**Functional Specification for Salary Prediction System**

**Date: October 22, 2024**

**Version: 1.0**

**Prepared by:**

Kirubel Temesgen

Student Number: C00260396

**Course:** Data Science and Machine Learning

**Supervisor:** Greg Doyle

Contents

[1. Introduction 2](#_Toc180507717)

[2. System Overview 2](#_Toc180507718)

[3. Functional Requirements 5](#_Toc180507719)

[4. Technical Specifications 6](#_Toc180507720)

[5. Non-Functional Requirements 6](#_Toc180507721)

[6. Limitations, Challenges, and Opportunities 7](#_Toc180507722)

[7. Conclusion 7](#_Toc180507723)

[8. References 8](#_Toc180507724)

[9. Appendices 8](#_Toc180507725)

## 1. Introduction

The Salary Prediction System helps predict employee salaries using Machine Learning (ML). It analyses data from different sectors and business sizes to forecast salary trends. The system is designed to assist organisations with budgeting and future salary planning by using Linear Regression to uncover relationships between salary, business size, and industry sector.

Users upload CSV files with salary data, select specific columns, and the system filters the data and generates salary predictions. These are then visualised through graphs, making the results easy to understand. The system is valuable for businesses looking to plan salary budgets, track trends, and make informed compensation decisions.

## 2. ***System Overview***

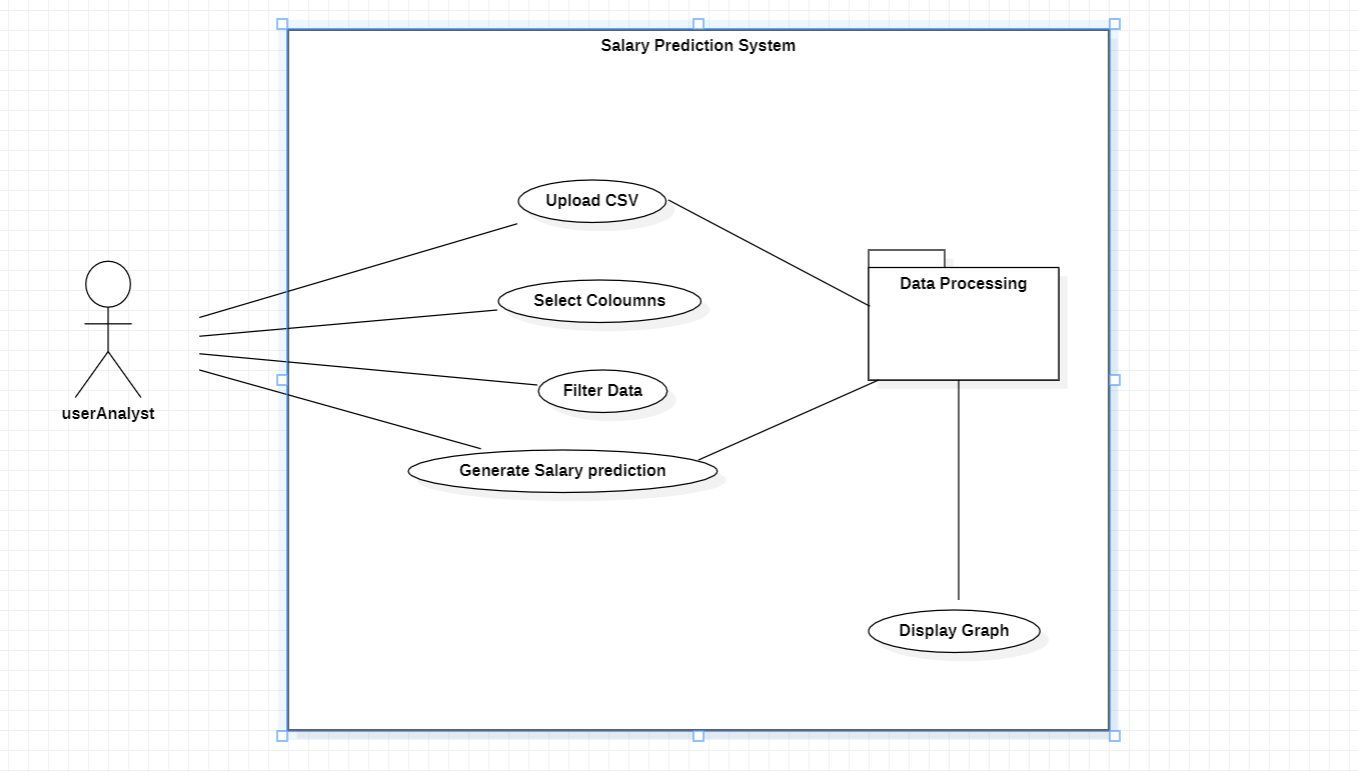
4.1 Use Cases and Diagrams

User Uploads CSV: Users upload CSV files containing salary data for different sectors and company sizes.

User Selects Sector and Business Size: After the CSV upload, the user selects the columns for salary, sector, and business size.

Prediction Output: The system processes the input data and generates salary predictions, visualised through color-coded graphs.

Figure1 - Use Case Diagram



4.2 System Architecture

This sequence diagram and system architecture diagram presents how the system works, from input (CSV file) to processing (data filtering, ML prediction) and output (graphical representation and report). This highlights user input and interactions within and between the system.

Figure2 - System Architecture Diagram

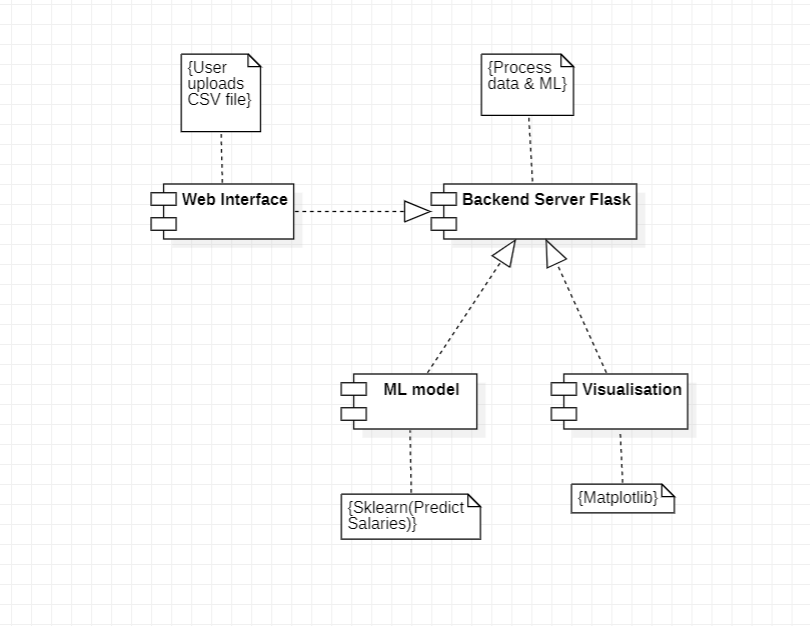


Figure3 – Sequence Diagram

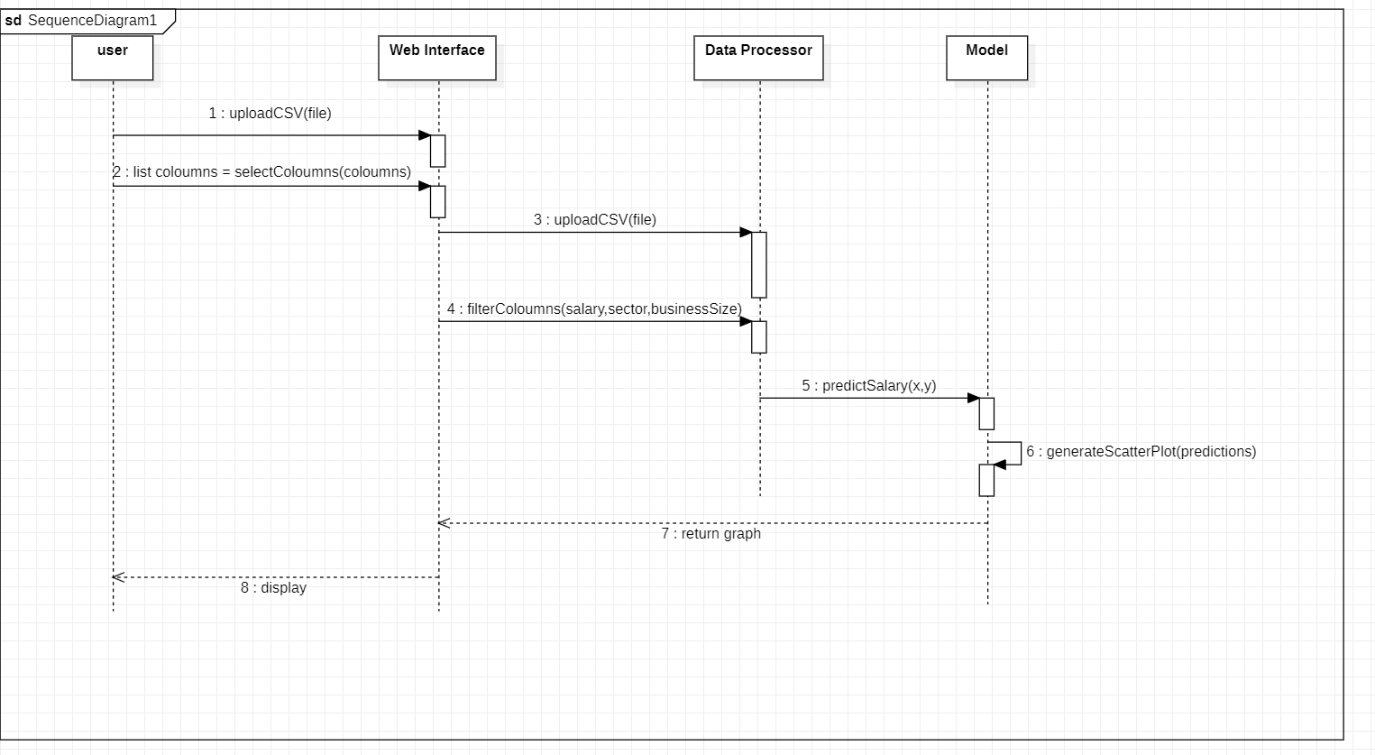


Figure4 - Actual vs Predicted Salary Graph (Sector-wise Colour Coded)

A screenshot of a computer screen

Description automatically generated

## 3. Functional Requirements

5.1 User Interface

Web Interface: A simple, user-friendly web interface built using Flask.

Upload Functionality: A file upload button allowing the user to import the CSV file.

Selection of Columns: Dropdown menus to allow users to specify the salary, sector, and business size columns.

Graphical Output: Color-coded scatter plots representing actual vs predicted salary values.

5.2 Data Inputs

CSV File Format: The input must be a CSV file, containing columns that match the types such as salary(Decimal value), sector(String), and business size.(String)

Dynamic Columns: The system should identify and adapt to different column names, allowing flexibility in data inputs.

5.3 Data Processing

Data Cleaning: Handling missing or invalid salary values, and ensuring data quality before processing. (strip)

One-Hot Encoding: The system uses one-hot encoding to process categorical data (e.g., economic sectors).

Linear Regression Model: Implements a Linear Regression model to predict salary based on business size and sector.

5.4 Output and Reporting

Scatter Plot: The system should provide a scatter plot with color-coded data points for each sector.

Prediction Report: A downloadable summary report, including predicted salaries and model evaluation metrics (e.g. R² score, mean squared error).

## 

## 4. Technical Specifications

6.1 Software

Python for machine learning and web development.

Flask for the web interface.

Pandas, scikit-learn for data processing and machine learning.

Matplotlib for plotting the results.

Use of PythonAnywhere for deployment

6.2 Hardware

Standard PC or cloud server capable of running Python applications.

Browser to access the Flask-based application.

6.3 Libraries

Pandas for data manipulation.

scikit-learn for machine learning algorithms (Linear Regression).

Matplotlib for generating scatter plots and other visualisations.

## 5. Non-Functional Requirements

Performance: The system should process datasets of varying sizes in less than 5 seconds.

Scalability: Ability to handle larger datasets(Over 1 million or so efficiently) or additional sectors and company sizes in the future.

Usability: A simple, intuitive user interface designed for non-technical users.

Reliability: Accurate salary predictions based on reliable data processing and ML models (mean score over 75%).

## 

## 6. Limitations, Challenges, and Opportunities

Limitations:

Prediction accuracy may vary based on the quality and completeness of input data.

Business sizes and economic sectors may vary across regions, causing discrepancies in the data.

Challenges:

Handling data discrepancies such as missing salary information or improperly formatted CSV files.

Ensuring that the predictions are valid for a wide range of sectors.

Opportunities:

Expanding the system to use more advanced models like Decision Trees or Random Forests for better predictions.

To be a general prediction model, not limited only to estimating salaries.

Integration with cloud services for larger datasets and faster processing.

## 7. Conclusion

The Salary Prediction System leverages Machine Learning to predict salary trends based on business size and sector, offering valuable insights for companies and HR departments. By analysing historical salary data, the system helps in forecasting future salaries, which supports more informed budgeting and compensation planning.

Key benefits include:

* Accurate salary predictions based on sector and business size.
* Valuable insights for HR teams and management to track salary trends.
* Ease of use, allowing non-technical users to upload CSV files, filter data, and generate salary predictions with minimal effort.
* The system’s use of Linear Regression ensures technical sophistication in predicting salary patterns, while the user-friendly interface makes it accessible to a broad range of users without requiring deep technical knowledge. This tool streamlines decision-making around salary management and future planning.

## 

## 8. References

* **Pandas**: McKinney, W., 2023. *Pandas: Python Data Analysis Library*. Available at: [General functions — pandas 2.2.3 documentation](https://pandas.pydata.org/docs/reference/general_functions.html#data-manipulations) [Accessed 22 October 2024].
* **Matplotlib**: Hunter, J.D., 2007. *Matplotlib: A 2D Graphics Environment*.[online] Available at: [Plot types — Matplotlib 3.9.2 documentation](https://matplotlib.org/stable/plot_types/index.html) [Accessed 22 October 2024].
* matplotlib.org. (n.d.). Quick start guide — Matplotlib 3.8.4 documentation. [online] Available at: <https://matplotlib.org/stable/users/explain/quick_start.html#types-of-inputs-to-plotting-functions> [Accessed 22 Oct. 2024].
* matplotlib.org. (n.d.). Tight layout guide — Matplotlib 3.8.4 documentation. [online] Available at: <https://matplotlib.org/stable/users/explain/axes/tight_layout_guide.html> [Accessed 22 Oct. 2024].
* scikit-learn.org. (n.d.). Linear Regression — scikit-learn 1.3.1 documentation. [online] Available at: [1.1. Linear Models — scikit-learn 1.5.2 documentation](https://scikit-learn.org/stable/modules/linear_model.html#ordinary-least-squares) [Accessed 22 Oct. 2024].
* MDN Web Docs. (n.d.). HTMLSelectElement: add(). [online] Available at: <https://developer.mozilla.org/en-US/docs/Web/API/HTMLSelectElement/add>
* MDN Web Docs. (n.d.). FormData documentation. [online] Available at: <https://developer.mozilla.org/en-US/docs/Web/API/FormData>

## 9. Appendices

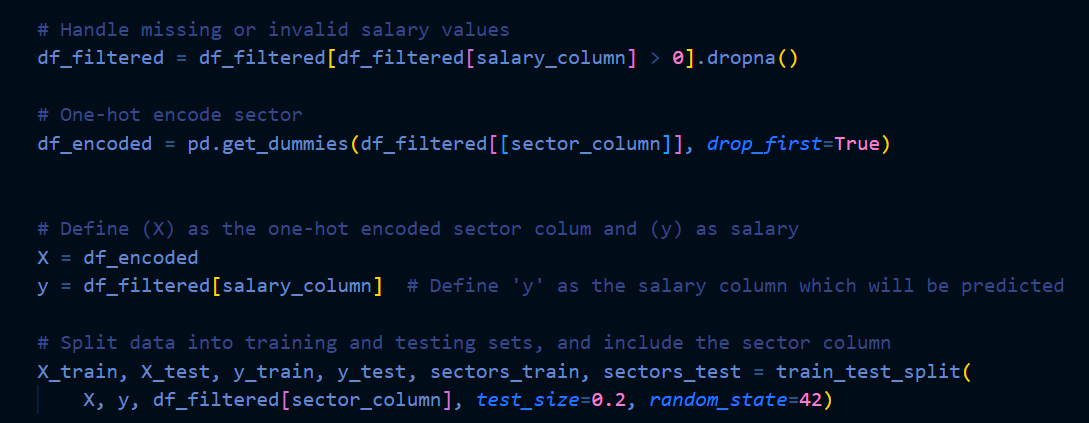
[Figure 1](#Figure1): Use Case Diagram for the Salary Prediction System

[Figure2](#Figure2): System Architecture Diagram

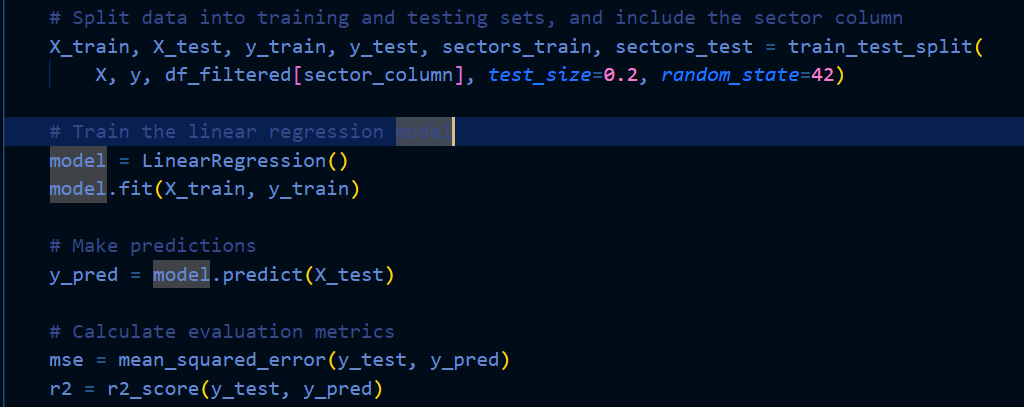
[Figure3](#Figure3): Sequence Diagram

[Figure4](#Figure4): Actual vs Predicted Salary Graph (Sector-wise Color Coded)

Code Snippet 1: Data Preprocessing - One-Hot Encoding for Sectors



Code Snippet 2: Linear Regression Model for Salary Prediction



Code Snippet 3: Visualisation Code - Scatter Plot of Actual vs Predicted Salaries with Color-Coded Sectors

