SIM3 Task 2: Design and Development

Anthony Mascari

Western Governor’s University

C964: Computer Science Capstone

Part A – Letter of Transmittal & Project Proposal

## Summary

The task of predicting competitive hourly rates for the roles of any profession but particularly (and more relevantly) the roles of data analyst and machine learning engineer across the globe can be extremely challenging. Submitting competitive bids is a key part of winning employment contracts on machine learning or data analysis projects. There are a number of factors that likely influence the rate which businesses and individuals charge making it difficult to submit competitive bids for contracts in regions the bidder lacks familiarity with. With some surrounding information about features such as location, company size, job title and experience we can create tools to allow more accurate predictions in a very fast manner compared to an individual having to research each region on their own and possibly gathering poor data to make their decision from.

## Benefits

The proposed product benefits customers in a variety of ways. First, the data product assists in predicting fair hourly rates in different regions around the world. The product is able to do this quickly without spending human time digging through the internet for tidbits of information. This supports the user in a number of critical ways for business success with faster turn-around time on bids and more competitive bids given the aggregated and analyzed data. Lastly, this product allows for the fast comparison of different opportunities in different regions so a business with limited resources can quickly make the best strategic decision possible.

## Product Outline

This product is a model that will be trained using machine learning. The algorithm that this model will use is the XGBoost algorithm. It will be trained on data such as location, job title, company size, and average pay. The user interface for the model will be a web-application where a user can enter criteria about the role that they would like to receive a prediction for. This interface will be written in HTMX with the backend API running on python’s FastAPI framework on a REST API.

## Data Description

The data that the model will be trained on contains a number of useful features such as job title, category, salary, location, experience, company location, and company size. There are over 9300 rows in the initial dataset which will need to be cleaned and verified to contain useful data. The dataset has the years 2020 through 2023 which means we will have up-to-date information once the model has been trained.

## Objective & Hypotheses

Developing an accurate and efficient tool which is easy for non-technical individuals to use to predict pay rates is the objective of this project. The hypothesis is that a model which is properly trained on the dataset will be able to accurately predict pay rates in different regions given certain input and details about the project.

## Methodology

The methodology in general will follow an AGILE style and begins with collecting and cleaning the data in phase one. The next phase is to visualize the data and look into any possible outliers or items that may incorrectly and drastically influence our result. After data visualization, the model will be trained using the XGBoost algorithm. Once the model is trained a thin backend web API will be developed to serve the http requests from a web app front end. After the backend is functioning the front end will be developed for user interaction. Once all pieces are working, they will be tested then documented.

## Funding Requirements

In order to develop this data product, it will require funding for a data scientist/machine learning engineer. There is also the cost of infrastructure and necessary hardware and software tools and tooling. The budget estimated for this project is $85,600 the largest portion of that being the machine learning engineer’s time.

## Impact on Stakeholders

This data product has a very positive impact on stakeholders by providing a means of quickly and accurately gathering information relative to submitting competitive bids. The business is likely to secure more contracts at rates they are comfortable with since they do not have to spend human resources to manually gather data anymore. The business may use this predictive tool to help save hours on every bid which means more bids can be analyzed by the business which in turn means that the business may pick and choose the projects that offer the biggest advantage for them.

## Ethical & Legal

The ethical and legal considerations for this project will include a variety of steps to ensure adherence. Data will be checked for anonymization and anonymized if it is not already. The integrity of the data will be protected throughout all steps of the process. Any data that is sensitive, highly personal, or legally regulated will not be included.

## Expertise

The required expertise in regard to the proposed solution includes data gathering or mining, data preparation, proficiency in implementing machine learning via libraries, software development of a backend REST API, software development of a front-end user interface (web-app), and the ability to design the system so it works cohesively as one experience for a non-technical end-user.

Part B – Executive Summary to IT Professionals

## Summary

This project proposes a means of addressing the challenging problem of predicting competitive hourly rates for data analysts and machine learning engineers across the globe. Different regions around the world have many different factors that affect the pricing of their services. When submitting bids for projects, a competitive bid makes the difference of keeping everyone employed and busy or paying idle human resources. The opportunity for improving the bidding process lies in developing a data product that enables the insight of what bids may be competitive in any given region around the globe. Factors that will be considered include location, experience, and job title, among others.

## Product Fit

Currently, in order to gain insight into what a competitive bid may be in a region it requires human hours to be spent manually researching for each proposal. In order to expedite the turn-around time for proposal creation and provide more reliable estimates a data product will be created. This product will enable users to enter a few details and instantly be provided with a competitive bid number or range.

## Existing Gaps

While there are some tools or products that allow for general estimations of salary ranges between different countries, there are none that exist that are as nuanced or specific to the data analysis and machine learning fields. This tool provides a more accurate estimate than general difference in cost of living from country to country.

## Data to Support Lifecycle

The initial set of data contains over 9000 entries with job titles, categories, salaries, location of residence, and location of company. The number of samples in north America outweighs the number of samples from other countries so additional data may need to be collected to remedy this. Additionally, as world economics are not static, the model will need to train on data quarterly as economic new is published and affects salary ranges.

## Methodology

In order to create a model quickly and efficiently, the project will implement the CRISP-DM Methodology. First, we will ensure a business understanding, then gather and gain understanding of the data through the preparation process, next we will build the model followed by an evaluation period and lastly the model will be deployed. This process will allow for fairly direct and rapid development and deployment of our model.

## Deliverables

The deliverables for this project will include the following items. A trained and fully functional predictive model will be delivered as part of the software package. A REST API will be delivered along with a user-friendly front-end. The front end and the backend will be integrated, and documentation will be provided including notes in the delivered source code as well as a comprehensive user manual.

## Implementation Plan

The implementation plan involves first developing the model, back end, front end interface, and linking everything together. Next the implementation will be tested, and feedback notes will be taken. The outcome that is anticipated from the implementation should be faster availability of more competitive bids and reduced time in bid-preparation for the end user.

## Validation & Verification

In order to be considered a success, the data product will need to hit a variety of targets. The model will need to be at minimum 80% accurate meaning that on average the estimate provided by the model should be within plus or minus 20% of the verification target. The response time will need to be under one second from user submission to a response from the program. User feedback will be the most reliable method for verifying an intuitive and easy-to-use interface on the front end.

## Programming Environments

The environment for development will begin based in the Python language. This language has been selected as it is industry standard and has many wonderful tools built with clean interfaces for us to use. The preferred IDE will be up to the developer but a recommendation of NeoVim will be given with a Python Language Server running as well as a linter for uniform formatting. The use of Jupiter will also be accepted if the engineers prefer. Development will take place on the Linux operating system as it is the only OS that supports multiple GPUs for training the XGBoost model. The front end will be built using HTMX. Testing will take place on Localhost. If the customer desires the model may be deployed on an AWS EC2 server with the REST API sitting behind NGINX, otherwise it will be able to run locally without the additional expense of cloud hosting. The majority of the cost will be human resources, which ironically include a data analyst and machine learning engineer. If cloud hosting is included the total estimate cost of the project up to deployment is around $85,500.

## Expertise

The required expertise in regard to the proposed solution includes data gathering or mining, data preparation, proficiency in implementing machine learning via libraries, software development of a backend REST API, software development of a front-end user interface (web-app), and the ability to design the system so it works cohesively as one experience for a non-technical end-user.

## Timeline

The projected timeline and milestones are laid out in a table below along with any dependencies or resources assigned to each task.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sprint** | **Start** | **End** | **Required  Dependency / Resources** | **Milestone** |
| 1 | 02/01/2024 | 02/14/2024 | Data Analyst | Business requirement research, initial data collection, preliminary analysis. |
| 2 | 02/15/2024 | 03/07/2024 | Data Analyst Machine Learning Engineer | Data preparation and cleaning, preliminary feature engineering |
| 3 | 03/08/2024 | 03/30/2024 | Machine Learning Engineer | Model development and initial training with XGBoost. |
| 4 | 04/01/2024 | 04/14/2024 | Machine Learning Engineer | Model testing and tuning. |
| 5 | 04/15/2024 | 04/28/2024 | Software Engineer (General) User Experience Eval | Develop interface for operations users |
| 6 | 04/29/24 | 05/14/24 | User Experience Eval | User Feedback |
| 7 | 05/15/24 | 05/30/24 | Machine Learning Engineer  Software Engineer (General) | Documentation |
| 8 | 06/01/24 | 06/14/24 | Devops Engineer  Software Engineer (General) | Deployment & Delivery |

Part C – Design & Development

## Overview

Below is a table referencing the required functionality and where to locate the functionality that has been implemented along with a brief description. Path beginning is at the source root.

|  |  |  |
| --- | --- | --- |
| **Requirement** | **Function or Location of Data** | **Description** |
| Descriptive Method | describeAll() | This method is descriptive and describes each column individually in the dataset. |
| Non-Descriptive Method | makePrediction() | This method uses the XGBoost model that was created to predict salary based on input features. |
| Datasets | loadData() | Reads in the csv ‘jobs\_in\_data.csv’ into a pandas dataframe and returns the dataframe. |
| Decision Support | Entire Application | The combination of user input with the many smaller functions in the program combine to provide decision support. |
| Data Wrangling | dropCols()  nullEmptyCheck() | Function created to drop a list of columns in a single call and another function created to verify that values inside our data are not empty or Null/NaN before attempting training a model. |
| Data Exploration | describeAll()  encode()  train\_test\_split() | These functions all contribute to getting a picture of the data and separating it for further analysis and use. |
| Interactive Queries | Estimator.html | Front end uses HTMX to dynamically load data into dropdown menus from the csv that the model was trained on. It then fetches predictions from a REST API which uses the stored XGBoost model to make a prediction from the input parameters and return a value to the user on the front end. Next there is a “visualize prediction” button that creates a variety of charts displaying what the predicted average is and where it falls relative to adjacent categories. |
| Machine Learning Methods | XGBoost / joblib | XGBoost was used to create the model and it was stored using joblib. |
| Accuracy Evaluation | RSME | The backend code calculates Root Mean Squared Error to measure the model’s accuracy. |
| Security | login() | Mandatory Login to access the front end user interface. |
| Monitor & Maintain | customMsg()  warn()  error() | Customized logging calls for monitoring activity on the API. Errors that are caught in try/catch are passed into the logging information for future maintenance and monitoring. |
| Dashboard | locationVsalary()  categoryVusd()  salaryDistrib() | Three front-end visualizations based upon input parameters. These functions create images and return bytecode that is temporarily stored in memory to be passed back from the API to display customized images for the user based upon their input parameters. |

Part D – Forms & Documentation

## Overview

Below is a table displaying the names of the requested forms and documentation.

|  |  |
| --- | --- |
| **Document Required** | **Location** |
| Business Vision | C964/Docs/BusinessVision.docx |
| Dataset | C964/jobs\_in\_data.csv |
| Analysis Code | C964/modules/visualize.py  C964/modules/train.py |
| Hypothesis Assessment | C964/Docs/HypothesisAssessment.docx |
| Visualizations | C964/Docs/Visualizations.docx |
| Accuracy Assessment | C964/modules/train.py |
| Results | C964/Docs/Results.docx |
| Source Code | C964 (root) |
| QuickStart Guide | C964/Dist/QuickStart.txt |

References

Dataset provided courtesy of:

Hummam Qaasim. 2024. Jobs in Data. Kaggle.

<https://www.kaggle.com/datasets/hummaamqaasim/jobs-in-data>

Documentation used:

Pandas - <https://pandas.pydata.org/docs/>

NumPy - <https://numpy.org/doc/>

SeaBorn - <https://seaborn.pydata.org/>

XGBoost - <https://xgboost.readthedocs.io/en/stable/>

MatPlotLib - <https://matplotlib.org/stable/index.html>

SciKitLearn - https://scikit-learn.org/stable/