**FINAL PROJECT – TEAM 3 PROPOSAL DOCUMENT**

**Project name:** [Flower-Identification-using-deep-learning](https://github.com/SriPenumatcha/Flower-Identification-using-deep-learning)

**Team members:** Lindsay McCulloch, Sri Penumatcha, Vrinda Patel, Yashada Kulkarni

**Repository:**

<https://github.com/SriPenumatcha/Flower-Identification-using-deep-learning>

**Google Collab file:**

<https://colab.research.google.com/drive/1rEDcHR7Swwjk4TG5KyduuMF3aKKUWtch?usp=sharing>

**Project description:**

The aim of this project is to prepare a data deep dive to show machine learning in the automated identification of flower species based on identifying features in a dataset.

**Dataset:**

* <https://www.kaggle.com/datasets/alxmamaev/flowers-recognition>

**Libraries/Frameworks:**

**NumPy:** for scientific computing in Python, provides support for large multi-dimensional arrays and matrices, along with a collection of mathematical functions to operate on these arrays.

**Pandas:** for data manipulation and analysis library. Providing data structures like DataFrames and Series that make working with structured data easier.

**Scikit-learn (sklearn):** is a machine learning library that supports various supervised and unsupervised learning algorithms. Also providing tools for model selection and evaluation.

**TensorFlow:** is an end-to-end open-source platform for machine learning, offering a comprehensive ecosystem of tools, libraries, and community resources to build and deploy machine learning models.

**Mnist:**  used as a benchmark for testing machine learning algorithms, particularly in the field of image recognition, for learning about neural networks and deep learning, to build and train models for image classification.

**Keras:** for building and training deep learning models with the creation of custom layers, loss functions, and metrics.

**Matplotlib:** is a plotting library for creating static, animated, and interactive visualizations in Python, providing a wide range of plotting options for data visualization.

**Plotly:** is another visualization library that specializes in creating interactive plots and dashboards. It can be used to visualize model accuracy, and data trends.

**\*Streamlit:** is a framework for building interactive web applications with Python, allowing the creation of data visualization dashboards and deploy machine learning models.*\*Potential framework to be used but not required*

**Method:**

**Data Collection:** Obtain a dataset that contains information about different flower species along with their identifying features.

**Data Exploration:** Explore the dataset to understand its structure, features, and the distribution of data. Use Pandas for data manipulation and analysis to gain insights into the dataset.

**Data Preprocessing:** Preprocess the data by handling missing values, encoding categorical variables, and scaling numerical features if needed.

**Feature Selection/Engineering:** Identify relevant features that can help in distinguishing between different flower species.

**Model Selection:** Choose a suitable machine learning algorithm for classification tasks. Common algorithms for this type of project include Decision Trees, Random Forest, Support Vector Machines, or Neural Networks.

**Model Training:** Split the dataset into training and testing sets. Train the selected model on the training data to learn the patterns in the features and their corresponding flower species labels.

**Model Evaluation:** Evaluate the model's performance using metrics like accuracy, precision, recall, and F1-score. This step helps in assessing how well the model can identify flower species based on the provided features.

**Visualization:** Use Matplotlib or Plotly to create visualizations that showcase the model's predictions, feature importance, or any other relevant insights from the data.

**\*Deployment:** Using Streamlit to build a user-friendly interface where users can input features and get predictions on the flower species. *\*Potential framework to be used but not required*

**Requirements**

**Data Model Implementation (25 points)**

* A Python script initialises, trains, and evaluates a model (10 points)

*IPYNB file on Google Collab*

* The data is cleaned, normalised, and standardised prior to modelling (5 points) *Completed within IPYNB file*
* The model utilises data retrieved from SQL or Spark (5 points)

*Chris confirmed not required as we are not expected to store all image data in AWS and we will not loose marks*

* The model demonstrates meaningful predictive power at least 75% classification accuracy or 0.80 R-squared. (5 points)

*Pre-prossessing notebook to test the model to see base accuracy then optimise the model in next steps.*

**Data Model Optimisation (25 points)**

* The model optimisation and evaluation process showing iterative changes made to the model and the resulting changes in model performance is documented in either a CSV/Excel table or in the Python script itself (15 points)
* Overall model performance is printed or displayed at the end of the script (10 points) use Leaflet or Plotly

**GitHub Documentation (25 points)**

* GitHub repository is free of unnecessary files and folders and has an appropriate .gitignore in use (10 points)
* The README is customised as a polished presentation of the content of the project (15 points)

**Group Presentation (25 points)**

* All group members speak during the presentation. (5 points)
* Content, transitions, and conclusions flow smoothly within any time restrictions. (5 points)
* The content is relevant to the project. (10 points)
* The presentation maintains audience interest. (5 points)