Project Title: PLANT INSIGHT

Cat-Safe Plant Identifier: A ML-based iOS Application for Detecting Toxic Plants

Team Members:

1. Milena Petrova
2. Jessenia Tsenkman
3. Aleksei Ganyukov

Task 2: Business Understanding

Background

With the rise in pet ownership, particularly among cat owners, ensuring the safety of household plants has become a critical concern. Many common indoor plants can be toxic to cats, leading to serious health complications or even fatality if ingested. Despite the risk, pet owners often lack awareness or tools to identify whether a specific plant is for their feline companions. This project addresses the issue by developing a machine learning-based Swift application that identifies plant species from user-uploaded images and indicates their toxicity to cats.

Business Goals

- *Primary goal:* Develop a robust machine learning model and prototype application capable of accurately identifying plant species from user-uploaded images and determining their toxicity to cats.
- *Secondary Goals:*
    - Enhance pet safety by providing reliable information about plant toxicity.
    - Educate users on plant care and safety for pets.
    - Gain practical experience in data preprocessing, model training, and integration with iOS applications.
    - Create a scalable framework that can be adapted for future projects involving different types of data or applications (e.g. extended to include toxicity information for other pets or additional plant species..)

Business Success Criteria

- Model Accuracy: Achieve at least 85% accuracy in plant species identification.
- Performance: Ensure the application processes images and returns results within 3 seconds on standard iOS devices.
- User Interface: Develop an intuitive and user-friendly interface that facilitates easy image uploads and result interpretation.

Assessing Your Situation

Inventory of Resources:

- Data: A dataset comprising 14,790 images across 47 plant species.
- Tools: Jupyter Notebook, TensorFlow/Keras for model development, GitHub for version control, and Xcode for iOS app development.
- Team Expertise: Proficiency in Python, machine learning, data preprocessing, and basic iOS development.
- Support: Access to online tutorials and machine learning communities for troubleshooting and guidance.

Requirements, Assumptions, and Constraints:

- Requirements:
    - A high-quality, labeled dataset of 14,790 images, accurately categorized into 47 plant species.
    - Reliable mapping of plant species to binary toxicity labels (toxic or non-toxic for cats).
    - An efficient machine learning model that achieves at least 85% accuracy and is optimized for mobile deployment using Core ML.
    - (Access to tools such as TensorFlow/Keras for model development, and Xcode for iOS prototyping.)
- Assumptions:
    - The dataset is representative of real-world variations in plant images.
    - Users will provide clear and focused plant images, ensuring effective identification by the model.
    - (External resources for toxicity data are accurate and sufficient for mapping plant species.)
- Constraints:
    - Limited time frame of one month to complete the project.
    - Potential class imbalance within the dataset.
    - Ensuring user privacy and data security within the app.
    - Limited expertise in advanced iOS development.

Risks and Contingencies:

- Risk: Model underperformance due to class imbalance.
    - Contingency: Implement data augmentation and class weighting techniques to mitigate imbalance. Select only a subset of the classes.
- Risk: Limited toxicity data for some plant species.
    - Contingency: Focus on the most common plants and clearly indicate when toxicity information is unavailable.
- Risk: Technical challenges in integrating the model into the iOS app.
    - Contingency: Utilize Core ML tools and seek support from online communities or documentation.

Terminology:

- Transfer Learning: Leveraging pre-trained models to expedite the training process on a new dataset.
- Class Imbalance: A scenario where some classes have significantly more samples than others, potentially biasing the model.
- Data Augmentation: Techniques used to increase the diversity of data available for training models without collecting new data.
- Core ML: Apple's machine learning framework for integrating models into iOS applications.

Costs and Benefits:

- Costs:
    - Time investment: Described in Task 4.
    - Computational Resources:
        - Use of personal laptops with CPU,GPU for model training and experimentation.
        - Potential reliance on cloud services like Google Colab for intensive training tasks, with estimated costs ranging from €12 to €50, depending on the computational demands.
- Benefits:
    - Enhanced understanding of machine learning workflows and mobile app integration.
    - Enhanced pet safety through reliable toxicity information.
    - Educational value in raising awareness about pet-friendly plants.

Defining Your Data-Mining Goals

Data-Mining Goals:

- Develop a machine learning model capable of identifying plant species from images with a high degree of accuracy.
- Integrate the trained model into an iOS application to provide real-time toxicity information to users.
- Ensure the model is optimized for performance on mobile devices, balancing accuracy and processing speed.
- (Enhance usability by ensuring that the model output is interpretable and actionable, providing clear toxicity information to the user.)

Data-Mining Success Criteria:

- Model Accuracy: The identification model should achieve at least 85% accuracy on the validation dataset.
- Performance Metrics: The application should process and return identification results within 3 seconds per image.

- Model Efficiency: The final model should be optimized to run smoothly on standard iOS devices without excessive battery consumption.

Task 3: Data Understanding

During this phase of the CRISP-DM methodology, we systematically gather, describe, explore, and verify the quality of the data to ensure it aligns with our project goals.

Gathering Data

Data Requirements

To achieve our goal of developing an accurate cat-safe plant identifier, the following data types are essential:

- Image Data: Images of 47 distinct plant species of various resolutions. Accepted formats are .jpg, .jpeg, .png
- Labels: Precise classification labels for each image, indicating the specific plant species.
- Toxicity Information: Binary labels denoting whether each plant species is toxic to cats (toxic/non-toxic).

Verify Data Availability

Our dataset comprises 14,790 images distributed across 47 plant species, with individual class sizes ranging from 66 to 547 images. This diversity provides a robust foundation for training our machine learning model but reveals an imbalance in class distribution.

Toxicity labels are not inherently included in the dataset. This information will be researched and compiled manually from external resources. Data integration will involve mapping plant species to their respective toxicity status.

Selection Criteria

To ensure the dataset's suitability for model training, we established the following selection criteria:

- Relevance: Images must clearly depict the plant species with identifiable features such as leaves, flowers, or stems.
- Quality: Images should maintain high resolution with minimal noise and artifacts to prevent model training issues.
- Diversity: A wide range of image conditions, including varying lighting and backgrounds, to enhance the model's generalization capabilities.
- Consistency: Uniform labeling format and structure to facilitate seamless integration and analysis.

Describing Data

Dataset Composition

- Total Images: 14,790
- Number of Classes: 47 plant species
- Image Size: Varies from 40KB to 1MB
- Image Types: Includes both whole plant images and close-ups of specific plant parts, captured in both indoor and outdoor settings.

Class Distribution

There is a notable imbalance in the dataset, with some species like *Monstera Deliciosa* having 547 images, while others like *Yucca* have only 66. This disparity necessitates strategies to address potential biases during model training.

Exploring Data

Visual Distribution

Initial visualization using bar charts and histograms highlighted the class imbalance. While the majority of plant species have a sufficient number of images for effective training, a few classes are underrepresented. Due to imbalance we choose a subset of plant classes with the most images, and implement the techniques of data augmentation and class weighting to ensure equitable model performance across all the selected classes.

Image Characteristics

- Quality Variation:
    - Wide range in image resolutions and clarity.
    - Presence of noise and varying degrees of image sharpness.
- Environmental Diversity:
    - Images captured in diverse backgrounds and lighting conditions.
    - Includes both controlled indoor environments and natural outdoor settings.

Preliminary Hypotheses

- Data augmentation can mitigate class imbalance and improve model robustness.
- High-quality, diverse images will contribute positively to model accuracy and generalization.

Verifying Data Quality

Consistency

Each image has been meticulously labeled with its corresponding plant species. However, the initial labeling was performed by non-experts, introducing the possibility of mislabeling. A small manual verification will be done.

Completeness

The dataset is complete in terms of image labels, with no missing classifications. Nevertheless, the imbalance in class sizes could lead to model bias favoring overrepresented species. Addressing this imbalance is critical to maintaining model fairness and accuracy.

Accuracy

While manual curation was conducted to ensure label accuracy. preprocessing steps, including resizing and noise reduction, will be applied to standardize image quality.

Task 4: Planning Your Project

Project Plan and Task Allocation:

Task	Description	Team Member(s)	Estimated Hours (h)
Data Analysis and Preparation	Select classes with the largest number of images for enhanced learning. Augment the dataset using libraries like ImageDataGenerator to ensure balance across classes.	Milena	4
		Jessenia	2
		Aleksei	2
Manual Toxicity Mapping	Research and compile toxicity information for each plant species.	Milena	4
		Jessenia	1
		Aleksei	1
Model Research and Selection	Research pre-trained models like MobileV2Net, evaluate datasets, and determine the most suitable model for the task.	Milena	7
		Jessenia	7
		Aleksei	7
Model Training and Evaluation	Use the chosen TensorFlow pre-trained model as the base. Train the model on the selected classes with proper data augmentation. Fine-tune the model to optimize the main performance metrics.	Milena	12
		Jessenia	12
		Aleksei	6
Model Optimization and Conversion	Convert the trained TensorFlow model into an MLModel format for Swift compatibility. Test the model to ensure no performance degradation during conversion.	Milena	6
		Jessenia	6
		Aleksei	1
Swift App Development	Create the app interface in Swift. Integrate the converted MLM model into the app.	Milena	8
		Jessenia	8

	Implement logic to display plant names and toxicity levels based on the model's output. Develop a feature to allow users to take or upload a photo for analysis.	Aleksei	4
App Testing	Test the performance and usability of the created app.	Milena	2
		Jessenia	2
		Aleksei	2
Documentation	Clean up the codes and the repository. Write the reports. Creation of the poster with the results of the project	Milena	6
		Jessenia	6
		Aleksei	10
Presentation	Presentation of the results during the poster session on December 13	Milena	3
		Jessenia	3
		Aleksei	3

Methods and Tools:

Model Training: TensorFlow, Keras, ImageDataGenerator, MobileNetV2
Model Conversion: TensorFlow Lite, CoreML
App Development: Swift, Xcode, CoreML
Data Augmentation: ImageDataGenerator
Testing: Xcode simulator, real iOS devices

Additional comments:

Not all parts of the project have been completed at this stage. Pending tasks may lead to small updates in the current plan and remarks.