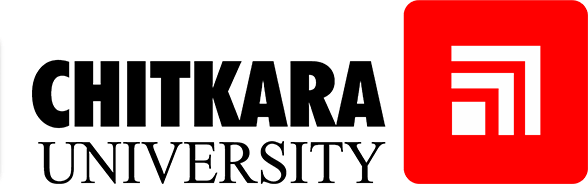
**Artificial Intelligence and Machine Learning**

Project Report Semester-IV (Batch-2022)

**Human Activity Recognition using Smartphone  Dataset**

**ML project**



**Supervised By:**Mrs. Rishu Taneja  Roll No. :  2210990858

**Submitted By:** Sneha Miglani Group: G-13

**Department of Computer Science and Engineering**

**Chitkara University Institute of Engineering & Technology, Chitkara University, Punjab**

**ABSTRACT**

In this work, Artificial Intelligence and Machine Learning (AIML) techniques are applied to Human Activity Recognition (HAR) using a dataset from smartphones. The techniques are implemented in Python. The dataset includes gyroscope and accelerometer readings taken from smartphones used for a variety of purposes. Our goal is to create an effective AIML model that uses sensor data to accurately classify activities like running, walking, sitting, and standing.

For the purposes of feature extraction, data preprocessing, and model implementation, we make use of Python libraries and frameworks. To extract pertinent features from the unprocessed sensor data, feature engineering techniques are applied. Python libraries like scikit-learn and TensorFlow are used to implement supervised learning algorithms like Support Vector Machines, Random Forests, and neural network architectures like Convolutional Neural Networks for training the HAR mode. utilized to assess.

The model's performance is assessed using cross-validation techniques and measures such as accuracy, precision, recall, and F1-score. The outcomes demonstrate how well the AIML method works to reliably identify human actions from smartphone sensor data.

By showcasing the effectiveness of Python-based AIML techniques in processing and analyzing smartphone sensor data for activity recognition, this study advances the field of HAR research. This research shows how AIML can be used to improve human-centric technologies, with applications in context-aware computing, fitness tracking, and healthcare among the practical implications.

**Table of Contents**

|  |  |  |
| --- | --- | --- |
| Sr.no | Section | Page No |
| 1. | Introduction | 4 |
| 2. | Problem Statement | 5 |
| 3. | Technical Details | 7 |
| 4. | File Structure | 8 |
| 5. | Result | 9 |
| 6. | References | 28 |

**INTRODUCTION**

1. **BACKGROUND**

With the increasing availability of smartphones equipped with sophisticated sensors such as accelerometers, gyroscopes, and magnetometers, human activity recognition, or HAR, has garnered significant attention. These gadgets provide ongoing behavioral tracking of people without the need for specialized hardware or wearable devices. HAR systems for security surveillance, sports analysis, and health monitoring can all be developed on smartphones. Smartphones are useful for data collection and analysis because of their widespread availability, sophisticated sensor technology, potent processors, and seamless connectivity.

**2.OBJECTIVE**

The goal of this project is to use data from smartphones' built-in sensors to develop a reliable system for identifying human activities. Large datasets will be gathered and preprocessed, feature extraction methods will be investigated, classification models will be put into practice and assessed, and transfer learning tactics will be looked into. The system's ability to be applied to various demographic groups, environmental conditions, and populations will be evaluated. Evaluation will also be given to the solution's possible effects on a range of applications, including safety monitoring, fitness tracking, and healthcare. To guarantee the project's relevance and practical applicability, stakeholders and domain experts will collaborate. The project's objectives are to investigate the significance of human activity recognition, its uses, and potential future paths for this field of study.

**Significance**

* **Healthcare Monitoring**: Accurate activity recognition can facilitate remote monitoring of individuals' health and well-being, particularly for elderly or patients with chronic conditions. It can enable healthcare providers to track patients' activity levels, detect anomalies, and intervene promptly if necessary.
* **Fitness Tracking:** In the realm of fitness and sports, activity recognition can power wearable devices and mobile applications to provide users with insights into their physical activities, helping them set goals, track progress, and make informed decisions about their fitness routines.
* **Safety and Security:** Activity recognition can play a vital role in enhancing safety and security, such as detecting falls or unusual movements in real-time, alerting caregivers or authorities, and ensuring timely assistance.
* **Context-Aware Computing:** By understanding users' activities and contexts, devices and systems can adapt their behavior and provide personalized experiences. For example, smartphones can automatically adjust settings based on users' activities or provide relevant suggestions and reminders.
* **Research and Development:** This project contributes to advancing research in machine learning, signal processing, and sensor technologies. It fosters innovation in activity recognition algorithms, data analysis techniques, and deployment strategies, leading to improvements in various applications and domains.

**Problem Statement**

Using data gathered from smartphone sensors, the problem statement for human activity recognition using smartphone dataset in Python attempts to create a machine learning model that precisely categorizes human activities. The dataset usually consists of gyroscope and accelerometer readings from people engaged in different types of activities. The objective is to develop a classification model that, using sensor data, can determine the activity being carried out. In order to classify activities with high accuracy, this entails preprocessing the data, extracting pertinent features, and training a machine learning algorithm. The collection of data, preprocessing, feature extraction, model selection, training, evaluation, optimization, and deployment are the main elements of the problem statement. For the model to be used in practical applications such as context-aware computing, fitness tracking, or healthcare monitoring, it must be strong, accurate, and efficient.

**Software Requirements**

* **Programming Language:** Python - Python is a versatile and widely-used programming language with extensive support for data analysis, machine learning, and web development.
* **Libraries and Frameworks:**

1. **Pandas:**

* + Pandas simplifies data manipulation and analysis with its DataFrame and Series data structures, offering tools for reading/writing data and performing operations like cleaning, filtering, and aggregation.

2. **Seaborn:**

* + Seaborn is a statistical visualization library that builds on matplotlib, providing a high-level interface for creating attractive plots with minimal code. It's particularly useful for exploring relationships in datasets.

3. **Matplotlib.pyplot:**

* + Matplotlib is a comprehensive plotting library, and pyplot is its interface for creating figures and plots. It offers extensive customization options for creating publication-quality visualizations.

4. **Scikit-learn (sklearn):**

* + Scikit-learn is a machine learning library offering tools for data preprocessing, model training, and evaluation. It provides various algorithms and utilities for building and deploying machine learning models.

5. **Joblib:**

* + Joblib is a library for efficient pipelining in Python, useful for saving/loading machine learning models. It simplifies the process of serialization, allowing for easy reuse of trained models.

6. **Tkinter:**

* + Tkinter is the standard GUI toolkit for Python, providing tools for creating desktop applications with graphical interfaces. It enables the creation of windows, buttons, and other GUI elements.

7. **Filedialog and messagebox from tkinter:**

* + These submodules of tkinter offer additional functionality for file handling and displaying messages in GUI applications, enhancing usability and interactivity.
* **Development Environment:** Jupyter Notebook or any Python IDE (e.g., PyCharm, Visual Studio Code) - Jupyter Notebook provides an interactive computing environment for writing and executing Python code, making it ideal for exploratory data analysis and prototyping machine learning models. Python IDEs offer integrated development environments with features like code editing, debugging, and version control integration.

* **Version Control Git** - Git is a distributed version control system that allows multiple developers to collaborate on a project efficiently. It enables tracking changes to the codebase, managing different versions of the project, and facilitating collaboration through features like branching and merging.

**DATASET**

It contains data collected from a group of 30 volunteers who performed various activities while wearing a smartphone(Samsung Galaxy S II) on their waist.

The smartphone was equipped with embedded inertial sensors such as an accelerometer and a gyroscope, which were used to capture the movements and vibrations associated with different activities.

**Training Dataset**: The training dataset comprises a portion of the overall dataset containing labeled sensor data collected from smartphones during various human activities. It consists of input features extracted from sensor readings (such as accelerometer and gyroscope data) and corresponding output labels indicating the activity being performed (e.g., walking, running, sitting, standing). The training dataset is used to train the machine learning model, allowing it to learn the underlying patterns and relationships between sensor data and activity labels.

**Testing Dataset:** The testing dataset is another portion of the overall dataset that is kept separate from the training dataset. Like the training dataset, it contains sensor data from smartphones but with withheld output labels. The testing dataset is used to evaluate the performance of the trained model by providing it with unseen data and assessing its ability to accurately predict activity labels. The predicted labels are compared against the true labels (if available) to measure the model's accuracy, generalization ability, and effectiveness in real-world scenarios.

**Proposed Design/ Methodology**

**Design and Working**

1. **User Interface (UI):**

* Tkinter is the standard GUI toolkit for Python, providing a way to create GUI applications with widgets such as buttons, labels, entry fields, and more.
* It comes pre-installed with Python, making it readily available for creating cross-platform GUI applications.
* Tkinter provides a simple and easy-to-use interface for building GUIs, making it suitable for creating both simple and complex applications.

1. **Data Analysis and Visualization:**

* **Matplotlib:**

Matplotlib is a comprehensive library for creating static, interactive, and animated visualizations in Python.

It provides a wide variety of plotting functions to create line plots, scatter plots, bar charts, histograms, and more.

Matplotlib is highly customizable, allowing users to fine-tune every aspect of their plots, including colors, labels, axes, and annotations.

It is the foundation for many other plotting libraries and tools in Python.

* **Seaborn:**

Seaborn is built on top of Matplotlib and provides a high-level interface for creating attractive and informative statistical graphics.

It simplifies the process of creating complex plots by providing functions for visualizing relationships between variables, including scatter plots, pair plots, violin plots, and heatmap plots.

Seaborn comes with built-in themes and color palettes that improve the aesthetics of plots, making them more visually appealing.

It also offers additional functionalities for statistical analysis, such as automatic estimation and plotting of linear regression models.

In the given project, Matplotlib and Seaborn might be used for visualizing various aspects of the data, such as:

* **Data Exploration:** Before preprocessing, visualizations can help understand the distribution of sensor data, identify outliers, and explore relationships between different features.
* **Feature Analysis:** Visualizing features selected by the feature selection techniques (e.g., SelectKBest, RFE) can provide insights into their importance and relationships with the target variable.
* **Model Evaluation:** After making predictions on the test data, visualizations can be used to assess the performance of the model, such as plotting confusion matrices, ROC curves, or precision-recall curves.
* **Results Visualization:** Visualizing the predicted activities alongside the original sensor data can help interpret the model's predictions and identify any patterns or discrepancies.

1. **Model Training and Testing:**
2. The human activity recognition project employs a Long Short-Term Memory (LSTM) model to predict human activities based on smartphone sensor data.
3. The LSTM model is trained separately in a Jupyter Notebook titled "LSTM\_model.ipynb".
4. Prior to training, the training data is scaled using MinMaxScaler to ensure numerical stability and convergence during the model training phase.
5. Upon completion of training, the LSTM model is saved as an HDF5 file named "LSTM\_model.h5".
6. **Prediction and Evaluation:**
7. Testing data is prepared by combining the past data of a specific duration (e.g., last 100 readings) from the training set with the remaining data.
8. The trained LSTM model is loaded from the saved HDF5 file.
9. Predictions are made using the loaded model on the testing data.
10. After making predictions, the predicted labels are rescaled back to their original values for evaluation and comparison with the actual human activity labels.
11. Predictions versus original activity labels are plotted and displayed using Matplotlib and Streamlit for visual evaluation, allowing for comprehensive analysis of the model's performance.

**Machine Learning Technique**

**Labeled Data:**

In human activity recognition, the labeled dataset consists of various features extracted from smartphone sensors such as accelerometer and gyroscope data. These features include measurements like acceleration, orientation, and rotational velocity, collected over time intervals representing different activities (e.g., walking, running, sitting). Each data point is associated with a specific activity label indicating the human activity being performed during that time period.

**SUPERVISED AND UNSUPERVISED**

**Supervised learning** is a method where an algorithm learns from labeled data, where each example is associated with a label or target variable. The goal is to learn a mapping from input features to the corresponding output labels, with the model adjusting its parameters to minimize the difference between predicted outputs and true labels. Examples of supervised learning tasks include classification and regression.

**Unsupervised learning**, on the other hand, learns from unlabeled data without explicit labels, aiming to find hidden structures or patterns without guidance. Examples of unsupervised learning tasks include clustering and dimensionality reduction.

**The key difference** between supervised and unsupervised learning lies in the presence of labeled data. Supervised learning is typically used for tasks where the output is known and the goal is to make predictions, while unsupervised learning is used for tasks like data exploration and clustering where the structure of the data is unknown.

In order to analyze smartphone sensor data (accelerometer, gyroscope) labeled with activities (walking, running, sleeping), this project makes use of AI-ML, more especially supervised learning. The objective is to develop a trustworthy model that recognizes these actions.

**Human activity recognition is a supervised learning problem**

Because of the availability of labeled data, the classification nature of the task, the requirement for performance evaluation metrics, and the possibility of iterative improvement through model refinement techniques, human activity recognition (HAR) using smartphone datasets is typically approached as a supervised learning problem. Models can learn from labeled data, accurately classify activities, assess performance, and improve model efficacy iteratively with supervised learning.

**Algorithm**

**Feature Selection**

When using smartphone data for human activity recognition, feature selection plays a critical role in enhancing model performance and lowering complexity. Following the collection and preprocessing of sensor data, features are selected and extracted through the application of strategies such as tree-based methods, recursive elimination, and correlation-based selection. While iterative refinement optimizes the feature set for best performance as measured by metrics like accuracy and F1-score, cross-validation guarantees generalizability.

**Techniques Used**

**Filter Methods:** These methods select features based on their statistical properties and are independent of any specific machine learning algorithm.

Correlation-based feature selection (CFS)**:** Select features that have high correlation with the target variable but low correlation with each other.

Chi-square test: Select features that are most dependent on the target variable.

**Wrapper Methods**: These methods select subsets of features and evaluate them using a specific machine learning algorithm.

Recursive Feature Elimination (RFE): Start with all features, recursively remove one feature at a time and select the subset of features that gives the best model performance.

**Regression Used**

**Random Forest Regression**

When using smartphone datasets for human activity recognition projects, Random Forest Regression is a useful tool. Relevant features are extracted and the dataset is divided into training and testing sets following the collection of sensor data and labeled activity information. After that, the model is trained using the training set of data to discover how features from the sensor data relate to continuous activity variables like intensity or duration. Model performance is optimized through hyperparameter tuning, which is measured with metrics such as mean squared error. After validation, the model can forecast fresh data, which helps with applications such as health monitoring and fitness tracking. Random Forest Regression is a good option for these kinds of projects because it can handle high-dimensional data, capture intricate relationships, and reduce overfitting.

**Logistic regression** is a statistical method used for binary classification tasks where the target variable is categorical and has only two possible outcomes, typically represented as 0 and 1. The model predicts the probability that a given input belongs to one of the two classes. It uses the logistic function, also known as the sigmoid function, to model the relationship between input variables and the probability of the binary outcome. The logistic function is defined as:

(z)=1+e^−z

**Training Process:**

During the training phase, the algorithm learns to recognize patterns and relationships between the input features extracted from smartphone sensor data and the corresponding human activities. It adjusts its parameters iteratively to minimize the difference between its predicted activities and the actual observed activities from the labeled dataset.

**Classification Task:**

Human activity recognition is typically formulated as a classification problem, where the goal is to predict the activity label (e.g., walking, running, sitting) based on the input features extracted from smartphone sensor data. The algorithm aims to approximate the mapping function between the sensor data and the corresponding human activities.

**Supervision:**

The labeled dataset serves as supervision for the learning process. The algorithm learns from past sensor data, guided by the known activity labels, and aims to generalize this knowledge to accurately predict human activities in unseen data.

**Sequence Modeling:** Human activity data collected from smartphone sensors can be treated as a sequence, where each data point represents sensor measurements over time. LSTM models are well-suited for modeling such sequential data due to their ability to capture temporal dependencies.

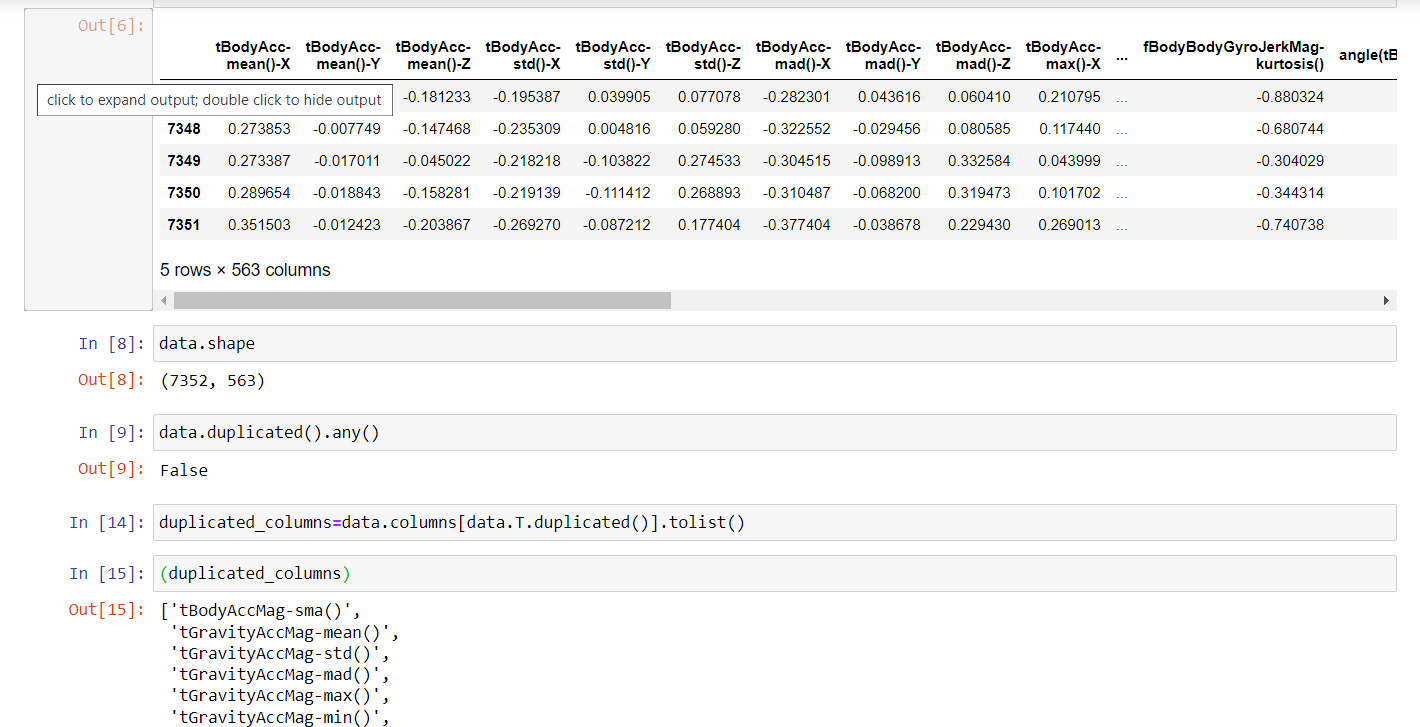
**Feature Extraction:** LSTMs can automatically extract relevant features from the sequential sensor data, such as patterns and trends in movement, without requiring manual feature engineering.

**Temporal Dynamics:** Human activities exhibit complex temporal dynamics influenced by factors like movement patterns and activity transitions. LSTMs can capture these dynamics and make predictions based on historical sensor data.

**Implementation in the Project:** In the project, an LSTM model is trained using smartphone sensor data to recognize human activities. The model learns from the sequential patterns in the sensor data to predict the corresponding activities. It captures temporal dependencies in the sensor data, allowing it to learn from past measurements and make accurate predictions. The trained LSTM model is evaluated on testing data to assess its performance, using metrics such as accuracy or F1 score to measure the effectiveness of activity recognition.

**Results**











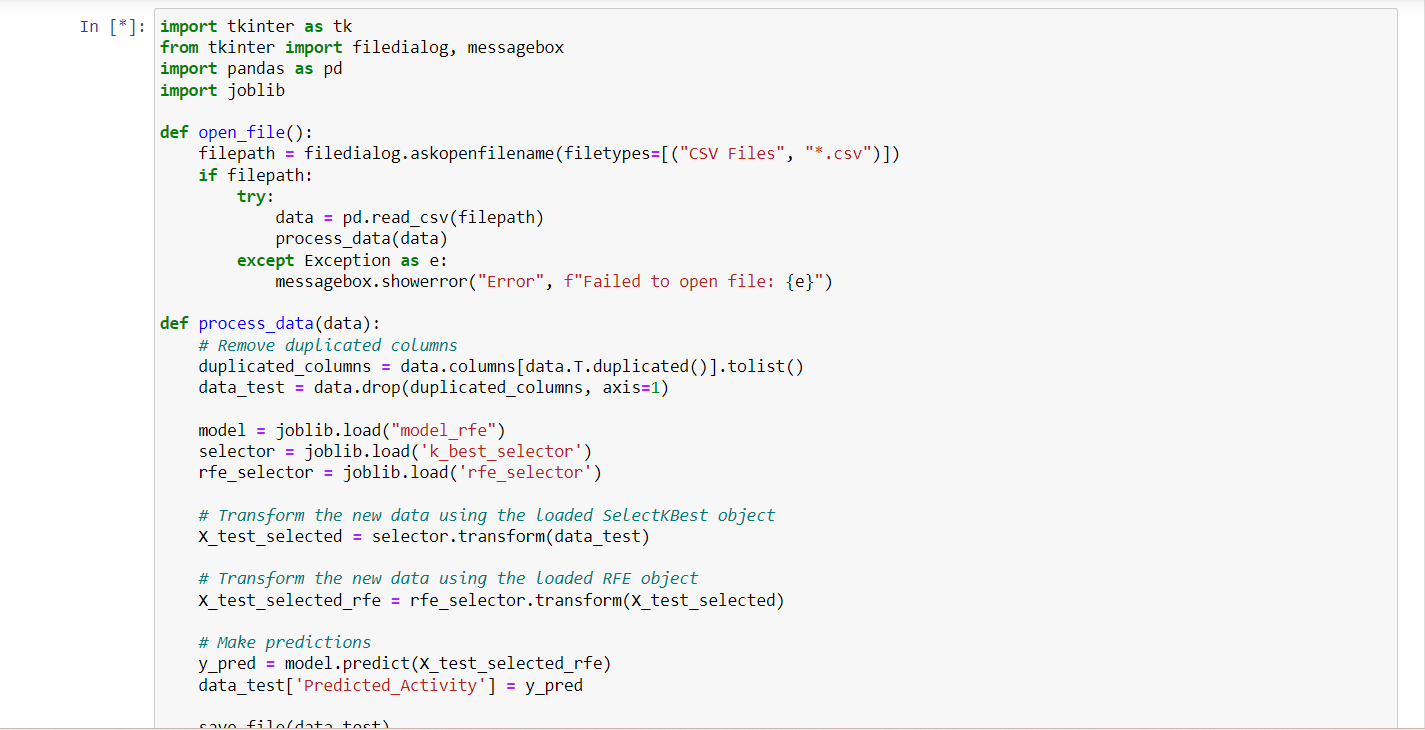


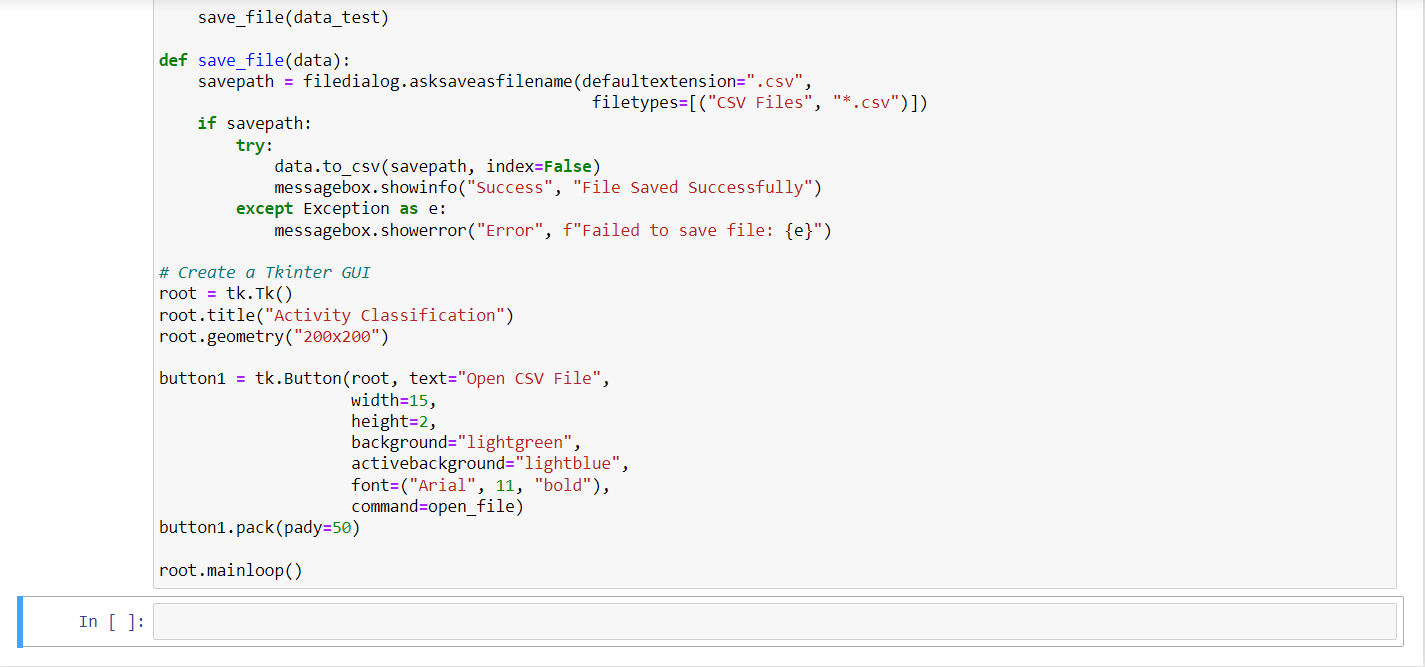


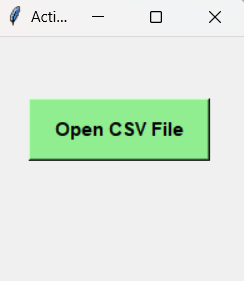




**GUI**







**References**

1. <https://www.kaggle.com/>
2. <https://www.geeksforgeeks.org/>