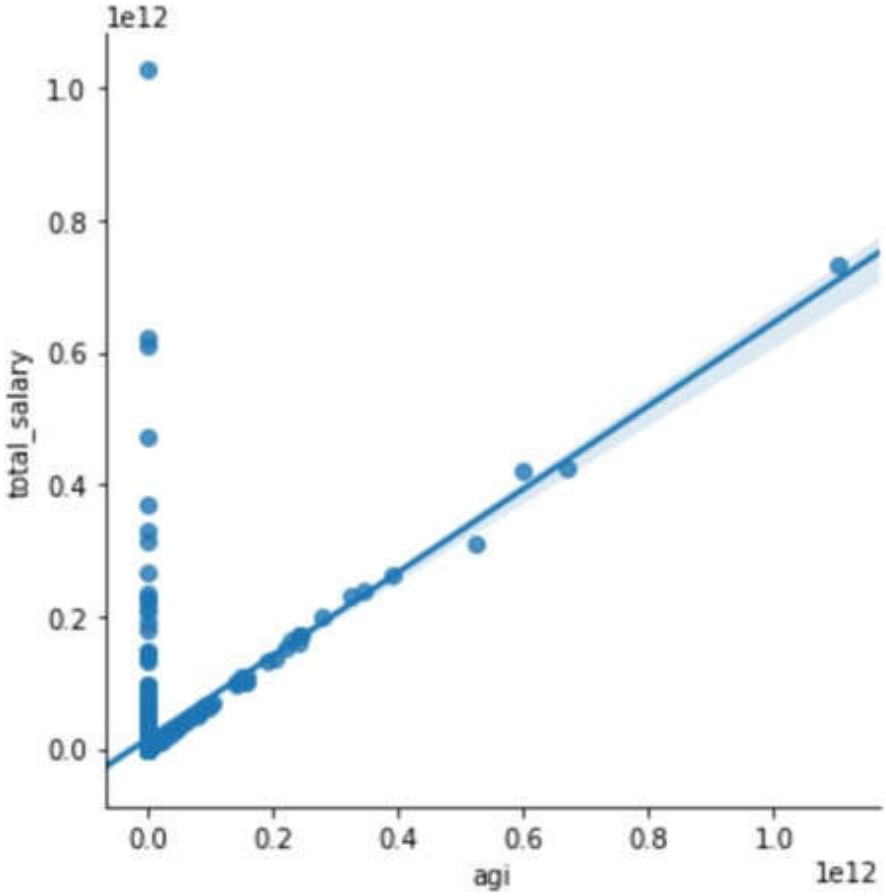
Correlation between Each Variable 2005–2015



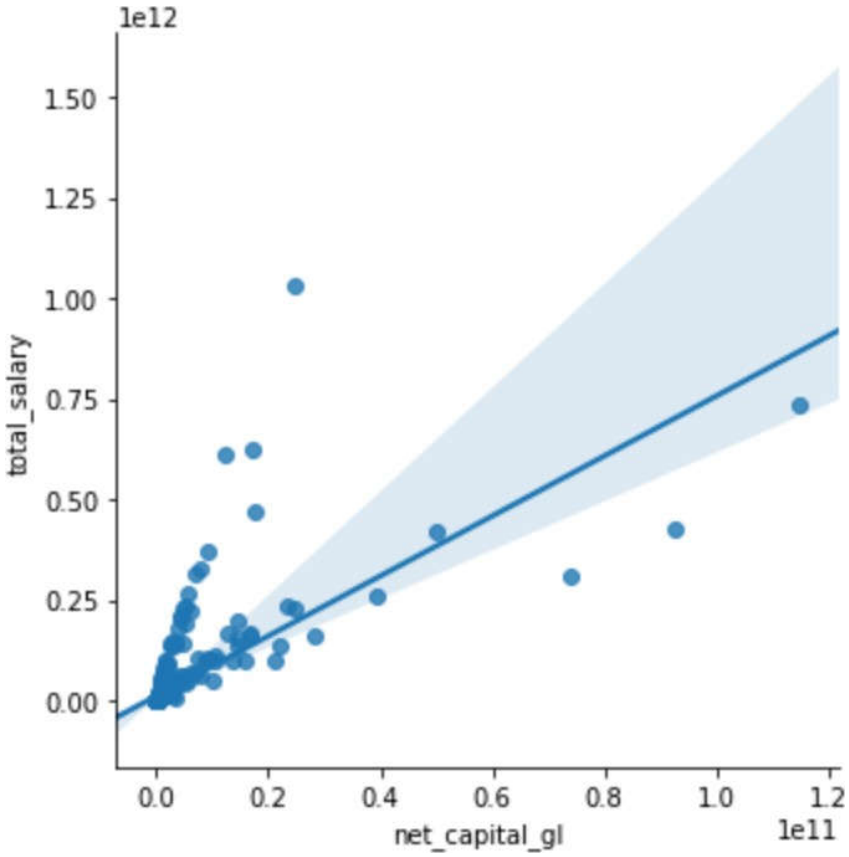
Correlation between Each Variable 2009–2015



Total Salary vs. Total Tax



Total Salary vs. AGI



Total Salary vs. Capital Gain

Although we see there are some inconsistency of the data in each year, the pattern is very similar. The total tax has a strong positive relationship with total salary amount. Surprisedly, it doesn't have a strong relationship with AGI amount.

There are certain amount of people have strong positive relationship between their total salary and their AGI, while the other doesn't have any relationship at all. This might because of alimony or tuition.

Total salary and net capital gain&loss doesn't have a strong relationship.

People with higher salary doesn’t have higher investment than others with lower salary.

Algorithms and Techniques

This is a regression problem. Therefore, I will use algorithms can predict continuous amount. The following models are good regression model, and they do not require hours to train:

Linear Regression

Linear regression is a linear approach for modeling the relationship between a scalar target variable and some explanatory variables. It estimated the parameters of a linear functions based on the original data, and tried to minimize the differences between the real value and the predicted value.

Decision Tree Regression Model

Decision tree is a tree-like model that has a lot of ‘if-then’ to predict the final value. If the target variables are continuous values, decision tree can be used to fit a sine curve with addition noisy observation. It learns local linear regression approximating the sine curve. It tries to maximum information gain.

Random Forest

Random forest is an ensemble learning method for classification and regression. It is a combination of many small decision trees. The basis is similar to decision tree, which is to maximum information gain. Random

forest can generate better results since it correct the issues of overfitting of decision tree. It is the combination of many decision trees. Each decision tree will make a vote of what the final result should be. Result with more votes will be returned.

The variable in the dataset can be easily included in these model, which can then predict the final result.

Benchmark

I will train a simple linear regression model without any tuning as a benchmark model. The detailed information about this benchmark model will be discussed more in the metrics section below.

III. Methodology

Data Preprocessing

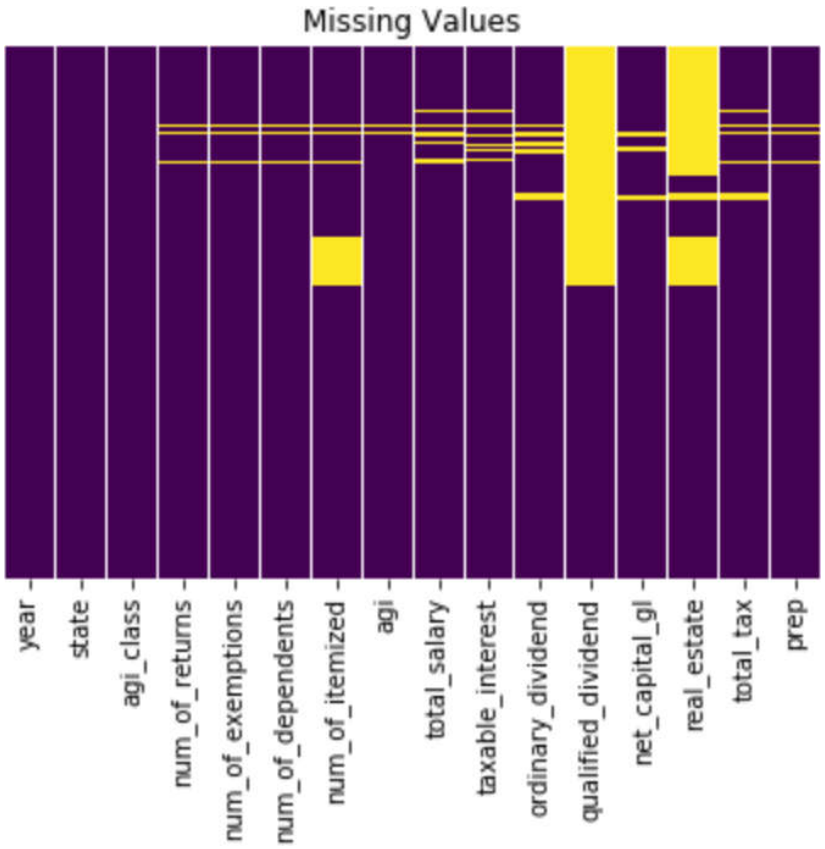
There is nothing too surprised in this step.We need to deal with missing values and outliers as we normally do with other datasets. However, we need to be very careful about outliers. We cannot drop them simply because some data points look “weird”. There are a lot of stories behind the tax returns, we need to retain as much information as possible for the model.

Missing Value & Outliers

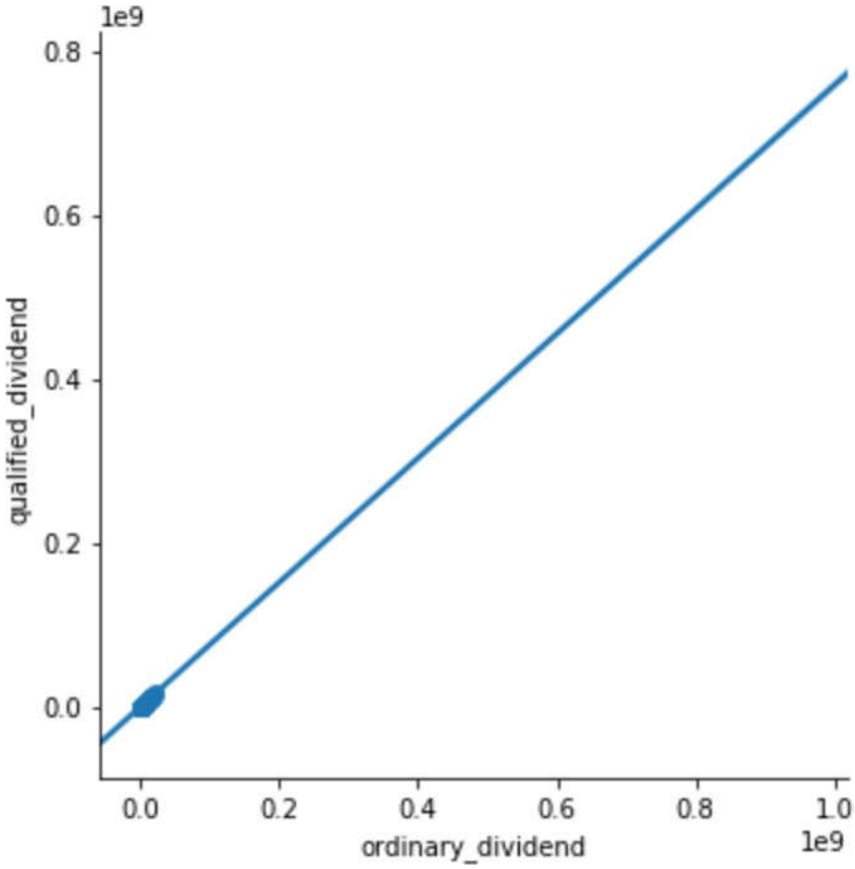
To train the model that can predict the final tax amount, I do not need the all the variables. For example, total number of returns per year is irrelevant for

the model. I will use the following steps to clean the data:

- Deal with missing values and outliers
- Deal with categorical variables
- Scale the data to make them under same range.
- Manually select the variables based on my domain knowledge



Missing Values

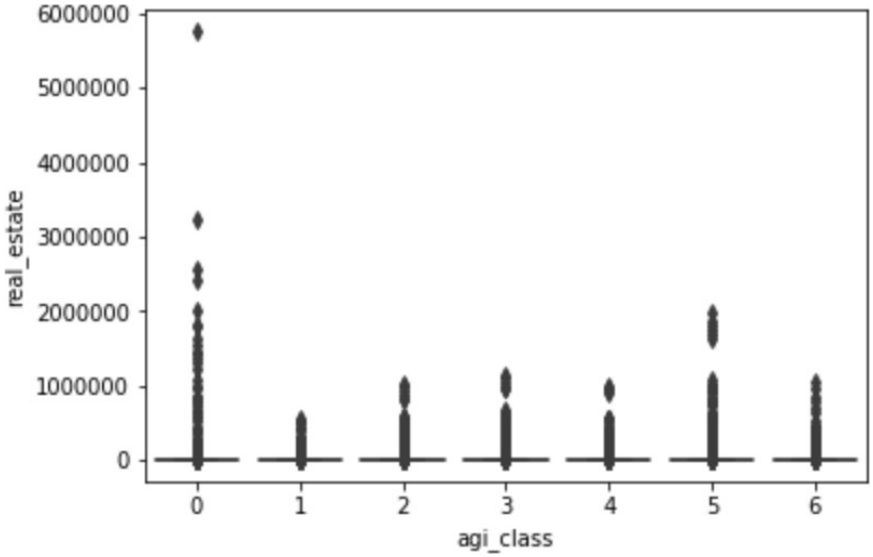


Dividend vs. Qualified Dividend

There are a lot of missing values under qualified_dividend and real_estate. As shown in previous visualization, qualified dividend and ordinary dividend are strongly related with each other. Therefore, to deal with missing values, and to deal with potential collinearity issues, I will drop qualified_dividend variable.

We miss around a third of real estate information and a little of other variables. I will fill in the based on the rest of the values. Based on the plot below, the average real estate tax amount for each agi class is different,

and there are some outliers as well. Therefore, to make it more accurate, I will fill in the missing value based on the agi class using median, which is not influenced by outliers.



AGI Class & Real Estate Tax

Total_tax is the target variable, and only a few of them are missing. I will directly drop the missing values to make it more accurate.

Implementation

This model is different from other machine learning models because we know what we want the user to input. Therefore, we will not use any techniques, such as select K-best features, to select the features. We will manually select the features we want the model to use, so non-tech people can play around with the model to predict their individual tax.

Since the scaling of each variable is different, we also need to use feature scaling.

Feature Selection

I will not use any algorithms for the feature selection in this model. My final goal is to get a model that can predict individual tax amount based on several amount people input. Therefore, I will manually choose the variables I want people to input based on my domain knowledge. I will choose the following variables:

- state
- number of dependents
- total salary
- taxable interest
- ordinary dividend
- net capital gain/loss
- real estate tax
- total tax

Feature Scaling

These variables are very easy to get, so people will not have any issues inputting these number to predict their tax. Since state is a categorical variable, which cannot be used directly by machine learning algorithms, I will use `pd.get_dummies` to change it to numerical variable.

Since the range of each variable is different, for example, number of

dependents will be significantly smaller than total salary, I will scale each variable to make them into a similar range.

Refinement

Linear Regression

I want to use linear regression as a benchmark or a base model. There are not many parameters can be tuned for linear regression model. The final score is 0.75.

Decision Tree

I trained a simple decision tree model without tuning parameters and cross validation. The score is 0.75. Later, I applied cross-validation technique here to make the final model more accurate. I tuned several parameters:

- min_samples_split: 2, 3, 5
- min_samples_leaf: 2, 3, 5

I want to try different combinations to find the best estimator. The final score is 0.84.

Random Forest

I applied cross-validation technique here to make the final model more accurate. I tuned several parameters:

- min_samples_split: 2, 3, 5
- min_samples_leaf: 2, 3, 5

The final score is 0.89.

KNN

KNN acted weird in this dataset. It runs for 5 hours and cannot return a single model. I will try to run on the cloud in the future to figure out what the issues might be.

IV. Results

Model Evaluation and Validation

The final model is the random forest model, with 2 minimum sample split and 2 minimum sample leafs. The final explained variance score is 0.89. I choose it because it has the highest score. Compared with the un-tuned decision tree, which has score of 0.75, and the tuned decision tree with score of 0.84, random forest achieves the best result.

The model has been trained through cross-validation. Each time, the training and testing sets are different. To verify the stability of my model, I used ‘cross_val_score’ to cross validate the final score. The average score is 0.86 with standard deviation of 0.008. I think the model is very stable.

Justification

I have one benchmark for this model - the simple linear regression I trained at the beginning of my project. Since the simple linear regression model is not as accurate as the random forest one based on the metrics score, the

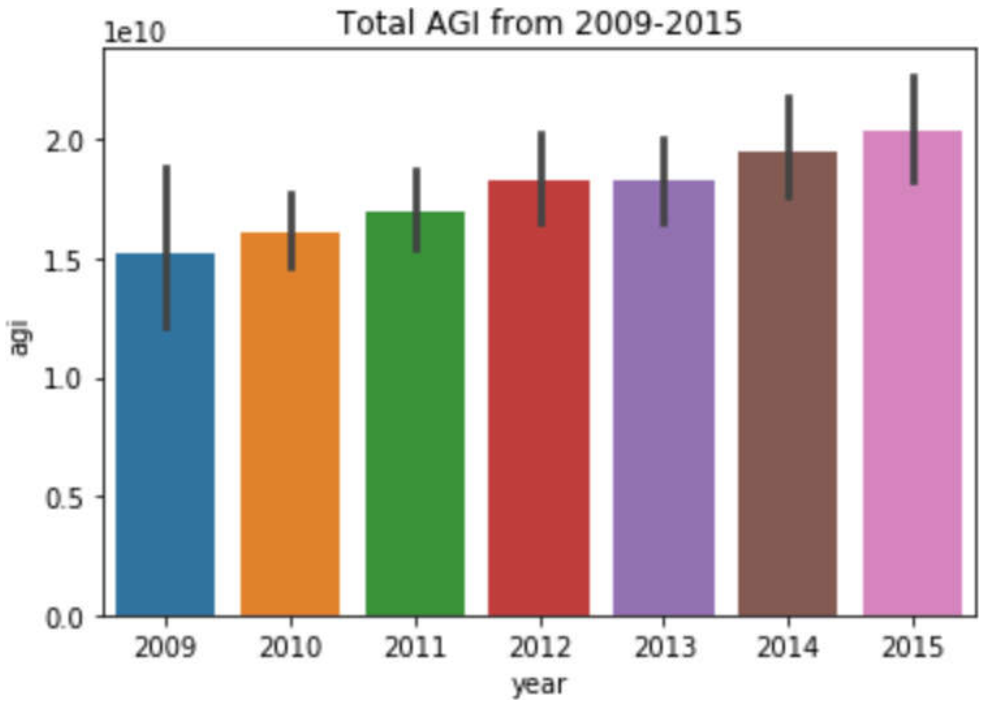
tuned random forest model is better.

V. Conclusion

Free-Form Visualization

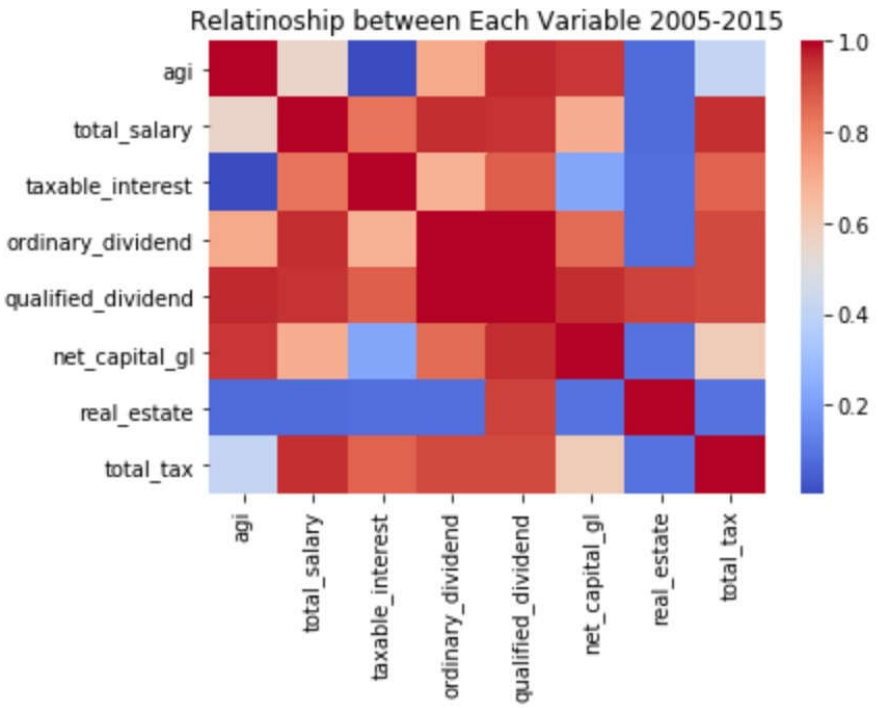
I have done several visualizations for this dataset, including univariate and bivariate variables. I noticed how messy the original dataset is, and why data scientists spend most of their time cleaning data, instead of actually tuning the models. The following two visualizations make an impression on me.

AGI amount shows the total income minus some deductible expenses. It shows that the financial crisis does not hurt the total income of US in general. The AGI amount increased every year since 2009.



Total AGI amount 2009–2015

As a tax analyst, I understand how important AGI is for calculating the final tax amount. It is the starting point to finish the rest of the tax returns. However, based on the correlation graph below, AGI does not have a strong relationship with the final tax amount. Although this is counterintuitive, it shows to me that knowledge and experience may not be accurate.



Correlation between each factors 2005–2015

Reflection

I went thought the following steps:

- Roughly cleaned up the dataset
- Did some simple dataset descriptions and exploration
- Explored the dataset and visualized the data
- Preprocessed data by dealing with outliers and missing values
- Implemented the algorithms

I think the most difficult part is data cleaning. I did two parts of data cleaning. I cleaned the data a little bit when I read the data. I did not expect reading the dataset at the beginning took me long since the dataset is very

messy and the columns name are not consistent. I did some initial cleaning and selecting from the original dataset. Later, through the exploration, I noticed some incorrect information. Although AGI class only has 6 choices, the original dataset has class as 14000. There are a lot of missing values as well. I spent 70% - 80% of my time cleaning the data.

The most interesting part is the data exploration through visualization. On the one hand, it showed potential errors in the dataset, such as the incorrect AGI class. On the other hand, it shows a lot of unknown information, such as the correlation between each variables.

Training model takes me very long since the dataset is huge. I should have used AWS to train it, which could save me some time.

Improvement

The model takes very long to train on my machine. For example, Random Forest trained for around 2 hours, and KNN cannot get any results back after training for 5 hours. The current accurate rate is below 90%. I want to apply more complicated algorithms for the prediction. For example, I think deep learning models can get a better results. To improve the model, I will uses AWS to decrease the training time so that I can try more ways to tune different parameters and get a more accurate model.