



Summary gUIDE

STUDENT\_PREDICTIVE\_GRADES.PY CODE



**Overview**

This Python script is designed to perform predictive grade outcomes using a graphical user interface (GUI) built with Tkinter. It allows users to load a dataset, train a machine learning model, and make predictions.

This script provides a simple yet effective way to perform predictive grade calculations with a user-friendly interface.

**Key Components**

1. **Libraries Imported**:
   * pandas for data manipulation.
   * tkinter for creating the GUI.
   * sklearn for machine learning functionalities.
2. **Functions**:
   * **load dataset()**: Opens a file dialog to load a CSV or Excel file into a pandas DataFrame.
   * **train\_model(df, features, target)**: Trains a RandomForestClassifier on the specified features and target from the DataFrame. It splits the data into training and testing sets, trains the model, and displays the accuracy.
   * **make\_predictions(model, df, features)**: Uses the trained model to make predictions on new data and displays the results in the GUI.
3. **GUI Elements**:
   * Buttons to load the dataset, train the model, and make predictions.
   * Entry fields for specifying features and target columns.
   * A text box to display prediction results.

**Workflow**

1. **Load Dataset**: The user loads a dataset through a file dialog.
2. **Train Model**: The user specifies the features and target columns, then trains the model. The accuracy of the model is displayed.
3. **Make Predictions**: The user can make predictions on new data using the trained model, and the results are shown in the text box.

**Summary**

When the script has been fully tested it should be a complete and functional implementation for predictive modelling using a GUI. The student\_predictive\_grades.py script is an unfinished version that lacks some critical components and functionality. The main problems are detailed below.

1. Global Variables for Dataset and Model
2. Label Encoding for Categorical Features
3. Integration of Global Variables in GUI Commands
4. Error Handling for Missing Dataset and Model
5. Formatting Predictions for Better Readability

**GUI Support**

In the context of the student\_predictive\_grades.py script, the **features** and **target** are key components used for training the machine learning model:

**Features**

* **Features** are the input variables that the model uses to make predictions. These are typically columns in your dataset that contain relevant information for the prediction task.
* In the script, features are specified by the user through a GUI entry field where they can input the names of the columns (comma-separated) that should be used as features.

**Target**

* **Target** is the output variable that the model is trying to predict. This is the column in your dataset that contains the labels or values you want to predict.
* In the script, the target is also specified by the user through a GUI entry field where they can input the name of the column that should be used as the target.

**Example**

If you have a dataset for predictive maintenance with columns like temperature, vibration, pressure, and failure, you might choose:

* **Features**: temperature, vibration, pressure
* **Target**: failure

The model will use the values in the temperature, vibration, and pressure columns to predict the failure column.