Technology Review

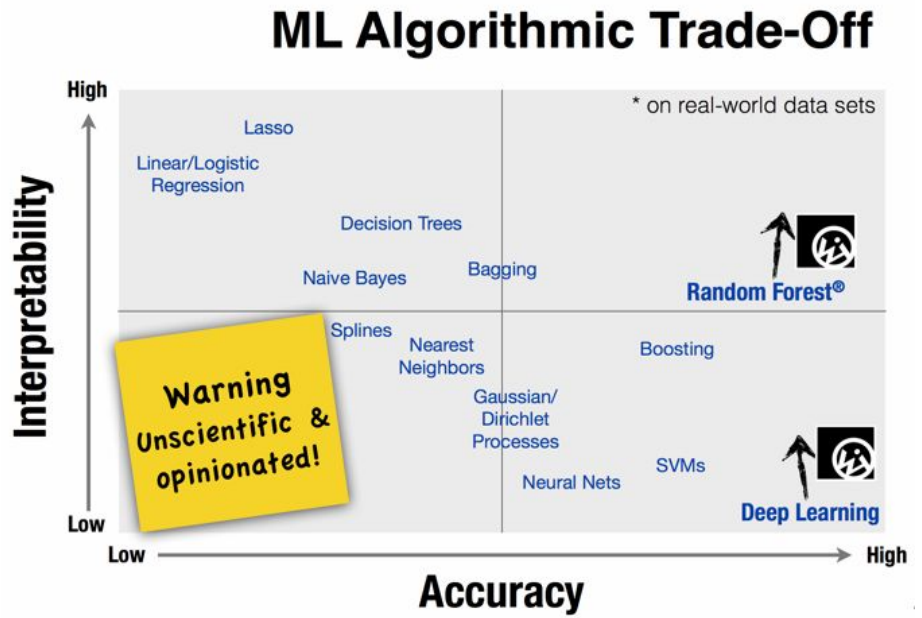
Arjun Singh, Cathy Jia, Joel Stremmel, Monique Bi, Yiming Liu

Background

- Understanding the Data
  - Know the data (summary statistics and visualizations)
  - Visualize the data (Outliers, spread, bivariate relationships, ideal number of principal components, ideal number of clusters for clustering)
  - Clean Data (Missing value, dealing with the outliers, aggregate data)
  - Augment the data (Reduce dimensions, rescale variables)
- Categorize the problem
  - By input (Supervised, unsupervised, reinforcement learning)
  - By output (Regression, Classification, clustering, anomaly detection)
- Understand the Constraints(complexity)
- Find the available Algorithms (models, accuracy)

Commonly Used Machine Learning Algorithms

- Linear Regression
- Logistic Regression
- K-means Clustering
- Principal Component Analysis(PCA)
- Support Vector Machines(SVM)
- Random Forest
- Neural Networks
- …



Picture credit to: https://hackernoon.com/choosing-the-right-machine-learning-algorithm-68126944ce1f

Use case

- You are a data scientist, software developer, or business analyst who works with predictive models.
- You need to analyze the performance of a model. It could be any model: logistic regression, random forest, neural network. All you need is the predicted outputs and true labels/values from your test set.
- Your time is limited and you don't want to write lots of custom code to diagnose the performance of your model
- You find **betas**…
- You follow our example notebooks and call methods from our classes to generate performance plots in one line of code.

```
bsp = binary_score_plot(scores, labels)
bsp.plot_hist()
bsp.plot_pr_by_threshold()
bsp.plot_roc()
```

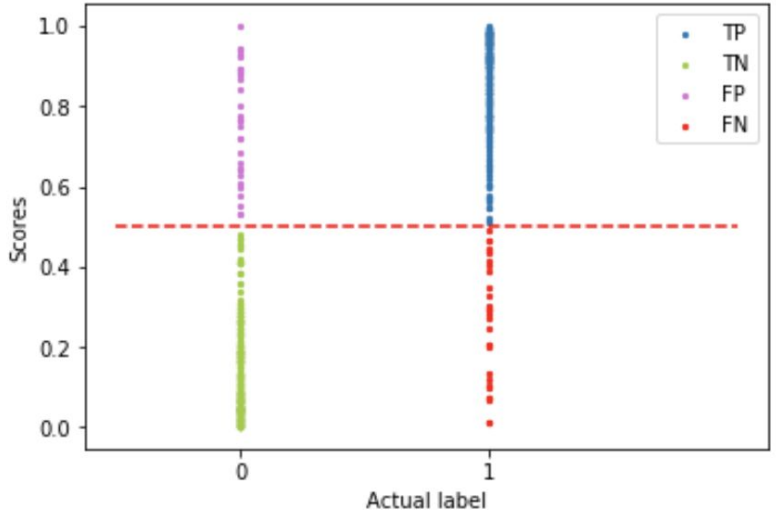
Python Libraries Required

- numpy >= 1.13.1
- pandas >= 0.23.1
- matplotlib >= 2.0.2
- seaborn >= 0.9.0
- scikit-learn >= 0.20.2

Example of using **matplotlib**.pyplot.scatter

- Using pyplot.scatter() to plot predicted scores against actual labels
- Problem is that the x value of each score is either 0 or 1

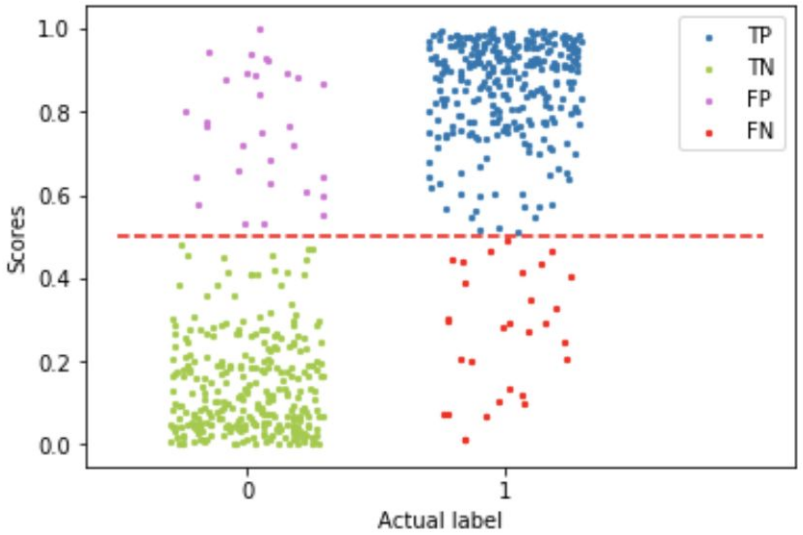
	scores	actual_label
0	0.993285	1.0
1	0.952882	1.0
2	0.932667	1.0
3	0.978911	1.0
4	0.953861	1.0
5	0.999428	1.0
6	0.901714	1.0



Example of using **numpy**.random.uniform

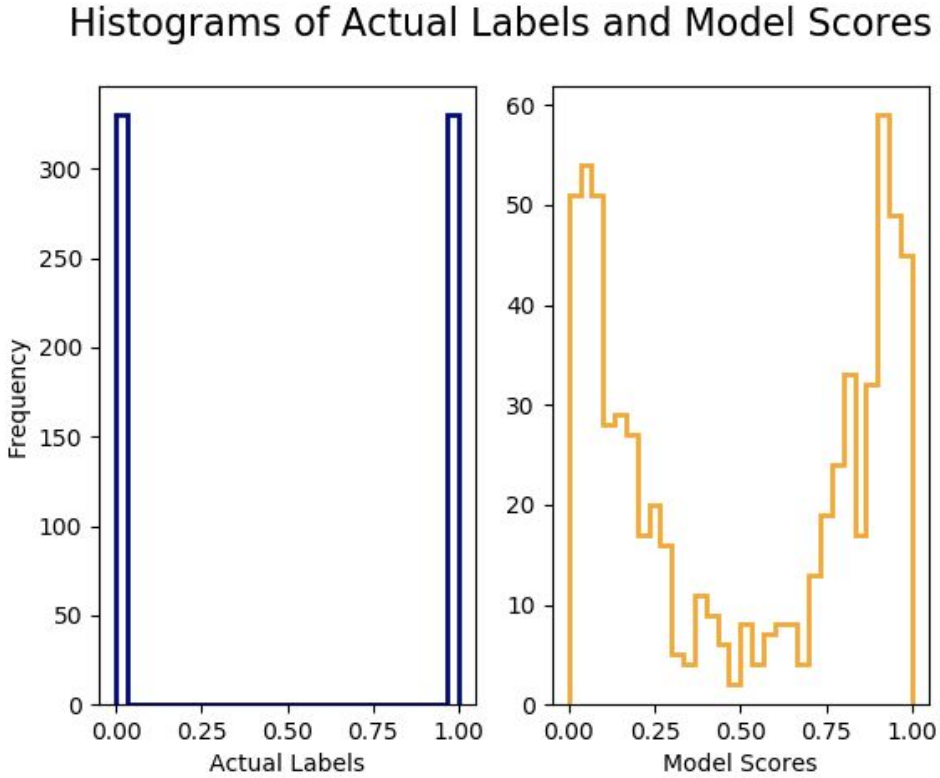
```
df['position'] = df['actual_label'] + \
                 np.random.uniform(low=-0.3, high=0.3, size=len(df))
```

	scores	actual_label	position
0	0.993285	1.0	1.029255
1	0.952882	1.0	0.921030
2	0.932667	1.0	1.067081
3	0.978911	1.0	1.058896
4	0.953861	1.0	1.079901
5	0.999428	1.0	0.794912
6	0.901714	1.0	1.171433



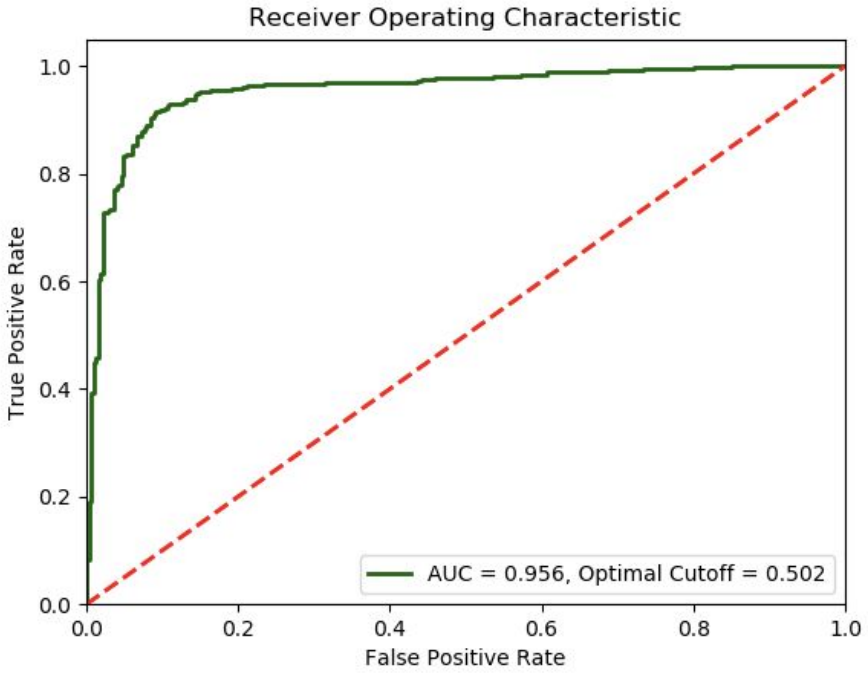
Example of using **matplotlib**.pyplot.hist

- Using pyplot.hist to compare the counts of actual labels and visualize the distribution of scores



Example of using **sklearn**.metrics

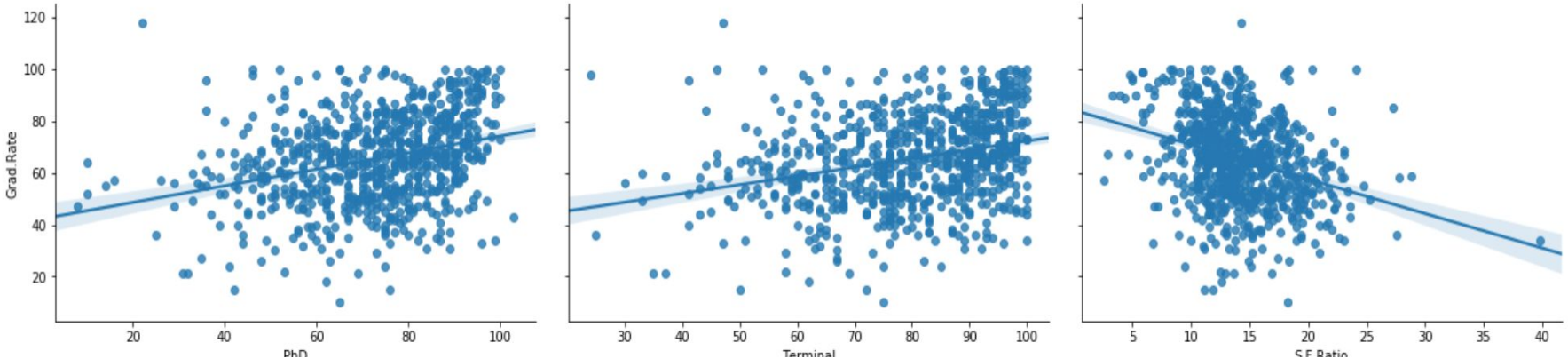
- Using metrics.roc_curve and metrics.auc to visualize the ROC curve and the corresponding area under curve
- It might be useful to also add the optimal threshold for converting scores to labels in the plot



Example 1 of using **seaborn**.pairplot

Scatter plot with linear regression

```
import seaborn as sns
pp = sns.pairplot(college,
                  y_vars=['Grad.Rate'],
                  x_vars=['PhD','Terminal','S.F.Ratio'],kind='reg',aspect=1.5,height=4)
```



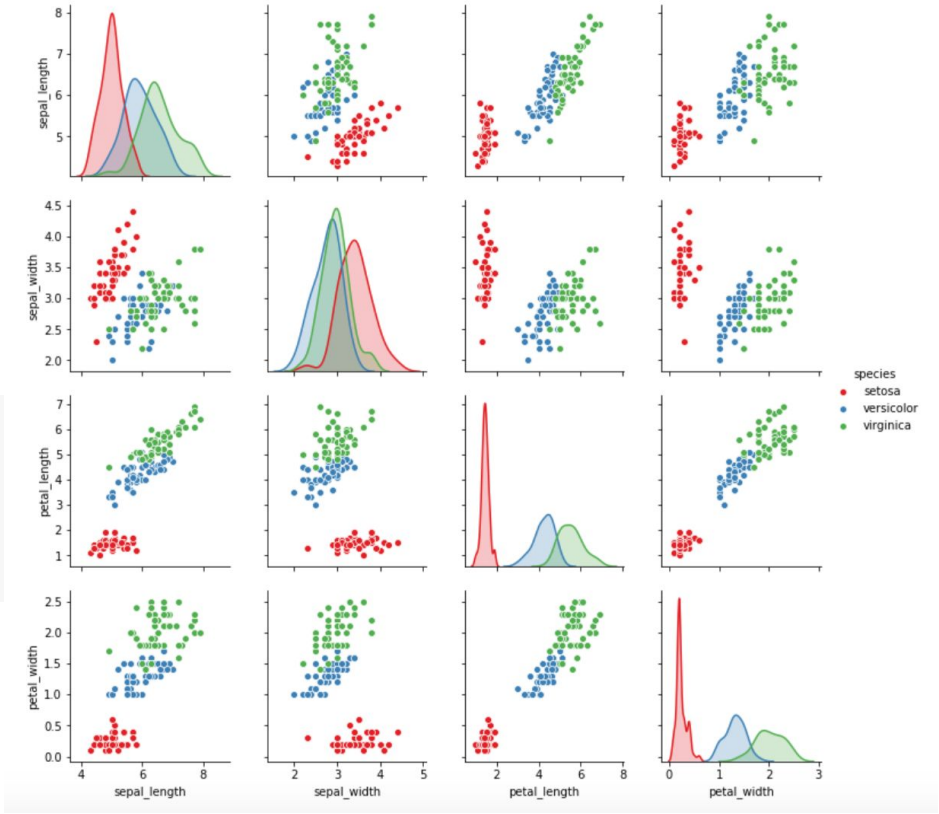
Example 2 of using **seaborn**.pairplot

Scatter matrix plot

- Flexible to select diagonal plots
- Response labels
- Color palette

```
def matrix_plot(dataframe, huelabel):
    '''
    matrix plot
    '''
    sns.pairplot(dataframe, hue=huelabel, palette='Set1')

matrix_plot(iris, 'species')
```



Example 3 of using **seaborn**.heatmap

Correlation Heat Map

- Annotated correlation
- Adjust color hue
- Color palette

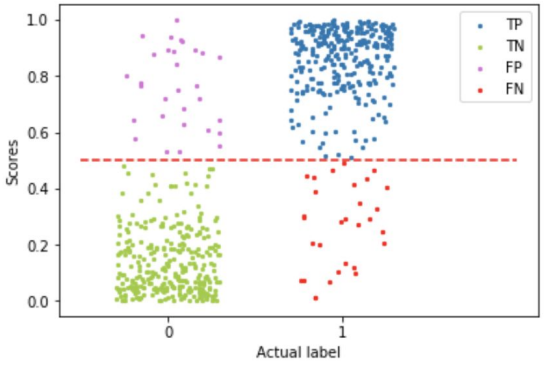

```
corrmat = auto_data.corr()
f, ax = plt.subplots(figsize=(9, 8))
shm2 = sns.heatmap(corrmat, cbar=True, annot=True, fmt='.2f', ax=ax)
```

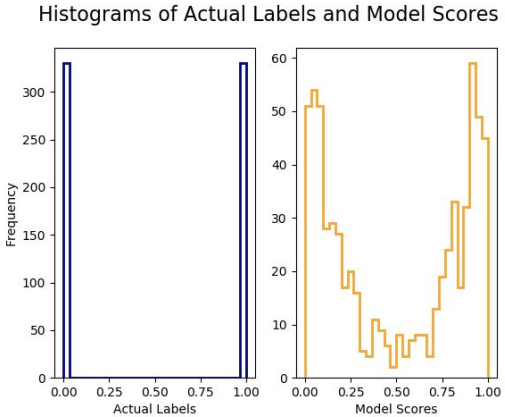
Side-by-side Comparisons

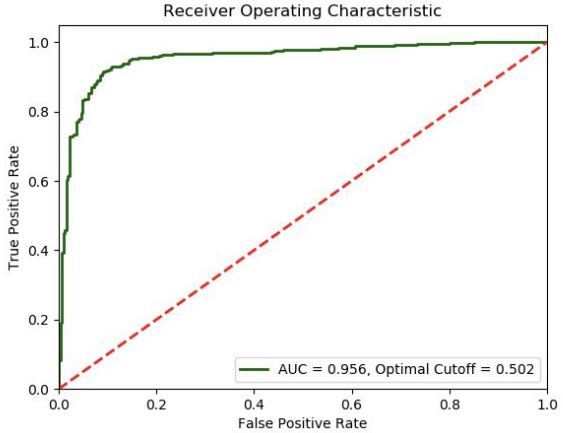
- matplotlib.pyplot.scatter()
- np.random.uniform()

- matplotlib.pyplot.hist()

- sklearn.metrics.roc_curve
- sklearn.metrics.auc





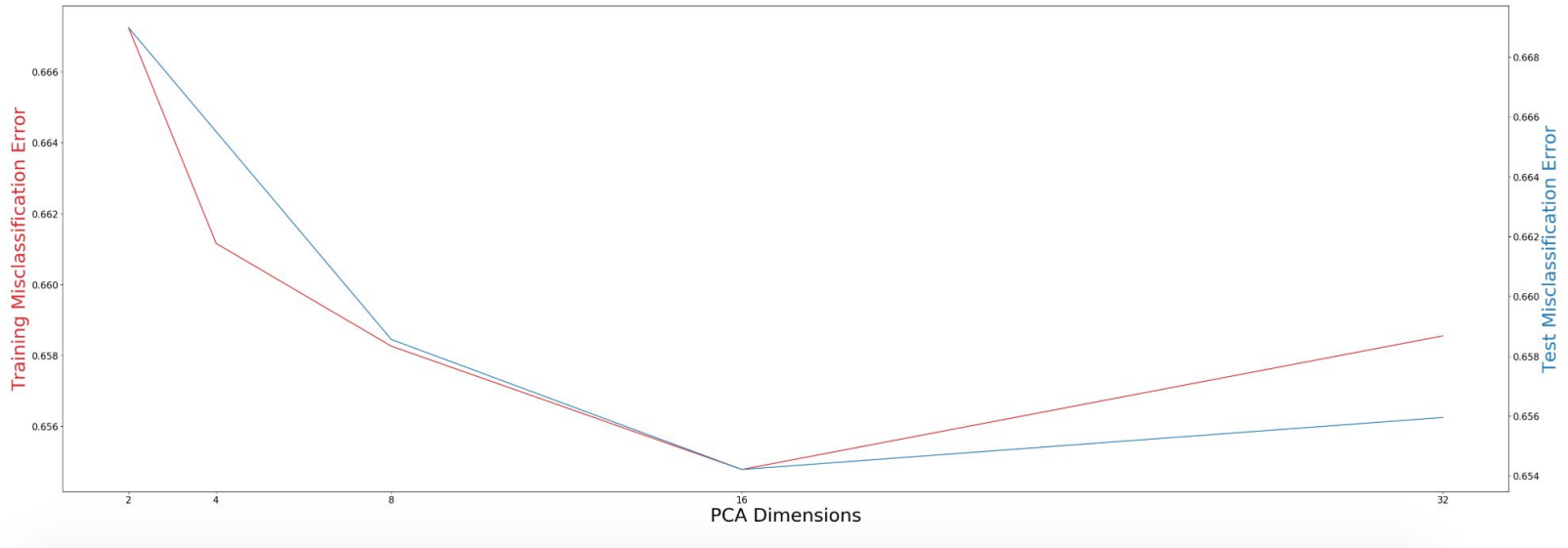


PCA Evaluation

```
In [4]: import PCA_evaluate
        fig = PCA_evaluate.Visualize_PCA(train_features, train_labels, test_features, test_labels)

In [5]: fig
```

Out[5]:




THANK
YOU
GROUP HUG
memegenerator.net