
# Functional Specification

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

## Background

Our project aim to create a simple and convenient visualization tool, Betas, for data scientists and data analysts to analyze model performance with visualizations in Python. Users are able to simply run one-line code to generate custom plots for analyzing linear regression model with assumptions diagnostics, computing model scores in binary classification, and presenting performance for principal component analysis (PCA) and clustering. This tool also helps users to fit machine learning models to datasets without a detailed understanding of how the models work. Our *betas* package is pip installable and easy to use by following our example IPython notebooks, in which we are using the Spam, Breast Cancer, Auto and College datasets for demonstration and testing. In addition, we have two interactive web dashboards designed for model diagnostics in linear regression and binary classification.

## User Profile

Our targeted users are data scientists and data analysts, specifically those who do not want to generate or implement visualizations for analyzing model scores on their own. This tool also caters to people who leverage machine learning models, without having a relevant background or understanding the machine learning model concepts, and would like to rely on a pre-configured solution for applying commonly used method. Users should be able to program in Python and how to run command lines. They need to be familiar with installing and importing Python libraries and following examples of

implementing Python functions. Users also need to browse the web to use our interactive model diagnostics dashboard.

## Data Sources

For our package, we aim to easy fit various datasets to the models and the interactive dashboards. Therefore, we collected several datasets with different types of attributes to test our library. The Spam and Auto data are popular datasets for respectively building binary classifier and regression models. Simple logistic regression and linear regression models can be fit on these datasets, and the predictions from models can be used as the data source for demonstrating the *betas* functionality of visualizing model performance and assessing model assumptions in interactive dashboards. We also use the Auto and Breast Cancer datasets for testing and demonstration.

| **Data Sources Structure** | | | | | |
|---|---|---|---|---|---|
| **Dataset** | | Spam | Breast Cancer | Auto | College |
| **Number of Instances** | | 4601 | 569 | 398 | 777 |
| **Number of Attributes** | **Continuous** | 57 | 30 | 5 | 19 |
| | **Discrete / Nominal** | 1 | 1 | 4 | 2 |
| **Source** | | ESL | scikit-learn | ISL | ISL |

## Use cases

The user will be able to install *betas* Python package and import it to make module functionality available. Given predictions from a binary classifier or regression model, the user will be able to create an instance of a plotting class. Plotting classes will include functionality such as binary plots or regression plots. The user will be able to get and set attributes of that class. The user will be able to use and reuse instances of each

plotting class by calling methods from that class to visualize model performance, model fit, or model assumptions with one line of code, in order to compare models and choose the one fitted the best. More details about use cases are listed below.

## ❑ Fit binary classification model

- Objective
    - Fit dataset with a binary classification model
    - Create binary score plots
    - Find the optimal threshold for converting probabilities to binary response
- Interactions
    - Install *betas* Python library and import *binary_score_plot*
    - Load dataset
    - Use library methods and write simple one-line code to generate plots of model scores
    - Run the interactive dashboard to explore the binary modeled scores

## ❑ Fit linear regression model

- Objective
    - Fit dataset with a linear regression model
    - Create plots for dataset overview and linear regression model
    - Assess linear regression model assumptions
- Interactions
    - Install *betas* Python library and import *regression_analysis_plot*
    - Load dataset
    - Use library methods and write simple one-line code to generate plots of linear regression model analysis
    - Use model diagnostics plots or run the interactive dashboard to assess model assumptions

## ❑ Evaluate PCA performance

- Objective
    - Create plots for PCA performance on different dimension reduced
    - Find the optimal dimension with least misclassification error
- Interactions
    - Install *betas* Python library and import *pca_evaluate*

    - Load dataset
    - Use library methods and write simple one-line code to generate plots of PCA performance evaluation
    - The optimal dimension can be observed from plot or by calling related method

## ❑ Evaluate clustering performance

- Objective
    - Fit dataset with k-means clustering with different number of clusters
    - Create plots for clustering performance on different number of clusters
    - Find the optimal number of clusters
- Interactions
    - Install *betas* Python library and import *clustering_evaluate*
    - Load dataset
    - Use library methods and write simple one-line code to generate plots of clustering performance evaluation
    - The optimal number of clusters can be observed from plot or by calling related method