# Case Study: Analyze Stack Overflow Developer Survey Data

## Data File

Dataset contains 2019 survey results from Stack Overflow developers.

survey\_results\_public.csv

survey\_results\_schema.csv

Data location: <https://insights.stackoverflow.com/survey>

## Overview

This analysis will look at the 2019 survey results from Stack Overflow. Throughout this analysis we will be looking for what aspects developers have. We’ll try to answer some questions such as, which gender developer tend to be, do they code for fun, is there a linear trend in years of development vs. pay. Do they participate in open source projects, and how their profession is related to programming (e.g., full time, student, hobby, etc.).

## Step-by-Step Instructions

### Part 1 Includes step 1 – 13

### Part 2 Includes step 14 -16

### Part 3 Includes step 17 -20

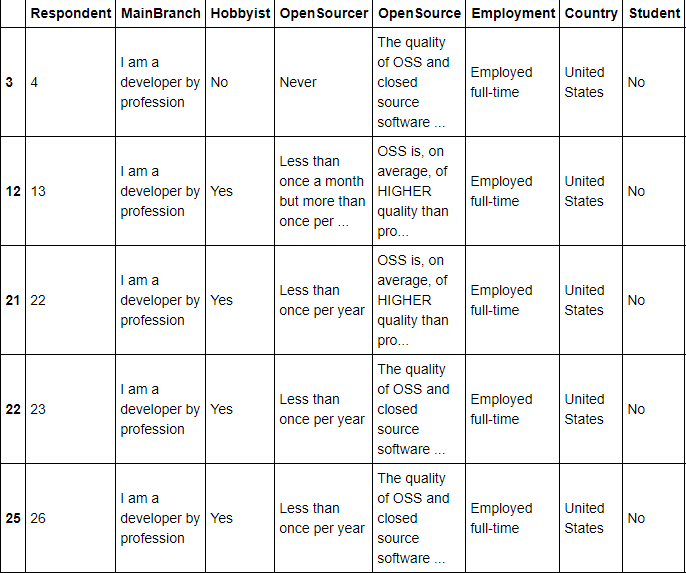
1. Load the data from *survey\_results\_public.csv* and *survey\_results\_schema.csv* files into a DataFrame.
   1. *survey\_results\_schema.csv* contains the header information, so we’ll want to include this to check headers later.
2. Display the dimensions of the dataset.
   1. There are a lot of columns that are not going to be directly applicable to this analysis, we’ll remove them later in Part 2.

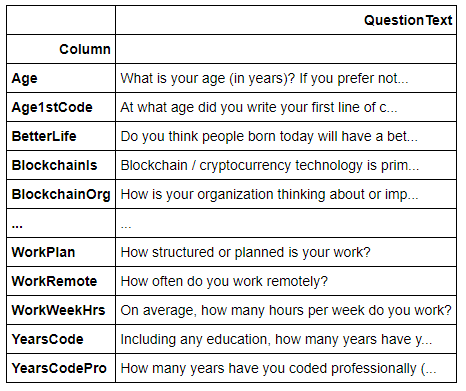
The dimension of the data is: (88883, 85)

1. Filter data to only include United States
   1. I’m limiting the survey results to be more applicable to me. Through this analysis, I would like to see what attributes others developers have and more directly compare them to myself.
2. Display the dimensions of the file.
   1. From the two filtering criteria we set in step 3, we were able to reduce the dataset to ~ 21,000 rows – this should help us more directly focus in features we care about.

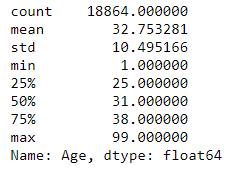
The dimension of the data is: (20949, 85)

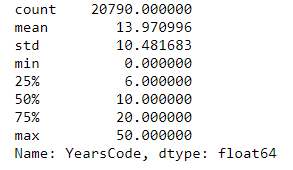
1. Display the first few rows of the new dataset and header schema.

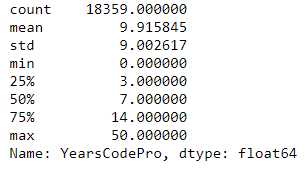




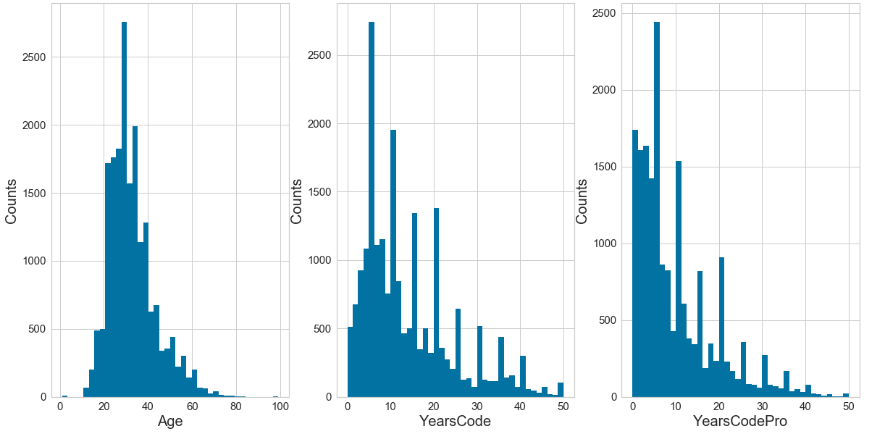
1. Replace values for select numerical features (YearsCode & YearsCodePro)
   1. Some options allow for more than 50 years or less than 1 years as an answer. For plotting, I’m lumping the less than 1 year as 0, and more than 50 with 50.
2. Convert select numerical features (YearsCode & YearsCodePro) to numerics
3. Display the summary of the data for features of interest (Age, YearsCode, YearsCodePro).
   1. Age – we can see non-realistic ages like 1 and 99, in Part 2 we’ll narrow the scope of the age.
   2. This also gives an idea of the experience level of the developers.







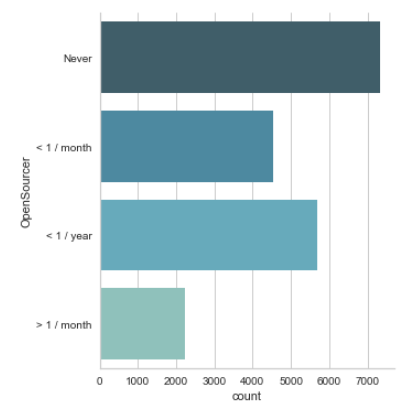
1. Plot histogram for features (Age, YearsCode & YearsCodePro).
   1. We can see by limiting the age range we will still be cover the majority of participants.

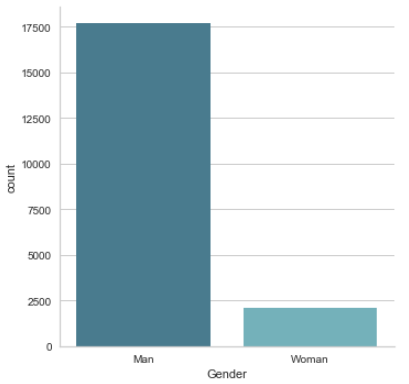


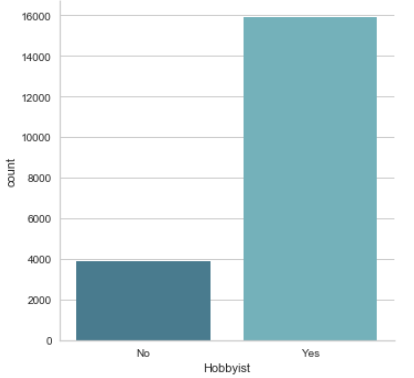
1. Replace values for OpenSourcer, the original survey has long text, to display better on the plot this should be converted to shorthand. Also replace text for Employment for similar reason.
2. Filter genders to Man & Woman; other genders are relatively small.

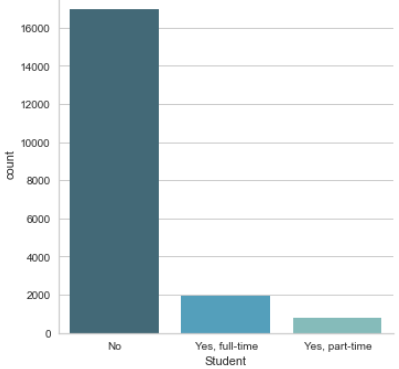
The dimension of the data is: (19792, 85)

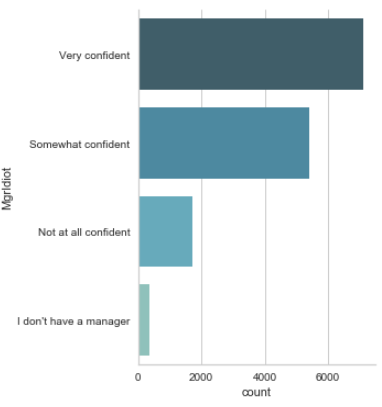
1. Make bar chart for other features of interest.
   1. This data enables us to better understand the distribution of various features. We can latter use these as filters and compare to salary. Additionally, we can use them for machine learning to see which are most important when considering salary.
   2. Open Source providers – the majority of developers rarely is at contribute.
   3. Gender – Overwhelming male dominated, this was expected since STEM fields follow the same trend. It will definitely be interesting to compare salary’s here when holding older features constant.
   4. Hobbyist – the majority of developers program for fun. It is surprising so few contribute regularly to open source programs when the majority like to program for fun.
   5. Students – only a small sample are part-time or full-time students.
   6. MgrIdiot- This feature is for the confidence developers have in their managers. It looks like most have somewhat or are very confident in their managers.
   7. Employment status – most respondents are full-time employed. Unemployed respondents will be removed since they skew the data on salary. I’m curious to see how independent compares to full-time salarys.

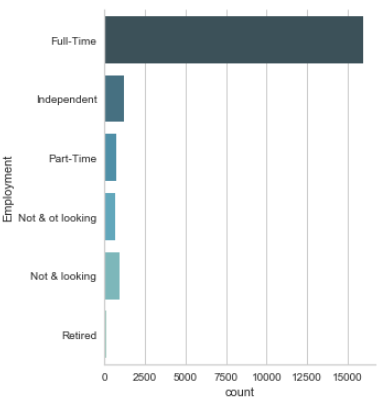




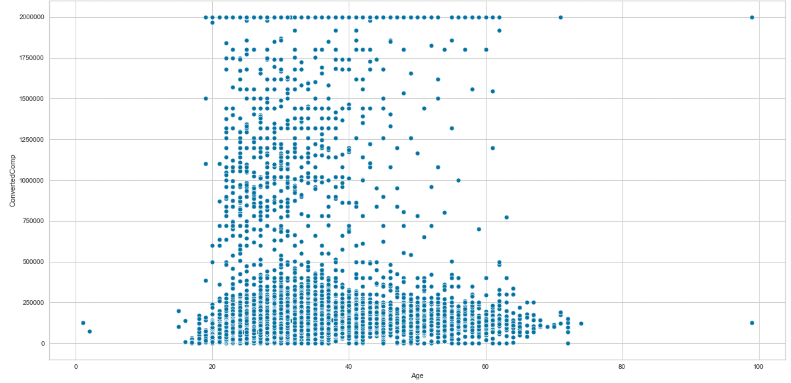


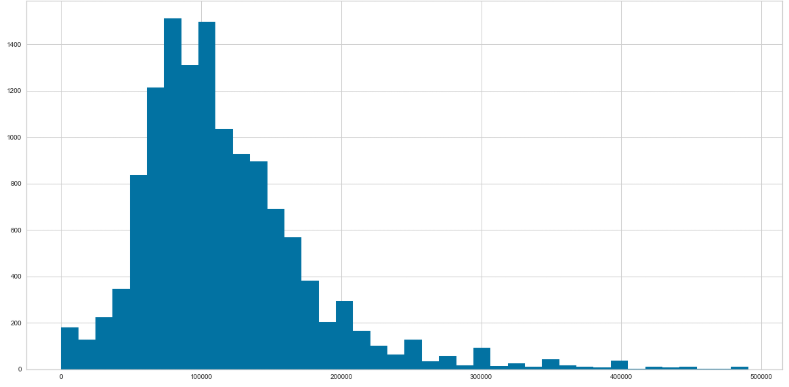


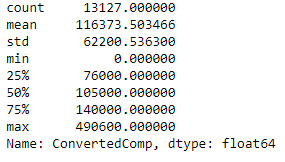




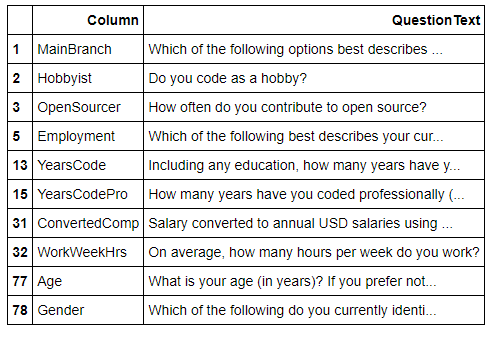
1. Scatter plot /Histogram of ConvertedComp (Salary) vs. Age
   1. We see there is a wide range of salary values (yearly).
   2. It doesn’t seem like a typical developer as much as the survey results show. We’ll only consider salary values of under 500K in step 14.







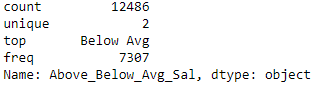
1. Filter additional data based on what we’ve seen from the graph analysis above.
   1. Step 3 removed all countries except the United States.
   2. Step 11 removed any gender that was not Man or Woman.
   3. Remove anyone that is not currently employed, retired folks or people not currently working won’t provide sufficient data relative to salary.
   4. Restrict age range to 18 – 65. This is the core working demongraphic.
   5. Restrict ConvertedComp (Salary) to under 500K.
2. Remove features not of interest.
   1. Since there are 85 features, it is easier to show the features that were included
   2. Features included relate to describe the type of people included and/or related to salary.
   3. May revisit this step for Part 3 tuning.



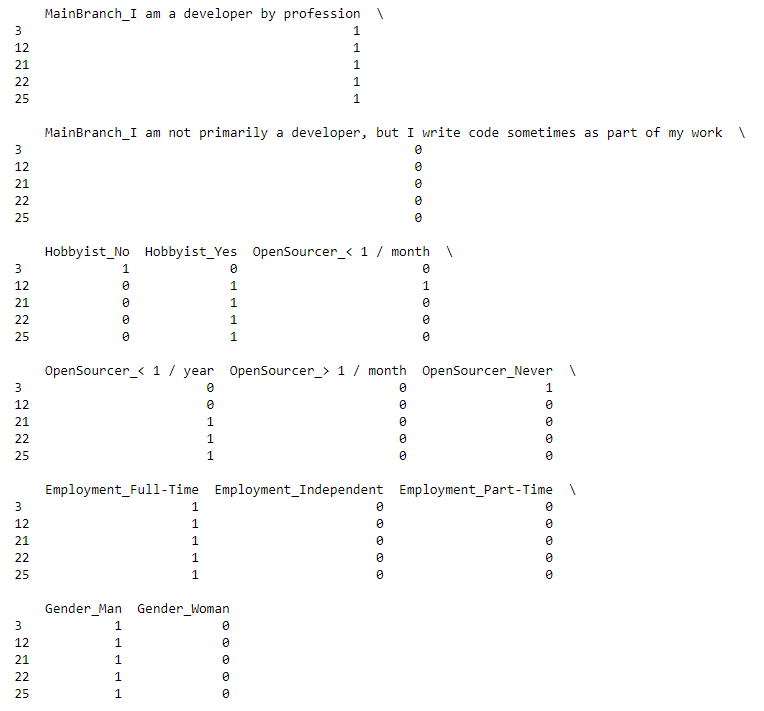
1. Drop na values
   1. Now that all filters and feature reduction have been applied we can remove any missing values.

The dimension of the data is: (12486, 11)

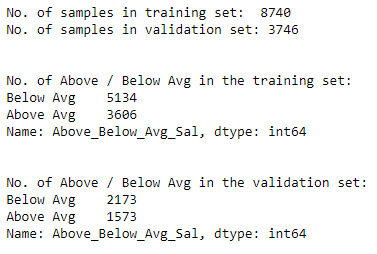
1. Create target column – above or below average Salary
   1. Using the ConvertedComp (yearly salary) we can set a new column to determine if the value is above or below the average – this will be the target for later steps



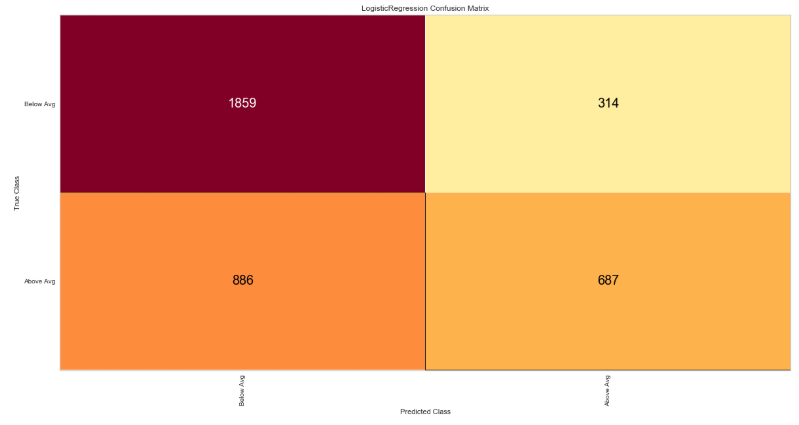
1. Convert categorical data to numbers (MainBranch, Hobbyist, OpenSourcer, Employment, Gender)

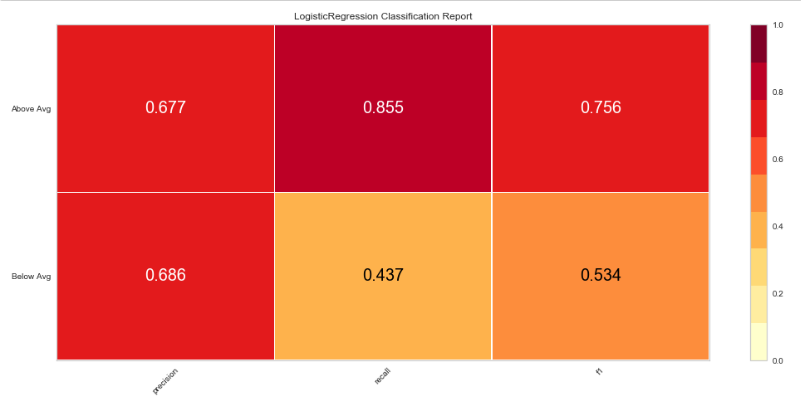


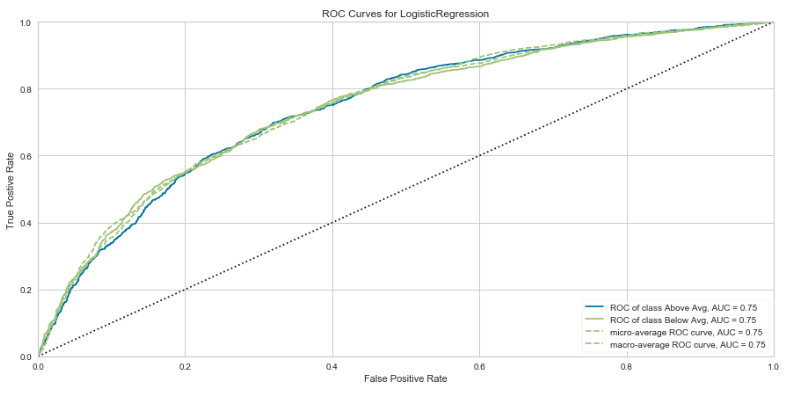
1. Training – split data into training and testing datasets
   1. As part of the tuning process I should try to even the distribution of below/above avg salary to help with the accuracy of the model.



1. Evaluation – The goal is to predict if the salary is above/below the average salary based on the features selected using logistic regression
   1. Metrics:
      1. Confusion Matrix - ~68%
      2. Precision, Recall & F1 score – fairly low-medium level of scores.
      3. ROC curve – All values above the dotted line but the model needs some improvements.







## Future Considerations

The confusion matrix show an accuracy of ~68%. This seems pretty low, with some parameter tuning or some refinement with the filtering considerations, this can probably be improved. Additionally, target was broken into above or below average salary. This is probably too coarse given the spread of salaries reported. Maybe additional classifications could be considered. With all the filtering and parameters refined, additional scatter/line plots should be added to demonstrate trends and expectation of salary.