



ARCHITECTURE DESIGN

Credit Card Default Prediction



INEURON
PRATHMESH JAGTAP

Contents

| | |
|---|---|
| Abstract | 2 |
| 1. Introduction | 3 |
| 1.1 Purpose | 3 |
| 1.2 Scope | 3 |
| 1.3 Definitions, Acronyms and Abbreviations | 3 |
| 2. Architectural Representation | 3 |
| 2.1 Scenarios | 3 |
| 2.2 Logical view | 4 |
| 2.3 Process View | 4 |
| 2.4 Development View | 4 |
| 2.5 Physical View | 5 |
| 3. Architectural Goals and Constraints | 5 |
| 3.1 Server side | 5 |
| 3.2 Client side | 5 |
| 3.3 Development Tools | 5 |
| 4. Use Case View | 6 |
| 4.1 Use case diagrams | 6 |

Abstract

Machine Learning is a category of algorithms that allows software application to become more accurate in predicting outcomes without being explicitly programmed. The basic premise of machine learning is to build models and employ algorithms that can receive input data and use statistical analysis to predict an output while updating outputs as new data becomes available. These models can be applied in different areas and trained to match the expectations of management so that accurate steps can be taken to achieve the organization's target. Taking various aspects of a dataset collected from people and the methodology followed for building a predictive model.

1. Introduction

1.1 Purpose

This document provides a comprehensive architectural overview of the system, using a number of different architectural views to depict different aspects of the system. It is intended to capture and convey the significant architectural decision which have been made on the system. This document elaborates the architecture of the system in 5 different views. Both the static and dynamic behavior of the system is described in this document, All the required diagrams and their descriptions are available in this document.

1.2 Scope

The software architecture document applies to each static and dynamic aspect of the system. This document is a component-level design process that follows a step-by-step refinement process. This process can be used for designing data structures, required software, architecture, source code, and ultimately, performance algorithms. Overall, the data organization may be defined during requirement analysis and then refined during data design work, And the complete workflow.

1.3 Definitions, Acronyms and Abbreviations

OOP -Object Oriented Programming

MVC – Model View Control Architecture

2. Architectural Representation

2.1 Scenarios

Audience:

All the stakeholders of the system, including the end-users.

Area:

Describe the set of scenarios and/or use cases that represent some significant, central functionality of the system. Describe the actors and use cases for the system, Other than the basic work flow the documents addresses the exception cases, exception outputs, and other related use cases.

2.2 Logical view

Audience:

Designers, Programmers, Testing staff

Area:

Functional requirements, object hierarchy, system layers

Describes the design of object model. Also describes the subsystems of the system and their relationships.

2.3 Process View

Audience:

Integrators, Programmers

Area:

Non-functional requirements, describe the design's concurrency and synchronization aspects. Elaborates the run time behavior of the system.

2.4 Development View

Audience:

Programmers, Code testers

Area:

Software components: describe the modules and subsystem divisions of the system.

2.5 Physical View

Audience:

Database administrators, System engineers, Deployment managers

Area:

Persistence: describes the architecturally significant persistent elements in the data model. Describe the mapping of the software onto the hardware and shows the system's distributed aspects.

3. Architectural Goals and Constraints

3.1 Server side

This system will be hosted at “**Heroku**” JSP server. Being a web-based application, this underlying client OS can be any PC operating system. All communication with the client has to comply with public HTTPS, TCP/IP communication Protocol standards.

3.2 Client side

Users will be able to access this system only online. Client/Users are expected to use a modern web browser such as Google Chrome or Safari to get full user experience.

3.3 Development Tools

The project incorporates many development tools.

Programming: VS Code, Adobe Dreamviewer.

Diagrams: Draw.IO, CREATLY, smartdraw.

Deployment: Amazon Web Service (AWS) EC2

4. Use Case View

4.1 Use case diagrams

