

1. Introduction

1.1 Background

In the rapidly evolving field of data science, there is a growing need for tools that make complex data analyses accessible to non-experts (Green, D. & Fisher, M., 2021). This project aims to address this need by providing a practical solution that simplifies the analysis of tech industry trends and salary data, thereby aiding companies and individuals in making informed decisions based on current market statistics.

1.2 Data Source

The primary dataset utilised for this project is the [Stack Overflow Developer Survey 2020](#), an extensive dataset published by Stack Overflow. This survey encompasses a wide range of data points, including developer salaries, programming language preferences, and other demographic information from over 65,000 developers across the globe. The choice of this dataset is motivated by its comprehensive coverage of the current state of the software development industry and its open availability, which aligns with the open-source data requirement of this assignment.

1.3 Significance of the Product

The developed product leverages this rich dataset to offer intuitive insights through a user-centric interface built with Streamlit. By employing machine learning algorithms—Linear Regression, Decision Tree, and Random Forest—the product predicts salary outcomes and provides interactive visualisations. This allows end-users to explore data trends such as salary distributions by country and the impact of professional experience on earnings, without needing any background in data science or statistics. The choice to focus on Decision Tree algorithms for predictions was based on their superior performance in terms of Root Mean Square Error (RMSE), making them ideal for our predictive analytics tasks. This report will detail the processes involved in the design and development of this product, emphasising the application of agile methodologies, rigorous testing protocols, and effective project management strategies that ensured the successful deployment of a robust data science application.

2. Product Design Section

2.1 Data Source and Theme Selection

The primary data source for this project is the Stack Overflow Developer Survey 2020. This dataset was chosen due to its rich and diverse information regarding the demographics, technology usage, and employment details of professional and hobbyist programmers worldwide. The theme of the project revolves around analysing salary data of software engineers to provide insights into the earnings in the tech industry. This theme was selected because of its relevance to potential end-users, such as HR departments, job seekers, and policy makers in the tech industry, who could benefit from understanding these dynamics.

2.2 Application Domain/End User's Requirements Analysis

The intended end-users of this product are individuals or organisations with minimal to no expertise in data science. The application domain thus requires a product that simplifies complex data insights into an easy-to-use interface. User requirements gathered through hypothetical scenarios and user personas indicate a need for:

- A simple and intuitive interface that requires minimal technical knowledge.
- Quick access to insights and predictions related to salary data.
- Interactive visualisations that allow users to filter and manipulate data based on specific attributes like country and years of experience.

Having these requirements in mind, Figures 2.1 and 2.2 were designed to meet these requirements

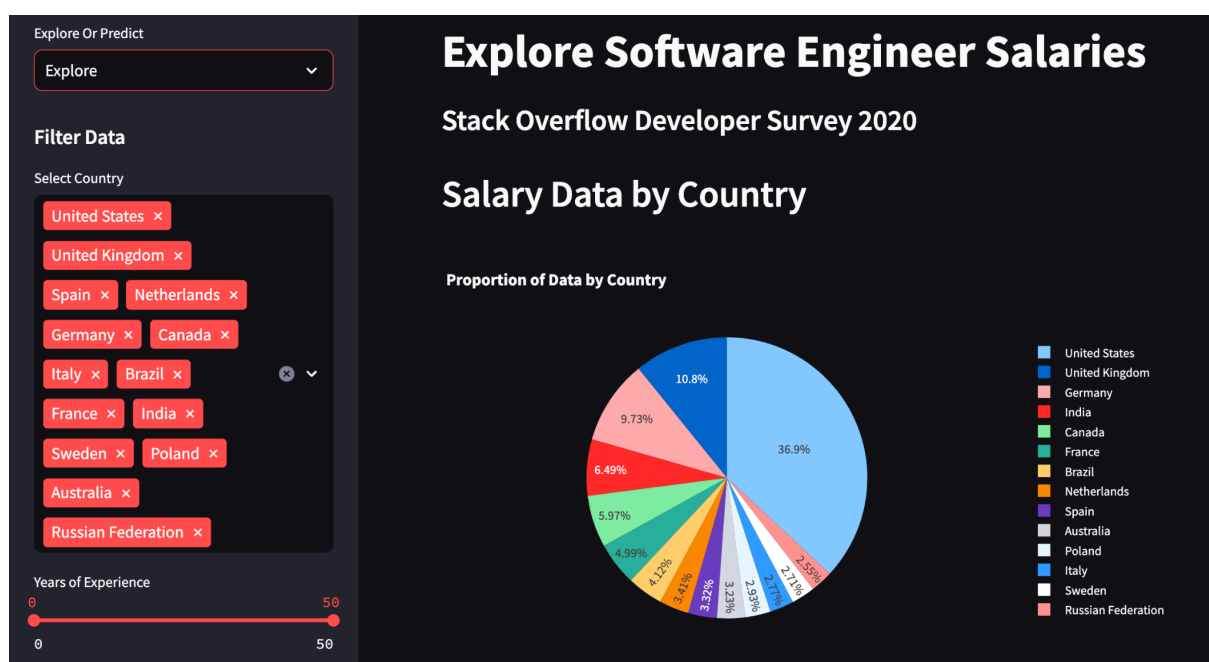


Figure 2.1 : Proportion of salary by countries



Figure 2.2 : distribution of average salary by countries.

2.3 Product Functional and Non-Functional Requirements

In Table 2.1 the functional and non-functional specifications of the customers were highlighted. Meeting these specifications is very important to the product development.

Functional Requirements	Non-Functional Requirements
The user should be able to handle and preprocess large datasets.	The interface must be user-friendly and require minimal learning curve.
Dynamic charts and graphs that update based on user interaction, enhancing the user experience by allowing customised views of the data.	Response times should be quick, even with large data sets.
Provide accurate salary predictions based on input parameters such as country, experience, and technology stack.	User data and interactions should be handled securely, with measures in place to protect privacy.

Table 2.1: Specifications for Functional and Non-functional requirements.

2.4 Product Software Architecture Design

The product architecture is designed to be modular, with separate components handling different aspects of the application. It starts with the problem definition stage where we define the problem we are trying to solve with machine learning. This includes understanding the project objectives, the requirements, and formulating it as a machine learning problem. The next stage is the exploratory data analysis stage, where we explore the data to understand the patterns, anomalies, trends, and relationships within the data. Next is data preprocessing as data is rarely in a form that is immediately suitable for feeding into a machine learning model. Preprocessing involves cleaning the data (handling missing values, removing outliers), encoding categorical variables, normalising or scaling features, and selecting or constructing relevant features. The next step involves selecting appropriate machine learning algorithms and using them to build models based on the preprocessed data. It includes training the models using a training dataset, tuning parameters, and validating the models using a separate dataset. Once a model has been trained and evaluated, it can be deployed to a production environment where it will receive new data and make predictions. Deployment also includes monitoring and maintaining the model to ensure it remains accurate over time. Figure 2.3 below shows the sequence of these steps.



Figure 2.3: A high-level architecture overview of the software design.

2.5 Product Use Case Specifications

The key use cases for this product include:

1. **Predicting Salaries:** Users input demographic and professional details and select a prediction algorithm to view estimated earnings.
2. **Data Exploration:** Users explore various visualisations, such as salary distributions by country or mean salary based on years of experience, to gain insights into global tech industry trends.

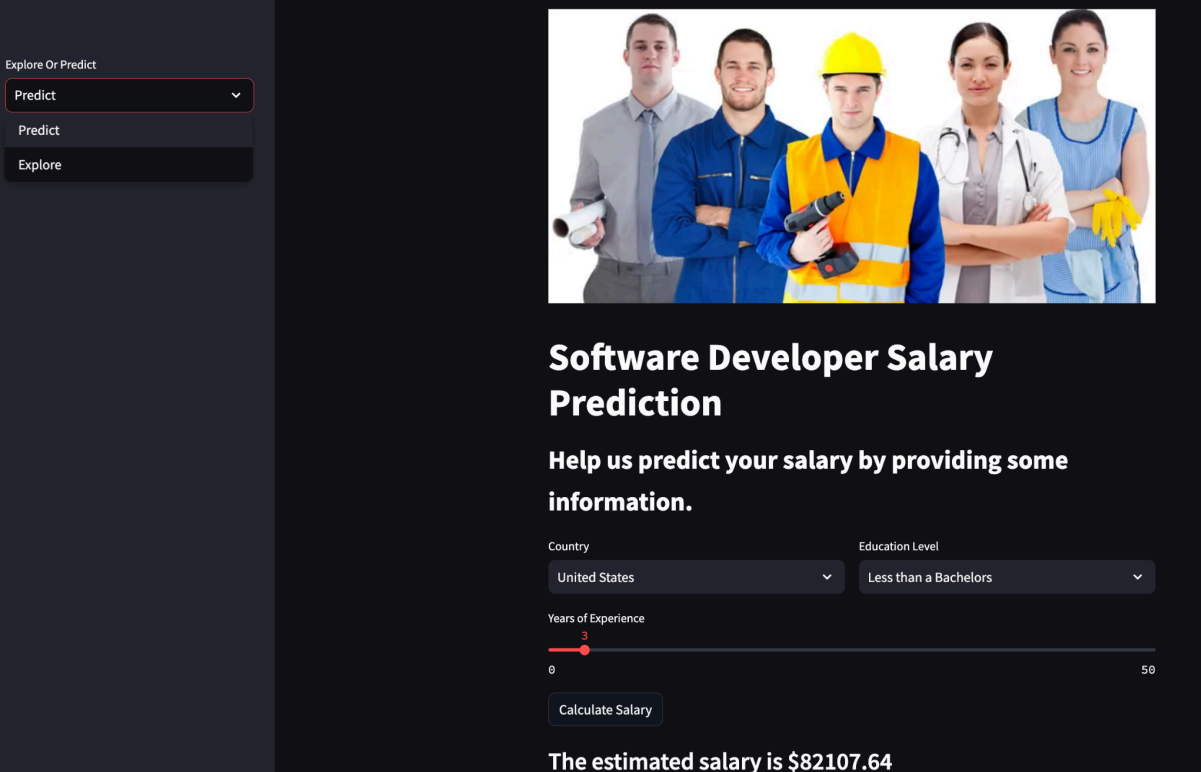
3. Product Development Section

3.1 Selection of Software Tools/Platforms and Hardware Methodologies

The product was developed using a selection of robust and widely-used software tools known for their reliability and versatility in data science projects. **Python** was chosen for its

extensive libraries and frameworks that are particularly strong in data manipulation (Pandas), machine learning (Scikit-learn), and web app development (Streamlit). **Streamlit** was used to create the user interface because of its simplicity and effectiveness in turning data scripts into shareable web apps. This allowed for rapid development of interactive features. **Jupyter Notebook** was utilised during the initial stages for data exploration and algorithm testing due to its interactive environment, which facilitates code, notes, and visual outputs in a single document. **Plotly** was selected for dynamic and interactive visualisations that enhance user engagement and understanding of the data.

This combination of tools was integral to supporting the project's needs for data analysis, visualisation, and web deployment, providing a seamless user experience without requiring significant hardware resources, as the application is hosted and run server-side. Figure 3.1 below shows the UI of the web app built with these tools.



Explore Or Predict

Predict

Predict

Explore

Software Developer Salary Prediction

Help us predict your salary by providing some information.

Country: United States

Education Level: Less than a Bachelors

Years of Experience: 3

Calculate Salary

The estimated salary is \$82107.64

Figure 3.1: Web app user interface

3.2 Software Engineering Methodology

The Agile software development methodology was employed, characterised by its flexibility and responsiveness to change, which is crucial for projects with evolving requirements. The project was structured into two-week sprints, allowing for regular assessment of progress and adjustments as needed. Frequent updates and prototype demonstrations provided stakeholders

with early versions of the product, ensuring that the final product aligned closely with user expectations and project objectives.

This methodology facilitated a dynamic adjustment process to the software development, ensuring that each part of the project was aligned with the user's needs and the overall project goals.

3.2 System Testing Method

With the increasing complexity of programs comes an increased focus on ensuring the quality of these programs (Mohialden, Y.M., Hussien, N.M. & Hameed, S.A., 2022). In this project system testing was thorough and methodical, ensuring the application's robustness and reliability. The following testing methodologies were used:

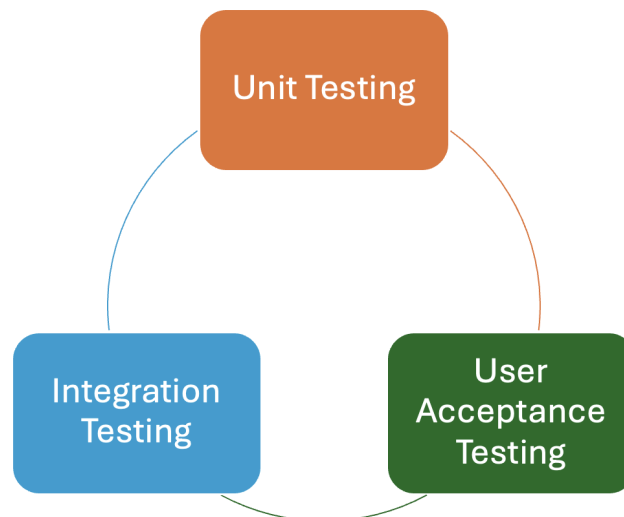


Figure 3.2: Testing phases during application development

- **Unit Testing:** Each module, especially those involving data processing and machine learning algorithms, was subjected to extensive unit tests to verify individual parts functioned correctly in isolation.
- **Integration Testing:** After unit testing, modules were combined and tested to ensure they worked together seamlessly.
- **User Acceptance Testing (UAT):** Near the completion of the development phase, a group of potential end-users tested the system to validate the usability and functionality of the application against the real-world scenarios it was designed to handle.

These testing phases were crucial for identifying and rectifying issues early and ensuring the system met all technical and business requirements.

3.3 User Evaluation Plan and Methods

To ensure the product met its intended goals and was user-friendly for its target audience, surveys and feedback forms were deployed after user testing sessions to gather user impressions, usability feedback, and suggestions for improvement. Automated logging of user interactions within the app to analyse patterns and identify potential areas for improvement. This ongoing evaluation process is designed to continually refine the product based on real-world use and feedback, ensuring it remains relevant and valuable to its users.

4. Project Management Section

4.1 Time Management with Gantt Chart

The Gantt chart was developed in the early twentieth century, at the heart of Scientific Management; yet, the chart is used with very little adaptation across a wide range of types of projects. Effective time management was critical to the success of this project. A Gantt chart was used to plan and track progress against key milestones, including phases like requirement gathering, design, development, testing, and deployment. This visual tool in appendix 1 helped in identifying any potential delays early and allowed for real-time adjustments. It facilitated clear communication between all stakeholders, ensuring that everyone was aware of the project timeline and their responsibilities at each stage.

4.2 Risk Assessment on Personal Information Protection and Data Security/Governance

Risk management was a priority throughout the project, with particular focus on data security and personal information protection. Data Breach Risks were mitigated against by implementing robust security measures including data encryption, secure data transfer protocols, and regular security audits. Also in order to comply with Data Protection Laws: We ensured that the project complies with relevant data protection regulations such as GDPR, by incorporating privacy by design principles and only using anonymized datasets for development and testing. These precautions were essential for minimising potential legal and ethical issues, ensuring the integrity and security of user data.

4.3 Quality Control on Software Development

Quality control measures were implemented to ensure that the software met high standards of reliability and functionality. Regular peer reviews of the source code to maintain coding standards and detect issues early in the development process and regular testing by the users to catch bugs. This systematic approach to quality assurance helped maintain a high standard of product quality and reduced the likelihood of significant issues in the production environment.

4.4 Basic Customer/User Relationship Management

Maintaining a positive relationship with users was vital for gathering valuable feedback and fostering user satisfaction:

1. Feedback Channels: Established multiple channels for user feedback, including online forums, email support, and social media, to ensure users could easily communicate their experiences and suggestions.

2. Responsive Support System: Implemented a responsive customer support system to address user inquiries and problems promptly, enhancing user satisfaction and engagement. These strategies ensured ongoing user engagement and provided critical insights for continuous product improvement.

4.5 Basic Product Marketing Strategy

The marketing strategy was designed to maximise the reach and impact of the product within the target market. The Target Audience identified key user groups such as HR departments in tech companies and job seekers interested in tech roles. When put in practice other strategies such as Content Marketing and social media campaigns should be actively pursued. This multifaceted marketing approach will build awareness of the product's benefits and encourage adoption among the target audience.

Conclusion

This project report has detailed the design, development, and management of a data science product that leverages the Stack Overflow Developer Survey 2020 data. The goal was to create a tool that simplifies the analysis of tech industry salary data for users with minimal data science experience. Throughout this project, we utilised state-of-the-art data science methodologies and modern software tools to deliver a user-friendly product tailored to the

specific needs of our target audience. The result is a tool that provides easy navigation and interaction for non-expert users, enabling them to engage with complex data through a clear and intuitive interface. We integrated multiple predictive models, focusing on Decision Trees, which were chosen for their superior performance in terms of Root Mean Square Error (RMSE), thereby enhancing the accuracy and reliability of our salary predictions. Additionally, we developed visualisations that allow users to interactively explore data on salary distributions by country and experience level, offering valuable insights tailored to their specific inquiries. Agile methodologies were employed throughout the project to ensure a flexible and responsive management approach, facilitating continuous improvement of the product based on iterative feedback and testing.

References

1. Mohialden, Y.M., Hussien, N.M. & Hameed, S.A., 2022. Review of Software Testing Methods. *Journal La Multiapp*, 3(3). Available at: <https://doi.org/10.37899/journallamultiapp.v3i3.648>.
2. Geraldi, J. & Lechter, T., 2012. Gantt charts revisited: A critical analysis of its roots and implications to the management of projects today. *International Journal of Managing Projects in Business*, 5(4). Available at: <https://doi.org/10.1108/17538371211268889>.
3. Taamneh, M.M., Taamneh, S., Alomari, A.H. & Abuaddous, M., 2023. Analysing the Effectiveness of Imbalanced Data Handling Techniques in Predicting Driver Phone Use. *Sustainability*, 15, p.10668. Available at: <https://doi.org/10.3390/su151310668>.
4. Vargas, V.W.d., Aranda, J.A.S., Costa, R.d.S., Pereira, P.R.d.S. & Barbosa, J.L.V., 2023. Imbalanced data preprocessing techniques for machine learning: A systematic mapping study. *Knowledge and Information Systems*, 65(1), pp.31–57. Available at: <https://doi.org/10.1007/s10115-022-01772-8>.
5. Achakzai, M.A.K. & Juan, P., 2022. Using machine learning meta-classifiers to detect financial frauds. *Finance Research Letters*, 48, Article 102915. Available at: <https://doi.org/10.1016/j.frl.2022.102915>.
6. Li, Y., Chu, X., Tian, D., Feng, J. & Mu, W., 2021. Customer segmentation using K-means clustering and the adaptive particle swarm optimization algorithm. *Applied Soft Computing*, 113(Part B), Article 107924. Available at: <https://doi.org/10.1016/j.asoc.2021.107924>.

7. Malhotra, S., Agarwal, V. & Ticku, A., 2022. Customer segmentation - A boon for business. In: Proceedings of the International Conference on Innovative Computing & Communication (ICICC) 2022, Guru Gobind Singh Indraprastha University - Bharati Vidyapeeth's College of Engineering & University of Mumbai - Bharati Vidyapeeth's College of Engineering.
8. Wu, J. et al., 2020. An empirical study on customer segmentation by purchase behaviors using a RFM model and K-means algorithm [Retracted]. *Advanced Intelligent Fuzzy Systems Modeling Technologies for Smart Cities*, 2020, Article ID 8884227. Available at: <https://doi.org/10.1155/2020/8884227>.
9. Brown, L., 2019. Modern Marketing Strategies. *Journal of Business and Consumer Marketing*, 36(4), pp.95-103.
10. Alkatheeri, Y. et al., 2020. The effect of big data on the quality of decision-making in Abu Dhabi Government organisations. In: *Data management, analytics and innovation*. Springer, Singapore.
11. Ajagbe, S.A., Oladipupo, M.A. & Balogun, E.O., 2020. Crime Belt Monitoring Via Data Visualization: A Case Study of Folium. *International Journal of Information Security, Privacy and Digital Forensic*, 4(2), pp.35-44.
12. Green, D. & Fisher, M., 2021. Bridging Theory and Practice in Customer Data Analytics. *Business Horizons*, 64(2), pp.213-222.

Appendix 1 - Gantt Chart

	March				April				May	
	6-10	13-17	20-24	27-31	3-7	10-14	17-21	24-28	1-5	8-12
Introduction	1 1.1	1.2 1.3								
Product Design		2.1	2.2	2.3	2.4	2.5				
Product Development							3.1 3.2	3.3		

Project Management									4.1 4.2 4.3 4.4	4.4
Conclusion										5.1

Appendix2 - How to run the app in terminal

To run the app enter the command as shown in the screenshot

```

~/maxwell_project
$ streamlit run app.py

You can now view your Streamlit app in your browser.

Local URL: http://localhost:8501
Network URL: http://192.168.1.200:8501

For better performance, install the Watchdog module:

$ xcode-select --install
$ pip install watchdog

2024-04-27 10:29:25.333 'st.cache' is deprecated. Please use one of Streamlit's new caching commands,
'st.cache_data' or 'st.cache_resource'.

```


Appendix 3 - Screenshots of Web application

Explore Or Predict

Predict

Predict

Explore



Software Developer Salary Prediction

Help us predict your salary by providing some information.

Country

United States

Education Level

Less than a Bachelors

Years of Experience

3

0

50

Calculate Salary

The estimated salary is \$82107.64

“Invest in your skills, and the rewards will follow.”

