Work done so far

Over the past few weeks, we have accomplished the following tasks:

- We cleaned, integrated, and analyzed the dataset from Stack Overflow, along with other data sources, and selected the 10 features that have the strongest correlation with salary.
- We conducted a series of visualizations and analyses on the selected features, and trained multiple models including XGBoost, Random Forest, Lasso regression, Ridge regression, and linear regression, evaluating the performance of each model.
- Based on our selected features, we built a frontend web page that can perform online salary predictions based on user input data (the features we selected).
- We have basically determined the framework and main content of the report and started preparing the draft.
- We have completed task distribution based on each team member's interests and strengths, and regularly hold meetings to discuss the project.

Challenges faced

Data Quality and Data Cleaning

Our dataset comes from Stack Overflow's user survey data. The survey page does not enforce validation of whether users have filled in data (for example, a user might be lazy and not select all programming languages they know in multiple-choice boxes), nor does it check whether users will enter obviously unreasonable data (such as users not wanting to fill in their real salary range, but instead entering extremely large values like hundreds of millions, or extremely small values like 100).

Therefore, our solution is to discard data whenever key features are missing, and we also removed some obviously unreasonable outliers, such as samples with annual salaries less than 100.

Issues Encountered in Dataset Integration

Our dataset, which is Stack Overflow's survey respondents, consists of global developer users. Therefore, compared to years of work experience and technology stack used, the country where users are located has a greater impact on salary levels. As a result, we also incorporated some public data including GDP of users' countries and cost of living indices, and integrated these additional data with our main dataset for prediction.

Demo Implementation

Since model training and testing are completely separate from the online web service, we needed to find a way to combine model predictions with the web application while keeping the training algorithms and web development separate. We found that after model training is completed, parameters can be exported as pkl files, and the web service can then read the pkl files to make predictions. However, since our model uses one-hot encoding, the number of features increased from 10 to several hundred, so when importing for prediction, we need to ensure that online data goes through correct one-hot encoding before making predictions.

Preliminary results

- We filtered out about 10 features from the 80 features in the dataset that have a significant impact on salary, including: years of work experience, years of programming experience, age, education level, company size, work location, employment type, job position, and programming languages used. Combined with the cost of living index of users' countries and the GDP of their countries, we can make good predictions.
- Currently, our best-performing model is Lasso regression. Since our predicted values are log-transformed salaries, the current MSE is 0.0806.
- We have completed the first version of the frontend page, which can correctly read model files and perform salary predictions. Here is the demo video: https://youtu.be/4DnVqufIzPc

Discussion

Over the past few weeks, we have basically completed project topic selection, dataset search and cleaning, model training, and frontend page development.

During this process, we found that the most challenging aspects of data science projects are two points:

1. The first is having a good idea for the project. We had many ideas at the beginning of topic selection, but the salary prediction idea is relatively good - if we can make it into a website, most people might be willing to use it.
2. The second issue is data quality. We did not use salary datasets from Kaggle but directly used user survey data from Stack Overflow, because we believe survey data is more original and better reflects the complexity of real-world work data. However, there are also many invalid data in the dataset, and we spent considerable time on data cleaning.

Our next step is to use explainable AI (SHAP) to interpret our salary prediction results, while continuing to optimize the frontend page and increase the interpretability of predictions.