Quick Start with EazyML APIs

1 Overview

Machine Learning (ML) is an involved science, its models often complex, not easy to understand. Transparent ML explains itself – its working, its prediction, its insights – so that the user understands it. The sequel describes how to use EazyML API based package to explore transparent machine learning with EazyML. The python package is called "eazyml_upload_build_predict.py". This package is offered as a quick start to dive into the comprehensive set of APIs described on eazyml.com.

2 Authenticate

EazyML authenticates you with username and API token. You can obtain your API token by logging into EazyML service portal and then navigating to "My Accounts" → "API Key".

Usage:

```
python eazyml_upload_build_predict.py -username <username> --api_key
<api_key>
```

Example:

```
python eazyml_upload_build_predict.py -u vikram.nunia@gmail.com --api_key
<api_key>
```

Authentication successful.

Authentication information is stored in authentication.json

3 Upload and Preprocess a Dataset

Post authentication, you can then upload a dataset to EazyML for pre-processing. The pre-processing steps include sanity checks, discarding empty or mostly empty rows, imputation and outlier detection/removal.

Usage:

```
python eazyml_upload_build_predict.py --train_file <train file> --outcome
<outcome_col>
```

Example:

```
python eazyml_upload_build_predict.py --train_file
Blood_Pressure_classification.csv --outcome "Systolic pressure"
```

Authentication successful.

Uploading the dataset ... Finished.

Data uploaded successfully on EazyML: Blood_Pressure_classification.csv

The response is stored in upload_data.json

The reference identifier for the dataset (dataset_id) is: 19292

4 Build Predictive Models

Once the dataset is uploaded and pre-processed, it's ready for building either predictive models or for building augmented intelligence model. The sequel shows how to build predictive models.

Usage:

```
python eazyml_upload_build_predict.py --dataset_id <dataset_id> --
predictive
```

Example:

```
python eazyml_upload_build_predict.py --dataset_id 19292 --predictive
```

Authentication successful.

Building predictive models ... Finished.

Predictive models built successfully.

The reference identifier for the model (model_id) is: 13939

Performance metrics are stored in predictive_model_performance_metrics.json

5 Build Augmented Intelligence Models

A pre-processed dataset can be used to build an augmented intelligence model as follows.

Usage:

```
python eazyml_upload_build_predict.py --dataset_id <dataset_id> --augi
```

Example:

```
python eazyml_upload_build_predict.py --dataset_id 19292 --augi
```

Authentication successful.

Building augmented intelligence models ... Finished.

Augmented Intelligence models built successfully.

The reference identifier for the model (model_id) is: 13940

Augmented intelligence insights are stored in augi_insights.json

6 Make Predictions

Once built, you may use the predictive model to make predictions as follows.

Usage:

```
python eazyml_upload_build_predict.py --model_id <model_id> --predict_file
<predict file>
```

Example:

```
python eazyml_upload_build_predict.py --model_id 13939 --predict_file
Blood_Pressure_classification.csv
```

Authentication successful.

Uploading the prediction dataset and making predictions ... Finished.

Predictions are ready.

The reference identifier for predictions (prediction_dataset_id) is: 8490

Predictions are stored in predictions.json

6 Execute Explainable AI

After the model has been executed on the prediction dataset to make predictions, here is how you can explain one or more predictions.

Usage:

```
python eazyml_upload_build_predict.py --model_id <model_id> --
prediction_dataset_id <prediction_dataset_id> --explain_rec_nums <comma
separated numbers if more than 1>
```

Example:

```
python eazyml_upload_build_predict.py --model_id 13939 --
prediction_dataset_id 8490 --explain_rec_nums 1,2
```

Authentication successful.

Executing Explainable-AI ... Finished.

Explanations is/are ready!!

Explanations are stored in explanations.json

7 Accelerate: Upload, Build, Predict and Explain

Here's how you can combine multiple of previously explained steps from start to finish. In the process you upload dataset, build predictive model, make predictions and explain them all with one command.

Usage:

```
python eazyml_upload_build_predict.py --train_file <train file> --outcome
<outcome_col> -id_col <ID Col> --discard_col_list <comma separated list if
more than 1> --predictive --predict_file <predict file> --explain_rec_nums
<comma separated row numbers if more than 1>
```

Example:

```
python eazyml_upload_build_predict.py --train_file forest_cover_train.csv -
-outcome Cover_Type --id_col Id --discard_col_list "Soil_Type4,Soil_Type5"
--predictive --predict_file forest_cover_test.csv --explain_rec_nums 1,2
```

Authentication successful.

Uploading the dataset ... Finished.

Data uploaded successfully on EazyML: Blood_Pressure_classification.csv

The response is stored in upload_data.json

The reference identifier for the dataset (dataset_id) is: 19295

Building predictive models ... Finished.

Predictive models built successfully.

The reference identifier for the model (model_id) is: 13939

Performance metrics are stored in predictive_model_performance_metrics.json

Uploading the prediction dataset and making predictions ... Finished.

Predictions are ready.

The reference identifier for predictions (prediction_dataset_id) is: 8490

Predictions are stored in predictions.json

Executing Explainable-AI ... Finished.

Explanations is/are ready!!

Explanations are stored in explanations.json

8 Accelerate: Upload and Build Augmented Intelligence Model

Here's how you can combine multiple of previously explained steps from start to finish. In the process you upload dataset, build augmented intelligence model and get actionable insights all with one command.

Usage:

```
python eazyml_upload_build_predict.py --train_file <train file> --outcome
<outcome_col> --augi
```

Example:

```
python eazyml_upload_build_predict.py --train_file
Blood_Pressure_classification.csv --outcome "Systolic pressure" --augi
```

Authentication successful.

Uploading the dataset ... Finished.

Data uploaded successfully on EazyML: Blood_Pressure_classification.csv

The response is stored in upload_data.json

The reference identifier for the dataset (dataset_id) is: 19295

Building augmented intelligence models ... Finished.

Augmented Intelligence models built successfully.

The reference identifier for the model (model_id) is: 13940

Augmented intelligence insights are stored in augi_insights.json

9 Custom Configuration

Here's how you could customize machine learning and transparency experiments with your own configuration parameters. Your configuration parameter choices could be specified in a configuration file in .ini format. List of all available configuration parameters can be found in the sample configuration file. The configuration parameters specifically for transparent machine learning are explained in a user guide here. The sequel describes how you upload your custom configuration for an experiment.

Usage:

```
python eazyml_upload_build_predict.py --config_file <configuration file>
```

Example:

```
python eazyml_upload_build_predict.py --config_file sample_config.ini
--config_file sample_config.ini
```

Authentication successful.
Uploading config file ...
Config file uploaded and set successfully.