***Lambton College at Queen’s College of Business, Technology and Public Safety***

**Program:** Big Data Analytics (DSMM)

**Project Name:** Stack Overflow Developer

**Course:** Advanced Python AI and ML Tools 01 (AML 2203)

**Professor:** Sagara Samarawickrama

**Group Members:**

Jennylynne Dominguez (C0929140)

Joyce Ann Murillo (C0927648)

Chaw Su Su Thinn (C0916347)

Hazel Portia Elaine Santos (C0915982)

**Project Overview:**

The objective of this project is to design and implement a predictive model using the Stack Overflow Developer dataset that contains structured and numerical survey responses from developers. Through this project, we aim to apply important concepts in machine learning, data preprocessing, visualization, and basic distributed computing using Scikit learn and PySpark.

**Description of dataset and project goal**

[jennylyne.dominguez@gmail.com](mailto:jennylyne.dominguez@gmail.com) #### for update

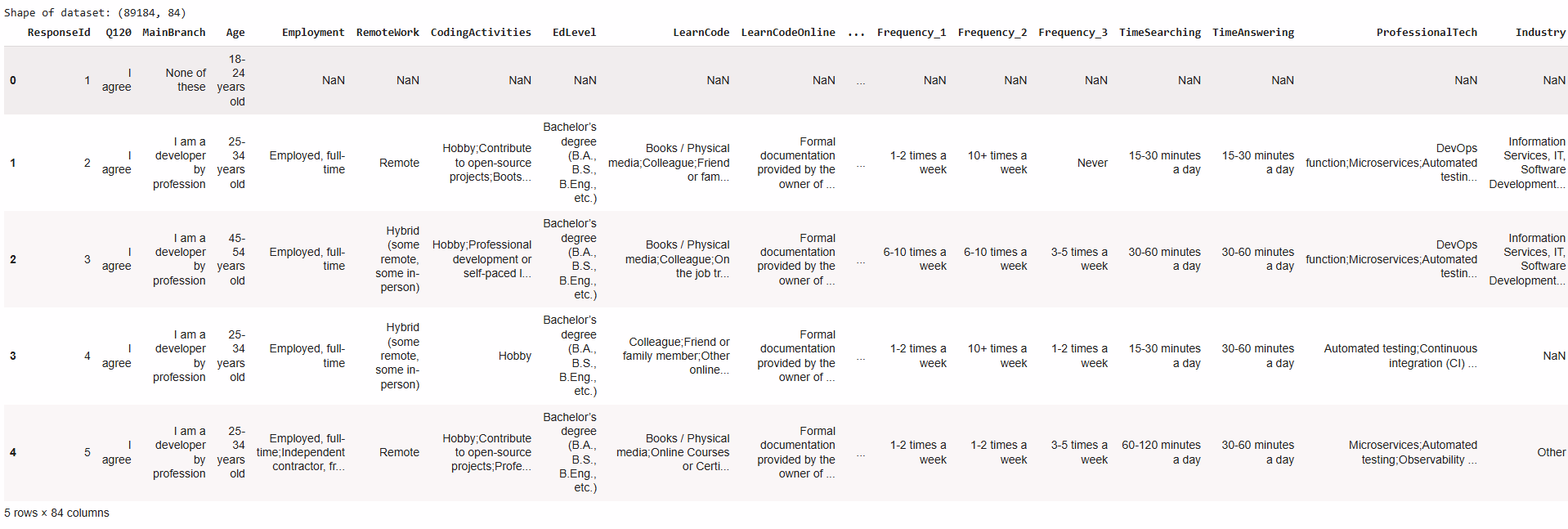
**Project Components & Instructions:**

1. **Data Selection & Setup**
   1. Loading of dataset in Google Collab using Pandas

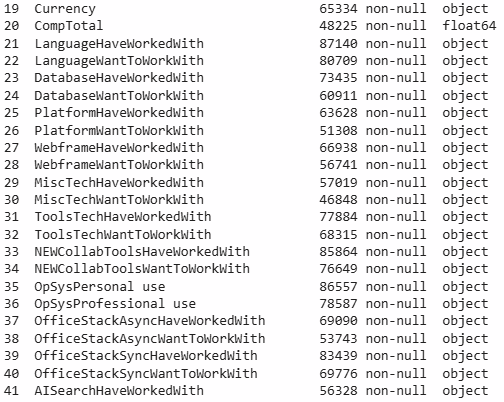
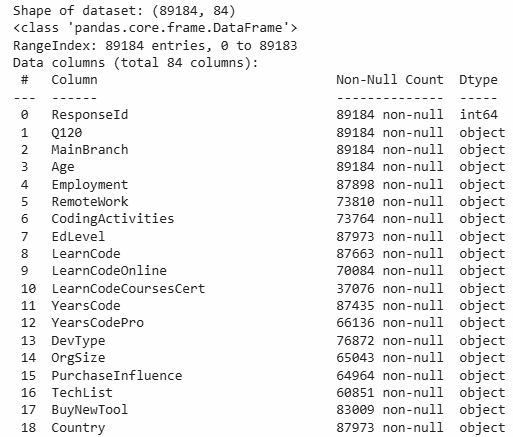




* 1. Understanding the data set, and defining prediction objective (Classification or Regression)

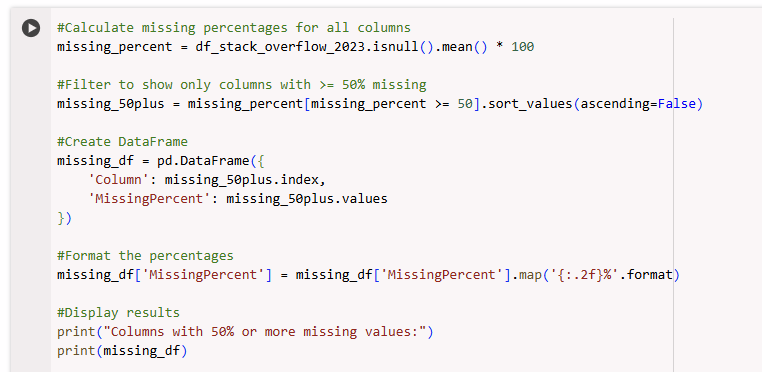


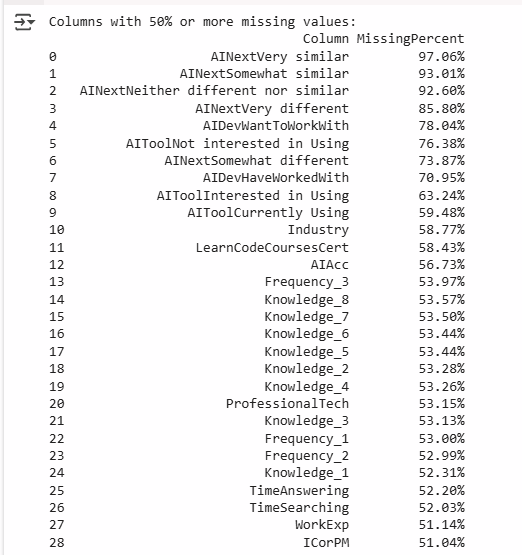
This shows the first 5 rows of the raw data, from survey\_results\_public.csv. It has 89,184 rows with 84 columns. It also show which columns are in numerical column or not.



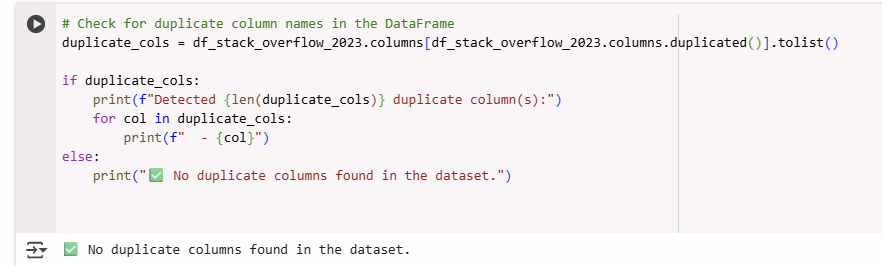
The data shows that majority of the fields are an object type, and some are on float

1. **Data Cleaning**
   1. Identifying and handling of missing, null or duplicate values



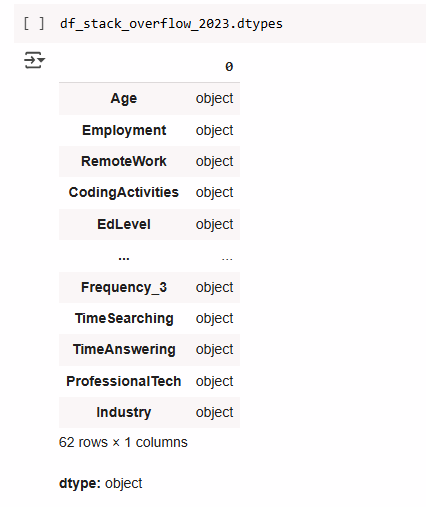


To identify how much data is missing in each column across all rows, we have sorted them by column; the isnull() codes creates a boolean condition to mask missing values (as True) then calculates the mean of missing values per column, after that it converts as a percentage as shown on the screenshot above. The code above outputs values in descending order, sorting it from the highest missing values, then only showing columns that have 50% or more missing values.

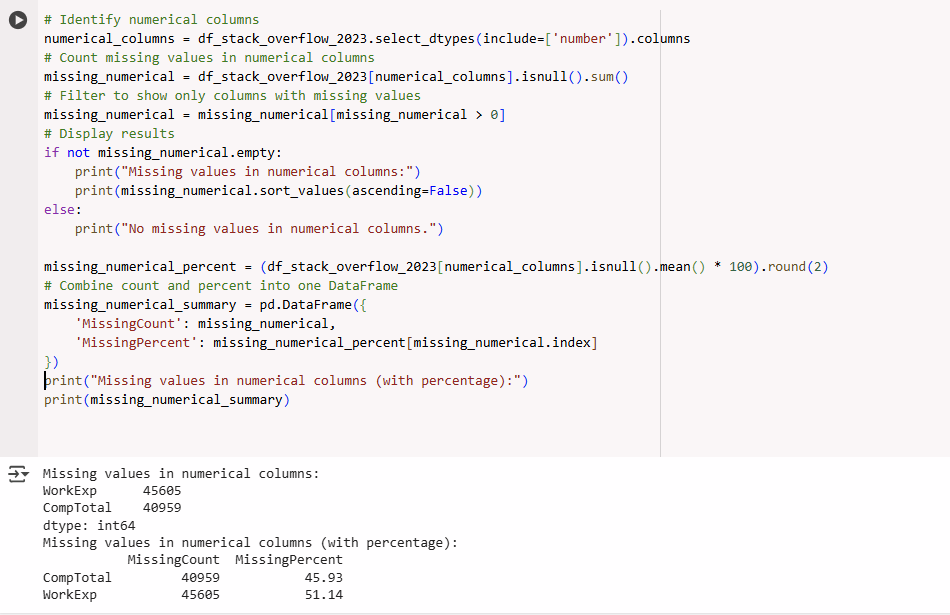


There were no duplicate columns in the dataset.

* 1. Check data types and convert as needed

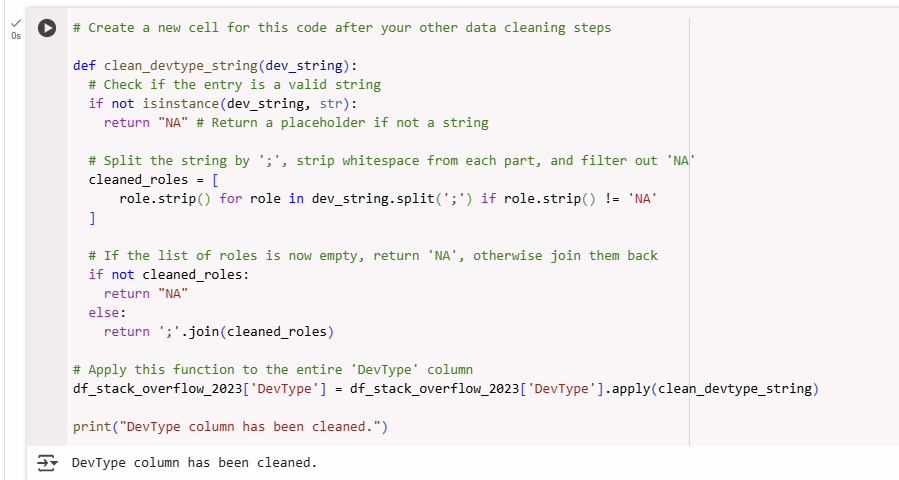


This shows that 62 rows are composed of non numeric values such as: combination of text data, or possibly mixed data types or numbers stored as strings.



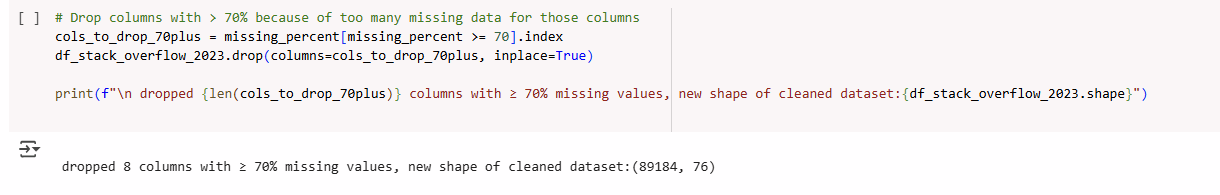
For the numerical columns, the code counts number of missing data, the percentage of missing values per numerical column, filters columns where the missing count is greater than 0, or were 100% complete. Then it calculates the percentage of missing values.

The output shows that out of all the numerical columns, CompTotal has 45.93% in missing values, and 51.14% for WorkExp

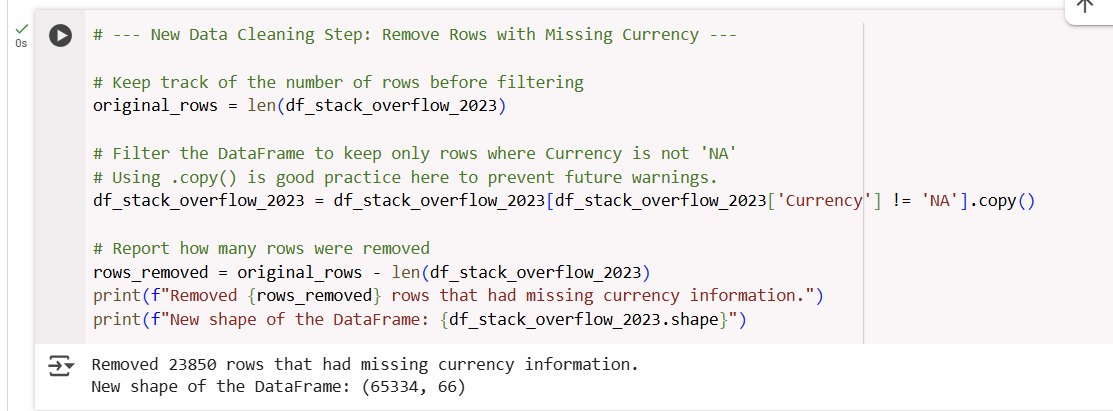


By understanding this distribution, it will assist in identifying which columns will need imputation or which needs to be excluded from modeling.

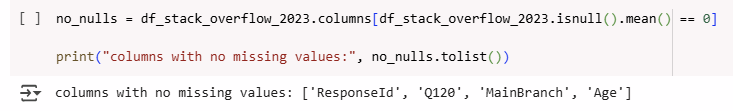
1. **Feature Selection & Engineering**
   1. Choosing of features to include in the model



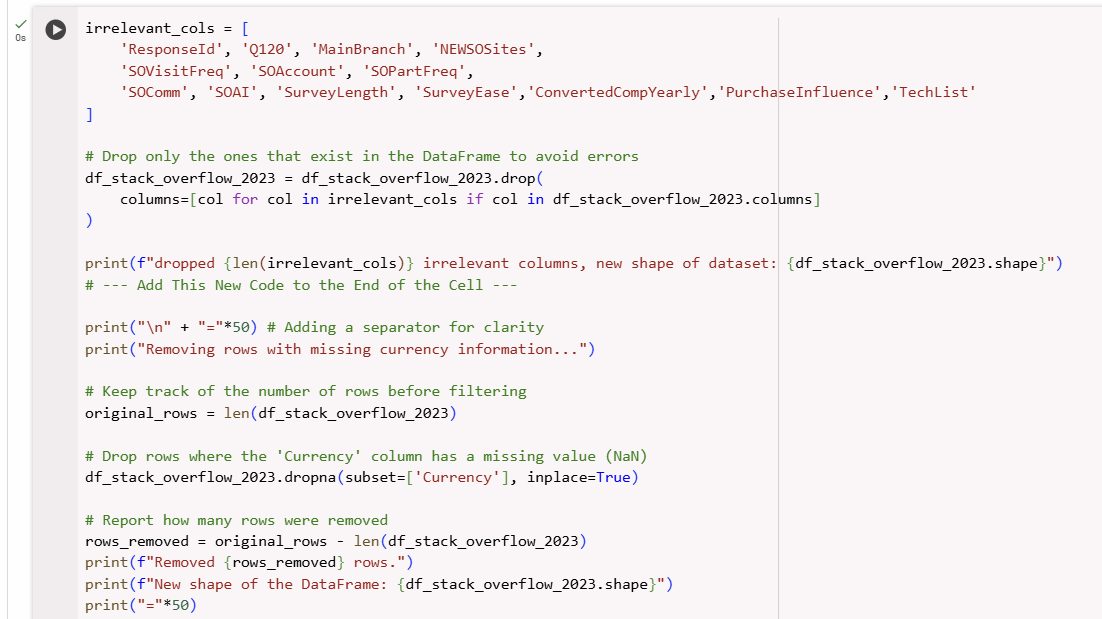
First step in feature engineering is understanding which columns we needed to drop. The criteria above drops columns that have more than 70% of missing values. The output shows that from 84 columns, it is now down to 76.

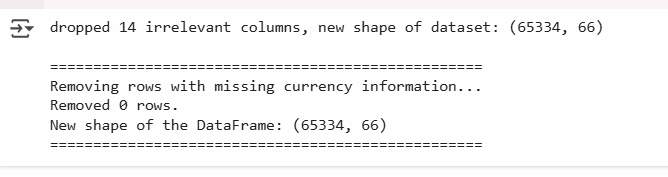


We have also dropped rows that had missing currency information. Because our analysis is based on predicting salaries in USD and if there are no currency information in the dataset, then we won’t know how to convert it in USD or if its already in USD.



The next step is identifying which columns are clean, or with no missing values, and the output shows that from all the columns, Response ID, Q120, MainBranch and Age were complete.





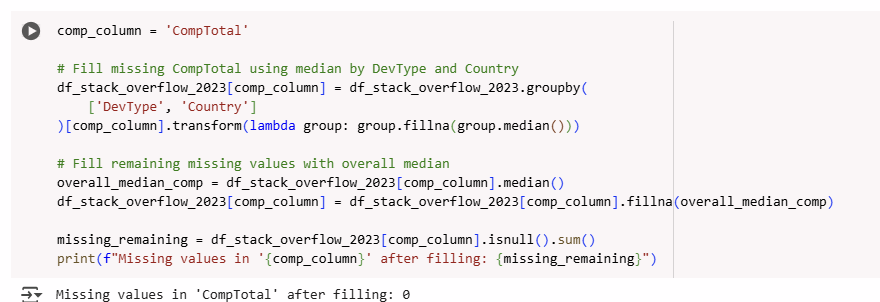
We have now further dropped columns that will not be used in modelling.

Why did we drop these columns?

* ResponseID - This is a unique identifier, if not dropped may result to data leakage
* MainBranch- Majority of the answers here states “I am a developer by profession” this will result to low variance
* ConvertedCompYearly - has the same values with CompTotal
* Purchaseinfluence - is not relevant for the analysis
* All other columns such as NEWSOSites,SOVisitFreq etc were only specific to Stack overflow community behavior, which is only relevant for Stack overflow usage studies

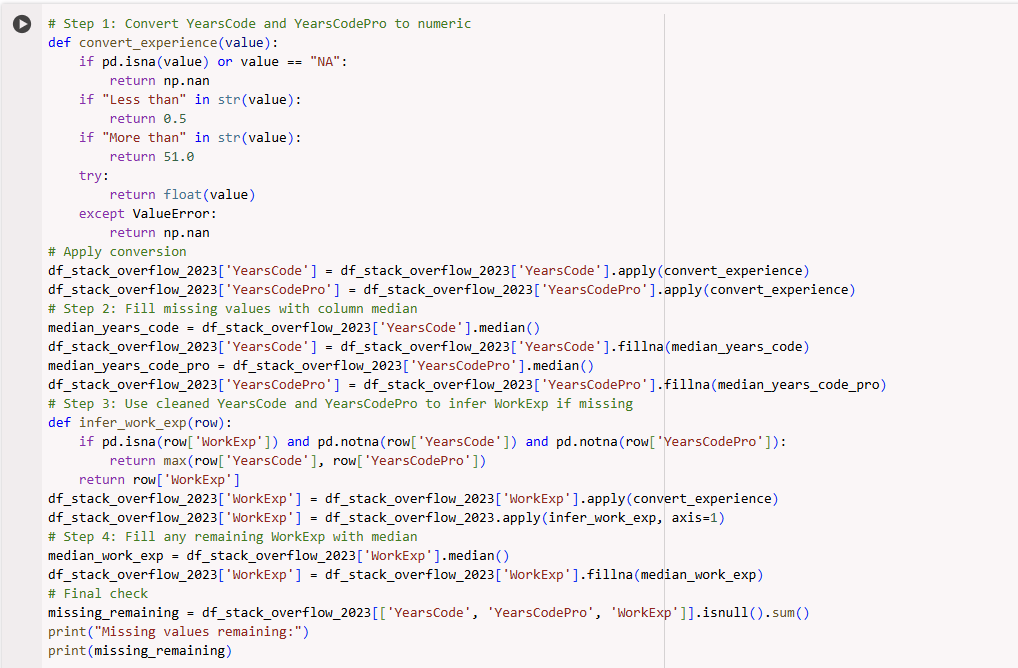
We have also dropped rows that have missing information.

* 1. Applying of transformations, imputations



CompTotal is one of the main features as we will be doing Salary prediction.

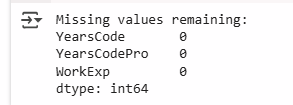
We have chosen to use median here for imputation instead of filling missing values with the mean because the compensation values are highly skewed. For example, some developers earn very high salaries ($1M+ some earns much lower for example <$20k). If we use mean, it may distort the imputation as it is sensitive to outliers. If median is used, it is robust to extreme values. Because this represents the central value in a group which is better used for skewed data.



Another most relevant column which will assist best in the analysis is the work experience of the developer (WorkExp). This is a combination of YearsCode and YearCodePro. This refers to the total years of coding and years of coding professionally. WorkExp column helps estimate the overall duration of employment and is inferred as the maximum of the two columns wherever a value is missing in WorkExp.

Step by Step procedure done here:

1. We called a function def convert\_experience, were we have converted values to numeric format
2. In that function, for Less than 1 year, convert to 0.5, if more than 50 years, convert to 51
3. Otherwise if NA or non numeric then np.nan
4. Then for the two columns, we have filled missing values with the median
5. Then to infer Work Exp with Max values from YearsCode and YearsCode Pro as a smart guess for WorkExp
6. After all the logic, if there are still missing values, this is the only time it fills it with the median

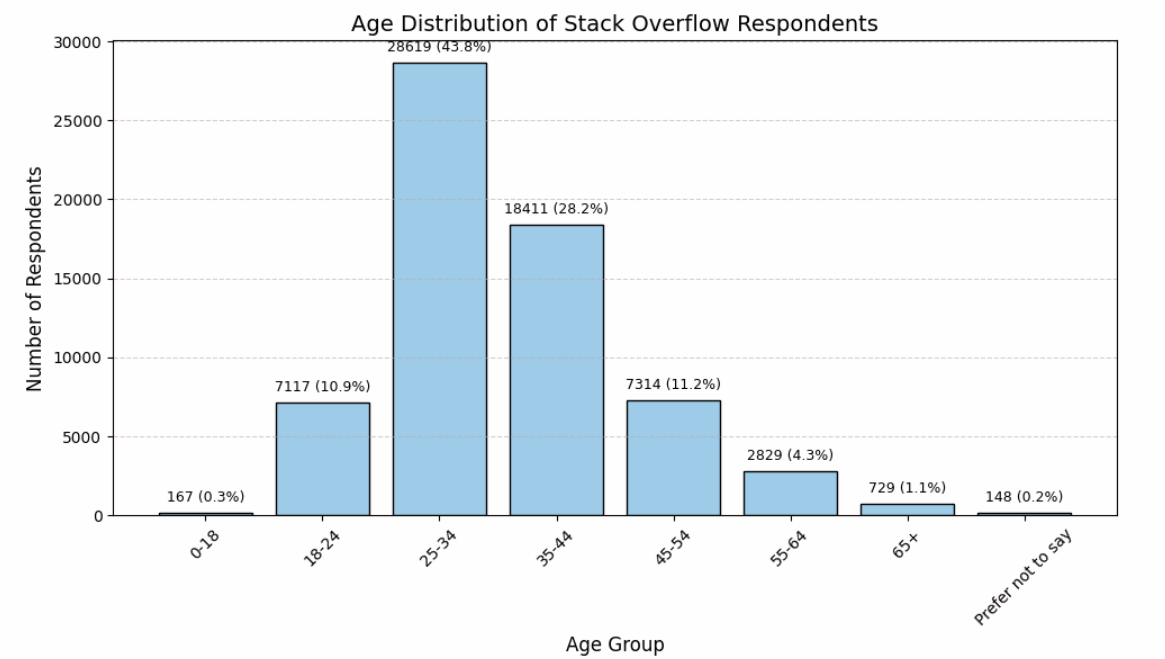


After the function was successfully integrated, there are no more missing values for all three columns

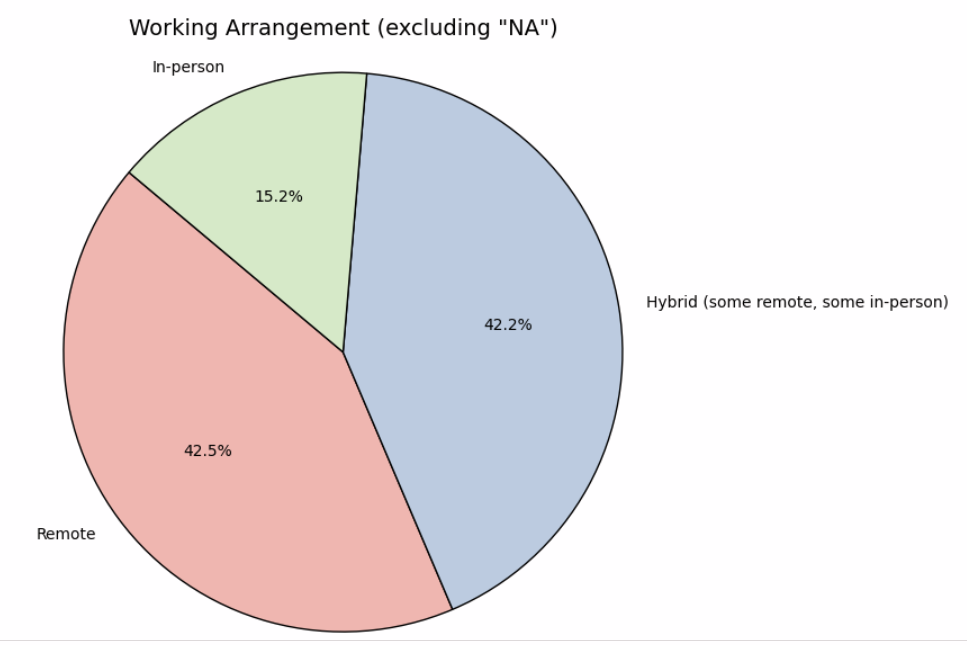


We have then checked if there were other currencies listed

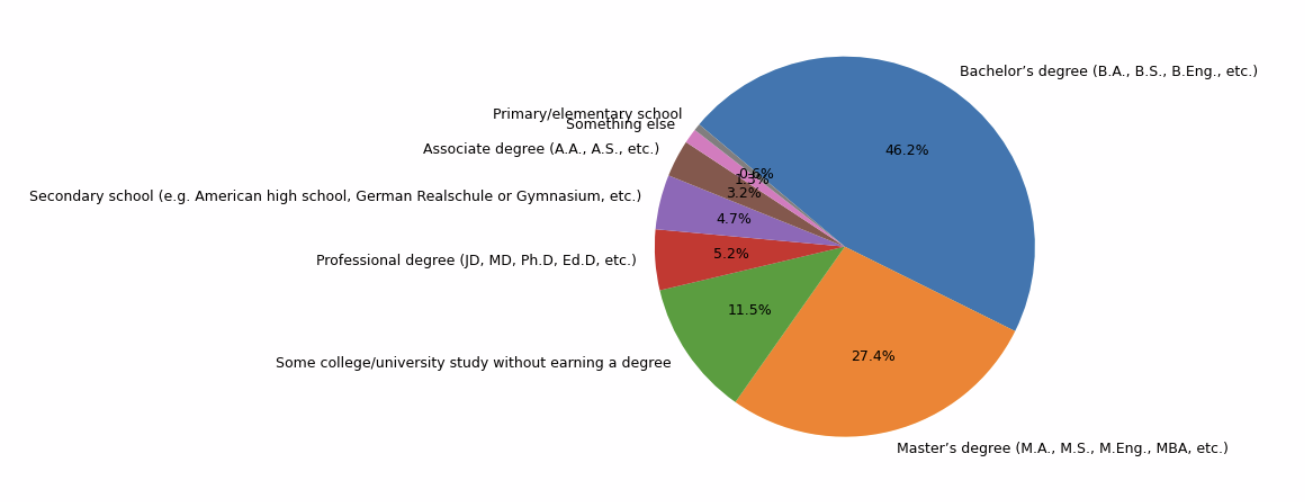
1. **Exploratory Data Analysis (Visualization)**
   1. Visualization of Feature Distributions



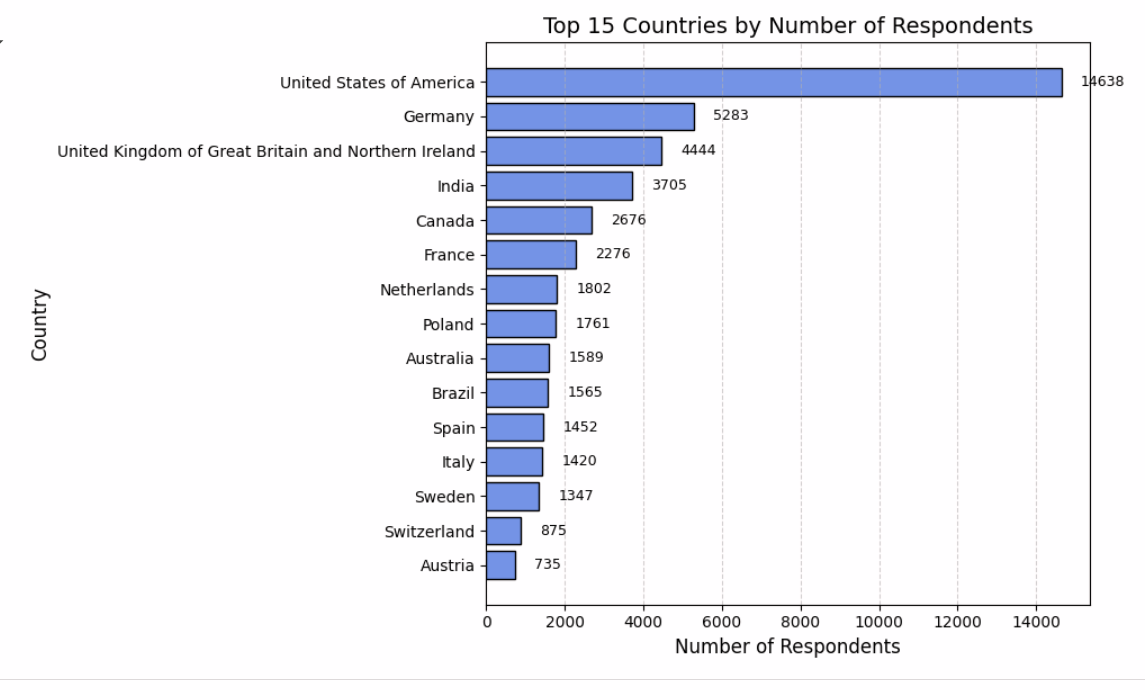
This shows the distribution of ages from the respondents



This chart indicates that majority of the respondents have a Hybrid Remote job arrangement



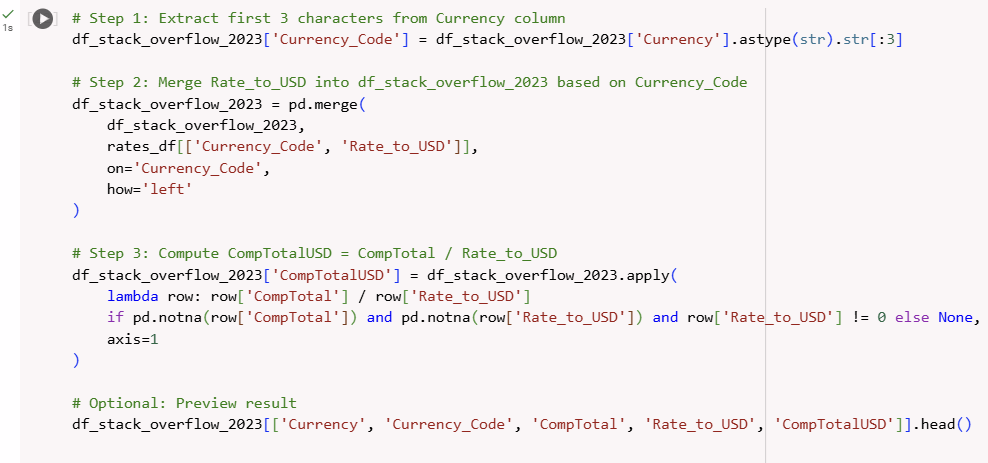
This shows that majority of the respondents have finished a Bachelors Degree in a Computer Science or Mathematical field. The second most are respondents that have completed a masteral degree also around the fields of Math and Sciences



This is a very important chart as it determines the nationality of the respondents. Did they list their CompTotal (our Target variable) in USD or in their own currency? If the respondents have listed their CompTotal under their own currency then there will be transformations done to standardize the Target variable.



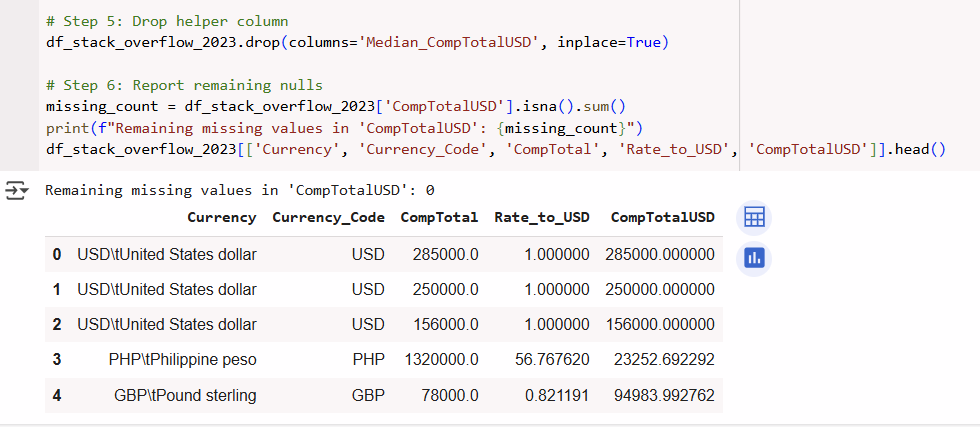
Inorder to understand if the CompTotal is listed under the respondent’s currency, we have then checked the first few rows in the currency column





These set of codes have created a new column named CompTotalUSD which represents the developer’s compensation in USD. This was done to standardized the target variable into one currency.

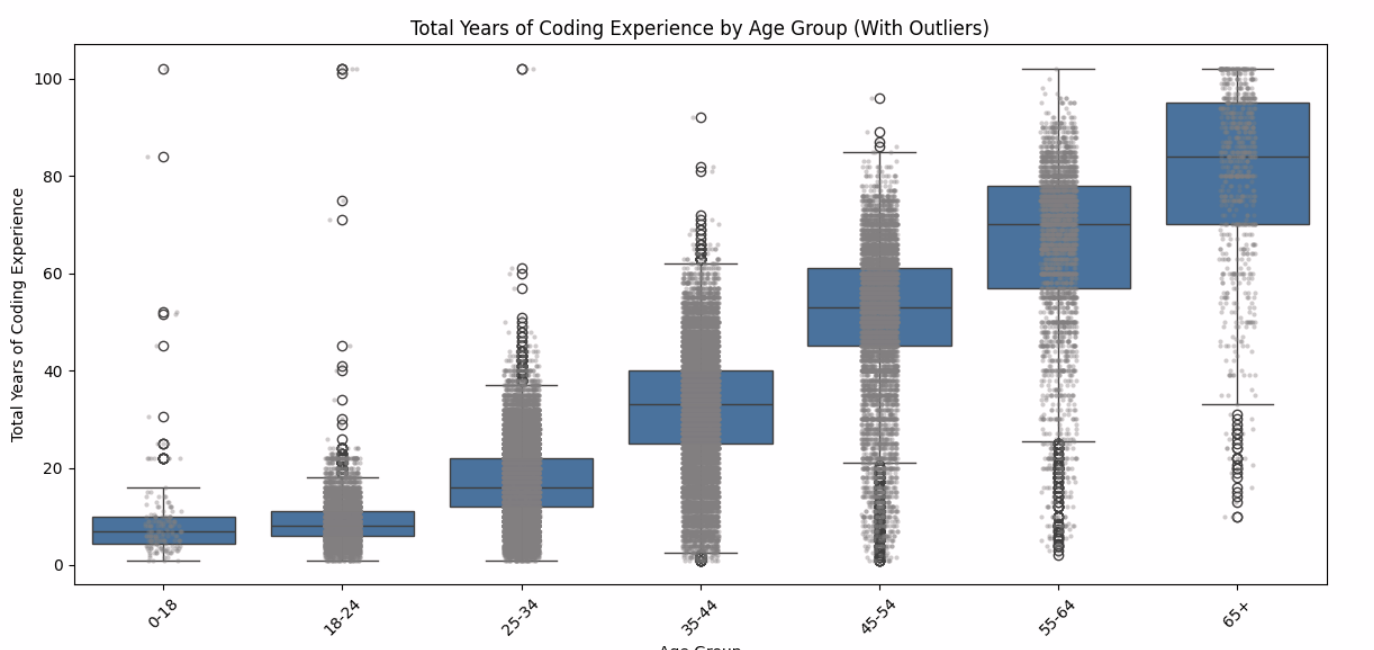




Now that we have created the CompTotalUSD, it will have missing values because some records don’t have exchange rates from currency\_rates.csv. To fill this up, we have created a more intelligent group-specific median instead of a global median or dropping of rows. In this scenario, we have imputed the missing values using the median salary for people in the same employment type and country.

After doing this step, and successfully imputing, there were remaining records with missing values.

* 1. Exploring class imbalances or outliers





* 1. Correlations between variables

1. **Modeling with Scikit-learn**
   1. Splitting of datasets
   2. Model Selection
   3. Why model was chosen
2. **Model Evaluation**
   1. Visualization on overfitting underfitting
   2. Classification: Evaluation of model, Accuracy, F1-Score, confusion matrix
   3. Or Regression: R2 score, RMSE, MAE
3. **Model Optimization**
   1. Trying of different models, comparing results
   2. Strategies on improving model performance (accuracy etc)
   3. Hyperparameter tuning
   4. Evaluation of impact from feature selection
4. **PySpark Integration**
   1. Use Pyspark to perform at least one basic task
   2. Load or explore data using a Spark Session
   3. Apply a transformation using a Spark DataFrame
   4. Clearly mark the PySpark section in your notebook