AutoML System Implementation Report

**Introduction:**

Automated Machine Learning (AutoML) aims to simplify the process of applying machine learning techniques by automating model selection, hyperparameter tuning, and feature engineering. This report details the implementation of an AutoML system using the Tree-based Pipeline Optimization Tool (TPOT) on the Iris dataset, a well-known dataset used for classification tasks.

**Objectives:**

The objective of this project is to implement an AutoML framework with TPOT to automatically select the best models and hyperparameters for the Iris dataset. This implementation serves as a foundational step towards exploring advanced meta-learning techniques.

**Dataset Selection:**

Source: UCI Machine Learning Repository

Dataset Description: The Iris dataset contains 150 samples of iris flowers, each described by four features: sepal length, sepal width, petal length, and petal width. The target variable represents the species of the iris, which can belong to one of three classes: Iris-setosa, Iris-versicolor, or Iris-virginica.

Data Preprocessing: The dataset was already clean and well-structured. An 80/20 split was performed to divide the data into training and testing sets, and no additional preprocessing was required.

**AutoML Framework Implementation:**

Tool Used: TPOT (Tree-based Pipeline Optimization Tool)

Justification: TPOT was selected due to its ability to automatically search through thousands of possible machine learning pipelines and its integration with scikit-learn, making it suitable for classification tasks.

**Implementation Steps:**

* Step 1 - Setup Environment:
  + Python version: 3.x
  + Libraries installed: TPOT, scikit-learn, pandas, numpy.
* Step 2 - Load the Iris Dataset:

from sklearn.datasets import load\_iris

from sklearn.model\_selection import train\_test\_split

iris = load\_iris()

X = iris.data

y = iris.target

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

* Step 3 - Initialize and Fit TPOT:

from tpot import TPOTClassifier

tpot = TPOTClassifier(generations=5, population\_size=20, random\_state=42)

tpot.fit(X\_train, y\_train)

* Step 4 - Evaluate the Model:

accuracy = tpot.score(X\_test, y\_test)

print(f'Test Accuracy: {accuracy:.2f}')

* Step 5 - Export the Best Pipeline:

tpot.export('best\_pipeline.py')

print(tpot.fitted\_pipeline\_)

**Results:**

**Model Selected:** The best pipeline discovered by TPOT is a stacking estimator that combines:

* **Base Model:** Multinomial Naive Bayes (MultinomialNB) with ‘alpha=10.0’ and ‘fit\_prior=False’.
* **Final Model:** Logistic Regression with ‘C=25.0’ and ‘random\_state=42’.
* **Accuracy:** The model achieved a test accuracy of 1.00 (100%) on the Iris dataset.

**Insights Gained:**

The results indicate that the combination of a stacking estimator leveraging Multinomial Naive Bayes and Logistic Regression provided excellent classification performance on the Iris dataset. The high accuracy demonstrates the effectiveness of the selected models, particularly in handling the relatively simple classification task presented by the Iris dataset.

**Conclusion:**

The implementation of the AutoML system using TPOT on the Iris dataset successfully identified a stacking estimator comprising a Multinomial Naive Bayes base model and a Logistic Regression final classifier. The model achieved a perfect accuracy of 100% on the test set, showcasing TPOT's capability to automate the model selection process effectively. This foundational work sets the stage for transitioning to more complex meta-learning techniques in future phases of the project.