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 --username vikr.nunia@gmail.com
--api_key <api_key>
```

Authentication successful.

Authentication information is stored in authentication.json

Please note that your authentication information gets stored in a local file and then gets used for all subsequent calls described in the sequel below. Authentication is the mandatory first step in any experiment.

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. If you are uploading a training data file then the outcome column name is mandatory. You may optionally provide a **prefix name** that will be prefixed to the auto generated output file names – so that you can keep your various output files organized for your datasets.

Usage:

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

Example:

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

```
Authentication successful.

Uploading dataset ...

Data uploaded successfully on EazyML: Blood_Pressure_classification.csv

The response is stored in BPCL_upload_data.json

Upload time: 2.81 secs

The reference identifier for the dataset (dataset_id) is: 5299
```

After a successful upload of your training data, a reference identifier is provided back to you for subsequent operations on this dataset – such as building predictive models or augmented intelligence.

4 Build Predictive Models

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

Usage:

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

Example:

```
python eazyml_upload_build_predict.py --prefix_name BPCL --dataset_id
5299 --predictive
```

```
Authentication successful.

Building predictive models ...

Predictive models: initialized successfully.

Model initialization time: 0.35 secs

Predictive models: feature selected successfully.

Selected features are stored in BPCL_predictive_model_feature_selected.json

Feature selection time: 0.89 secs

Predictive models built successfully.

Performance metrics are stored in BPCL_predictive_model_performance_metrics.json
```

```
Model building time: 20.21 secs

The reference identifier for the model (model_id) is: 3373
```

After a model is built successfully, a reference identifier is provided back to you for subsequent operations on this model – such as making predictions. There are two output files generated – one for selected features and other listing the models built along with their performance metrics. The best model in terms of performance is the one recommended by EazyML and is used for making predictions by default.

5 Build Augmented Intelligence Models

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

Usage:

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

Example:

```
python eazyml_upload_build_predict.py --prefix_name BPCL --dataset_id 5299 --augi
```

```
Authentication successful.

Building augmented intelligence models ...

Augmented Intelligence models: initialized successfully.

Model initialization time: 0.37 secs

Augmented Intelligence models: feature selected successfully.

Selected features are stored in BPCL_augi_insights_feature_selected.json

Feature selection time: 0.90 secs

Augmented Intelligence models built successfully.

Augmented Intelligence insights are stored BPCL_augi_insights.json

Model building time: 1.76 secs
```

There are two output files generated here too – one for features selected during the execution of augmented intelligence functionality and the other for the extracted insights. The insights are ordered by their confidence scores. You may choose to accept insights only above certain threshold.

6 Make Predictions

Here's how to make predictions on a predictive model. You will need the reference model identifier and a prediction dataset filename. The best model recommended by

EazyML is used for making predictions – best in terms of performance metrics. The predictions are stored in an output file in the same order of data records as in the input file.

Usage:

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

Example:

```
python eazyml_upload_build_predict.py --prefix_name BPCL --model_id
3373 --predict_file Blood_Pressure_classification.csv
```

Authentication successful.

Uploading the prediction dataset and making predictions ...

Predictions are ready.

Predictions are stored in BPCL_predictions.json

Prediction time: 1.94 secs

The reference identifier for predictions (prediction_dataset_id) is: 882

A reference identifier is provided so that you can execute Explainable-AI for each of those predictions – as the sequel explains.

7 Execute Explainable AI

After the predictive model has been executed on the prediction dataset file to make predictions, here is how you can explain one or more predictions. You need the model identifier and the prediction dataset identifier for reference. And then you can specify a comma separated list of record indices for which predictions need to be explained.

Usage:

```
python eazyml_upload_build_predict.py --prefix_name <file_prefix> --
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 --prefix_name BPCL --
model_id 3373 --prediction_dataset_id 882 --explain_rec_nums
1,2
```

Authentication successful.

Executing Explainable-AI ...

Explanation/s is/are ready

Explanations are stored in BPCL_explanations.json

```
Explanations time: 20.43 secs
```

The output file includes not only explanations for predictions, but also the confidence score expressing "goodness of reasons explained". The file also includes importance of features for each specific prediction data point – different data points may have different reasons for their predictions.

8 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 --prefix_name <file_prefix> --
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 --prefix_name "Forest_Cover" --
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 dataset ...

Data uploaded successfully on EazyML: forest_cover_train.csv

The response is stored in Forest_Cover_upload_data.json

Upload time: 18.17 secs

The reference identifier for the dataset (dataset_id) is: 5300

Building predictive models ...

Predictive models: initialized successfully.

Model initialization time: 0.83 secs

Predictive models: feature selected successfully.

Selected features are stored in Forest_Cover_predictive_model_feature_selected.json

Feature selection time: 10.50 secs

Predictive models built successfully.

Performance metrics are stored in
Forest_Cover_predictive_model_performance_metrics.json
```

```
Model building time: 1413.07 secs

The reference identifier for the model (model_id) is: 3375

Uploading the prediction dataset and making predictions ...

Predictions are ready.

Predictions are stored in Forest_Cover_predictions.json

Prediction time: 1.94 secs

The reference identifier for predictions (prediction_dataset_id) is: 883

Executing Explainable-AI ...

Explanation/s is/are ready

Explanations are stored in Forest_Cover_explanations.json

Explanations time: 40.43 secs
```

9 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 --prefix_name <file_prefix> --
train_file <train file> --outcome <outcome_col> --augi
```

Example:

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

```
Authentication successful.

Uploading dataset ...

Data uploaded successfully on EazyML: Blood_Pressure_classification.csv

The response is stored in BPCL_upload_data.json

Upload time: 2.75 secs

The reference identifier for the dataset (dataset_id) is: 5301

Building augmented intelligence models ...

Augmented Intelligence models: initialized successfully.

Model initialization time: 0.38 secs

Augmented Intelligence models: feature selected successfully.
```

```
Selected features are stored in BPCL_augi_insights_feature_selected.json

Feature selection time: 0.98 secs

Augmented Intelligence models built successfully.

Augmented intelligence insights are stored BPCL_augi_insights.json

Model building time: 1.63 secs
```

1 Custom Configuration

Here's how you can 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
```

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

You are welcome to enhance this package with various other configurability options that EazyML provides and build powerful applications.