MACHINE LEARNING PROJECT

By Shivam Arora and Dhrumil Patel

Problem Statement

- Weka requires the installation and is platform dependent.
- Weka UI is outdated and does not have proper documentation.
- Difficult to troubleshoot
- Limited feature engineering options.

Benefits of Trainify

WEB BASED MACHINE LEARNING APPLICATION

Trainify can run on any device that has access to internet

COMPREHENSIVE MACHINE LEARNING TOOLKIT

As a web-based platform, Trainify can be updated regularly with new features, improvements, and bug fixes, ensuring users always have access to the latest and most powerful tools without installing anything

CONTINUOUS UPDATES AND SUPPORT

Trainify offers a wide range of machine learning algorithms, data preprocessing tools, feature selection methods, model evaluation and validation techniques

Machine Learning Algorithms and Neural Networks

KNN

Logistic Regression

Decision Trees

Random Forests

SVM

Naive Bayes

MLP Classifier Neural Network

How to use Trainify?

STEP 1

Upload your Dataset

Datasets upto 200MB only accepted

STEP 2

Perform Data Preprocessing Techniques

- Drop specific columns
- Label Encode columns

STEP 3

Perform Data Scaling.

- StandardScaler
- Or MinMax Scaler

STEP 4

Handle Missing Values.

- Drop the rows
- Or fill in the values using mean, median or mode

STEP 5

Choose a Model or Benchmark All Models

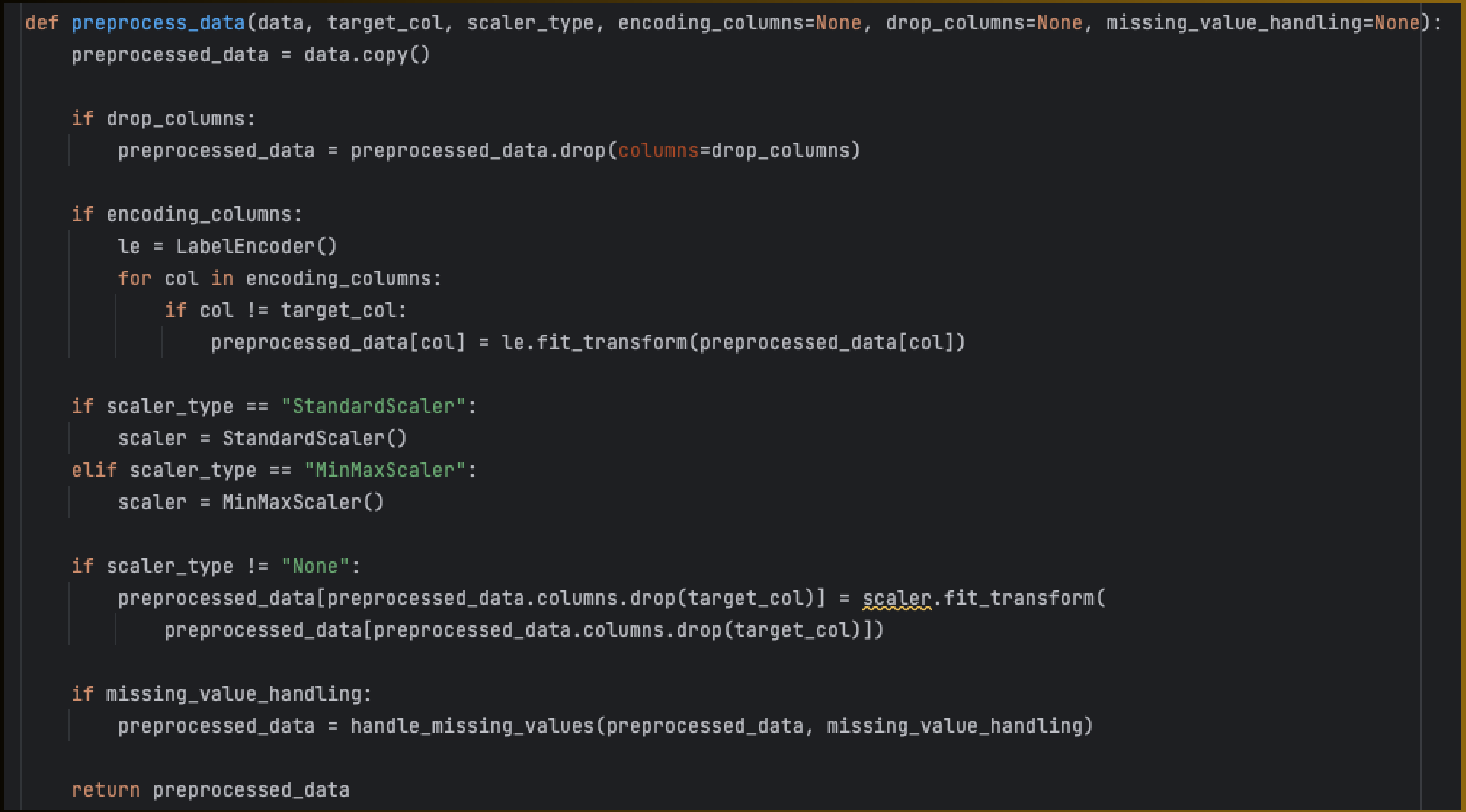
TRAINIFY DEMO

https://shivamaroraa-machine-learning-streamlit-app-cgttdx.streamlit.app

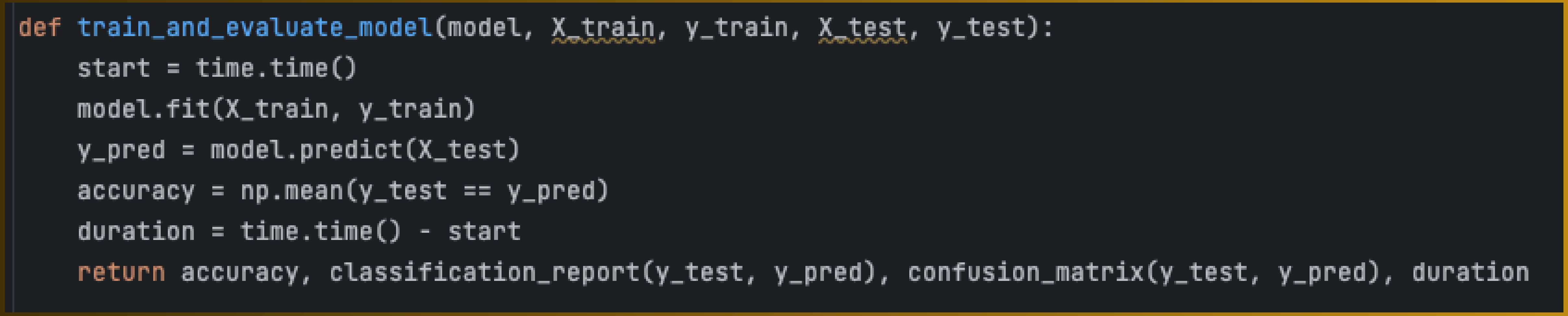
How Trainify is built?

Trainify is built completely in Python and uses the following libraries and frameworks:-

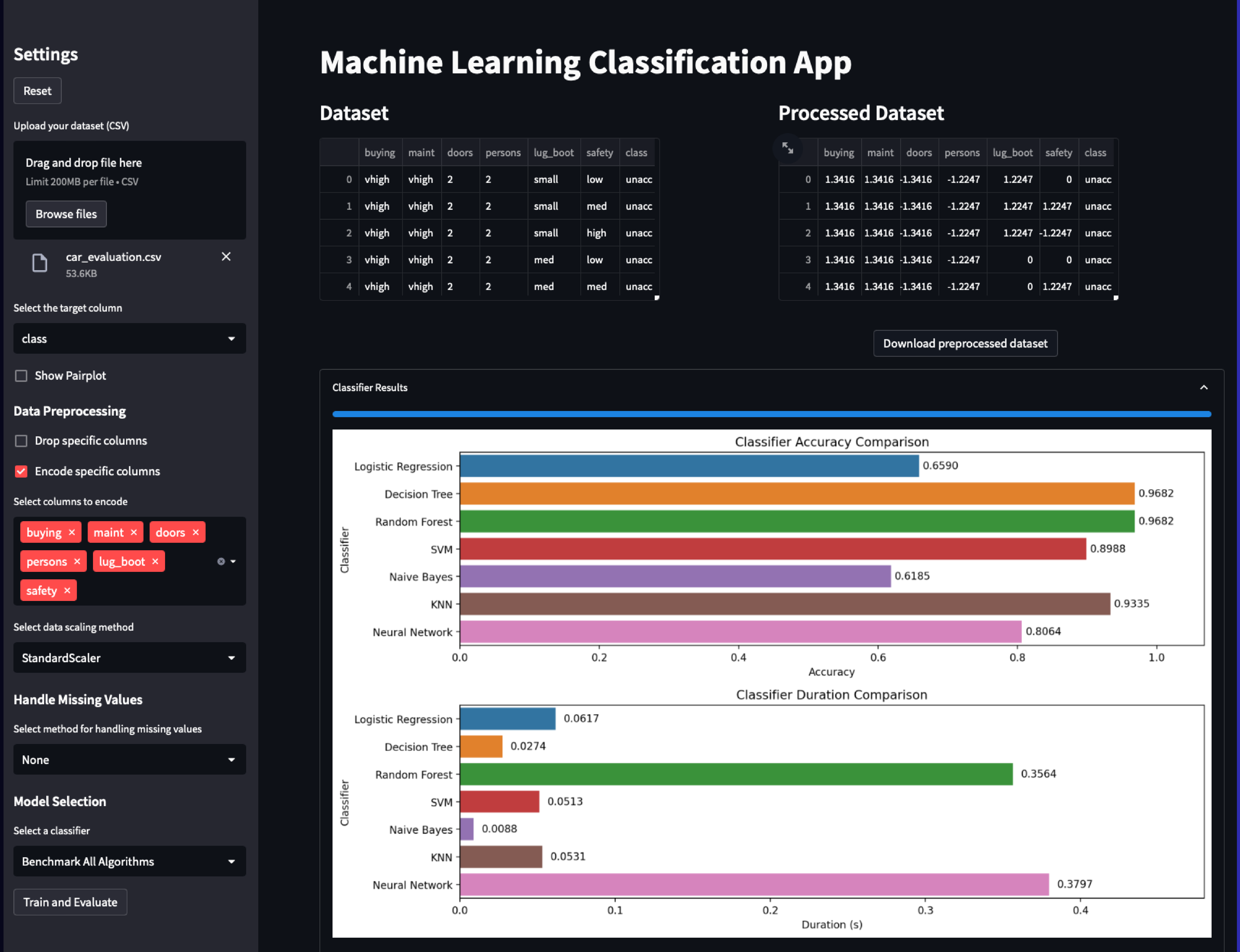
1. sklearn
2. Matplotlib
3. Seaborn
4. Pandas and Numpy
5. Streamlit



```python
def preprocess_data(data, target_col, scaler_type, encoding_columns=None, drop_columns=None, missing_value_handling=None):
    preprocessed_data = data.copy()

    if drop_columns:
        preprocessed_data = preprocessed_data.drop(columns=drop_columns)

    if encoding_columns:
        le = LabelEncoder()
        for col in encoding_columns:
            if col != target_col:
                preprocessed_data[col] = le.fit_transform(preprocessed_data[col])

    if scaler_type == "StandardScaler":
        scaler = StandardScaler()
    elif scaler_type == "MinMaxScaler":
        scaler = MinMaxScaler()

    if scaler_type != "None":
        preprocessed_data[preprocessed_data.columns.drop(target_col)] = scaler.fit_transform(
            preprocessed_data[preprocessed_data.columns.drop(target_col)])

    if missing_value_handling:
        preprocessed_data = handle_missing_values(preprocessed_data, missing_value_handling)

    return preprocessed_data
```



```python
def train_and_evaluate_model(model, X_train, y_train, X_test, y_test):
    start = time.time()
    model.fit(X_train, y_train)
    y_pred = model.predict(X_test)
    accuracy = np.mean(y_test == y_pred)
    duration = time.time() - start
    return accuracy, classification_report(y_test, y_pred), confusion_matrix(y_test, y_pred), duration
```



Settings

Upload your dataset (CSV)

Drag and drop file here
Limit 200MB per file • CSV

car_evaluation.csv
53.6KB

Select the target column: class

□ Show Pairplot

Data Preprocessing

□ Drop specific columns

☑ Encode specific columns

Select columns to encode: buying, maint, doors, persons, lug_boot, safety

Select data scaling method: StandardScaler

Handle Missing Values

Select method for handling missing values: None

Model Selection

Select a classifier: Benchmark All Algorithms

Machine Learning Classification App

Dataset

	buying	maint	doors	persons	lug_boot	safety	class
0	vhigh	vhigh	2	2	small	low	unacc
1	vhigh	vhigh	2	2	small	med	unacc
2	vhigh	vhigh	2	2	small	high	unacc
3	vhigh	vhigh	2	2	med	low	unacc
4	vhigh	vhigh	2	2	med	med	unacc

Processed Dataset

	buying	maint	doors	persons	lug_boot	safety	class
0	1.3416	1.3416	-1.3416	-1.2247	1.2247	0	unacc
1	1.3416	1.3416	-1.3416	-1.2247	1.2247	1.2247	unacc
2	1.3416	1.3416	-1.3416	-1.2247	1.2247	-1.2247	unacc
3	1.3416	1.3416	-1.3416	-1.2247	0	0	unacc
4	1.3416	1.3416	-1.3416	-1.2247	0	1.2247	unacc

Future Works

- Adding more preprocessing steps
- Adding support for performing regression
- Adding steps such as choosing the number of layers, number of neurons, choosing activation functions in Neural Networks
- Deployment on a better cloud VPC and support for GPU for Neural Networks.

Awesome right?

Any Questions?