11/26/2024

Sandra Fernandes Fernandes

COIL TEAM

PredictEase

Documentation

**Table of Contents**

[**1. Project Overview** 3](#_Toc184203239)

[1.1 Key features of the application 3](#_Toc184203240)

[1.2 Technological Integration 3](#_Toc184203241)

[**2. Purpose and Goals** 4](#_Toc184203242)

[2.1 Purpose 4](#_Toc184203243)

[2.2 Goals 4](#_Toc184203244)

[2.2.1 Empower Target Audience to Perform Predictive Analysis 4](#_Toc184203245)

[2.2.2 Highlight AI and Machine Learning Applications 5](#_Toc184203246)

[2.2.3 Technologies Powering PredictEase 5](#_Toc184203247)

[2.2.4 Enhance Learning Through Global Collaboration 6](#_Toc184203248)

[2.2.5 Deliver a Scalable Solution 6](#_Toc184203249)

[**3. Agile Development Approach** 6](#_Toc184203250)

[3.1 Development Phases 7](#_Toc184203251)

[**4. AI and Machine Learning Overview** 8](#_Toc184203252)

[4.2 Machine Learning in PredictEase 8](#_Toc184203253)

[4.3 Machine Learning Concepts 8](#_Toc184203254)

[**5. Linear Regression Overview** 9](#_Toc184203255)

[5.1 Data Handling and Regression Modeling 9](#_Toc184203256)

[**6. Installing and Setting Up PredictEase** 10](#_Toc184203257)

[Installing Python 10](#_Toc184203258)

[Using an Integrated Development Environment (IDE) 11](#_Toc184203259)

[**7. Developer Insights: Architecture, Roles, and Considerations** 12](#_Toc184203260)

[7.1 Team Members 12](#_Toc184203261)

[**8. GitHub Repository Information** 14](#_Toc184203262)

[**9. Contact Information** 14](#_Toc184203263)

[**10. Use Case Example** 14](#_Toc184203264)

[**Glossary Terms** 15](#_Toc184203265)

# **1. Project Overview**

As part of the COIL (Collaborative Online International Learning) Project, teams from Seneca Polytechnic (Toronto, Canada) and Universidad da Coruña (A Coruña, Spain) developed the **‘PredictEase’**. Designed to simplify simple analysis, the [PredictEase](#_Glossary_Terms) provides a user-friendly Python-based platform for data analysis, visualization, and managing [regression models](#_Glossary_Terms).

The application offers seamless functionality, allowing users to create, save, load, and interpret regression models. The team adhered to best coding practices, ensuring the application is scalable and maintainable, making it an effective tool for both educational purposes and real-world applications. Targeting a diverse audience, it bridges the gap between technical complexity and accessible predictive modelling.

This project combines academic objectives with professional software development practices. The team applies agile methodologies and fosters international collaboration, gaining valuable hands-on experience in cross-border, team-based development.

## 1.1 Key features of the application

* The application imports data from [**CSV**](#_Glossary_Terms), **Excel**, and [**SQLite**](#_Glossary_Terms) databases.
* It enables users to build **simple** [linear regression](#_Glossary_Terms) models.
* Users can **save** and **load** models for future use with its model management feature.
* The application provides an **intuitive interface**, catering to technical and non-technical users.

## 1.2 Technological Integration

The collaboration integrates advanced technologies, including artificial intelligence (**AI**) and machine learning (**ML**), to enhance regression analysis. It includes features such as error analysis and model accuracy metrics to improve performance and reliability.

For example, Imagine teaching a computer how to recognize handwritten numbers. You show it thousands of examples, and over time, it learns how to identify new, unseen numbers. That's how [machine learning](#_Glossary_Terms) works.

# **2. Purpose and Goals**

## 2.1 Purpose

The key purposes of the project are:

* Develop a **user-friendly** application to perform simple linear regression analysis.
* Support **multiple data formats**, including CSV, Excel, and SQLite.
* Enable dataset handling, including **loading**, **preprocessing**, and **analyzing data**.
* Incorporate data **preprocessing capabilities** such as handling missing data and applying mean or median values.
* Provide **clear visualizations** of regression results with an intuitive interface.
* Allow users to **save and reload models** for future predictions.
* Ensure [**cross-platform compatibility**](#_Glossary_Terms) for Windows, macOS, and [Linux](#_Glossary_Terms).
* Practice **agile software development** principles to foster iterative progress and team collaboration.
* Maintain **comprehensive documentation** to support developers and end-users.

## 2.2 Goals

### 2.2.1 Empower Target Audience to Perform Predictive Analysis

It serves a diverse audience, including:

* **Data Analysts and Researchers** can simplify dataset analysis and extract actionable insights.
* **Educators and Students** can teach and learn regression modelling techniques with a user-friendly interface.
* **Developers** can contribute to a modular, well-documented, and community-driven open-source project.
* **Organizations** can use a lightweight tool for regression analysis without the cost or complexity of commercial software.
* **Business Managers** can use regression models to make data-driven decisions and forecast trends.
* **Small and Medium Enterprises (SMEs)** can leverage a cost-effective solution for basic predictive analytics without needing specialized staff.
* **Freelance Analysts** can access a versatile and portable tool to handle client projects efficiently.
* **Data Science Enthusiasts** can practice building and interpreting regression models as part of their learning joney.
* **Policy Makers and Government Analysts** can evaluate data trends to inform policy decisions and resource allocation.
* **Nonprofits and Community Organizations** can analyze data to measure the impact of initiatives and improve program effectiveness.

### 2.2.2 Highlight AI and Machine Learning Applications

* Demonstrate how artificial intelligence and data-based prediction techniques enhance problem-solving.
* Apply linear regression models to address real-world challenges effectively.
* Simplify AI concepts to increase understanding and usability for all users.
* Showcase the potential of machine learning in predictive analytics and decision-making.

### 2.2.3 Technologies Powering PredictEase

We designed this project for broad compatibility and use robust technologies to ensure efficiency and reliability:

* **Supported Platforms**: Windows, macOS, and Linux.
* **Programming Language**: Python 3. x.

**Key Libraries**:

* + [Pandas](#_Glossary_Terms): Handles data manipulation.
  + [Scikit-learn](#_Glossary_Terms): Builds regression models.
  + [Matplotlib](#_Glossary_Terms): Creates data visualizations.
  + [SQLite3](#_Glossary_Terms): Manages databases.
  + [PyQt5](#_Glossary_Terms): Develops the graphical user interface (GUI).
  + [Joblib](#_Glossary_Terms): Saves and loads regression models.

**Development Tools**:

* + [**GitHub**](https://www.bing.com/ck/a?!&&p=3f1c7c0afcc5fb0f03a9015593f0a6e03e4b709f0952a0206123ccce8aa227bfJmltdHM9MTczMjU3OTIwMA&ptn=3&ver=2&hsh=4&fclid=08fb7364-6e62-6184-3637-67546fd06015&psq=github&u=a1aHR0cHM6Ly9naXRodWIuY29tLw&ntb=1): Manages version control and team collaboration.
  + [**Visual Studio Code**](#_Glossary_Terms): Facilitates coding and debugging.
  + [**Taiga**](https://taiga.io/): Tracks tasks and manages Scrum workflows.

### 2.2.4 Enhance Learning Through Global Collaboration

* Promote international teamwork by fostering collaboration among diverse team members.
* Provide hands-on experience in software development, technical writing, and content strategy.
* Enable team members to apply theoretical knowledge to practical, real-world scenarios.
* Encourage skill-building through experiential learning in a collaborative, global environment.

### 2.2.5 Deliver a Scalable Solution

* Design the application for scalability to support future integration of advanced data-based prediction models.
* Simplify predictive analytics for non-expert users, making the tool accessible and impactful.
* Create a functional application that addresses real-world needs while promoting global teamwork and skill development.

# **3. Agile Development Approach**

The [Agile methodology](#_Glossary_Terms) drives the development of the Linear Regression Application by promoting flexibility, collaboration, and continuous improvement. The project follows these key principles:

**Iterative Development**: The project is divided into one-week sprints, each focused on different features. This allows the team to deliver functional components quickly and adjust based on feedback.

**Real-time Feedback**: The team reviews progress and refines features during each sprint, ensuring continuous alignment with project goals.

**Cross-Country Collaboration**: Agile fosters seamless communication among cross-border teams, ensuring effective coordination despite geographic distances.

**Clear Communication**: Regular updates and retrospectives help the team address challenges, improve processes, and keep stakeholders informed throughout the development cycle.

## 3.1 Development Phases

**Sprint Duration and Schedule:**

* **Duration**: Each sprint lasts one week, with sessions held for approximately 45 minutes.
* **Number of Sprints**: The project includes seven sprints, spanning from October to November 2024.
* **Targets:** Each sprint targets specific features or deliverables, such as GUI development, functionality implementation, and documentation.

**Sprint Meetings:**

* **Planning**: The team assigns tasks based on priority and scope at the start of each sprint using [Taiga](https://taiga.io/), an agile project management tool.
* **Check-ins**: Daily check-ins ensure the team is up-to-date and provide a platform for resolving blockers or addressing questions immediately.
* **Reviews**: The team demonstrates completed work to the Collaborators and gathers valuable feedback.
* **Retrospectives**: The team reflects on successes, challenges, and areas for improvement at the end of each sprint.

**Collaboration Tools**

* **Taiga:** Used for task management, backlog tracking, and sprint planning.
* **GitHub:** Manages version control and repository updates.
* **Teams/WhatsApp:** Facilitates easy communication among team members.
* **Zoom:** Used for Sprint Retrospectives and team meetings.

This structure ensures efficient communication, task management, and continuous feedback, all essential for successful project delivery.

# **4. AI and Machine Learning Overview**

[**Artificial Intelligence (AI)**](#_Glossary_Terms) **and Machine Learning (ML)**: AI systems simulate human intelligence, while ML, a subset of AI, enables systems to learn from data, improve over time, and make predictions.

**Application in the Project**: This project uses linear regression, a data-based prediction technique, to build predictive models based on historical data.

## 4.2 Machine Learning in PredictEase

PredictEase incorporates core data-based prediction principles to simplify predictive modelling with linear regression. Here’s how it works:

* **Automates Data Preparation**: The application automatically handles tasks like cleaning missing values, saving users time on data preprocessing.
* **Builds Predictive Models**: Users can input data (e.g., sales figures and marketing spending) to create models that predict future outcomes.
* **Evaluates Model Performance**: ‘PredictEase’ provides key performance metrics, such as Mean Squared Error (MSE) and [R²](#_Glossary_Terms), to assess the reliability and accuracy of predictions, boosting confidence in decision-making.

Although not an advanced AI tool, ‘PredictEase’ leverages data-based prediction to make predictive analysis accessible to users with varying technical expertise.

## 4.3 Machine Learning Concepts

* **Training Data**: The dataset used to train the model and identify patterns.
* **Model**: The algorithm that analyzes relationships in the data and generates predictions.
* **Prediction**: The model’s output, based on new input data, is used for decision-making.

This streamlined approach ensures clarity and ease of understanding for users of varying technical expertise.

# **5. Linear Regression Overview**

Linear regression models the relationship between a dependent variable and one or more independent variables. It is a core technique in predictive analytics and data-based prediction, commonly used for making data-driven predictions.

* **Purpose:** Predicts outcomes by analyzing the relationship between variables.
* **Application:** Widely used in both predictive analytics and data-based prediction for tasks such as forecasting and trend analysis.

## 5.1 Data Handling and Regression Modeling

**Data Loading and Preprocessing**:

* + Users upload datasets in **CSV**, **Excel**, or **SQLite** formats.
  + The application preprocesses data using **mean or median imputation** to address missing values, ensuring data readiness for regression analysis.
  + This automation simplifies data preparation, a crucial step in any machine learning workflow.

**Linear Regression Modeling**:

The application builds linear regression models using Scikit-learn (Scikit-learn is a Python library that simplifies machine learning by providing tools for data analysis and building predictive models.), a popular machine-learning library (ML library). It simplifies the model creation process, enabling users to generate predictive models.

**Model Evaluation**:

* + The application calculates performance metrics, such as **R²** (coefficient of determination) and [**MSE**](#_Glossary_Terms) (Mean Squared Error), to evaluate model accuracy.
  + These metrics provide insights into the model's effectiveness in predicting the dependent variable based on the independent variables.

**Prediction Capabilities**:

* + Users can generate predictions using newly created and previously saved regression models.
  + For Instance, Users can predict future sales by analyzing historical data, including factors such as advertising spend and seasonal trends. The application enables users to apply linear regression models to understand key factors influencing sales performance and make data-driven forecasts.

**Model Saving and Reloading**:

* + The application saves models using **Joblib**, allowing users to reload and reuse them for future predictions or analysis.

|  |
| --- |
| Note: This section outlines the key aspects of the ‘PredictEase’ project, focusing on its programming architecture, team roles, and development considerations. |

# **6. Installing and Setting Up PredictEase**

Install and configure PredictEase to run the Linear Regression model. This includes installing Python, setting up an IDE (optional), and installing necessary Python libraries.

## Installing Python

To run the PredictEase using Python ensure to follow the steps:

1. Download Python from the [official Python website](https://www.python.org/).
2. Open the downloaded Python installer.
3. Choose **Custom Installation** in the setup wizard.
4. Check the following options:
   * Add Python to environment variables
   * Install pip
5. Select **Install** to complete the setup.

Python is installed and ready to use.

## Using an Integrated Development Environment (IDE)

Although not required, an IDE like Visual Studio Code (VS Code) can simplify development with helpful features such as syntax highlighting, debugging tools, and project organization.

**Setting up VS Code for Python**:

1. Restart VS Code after installing Python.
2. Install the Python extension in VS Code:
   * Select **Extensions** from the left navigation bar.
   * Search for **ms-python.python** and install the extension.

VS Code is configured for Python development.

**Installing Python Libraries**

Install the required libraries for PredictEase. Use a terminal, such as:

* Command Prompt (Windows)
* PowerShell (Windows)
* Terminal (macOS)
* VS Code Terminal

**Steps**:

1. Open your terminal and navigate to the **PROJECT-GROUP-8** folder.
2. Run the following commands one by one, pressing Enter after each:

Copy code

* pip install pandas (in the terminal)
* 'pip install scikit-learn (in the terminal)

All required libraries are installed.

1. And then execute the main.py file.

The graphical interface is now ready to be used.

# **7. Developer Insights: Architecture, Roles, and Considerations**

The ‘**PredictEase’** project adopts a clear and structured approach to ensure scalability, maintainability, and user-friendliness. Below are the key aspects of the development process:

**Programming Architecture:** [**Model-View-Controller**](#_Glossary_Terms) **(MVC)**  
The development team used the MVC **Model-View-Controller (MVC)** architecture to organize the code into three layers, ensuring modularity[Glossary Terms](#_Glossary_Terms) and ease of maintenance:

* **Model**: Handles data management, including data loading and processing with **Pandas** and creating regression models using **Scikit-learn**.
* **View**: Manages the graphical user interface (GUI) built with **PyQt5**, incorporating tables, buttons, charts, and plots.
* **Controller**: Manages user interactions, linking the **Model** and **View**, processing user inputs, and navigating between interface steps.

**Selection of Library**   
The team selected reliable and robust libraries to enhance functionality:

* + **Pandas**: For data manipulation and processing.
  + **Scikit-learn**: For building regression models.
  + **PyQt5**: For developing the graphical interface.

**Future Developments**   
The project’s modular structure allows for easy future upgrades, such as the addition of new regression models or data preprocessing tools.

This architecture ensures the application remains flexible and scalable for future features and enhancements.

## 7.1 Team Members

**Sandra Fernandes (Seneca Polytechnic)**Role: Technical Writer

* Responsible for creating and maintaining project documentation, including installation guides, user manuals, and GitHub updates.
* Collaboration with developers to ensure documentation aligns with project features and updates.
* Responsible for maintaining up-to-date documentation in GitHub.

**Alvaro Carpio (Universidade da Coruña)** Role: Developer

* Focused on backend functionalities, including data validation and regression algorithms.
* Contributed to interface improvements, including enhancing the user experience ([UX](#_Glossary_Terms)) and user interface ([UI](#_Glossary_Terms)) design.
* Responsible for organizing the codebase and improving its modularity and maintainability.

**Alba González Peña** **(Universidade da Coruña)** Role: Code Structure Specialist

* Responsible for organizing the codebase and improving its modularity and maintainability.
* Focused on backend functionalities, including data validation and regression algorithms.
* Contributed to interface improvements, including enhancing the user experience (UX) and user interface (UI) design.

**Irene Tembrás Díaz (Universidade da Coruña)** Role: Library Researcher

**‘PredictEase’ Application GitHub Documentation** Number: EFGH-529 Version: 1.0

* Conducted research to identify and select appropriate ML libraries for the project.
* Implemented prediction functionalities based on the chosen libraries.
* Contributed to interface improvements and code structuring
* Assisted with project documentation, ensuring clarity and consistency.

**Nawfal Heiloua (Universidade da Coruña)** Role: Developer and Scrum Leader

* Facilitated team meetings, sprint planning, and retrospectives to maintain alignment with Agile methodologies.
* Focused on backend functionalities, including data validation and regression algorithms.
* Assisted with project documentation, ensuring clarity and consistency.

# **8. GitHub Repository Information**

* **Repository Name**: PredictEase Application
* **Branching**: The repository follows a branching strategy with a main branch for stable releases and feature branches for ongoing development.

# **9. Contact Information**

For questions or contributions, please contact:

* **Project Manager**: alberto.alvarellos@udc.es
* **Developer Team**: [nawfal.heiloua@udc.es](mailto:nawfal.heiloua@udc.es), [alvaro.carpio@udc.es](mailto:alvaro.carpio@udc.es), [alba.gonzalez.pena@udc.es](mailto:alba.gonzalez.pena@udc.es), [irene.tembras@udc.es](mailto:irene.tembras@udc.es)
* **Documentation Team**: snfernandes@myseneca.ca

# **10. Use Case Example**

* **Scenario**: A data analyst is tasked with predicting the sales of a product based on historical data. By using the Linear Regression Application, the analyst imports a dataset containing product sales, advertising costs, and market trends. The analyst creates a linear regression model to predict future sales and evaluates the model's accuracy using MSE and R². The predictions generated are used for decision-making in marketing strategies.

# **Glossary Terms**

**Technical Terms and Definitions**

* **PredictEase**

A Python-based platform designed for regression modeling and data analysis.

* **Regression Models**

Models that predict outcomes by analyzing relationships between dependent and independent variables in datasets.

* **Linear Regression**

A method to model the relationship between a dependent variable and one or more independent variables using a straight line.

* **Machine Learning (ML)**

A type of artificial intelligence (AI) that enables systems to learn from data and improve over time without explicit programming.

* **Artificial Intelligence (AI)**

The simulation of human intelligence in machines that can learn, reason, and make decisions.

* **CSV (Comma Separated Values)**

A file format for storing tabular data where each row is separated by commas.

* **SQLite**

A lightweight, serverless database that stores data in files and uses a variant of SQL for data manipulation.

* **Python 3.x**

A widely used programming language for data analysis, machine learning, and web development.

* **Pandas**

A Python library that provides data structures for manipulating and analyzing large datasets.

* **Scikit-learn**

A Python library for machine learning that offers tools for data mining and analysis, built on NumPy, SciPy, and matplotlib.

* **Matplotlib**

A Python library is used for creating static, animated, and interactive data visualizations.

* **PyQt5**

A Python binding for Qt that enables the development of cross-platform graphical user interfaces (GUIs).

* **Joblib**

A Python library for efficiently saving and loading large machine learning models and other objects.

* **Model-View-Controller (MVC)**

A software design pattern that separates an application into:

* + - **Model**: Data and business logic
    - **View**: User interface (UI)
    - **Controller**: Manages user input and updates the model and view.
* **Mean Squared Error (MSE)**

A measure of model accuracy, calculating the average squared difference between actual and predicted outcomes.

* **R² (Coefficient of Determination)**

A statistical metric that indicates how well independent variables predict the dependent variable, ranging from 0 (poor) to 1 (perfect).

* **GitHub**

A platform for version control and collaboration, using Git to manage code changes.

* **Visual Studio Code**

A free, open-source code editor that supports multiple programming languages and integrates with version control systems.

* **Taiga**

A project management tool for agile development that helps teams organize tasks and track progress.

* **Agile Development**

A flexible software development methodology that focuses on iterative progress, collaboration, and rapid delivery of functional software.

* **Sprint**

A fixed period (usually 1-4 weeks) where a development team works on specific tasks or features, is a key aspect of Agile development.

* **Iteration**

A cycle in Agile development that involves repeating processes to improve or add features in the project.

* **Cross-Platform Compatibility**

The ability of software to run on different operating systems (Windows, macOS, Linux) without modification.

* **Modularity**

The design approach of breaking a system into smaller, self-contained components for easier development and maintenance.

* **Preprocessing**

The process of cleaning, transforming, and organizing data before analysis, including handling missing values and normalizing data.

* **Back-End Functionality**

The server-side logic, database management, and core processing of an application, which users interact with indirectly.

* **User Interface (UI)**

The visual elements (buttons, text fields) through which users interact with the application.

* **User Experience (UX)**

The overall satisfaction and ease of use users experience when interacting with the application, including design and functionality.