Sandra Fernandes Fernandes

Abstract

Welcome to PredictEase, a user-friendly application designed to make data analysis and regression modeling simple and accessible. Created through an international collaboration between teams from Seneca Polytechnic in Toronto, Canada, and Universidade da Coruña in Spain, this tool is part of the COIL (Collaborative Online International Learning) Project.

PredictEase

Readme File

[Introduction 3](#_Toc183984988)

[About this User Guide 3](#_Toc183984989)

[Getting Started 3](#_Toc183984990)

[Project Features 4](#_Toc183984991)

[Hardware/System Requirements 4](#_Toc183984992)

[Minimum System Requirements 4](#_Toc183984993)

[Recommended System Requirements 5](#_Toc183984994)

[Installing Python 5](#_Toc183984995)

[Integrated Development Environment (IDE) 6](#_Toc183984996)

[Installing Python Libraries 6](#_Toc183984997)

[User Interface 7](#_Toc183984998)

[UI Elements in PredictEase 7](#_Toc183984999)

# Introduction

Welcome to “**PredictEase”**, a user-friendly application designed to make data analysis and regression modelling simple and accessible. Created through an international collaboration between teams from Seneca Polytechnic in Toronto, Canada, and Universidade da Coruña in Spain, this tool is part of the COIL (Collaborative Online International Learning) Project.

PredictEase lets you import data, analyze it, visualize results, and manage regression models—all on an intuitive, **Python-powered** platform. Whether you're a beginner or an experienced analyst, PredictEase bridges the gap between complex technical processes and straightforward predictive modelling, making data-driven decisions easier than ever

### About this User Guide

This user guide helps you understand and use the Linear Progression Software, designed for creating and analyzing predictive models easily. Whether you’re a business manager, data analyst, or general user, this guide will walk you through setting up, using, and maximising the software’s features.

# Getting Started

# Project Features

|  |  |
| --- | --- |
| **Features** | **Description** |
| **Data Import and Management** | * Import data from multiple formats, including CSV, Excel, and SQLite. * Select and manage variables to tailor your model to specific needs. |
| **Model Creation and Customization** | * Create simple linear regression models to identify relationships in data. * Adjust model parameters as needed for tailored results. |
| **AI-Enhanced Predictions** | * Use AI to refine model accuracy and make predictions that adapt to new data inputs. * Enable automatic updates to model predictions with AI-driven improvements. |
| **Interactive Data Visualization** | * View data trends and model results through easy-to-interpret charts and graphs. * Explore visualizations with interactive features for better insights and data storytelling. |
| **Model Management** | * Save and load models for ongoing analysis or future reference. * Export model reports in PDF or Excel formats for easy sharing and presentation. |
| **User-Friendly Interface** | * Simple, intuitive design for easy navigation. * Step-by-step guidance through model creation and analysis. |
| **In-App Support and Resources** |  |

## Hardware/System Requirements

To use PredictEase effectively, ensure your system meets the following hardware and software requirements:

### Minimum System Requirements

* **Operating System**: Windows 10 (64-bit), macOS 11.0 or later, or Linux (Ubuntu 20.04 or equivalent)
* **Processor**: Dual-Core CPU (2.0 GHz or higher)
* **RAM**: 4 GB
* **Storage**: 500 MB of free disk space
* **Graphics**: Integrated graphics capable of displaying 1280 x 720 resolution

**Software**:

* Python 3.9 or higher
* Required Python libraries (install via requirements.txt)

### Recommended System Requirements

* **Operating System**: Windows 11, macOS 12.0 (Monterey) or later, or Linux (Ubuntu 22.04 or equivalent)
* **Processor**: Quad-Core CPU (2.5 GHz or higher)
* **RAM**: 8 GB or more
* **Storage**: 1 GB of free disk space
* **Graphics**: Dedicated GPU with OpenGL 3.3 support or higher

**Software**:

* + Python 3.10 or higher
  + Required Python libraries with the latest updates

Additional Notes

* **Internet Connection**: Required for downloading dependencies, updates, and certain online data integrations.
* **Supported File Formats**: CSV, Excel (.xlsx), SQLite databases.
* **Environment**: A stable and secure computing environment is recommended to avoid data loss or application interruptions.

For installation guidance and troubleshooting, refer to the **Installation Guide** section of the user documentation.

## Installing Python

To run the Linear Regression model, you need Python since the software is developed in this language. Follow these steps to install Python:

1. Download Python from the official [Python website](https://www.python.org/downloads/).
2. Open the Python installer.
3. Select **"Custom installation"** during the setup.
4. Ensure the following options are checked:
   * **Add Python to environment variables**
   * **Install pip**
5. Complete the installation by clicking **"Install"**.

After installation, you’ll be ready to run the Linear Regression model.

**Downloading and Installing the Model**

### Integrated Development Environment (IDE)

You don’t need an Integrated Development Environment (IDE) to run the Linear Regression model but, using one can make development easier. IDEs like Visual Studio Code (VS Code) offer useful features such as:

* **Code highlighting and suggestions**: IDEs help you write clean and error-free code by offering syntax highlighting and autocomplete suggestions.
* **Debugging tools**: You can easily find and fix errors with integrated debugging features.
* **Project organization**: IDEs help manage and organize files in complex projects.

If you decide to use VS Code, here are some tips for running Python programs:

1. After installing Python, restart VS Code to ensure it recognizes the Python installation.
2. Install the Python Extension in VS Code:
   * Select **Extensions** on the left navigation bar.
   * In the search bar, type **ms-python.python** and press **Enter** to locate and install the extension.

This setup will make it easier to work with Python in VS Code.

### Installing Python Libraries

To complete this section, you’ll need to use a terminal. Some options include:

* **Command Prompt** (Windows)
* **PowerShell** (Windows)
* **Terminal** (MacOS)
* **Visual Studio Code Terminal**

Follow these steps to install the required libraries:

1. Open your terminal and navigate to the Linear Regression project folder.
2. Type each command line below, pressing **Enter** after each one to install the library.
3. Repeat step 2 until all libraries are installed.

# User Interface

## UI Elements in PredictEase

Main Window (root):

* + Name: root (The main window of the app)

Input Selector (self.input\_selector):

* + Name: input\_selector
  + Function: Allows users to select the input data.

Output Selector (self.output\_selector):

* + Name: output\_selector
  + Function: Allows users to select the output data.

Text Area (self.dtext):

* + Name: dtext
  + Function: A text area where results or logs are displayed.

Results Table (self.results\_table):

* + Name: results\_table
  + Function: Displays results in a table format.

Prediction Button (self.make\_prediction\_button):

* + Name: make\_prediction\_button
  + Function: Button to trigger prediction functionality.

Clear Results Button (self.clear\_button):

* + Name: clear\_button
  + Function: Button to clear the results or reset the data.

Clear Graph Button (self.clear\_graph\_button):

* + Name: clear\_graph\_button
  + Function: Button to clear the graph display.

File Selection Button (self.file\_button):

* + Name: file\_button
  + Function: Opens the file dialog to load data.

Graph Frame (self.graph\_frame):

* + Name: graph\_frame
  + Function: A frame used to hold the graphical elements (e.g., charts, plots).

Prediction Result Label (self.prediction\_result\_label\_loaded):

* + Name: prediction\_result\_label\_loaded
  + Function: Displays the result of the prediction.

File Dialog (file\_dialog):

* + Name: file\_dialog
  + Function: Opens a file dialog for selecting input data files.

Messagebox (self.show\_error, self.show\_warning, etc.):

* + Name: show\_error, show\_warning
  + Function: Display error or warning messages.

Input Value Dialog (self.input\_value):

* + Name: input\_value
  + Function: Prompts the user to enter a numeric value for prediction.

Input Validation Message (messagebox.showerror, etc.):

* + Name: messagebox.showerror

Function: Show error messages when input values are invalid.

**Overview of User Interface Components**

**Beginning Interface**

The user Interface contains three parts:

* The beginning interface (welcome screen): Showing the information about the requirements of our software’s file formats and importing files.
* Importing a file main interface (preprocessing data screen): A spreadsheet screen with buttons and functionalities to adjust and collect data before predicting.
* Result Interface

**Importing File Main Interface**: Spreadsheet and data adjustment functionalities

**Results Interface**

**4. Functions and Usage**

* **Overview**
  + Key functionalities and execution process
* **Main Stages**

**Importing Files**

In this section, you upload the database file necessary to work on the following stages. You can import a file from these two interfaces:

**Interface options for file imports**

An initial interface opens when you first run the software. (Figure)

A second interface with a built-in section in the main interface for importing a file with the same functionality. (Figure)

**Steps for importing a file and selecting an output column**

To add your file, create a linear regression output

1. Click **Open File Explorer** in the initial interface(figure).

The select File dialog window opens.

1. Navigate to the folder you need to import the file button.
2. Click **Open** to import the file.

The spreadsheet appears in the input variable interface(figure)

1. Select the output column.

Note: you can only select one output.

1. Select one of the options: “Replace with a number”, “Replace with mean”, replace with median”, or “Remove row”.
2. Click Generate model.

A pop-up with the message “Linear regression model has been generated and plotted” appears.

1. Click OK.

Figure 2 shows that the linear regression, along with the selected data option, appears.

**File formats supported**

The software can import from local storage various database file formats:

* CSV
* Microsoft Excel
* SQLite
* .db
* .xls

**Preprocessing Data**

**Spreadsheet Section**

This section contains all the granular data from the imported file. The interface shows the information in cells, columns and rows with specific values within them. Values can be textual or numerical. (Figure)

**Variables and Target Selection**

This section comprises of two parts

Input variable selection

The variables selection section – allows you to select the columns from the imported file for the input variable(s) to predict the results.

Target selection for output values

Target selection – allows you to choose the columns from the imported file for the output value. The software will display the output value in this column after processing the input variables(s).

**Missing Data Handling**

This section is a drop-down list that contains options to handle empty cells with no value, such as “n/a”, “na”, “non” and “nan” The software offers different options.

|  |  |
| --- | --- |
| Options | Functions |
| Remove rows | To delete the entire row that contains the empty cell(s). |
| Fill with mean | To fill all empty cells with the mean of the remaining values of the rows. |
| Fill with median | To fill all empty cells with the median of the remaining cells’ value of the row. |
| Fill with a constant value | To fill all empty cells throughout the row with specific values. You must manually enter the constant value. |

Table : Options in the Handle missing Data drop-down list.

**Getting Results**

* + - Steps to generate and display the linear regression model

**5. Additional Information**

* **Support and Reporting Issues**
  + Contact information for the technical writing team
* **Acknowledgments**
  + Developers
  + Authors
* **Contribution Guidelines**
  + Instructions for contributions through GitHub
* **Licenses**