

## Front-end Web Framework Analysis

Poomrapee Chuthamsatid

Department of Software Engineering, University of Victoria, pchuthamsatid@uvic.ca

Min Kim

Department of Computer Science, University of Victoria, ming0505@uvic.ca

Soyun Lee

Department of Computer Science, University of Victoria, soyunlee@uvic.ca

**Abstract:** With the rapid increase of web frameworks in the present day, this paper aims to offer insights and guidance for choosing the most appropriate front-end web framework. The study employs various methods, including graph virtualization and model training, to determine an influential score that aids in the decision-making process. It analyzes three prominent web frameworks - React, Angular, and jQuery - to identify their respective trends and explores the key factors contributing to their usage. Furthermore, the research delves into a detailed examination of the evolution of these three web frameworks from 2020 to 2022.

**CCS CONCEPTS** Computing methodologies -> Machine learning -> Learning paradigms -> Supervised learning -> Supervised learning by classification

**Additional Keywords:** Accuracy, Attribute, F-1, Cross Validation, Correlation, Causal Diagram, Heatmap, Histogram, Decision Tree, Random Forest, LightGBM, AdaBoost, Pipeline, Data Cleaning, Data Exploration, Data Visualization, Model Training, Model Selection, Model Evaluation

## 1 INTRODUCTION

As technology continues to evolve and people's interest in programming grows, the programming market is expanding rapidly. Consequently, there is an increasing number of programming languages and web frameworks available today. In light of this, our research aims to provide valuable insights into the realm of front-end web frameworks, with a specific focus on three prominent frameworks: React, Angular, and jQuery. The study seeks to address the following three research questions:

**RQ1:** What are the current trends among React, Angular, and jQuery in the programming market?

**RQ2:** How has the usage of these three web frameworks evolved over the past three years, 2020-2022?

**RQ3:** What are the potential key factors influencing the usage of a particular web frameworks?

To support our investigation, we will employ several models, including decision trees, random forests, gradient boosting, and AdaBoost. By exploring these research questions and utilizing diverse analytical approaches, we aim to gain a comprehensive understanding of the dynamics and factors that drive the adoption of web frameworks. Through this research, we hope to contribute valuable insights to the programming community and aid developers in making informed decisions when selecting front-end web frameworks for their projects.

### 1.1 Motivation

The primary motivation behind this research is to provide comprehensive insights and practical guidance to developers and programmers, aiding them in selecting the most suitable front-end web framework for their specific project requirements. This study aims to play a crucial role in helping developers make well-informed decisions that align with industry standards and best practices. Additionally, by offering information on modern trends, the findings will empower developers to stay up-to-date and deliver cutting-edge solutions that meet the demands of the rapidly evolving web development landscape.

### 1.2 Terminologies

The following table is designed to aid in understanding some of the terminologies that will appear frequently (See Table 1). The terms are arranged in alphabetical order.

Table 1: Frequently used terminologies and their definitions

Terminology	Definition
Age	Age of participant
ConvertedCompYearly	Yearly compensation converted to USD
Country	Country the participant is currently residing
EdLevel	Highest level of formal education completed
LanguageHaveWorkedWith	Programming, scripting, and markup languages that have been worked in over the past year
Num_languages	Total number of programming, scripting, and markup languages that have been worked in over the past year
OrgSize	Number of people employed by the company or organization the participant is currently working for
WebframeHaveWorkedWith	Web frameworks that have been worked in over the past year
WebframeWantToWorkWith	Web frameworks want to work in over the next year
YearsCode	Number of years of programming in total, including any education
YearsCodePro	Numer of years of programming professionally, not including any education

### 1.3 Dataset

The datasets used for our research were obtained from Stack Overflow, where they conduct an annual survey each year. In 2020, there were nearly 65,000 participants, while in 2021, there were 83,439 participants, and in 2022, there were 73,268 participants from 180 different countries worldwide. It is important to note that respondents from Crimea, Cuba, Iran, North Korea, and Syria were unable to access the survey due to traffic being blocked by its third-party survey software. The salaries were converted from participant currencies to USD using the exchange rate on May 24, 2022, for the dataset from 2022 and June 16, 2021, for the dataset from 2021 by Stack Overflow. Unfortunately, relative information for the 2020 dataset was not available from the Stack Overflow website.

Each dataset contained more than 80 attributes, and we carefully cleaned the data to narrow it down to only 10 attributes that were of particular interest to our research. The target attribute we decided on was "WebframeHaveWorkedWith," and we also considered the following related attributes: "EdLevel," "YearsCode," "YearsCodePro," "OrgSize," "LanguageHaveWorkedWith," "Age," "WebframeWantToWorkWith," "Num\_languages," "Country," and "ConvertedCompYearly." Detailed explanations for these attributes can be found under the "Keywords" section.

By utilizing these datasets and refining our selection of attributes, we aimed to gain valuable insights into the trends and characteristics of developers worldwide, particularly regarding their web framework usage and related factors.

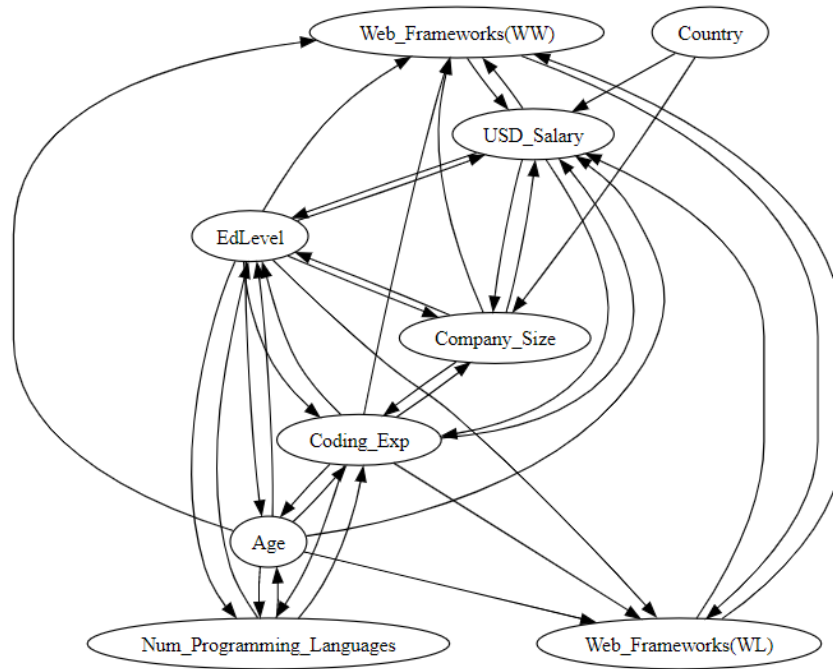


Figure 1: Causal diagram of the datasets

The causal diagram above illustrates the relationships between each attribute and their potential impact on the target attribute (See Figure 1). "Web\_Frameworks (WW)" refers to web frameworks that have been worked with, and "Web\_Frameworks (WL)" refers to web frameworks that developers want to work with. It is important to note that the colors used are arbitrary and do not carry any specific meaning. For instance, we assume that "USD\_Salary" (Yearly compensation converted to USD) is linked to "EdLevel," as the highest education level attained can significantly influence salary decisions within the industry. Similarly, we anticipate that "Company\_Size" and "Country" will also play vital roles; companies with a larger workforce are expected to offer higher compensation compared to smaller companies, and middle to high-income countries are likely to provide better pay compared to developing countries. Additionally, we predict that knowledge or experience with a particular "Web\_Framework (WW)" can make a developer more competent, ultimately leading to higher compensation.

## 2 DATA SCIENCE

### 2.1 Data Cleaning

Data cleaning is an initial step in analyzing data and constructing a machine-learning model. This paper elaborates on the data cleaning process to enhance transparency regarding decision-making within the process, as well as to promote the reproducibility of this research and a comprehensive understanding of the data column attributes. Of the more than 80 column attributes found within the datasets, a causal diagram identifies 10 attributes that may be correlated with "WebframeHaveWorkedWith," a specific target attribute. The initial step in cleaning raw datasets involves filtering out irrelevant column attributes to create a more manageable and comprehensible data set. Three datasets from the years 2020, 2021, and 2022 are processed in this manner, retaining only the columns of interest. Since these datasets originate from surveys conducted in three different years, all corresponding column attributes are renamed to ensure uniformity across the datasets.

Once the column headers are properly formatted, the data preprocessing begins. This stage involves cleaning, standardizing, and transforming each individual data column. Among the over 20 different web frameworks found in the columns "WebframeHaveWorkedWith" and "WebframeWantToWorkWith," this research focuses on extracting only specific frontend web frameworks of interest, namely Angular, jQuery, and React. The survey participants may have listed more than one unique frontend web framework within these columns. To address this, we duplicate the rows and split them in a way that each row contains only one mentioned frontend web framework. Finally, we drop any duplicates and remove any rows containing NaN values in these two columns, ensuring a refined dataset for analysis. The preprocessing of the "OrgSize" column requires the removal of any rows containing NaN or 'I don't know,' as these entries do not contribute to meaningful analysis. Finally, we categorize the "OrgSize" data into five distinct bins, grouping them by size to achieve a normal distribution.

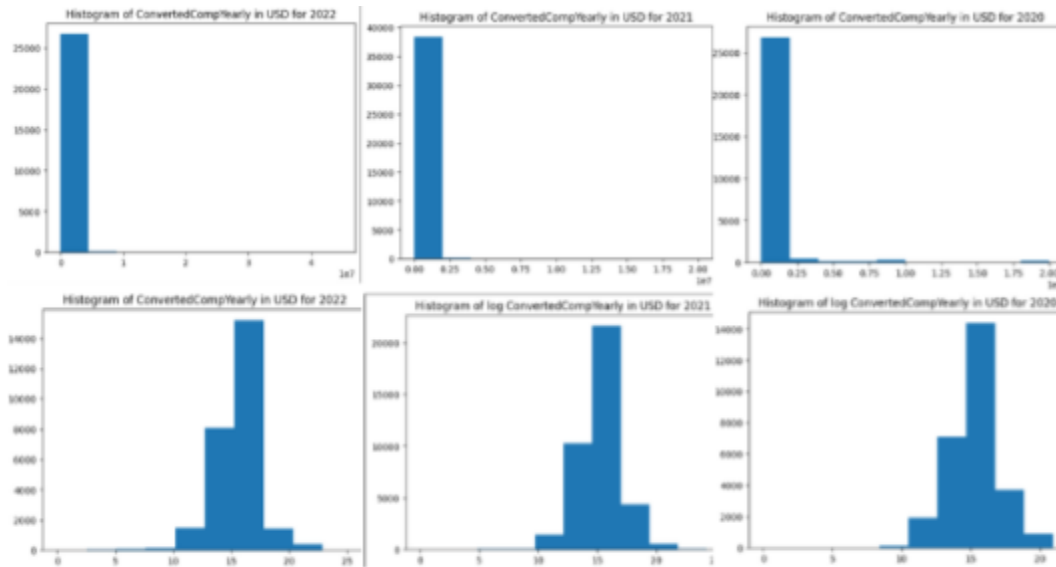


Figure 2: Logarithm transformations of ConvertedCompYearly in 2022, 2021, and 2022

The three surveys involve partial formatting and conversion of the "ConvertedCompYearly" column attribute into USD currency. However, additional preprocessing is necessary to remove data entries containing zero values, as these were identified as outliers. The data distribution of "ConvertedCompYearly" displays a strong right skew, so a logarithmic transformation is applied to each year within the data column, aiming to achieve a normal distribution (See Figure 2). The preprocessing for the "EdLevel" column involves renaming the descriptions of degrees. Additionally, any rows containing 'Something else' as a degree description are dropped, as they do not provide meaningful data for analysis. The level of experience combines the "YearsCode" and "YearsCodePro" columns. Rows with NaN in "YearsCode" are dropped, and NaN in "YearsCodePro" are replaced with zero to account for those with no professional experience. The research replaces any non-numerical entries with numerical values such as 'Less than 1 year' to zero and 'More than 50 years' to 51. The "Age" columns are represented in categorical ranges. We eliminated any rows with the value 'Prefer not to say' and grouped the ranges into 7 bins to achieve a normal distribution. The

"num\_languages" column tallies the total number of programming languages from the "LanguageHaveWorkedWith" column as provided by participants. The "Country" column requires no cleaning. We also add a new column named "yearOfSurvey," assigning numerical values corresponding to the survey years 2022, 2021, and 2020. Ultimately, the three datasets are concatenated into a single dataset, preparing it for further analysis and model training.

## 2.2 Data Exploration and Visualization

The dataset that we are using is intricate. And a quick glance at the data does not allow us to grasp what is happening. By investigating the data, we can determine which attributes require improvement and provide depth of understanding. despite this, the data itself contains the same information, the manner in which it is visualized will ultimately determine the final results.

To effectively address its complexity, employing data visualization is a favorable method. Our approach involves using causal diagrams to identify and explore interesting columns within the dataset. Initially, we focus on determining the total occurrences of React.js, jQuery, and Angular throughout the dataset.

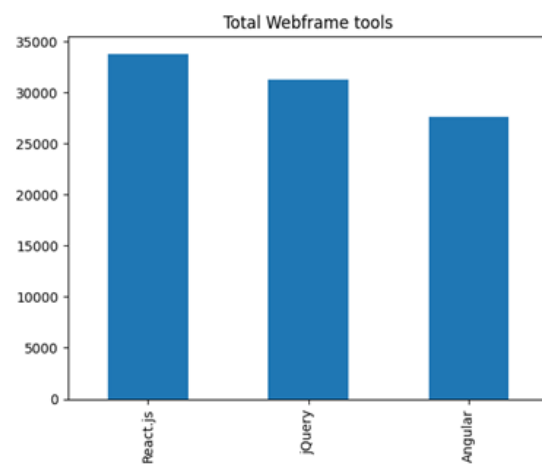


Figure 3: Total Web Framework Tool Frequency from 2020 to 2022

Three separate surveys' worth of data variables were combined, and the rows containing React.js, jQuery, and Angular are chosen. To get the full dataset, the dataset was counted but without the survey year. After that, a code is written to produce the graph's histogram (See Figure 3). First impressions indicate that React.js is the most widely used web framework of the three. But the causal diagram suggests that more study is required. Every year, the distribution of age, coding experience years, and converted pay will be examined.

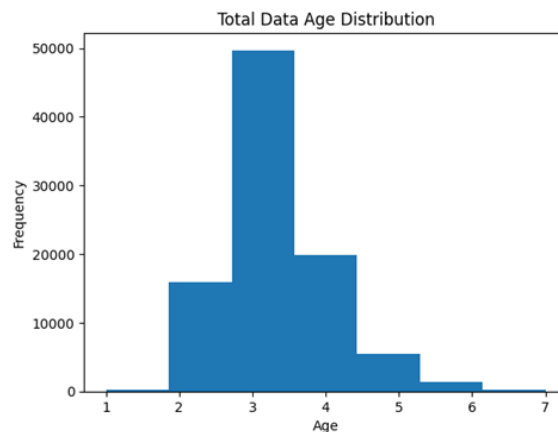


Figure 4: Total Age Distribution Frequency from 2020 to 2022

The data was then expressed into a histogram based on the redefined Age ordinal encoding. This developed visualization provides information on every age group and the number of developers in each range (See Figure 4). According to the dataset's age distribution, most respondents are in the age ranges 2 to 5, with age group 3 (those between the ages of 25 and 34) having the highest proportion of respondents. This demographic, which accounts for more than half of the poll respondents, is important for future study.

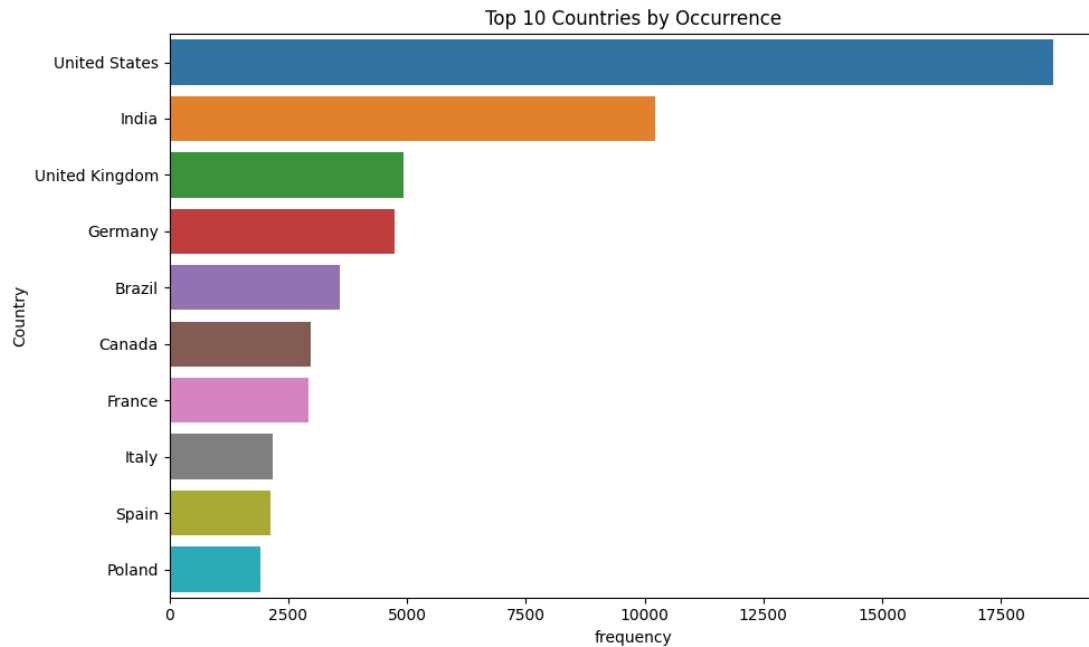


Figure 5: Top 10 Countries Frequency from 2020 to 2022

The list of countries is considered an essential attribute for analysis. The horizontal bar graph is generated by setting up the figure size and utilizing the bar plot function, displaying information on the top ten countries where developers are currently working (See Figure 5). The graph reveals that the United States has the highest number of developers, followed by India in second place and the UK in third place. It is worth noting that these countries have a high English fluency level.

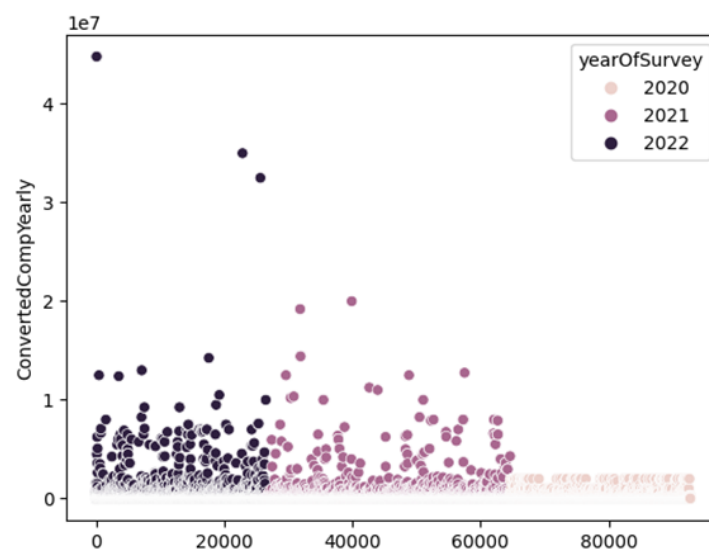


Figure 6: Total Yearly Salary (ConvertedCompYearly) from 2020 to 2022

The total annual salary is one of the principal key factors of interest. A scatter plot was made using the 'ConvertedCompYearly' column, which corresponds to the survey year, in order to investigate and illustrate this aspect (See Figure 6). Different colours are used to distinguish the data points in order to improve clarity. The Y-axis displays the dollar amount for each data point, while the X-axis displays the total annual salary's value range.

According to the analysis, the data tends to have lower Y-values in 2020 than it does in 2021 and 2022. Additionally, the variety of outliers in the data broadens over time.

## 2.3 Data Analysis

After exploration and visualization of the data, there are several interesting key factors and meaningful outputs. The Stack Overflow dataset is in CSV format and spans the years 2020 to 2022. For each participating developer, there are columns and rows with various developer survey questions. We select three front-end framework tools for analysis (Angular, React.js, and jQuery) and condense the dataset as a result. At first, we concentrate on the distribution of these three framework tools over time (See Figure 7).

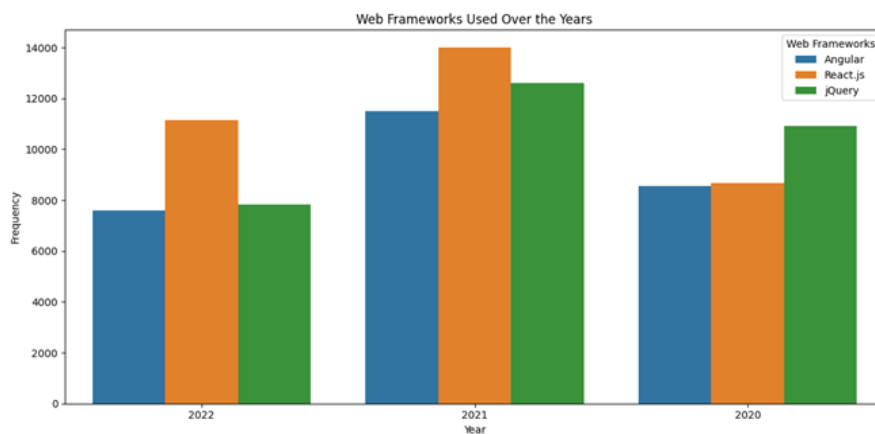


Figure 7: Web Frameworks Usage Frequency from 2020 to 2022

The bar graph displays the annual usage rates for each framework tool. It demonstrates a reasonable distribution with a sizable user base over time.

We continue with our causal diagram by looking at correlations. There are many variables that can affect the framework tool selection. We assume that factors such as education level, company size, coding experience, age, and salary directly affect the decision of which web framework to use based on the generated graph. The number of programming languages and the country are additional sub-variables that affect the primary influencing factors.

We concentrate our analysis of the dataset on identifying numerical trends that have a direct bearing on the selection of web frameworks. In this report, we pay particular attention to the columns Age and Coding Experience because they exhibit strong relationships with the web frameworks used.

We begin by converting each framework tool's percentage value growth from 2020 to 2021 and 2021 to 2022 (See Figure 8).





The heatmap reveals a substantial correlation coefficient of 0.75, indicating a significant relationship between a developer's age and their coding experience (YearsCode). As observed in figure 8, the middle row corresponding to React.js also contains higher numbers compared to the other frameworks (See Figure 9). This suggests that React.js is generally more accessible and easier for novice developers to learn compared to jQuery and Angular.

These results are consistent with the general pattern seen in the graph showing the usage of web frameworks over time. React.js has been gaining users in comparison to Angular and jQuery, making it (from a percentage standpoint) the preferred choice for many developers.

## **2.4 Model selection**

Model selection is a process of constructing and selecting the most suitable predictive model from a set of models for a specific problem. This paper elaborates on the model selection to explore the rationale behind the construction of different models, provide insight into the specifications of each model, and justify the choices made in tuning hyperparameter settings. This research investigates four different models, including Decision Tree, Random Forest, LightGBM, and AdaBoost classifiers, with the aim of identifying the model that performs best in predicting a front-end web framework. Additionally, the models are employed to find the key factors that may influence the selection and utilization of a specific web framework.

The Decision Tree Classifier is chosen for its simplicity and popularity. It divides data into subsets based on feature values, allowing for clear decision-making. However, it can be prone to overfitting if the tree grows too deep. Default hyperparameters are used, with the `random_state` set to 42 to ensure consistency. The Random Forest classifier is chosen for its robustness and accuracy, as it builds on multiple decision trees. The hyperparameters are optimized, with `n_estimators` set to 250 for constructing 250 trees, `n_jobs` set to 4 to expedite training using four cores and `random_state` set to 42 for result consistency. LightGBM, a gradient boosting framework, is chosen for its speed and efficiency, especially with large datasets. Its default hyperparameters yield satisfactory results for this application, except for the `random_state`, set to 42 to ensure consistent results. AdaBoost is chosen for its robust modeling through the combination of weak learners. Using DecisionTreeClassifier as the base with a max depth of 5, it trains 250 weak learners (`n_estimators=250`), controlled by a learning rate of 0.1. The `random_state` is set to 42 for consistent results.

## **2.5 Model Training**

Model training is the core of data processing that defines the model performance and accuracy of results and findings. This paper elaborates on the model training process to provide insight into detailed methodology and transparency into the techniques used. The dataset needs two special preprocessing before proceeding with the model training. The first step encodes categorical attributes into numerical ones using OrdinalEncoder, preparing the data in a format more suitable for the models. The second step randomly shuffles rows (using `random_state=42`) and removes rows with identical predictor values, with the exception of differences in the target value "WebframeHaveWorkedWith." This step improves model performance, as earlier data cleaning duplicated and split rows based on the number of frontend web frameworks, like Angular, jQuery, and React. This created identical predictor values with different targets, which could confuse the model. By eliminating duplicates after shuffling the data, we ensure the integrity of the data distribution (See Figure 10).

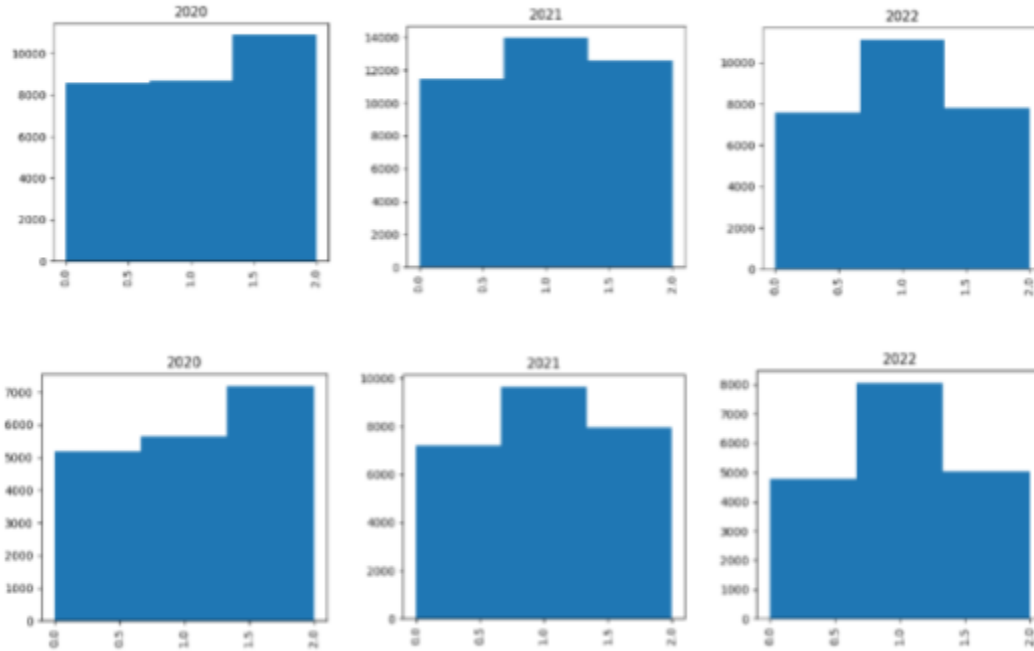


Figure 10: Distributions of Angular (0), React (1), and jQuery (2) for the years 2020, 2021, and 2022, before and after removing duplicates in predictor values (excluding the target "WebframeHaveWorkedWith").

The first phase of model training involves dividing the dataset into training and testing subsets. Utilizing the `train_test_split()` function from the SKlearn library, the dataset is partitioned into 80% for training and 20% for testing. Noting from Figure 2 that the number of records from each year is unevenly distributed, we employ 'stratify' on the "yearOfSurvey" column. This approach ensures a more balanced split, reducing bias in the distribution between the training and testing datasets. After dividing the data into training and testing datasets, we further separate the target column attribute from both sets. This results in the creation of `X_training`, `X_testing`, `y_training`, and `y_testing`. This separation is to ensure that the model is not exposed to the actual outcomes during the learning and evaluation stages, maintaining the integrity of the process.

The next phase in the process is the creation of Pipelines, which facilitate a sequence of transformations on multiple data attributes for machine learning models. Three distinct pipelines are created to apply different preprocessing techniques to three sets of data column attributes.

- **Numerical Pipeline:** Utilizes 'StandardScaler' to normalize numerical features to a mean range between 0 and 1. This scaling is typically preferred by machine learning models.
- **Categorical Pipeline:** Takes care of missing categorical values by imputing them with the 'most\_frequent' strategy and then converts the categorical values into numerical form using OneHotEncoder.
- **Logarithm Pipeline:** Applies a logarithm transformation coupled with 'StandardScaler' to normalize attributes with exponential distribution.

Finally, we apply these pipelines to the selected models along with `X_training` and `y_training` datasets and run the `fit()` function to train the models.

## 2.6 Model evaluation

Model evaluation is an essential stage in the machine learning process serving to provide a comparative analysis of performance and validate the results. This paper elaborates on the model evaluation to offer insight into the techniques used, an understanding of the model's performance, and an interpretation of the evaluation outcomes. The evaluations are designed to measure the model predictability of the target variable, `WebframeHaveWorkedWith`, utilizing measures such as accuracy and F1 scores (See Table 2). We conduct assessments on both training and testing datasets to compare the model's performance under different scenarios. Additionally, we use cross-validation, a technique that divides the data into multiple subsets for both training and testing. This method facilitates a more robust assessment of the model's performance and its ability to generalize.

Table 2: Result for models' evaluation in Accuracy and F-1 measures

Model	Accuracy (Training Sets)	Accuracy (Cross Validation)	Accuracy (Testing Sets)	F1-score (Training Sets)	F1-score (Cross Validation)	F1-score (Testing Sets)
Decision Tree	1.000	0.4799	0.5019	1.000	0.4743	0.4807
Random Forest	1.000	0.5836	0.5925	1.000	0.5785	0.5869
LightGBM (Gradient Boosting)	0.6256	0.6008	0.6091	0.6204	0.5948	0.6021
AdaBoost (based Decision Tree)	0.6582	0.5816	0.6005	0.6538	0.5874	0.5944

The decision tree model appears to be the least effective with a 48% score in accuracy and F1 metrics, and an overfitting issue is apparent with 100% scores in the training dataset. This discrepancy between training and testing results highlights a potential concern regarding the model's generalizability and performance consistency. The random forest model improves to 58% in cross-validation and testing dataset evaluation, but overfitting remains a concern. The Gradient Boosting model consistently achieves over 60% in both accuracy and F1 scores across various scenarios, making it the most suitable option. Finally, the AdaBoost model performs well at 65% in the training dataset but is slightly outperformed by the Gradient Boosting model in testing and cross-validation scenarios.

### 3 RESULT

#### RQ1: What are the current trends among React.js, Angular, and jQuery in the programming market?

While the absolute value of usage cannot be guaranteed, in terms of percentage, React.js shows a consistent and continuous increase compared to Angular and jQuery. There is a strong correlation of 75% between the age of developers and their years of coding experience. According to the heatmap in figures 8 and 9, it is evident that younger developers and who are less year of coding experience are more inclined to learn React.js compared to jQuery and Angular.

#### RQ2: How has the usage of these three web frameworks evolved over the past three years, 2020-2022?

Between the years 2020 and 2021, there was a noticeable increase in the number of users for React.js, Angular, and jQuery. Among these frameworks, React.js experienced the most significant increase in user adoption (React.js increases by 28.41% ). However, from the year 2021 to 2022, there was a general trend of decreased usage for most frameworks. Despite this decline, React.js showed the lowest rate of decrease compared to the other frameworks (React.js decreases by 20.34%).

#### RQ3: What are the potential key factors influencing the usage of web frameworks?

All four models identify "ConvertedCompYearly" as the key factor influencing web framework usage. "Num\_language," "YearsCode," "YearsCodePro," and "Country" have a moderate impact, while "OrgSize," "EdLevel," "Age," and "yearOfSurvey" show little influence (See Figure 11). Interestingly, Decision Tree and Random Forest highlight "WebframeWantToWorkWith" as significant, but LightGBM and AdaBoost disagree, indicating otherwise.

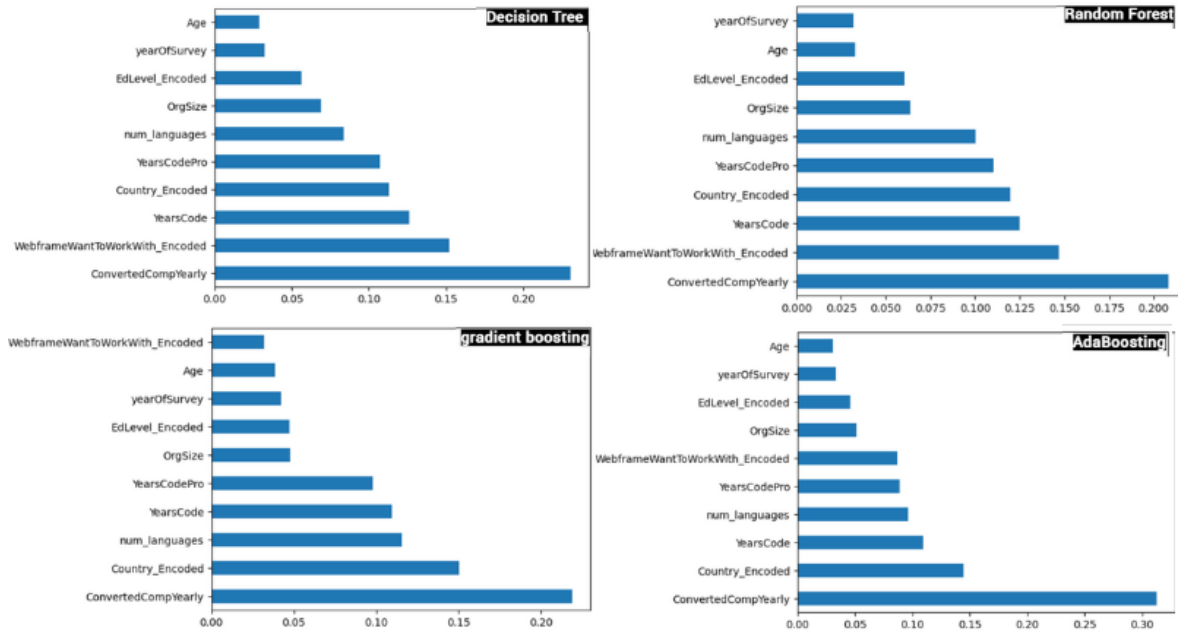


Figure 11: Four bar chart indicate influential score of each column attributes

#### 4 DISCUSSION

In the discussion section, we conduct a comprehensive analysis of the findings, with the aim of interpreting and contextualizing the results obtained in our research. Additionally, we compare and contrast different models to determine their relative strengths and weaknesses, and finally, we thoroughly examine the key factors contributing to overfitting occurrences.

##### RQ1: What are the current trends among React, Angular, and jQuery in the programming market?

The data indicates that developers with less experience in programming languages tend to show a preference for React.js over Angular and jQuery. While the dataset lacks specific columns to fully support this observation, there could be other reasons beyond the scope of the Stack Overflow dataset that influence this trend. Some potential reasons could be related to React.js itself, such as its good documentation, beginner-friendly environment, and ease of learning. These factors might contribute to the higher adoption of React.js among less experienced developers.

##### RQ2: How has the usage of these three web frameworks evolved over the past three years, 2020-2022?

The usage of the frameworks React.js, Angular, and jQuery shows fluctuations as we progress through the years from 2020 to 2022. Among the three frameworks, React.js experiences the most significant increase and the least decrease. However, the dataset does not provide a clear explanation for these trends. Specifically, there is a general decrease in the usage of all frameworks from 2021 to 2022, and one possible contributing factor could be the impact of Covid-19 and related lockdowns. Further investigation is necessary to understand the underlying reasons for these changes.

Although the dataset includes information on company size, age, salary, and other factors, it does not distinctly guide us towards a specific cause for the observed trends. Additional research and analysis would be required to gain a deeper understanding of the situation.

##### RQ3: What are the potential key factors influencing the usage of web frameworks?

Most of this finding is understandable, as we do expect a little impact from "OrgSize," "EdLevel," "Age," and a moderate impact from "Num\_language," "YearsCode," "YearsCodePro," and "Country" on the selection or use of a particular web framework.

Surprisingly, despite a 10% to 20% increase in React usage from 2020 to 2022, the four models do not emphasize the "yearOfSurvey." This could be due to React's rise being overshadowed by the consistently higher usage of Angular and jQuery, which exceeded 10,000 each year. Consequently, the models may err in predicting the influential factor tied to a specific web framework based solely on the "yearOfSurvey", as the relative growth in React's usage is not substantial enough to influence a firm prediction.

The result of "WebframeWantToWorkWith" is worthwhile to discuss. Initially, we expect this attribute to substantially impact web framework usage, as it could foreshadow future learning trends. Upon closer examination of the model evaluations, we find a notable discrepancy. Both Decision Tree and Random Forest models, which attribute significant influence to "WebframeWantToWorkWith," are overfitting, whereas LightGBM and AdaBoost, which assign little weight to this factor, are not. This leads us to conclude that "WebframeWantToWorkWith" might exert undue influence on predictions, and some model algorithms can detect and reduce this effect to mitigate overfitting.

## 5 CONCLUSION

Through data visualization and analysis, we explore three selected attributes: "Age," "YearsCode," and "ConvertedCompSalary," which are found to be correlated with each other. Among these attributes, it is evident that younger developers with less experience in 'YearsCode' show a strong inclination towards learning React.js compared to other frameworks. However, the specific reasons for the fluctuations in the usage numbers each year remain unclear. A careful assumption is that the increase in React.js usage may be attributed to its well-documented nature, beginner-friendly environment, and ease of learning. However, further investigation is required to gain a deeper understanding of these trends and the contributing factors behind them.

Upon further analysis with the machine learning process, we conducted an analysis of various machine learning models for predicting a specific frontend web framework. We found that LightGBM is the most suitable model in predicting a particular frontend web framework. The LightGBM model suggests that "ConvertedCompYearly" is the most influential factor contributing to the use of a particular frontend programming language. The "Num\_language," "YearsCode," "YearsCodePro," and "Country" attributes moderately impact the decision in the use of the web framework. The "OrgSize," "EdLevel," "Age," do a little effect to the decision-making. To address potential overfitting concerns, we observed a decrease in the significance of the "WebframeWantToWorkWith" attribute, indicating its role in mitigating overfitting issues.

## 6 LIMITATION

Some limitations are considered while interpreting the survey results. As mentioned earlier in the paper, one limitation stems from the fact that the survey was not accessible to prospective respondents in certain regions, which limits the diversity of ethnicity representation. Additionally, participants' salary responses in different currencies were converted to USD using a specific date of the year, as previously mentioned. To address potential outliers, the top 2% of salaries were trimmed and replaced with threshold values. Moreover, the varying number of respondents in each year could introduce some biased data. Lastly, we had to handle and remove unclear and invalid data, which might have affected the overall analysis.

## 7 FUTURE WORK

Due to time constraints, we regret that we are unable to expand the project further. However, given the opportunity of additional time, we would aim to procure similar datasets from different sources, such as GitHub or JetBrains. Additionally, incorporating supplementary datasets with distinct characteristics could prove highly advantageous, as it would enhance the accuracy and depth of our predictions and analysis.

## ACKNOWLEDGEMENT

The work was enhanced by ChatGPT, with a specific focus on improving the English grammatical syntax in the paper *only*.



## REFERENCES

- [1] Stack Overflow. 2022. Stack Overflow Developer Survey 2022. Retrieved June 25, 2023 from <https://survey.stackoverflow.co/2022>
- [2] Stack Overflow. 2021. Stack Overflow Developer Survey 2021. Retrieved June 25, 2023 from <https://insights.stackoverflow.com/survey/2021>
- [3] Stack Overflow. 2020. Stack Overflow Developer Survey 2020. Retrieved June 25, 2023 from <https://insights.stackoverflow.com/survey/2020>
- [4] Google Colab. 2023. Feature Engineering and Model Tuning. Retrieved June 25, 2023 from `02_chap.ipynb`
- [5] Google Colab. 2023. Classification and Evaluation. Retrieved June 25, 2023 from `03_chap.ipynb`
- [6] Data Maven. 2023. Sof\_survey\_analysis. Retrieved June 25, 2023 from <https://www.kaggle.com/code/crownedhead06/sof-survey-analysis>
- [7] Cory Kapser and Michael W. Godfrey. 2006. “Cloning Considered Harmful” Considered Harmful: patterns of cloning in software. Empirical Software Engineering, vol. 13, no. 6, pp. 645-692, doi: <https://doi.org/10.1007/s10664-008-9076-6>
- [8] Mark Harman and Bryan F. Jones. 2001. Search-based software engineering. Information and software technology, vol. 43, no. 14, pp. 833-839, doi: [https://doi.org/10.1016/s0950-5849\(01\)00189-6](https://doi.org/10.1016/s0950-5849(01)00189-6)

## Appendix A

describe and provide links to any supplementary materials or replication packages.

The list below elaborates on how the supplementary materials were used for the project.

1. Stack Overflow Developer Survey 2022 - <https://survey.stackoverflow.co/2022>
2. Stack Overflow Developer Survey 2021 - <https://insights.stackoverflow.com/survey/2021>
3. Stack Overflow Developer Survey 2020 - <https://insights.stackoverflow.com/survey/2020>
  - Main datasets for the project
4. Feature Engineering and Model Tuning -  02\_chap.ipynb
  - Guidance on programming code for feature engineering, data cleaning, data visualization, data analysis, Model selection, model training
5. Classification and Evaluation -  03\_chap.ipynb
  - Guidance on programming code for model Evaluation
6. Sof\_survey\_analysis
  - Guidance on programming for graphs and charts in data visualization and data analysis
7. “Cloning Considered Harmful” Considered Harmful
8. Search-based Software Engineering
  - Guideline on writing a formal report paper

## Appendix B

The following table summarizes the contributions made by each team member for the final report.

Team member	Contribution
Poomrapee	<ul style="list-style-type: none"><li>- Causal Diagram programming code</li><li>- Data Clean-up programming code</li><li>- Model Training programming code</li><li>- Model Selection programming code</li><li>- Model Evaluation programming code</li><li>- Data Cleaning</li><li>- Model Selection</li><li>- Model Training</li><li>- Model Evaluation</li><li>- Result</li><li>- Discussion</li><li>- Conclusion</li><li>- References</li><li>- Appendix A</li><li>- Appendix B</li></ul>
Min	<ul style="list-style-type: none"><li>- Data Clean-up programming code</li><li>- Data Visualization programming code</li><li>- Data numerical value calculation programming code</li><li>- Data Analysis</li><li>- Data Explore</li><li>- Data Visualization</li><li>- Result</li><li>- Discussion</li><li>- Conclusion</li><li>- References</li><li>- Appendix A</li><li>- Appendix B</li></ul>
Soyun	<ul style="list-style-type: none"><li>- Introduction</li><li>- Motivation</li><li>- Terminologies</li><li>- Dataset</li><li>- Data Clean-up programming code</li><li>- Limitation</li><li>- Future Work</li><li>- Acknowledgement</li><li>- References</li><li>- Appendix A</li><li>- Appendix B</li><li>- Documentation following ACM format</li></ul>