Salary prediction



Name : Pretty Abraham

Batch Name : Data science

Project Name : Build A Machine Learning Web App From Scratch

Date : 09/01/2024

INTRODUCTION:

The integration of machine learning into user-friendly applications has become increasingly crucial in today's technological landscape. This project serves as a practical guide for developers looking to build a Machine Learning web application from scratch using the Python programming language and Streamlit, a powerful and intuitive web app framework.

In the initial phase, the project focuses on leveraging real-world data for analysis and machine learning model development. This includes exploring data, conducting feature engineering, and employing popular Python libraries such as pandas, NumPy, and scikit-learn. Developers will gain insights into the entire process of preparing and fine-tuning a machine learning model.

The second part of the project shifts the focus to Streamlit, a user-friendly framework designed to simplify web application development. With Streamlit, developers can seamlessly transform their machine learning models into interactive and visually appealing web applications with minimal code. The emphasis is on creating a compelling user interface that effectively communicates the insights derived from the machine learning model.

Throughout this project, the goal is to empower developers with the skills to bridge the gap between data science and web application development. By providing a step-by-step guide, we aim to demystify the complexities of both machine learning and web development, enabling developers to create impactful and accessible applications that leverage the power of data-driven insights.

PROJECT OBJECTIVE:

The project is designed to achieve the following specific goals:

· **Data Analysis and Model Development:**

* · Utilize real-world data to guide developers through the process of data analysis, feature engineering, and the development of a machine learning model.
* Demonstrate the use of popular Python libraries, including pandas, NumPy, and scikit-learn, for efficient data manipulation and model training.

· **Web Application Design with Streamlit:**

* · Introduce Streamlit as a powerful and user-friendly web app framework for Python.
* Provide hands-on experience in designing an intuitive user interface using Streamlit's simple syntax and features.
* Seamless integration of the previously developed machine learning model into the Streamlit web application.

· **User Interaction and Visualization:**

* · Showcase the creation of interactive elements within the web app for user input and model interaction.
* Implement visualizations to effectively communicate the insights derived from the machine learning model.

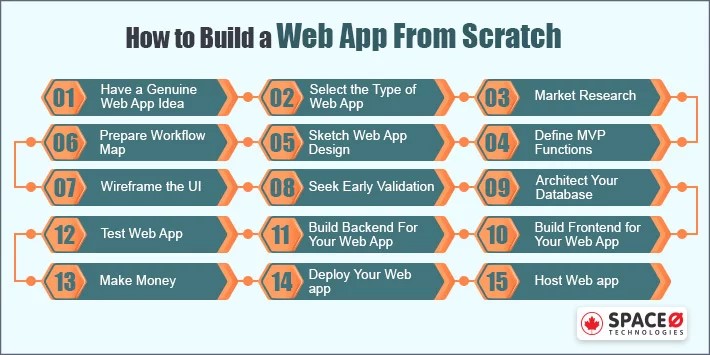
· **End-to-End Deployment:**

* · Guide developers through the deployment process, ensuring the web application is accessible to users.
* Explore deployment options, including cloud platforms, to showcase real-world accessibility and scalability.

· **Practical Application:**

* · Emphasize real-world application scenarios, demonstrating how machine learning models can be seamlessly integrated into web applications for practical use.
* Encourage developers to adapt the project to their specific use cases and datasets

By achieving these objectives, this project aims to provide a comprehensive and hands-on learning experience, enabling developers to confidently build and deploy Machine Learning web applications using Python and Streamlit. The end result is a skill set that combines data science and web development, facilitating the creation of impactful applications that harness the capabilities of machine learning in a user-friendly manner.



**CONTENTS:**

Building a machine learning web app from scratch involves several steps, and it requires a combination of programming, web development, and machine learning skills.

* **Frontend**: HTML, CSS, JavaScript (React, Angular, or Vue.js)
* **Backend**: Flask (Python), Django (Python), Express (Node.js), or others
* **Database**: SQLite, PostgreSQL, MongoDB, etc.
* **Machine Learning Framework**: TensorFlow, PyTorch, Scikit-learn (Python)
* **Deployment**: Docker, Heroku, AWS, Google Cloud, or Azure

### Table of Contents

1. **Introduction**

1.1 Purpose of the Document  
1.2 Scope of the Machine Learning Web App  
1.3 Objectives  
1.4 Document Overview

**2. Overall Description**

2.1 Product Perspective  
2.2 Product Features  
2.3 User Classes and Characteristics  
2.4 Operating Environment  
2.5 Design and Implementation Constraints  
2.6 User Documentation  
2.7 Assumptions and Dependencies  
2.8 Future Enhancements

**3. Specific Requirements**

3.1 External Interface Requirements  
3.1.1 User Interfaces  
3.1.2 Hardware Interfaces  
3.1.3 Software Interfaces  
3.1.4 Communication Interfaces  
3.2 Functional Requirements  
3.2.1 Data Analysis Module  
3.2.2 Model Development Module  
3.2.3 Streamlit Web Interface  
3.2.4 User Input and Interaction  
3.2.5 Visualization Module  
3.2.6 Deployment Module  
3.3 Performance Requirements  
3.3.1 Response Time  
3.3.2 Scalability  
3.4 Security Requirements  
3.4.1 Data Encryption  
3.4.2 Access Control  
3.5 Software Quality Attributes  
3.5.1 Usability  
3.5.2 Reliability  
3.5.3 Maintainability  
3.5.4 Portability  
3.6 Constraints  
3.6.1 Legal and Regulatory  
3.6.2 Ethical Considerations

**4. System Models**

4.1 Use Case Diagrams  
4.2 Activity Diagrams  
4.3 Sequence Diagrams  
4.4 Class Diagrams  
4.5 Data Flow Diagrams

**5. External Interface Requirements**

5.1 User Interfaces  
5.1.1 Landing Page  
5.1.2 Data Input Interface  
5.1.3 Model Output Interface  
5.2 Hardware Interfaces  
5.3 Software Interfaces  
5.4 Communication Interfaces

**6. Non-functional Requirements**

6.1 Usability  
6.2 Reliability  
6.3 Performance  
6.4 Supportability  
6.5 Security  
6.6 Cultural and Political Requirements

**7. Appendix**

7.1 Glossary  
7.2 References  
7.3 Revision History

This simplified SRS outline provides a structure to define the requirements for a Machine Learning Web App, covering aspects from introduction and overall description to specific functionalities, system models, and non-functional requirements. Each section can be expanded upon with detailed information as needed.

Conclusions:

This Software Requirements Specification (SRS) document serves as a comprehensive guide, outlining the system's objectives, scope, and detailed specifications to ensure the successful development and deployment of the salary prediction model.

The primary aim of our Salary Prediction System is to empower organizations and individuals with a reliable tool that leverages machine learning algorithms to forecast salaries accurately. By focusing on intricate details such as data sources, model training, and performance metrics, we are committed to delivering a solution that facilitates informed decision-making in hiring processes and career planning.

Throughout this document, we have meticulously defined the system's purpose, delving into specific requirements ranging from data collection and preprocessing to the implementation of machine learning models. Attention has been given to non-functional aspects, including security, scalability, and usability, to ensure that the system not only meets technical standards but also aligns with user expectations.

We have included a set of system models, such as use case diagrams, activity diagrams, sequence diagrams, and data flow diagrams, to provide a visual representation of the system's functionality and interactions. These models serve as essential tools for both developers and stakeholders, fostering a shared understanding of the system's architecture and processes.

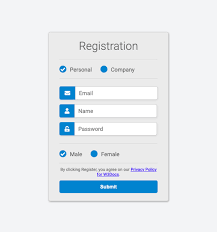
External interface requirements, encompassing user interfaces and communication interfaces, have been thoroughly detailed to ensure a seamless and user-friendly experience. By prioritizing clear communication and intuitive design, we aim to create a system that is accessible and beneficial to a diverse user base.

As we transition into the development phase, we are excited about the potential impact of our Salary Prediction System. This tool has the capacity to streamline human resources processes, guide job seekers in making informed decisions, and contribute to a more transparent and equitable job market. With the insights gained from this SRS document, we are well-prepared to transform our vision into a tangible and effective solution, revolutionizing the way salary predictions are made in the professional landscape.

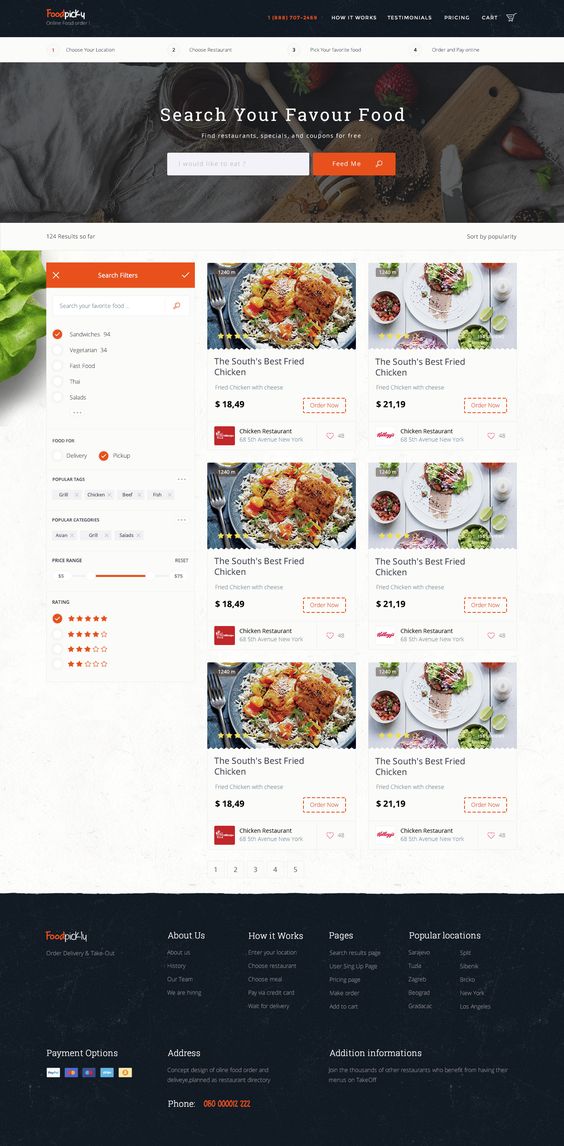
Step1:



STEP 2:



STEP 3:



STEP 4:

A collage of different types of oats

Description automatically generated