
# Component Specification

Joel Stremmel, Yiming Liu, Cathy Jia, Monique Bi, Arjun Singh

## Software Components

### ❑ Binary Score Plot

A wrapper class for visualizing model performance plots for binary classification tasks given model scores and true labels. It includes commonly used performance plots such as the Receiver Operating Characteristic curve. Getter and setter methods support adding reusable plotting options.

**User Input**

- A dataframe

**Output Plots**

- Overall distribution of predicted probabilities
- A plot showing trade-off between false positive and true positive rate for binary classifiers
- A plot showing the Receiver Operating Characteristic curve
- Precision and recall plot

### ❑ Regression Analysis Plot

A wrapper class for visualizing model fit for regression tasks. It allows a user to assess regression model assumptions and judge the quality of the fit to the data use plots such as residuals vs fitted values and scatter plots with the fitted model.

**User Input**

- A dataframe

**Output plots**

- Scatter matrix plot
- Correlation heatmap
- Box plot
- Distribution plot
- Scatter plot with regression line
- Model diagnostics plots
    - Residual vs fitted plot
    - Normal QQ plot
    - Scale-Location plot
    - Residual vs leverage plot

## ❑ PCA Evaluate

A module for visualizing model fit for PCA performance. It provides user with direct presentation on misclassification error on different dimension reduced, therefore suggests the optimal dimension by picking the one with least misclassification error.

**User Input**

- A dataset splitted into *train_features*, *test_features*, *train_labels* and *test_label* sets.

**Output**

- A plot showing misclassification error vs PCA dimensions
- The optimal dimension

## ❑ Clustering Evaluate

A module for visualizing model fit for clustering performance. It allows user to fit dataset with k-means clustering using different number of clusters in order to find the optimal number of clusters.

**User Input**

- A dataset splitted into predictors set X and response set y

**Output**

- A plot of objective value from k-means++ vs number of clusters
- The optimal number of cluster

### ❑ Example Notebooks

Jupyter notebooks which will load example data and fit basic models from sklearn to obtain model scores. These notebooks will then step through how to initialize plotting objects, and use plotting methods from the classes above to analyze performance or fit of the example models.

### ❑ Interactive Dashboards (Using Bokeh and Dash)

Interactive model diagnostics dashboards served for binary classification and linear regression models. These two interactive dashboards allow users to input their designated dataset and create plots, which are referenced to *betas* library.

## Interactions

Two of our use cases are allowing users to fit binary classification model and linear regression models. The above software components interact to accomplish these tasks by allowing users to instantiate an instance of a plotting class and pass arguments to the object such as model scores and true labels.

The users are required to provide a dataframe and install the *betas* python package. After import *betas*, the users can choose to call different methods in the plotting object, and will be able to visualize model performance by looking at plots like the Receiver Operating Characteristic curve. If using the regression model class, the plotting object for that class has methods for analyzing properties like model fit or model assumptions such as constant variance. In this way, all interactions with this software will come by writing code using classes and methods from our package to quickly and efficiently analyze model performance and assumptions.

Also, users are able to use our interactive dashboards designed for these two models to explore model fitting and model assumptions, without writing any Python code. They need to run the dashboard files, open the dashboard in a web page and input a dataset source. On the dashboard pages, users can select model parameters in a dropdown box or change threshold by adjusting a slider.

## Preliminary Plan

Our preliminary plan consists of accomplishing the following tasks in order of priority:

- Identifying example datasets and fitting basic classifiers from sklearn to these datasets to generate example model scores to be compared with true labels.
- Creating plotting classes to enable easy analysis of various types of model performance we intend to analyze.
- Create methods within each class to generate the most critical plots first, such as plots of precision and recall, as well as plots of the Receiver Operating Characteristic curve. We will repeat this process for regression analysis, focusing on the most important plots for analyzing regression model performance and assumptions.
- Create unit tests for quality assurance.
- Enforce code style using pylint and flake8.
- Create and polish example notebooks to demonstrate the functionality of our package.
- Make our package pip installable for easy access.
- Write documentation and tutorial for our package.