**Machine Learning Engineer Nanodegree**

**Capstone Project**

**Classification of Objects in 3D Point Cloud Data**

Stephan Roth  
March 8, 2017

**I. Definition**

**Project Overview**

Images of things are no longer restricted to 2D pictures. Three dimensional images are becoming more available because of low cost commercial and industrial sensors. Sensors like the Microsoft Kinect allow anyone to create a 3D image of anything [Figure 1].

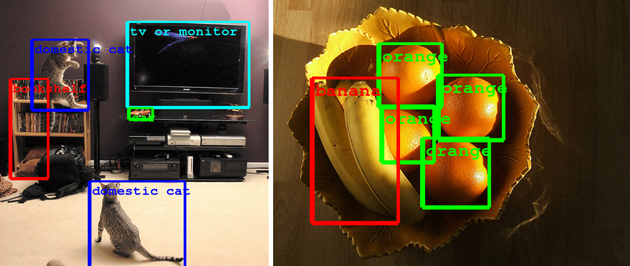


Figure 1: (left)3D Point cloud collected using a Microsoft Kinect (ref. <http://123kinect.com/wp-content/uploads/2012/08/Kinect-1-point-cloud-example.jpg>) (right) Google’s object recognition algorithm (ref. https://www.engadget.com/2014/09/08/google-details-object-recognition-tech/)

Just like with 2D images, it is now possible to do object recognition on 3D structures. Google, Microsoft and other tech companies have long had the ability to take a picture of a scene and identify the objects it contains [Figure 1]. It is now possible to do the same thing with these new 3D images. This has applications to things like gesture recognition for next generation user interfaces or industrial applications such as identifying structures from 3D survey data.

One application of 3D object recognition is the automatic identification of structures in a building survey. Surveys are no longer limited to measuring distances to single points on a job site. It is now possible to capture highly accurate 3D structural data [Figure 2]. The data collected during one of these surveys is what is termed a point cloud. A structure is represented as a dense collection of points in (x,y,z) possibly with brightness and color information. However, just as with images, there is no label attached to each point. The point cloud by itself does not identify walls, chairs, roofs, etc. It would be very useful to have these objects automatically identified within a point cloud. That is the goal of this project, to train a classifier to recognize a 3-dimensional object within a point cloud.

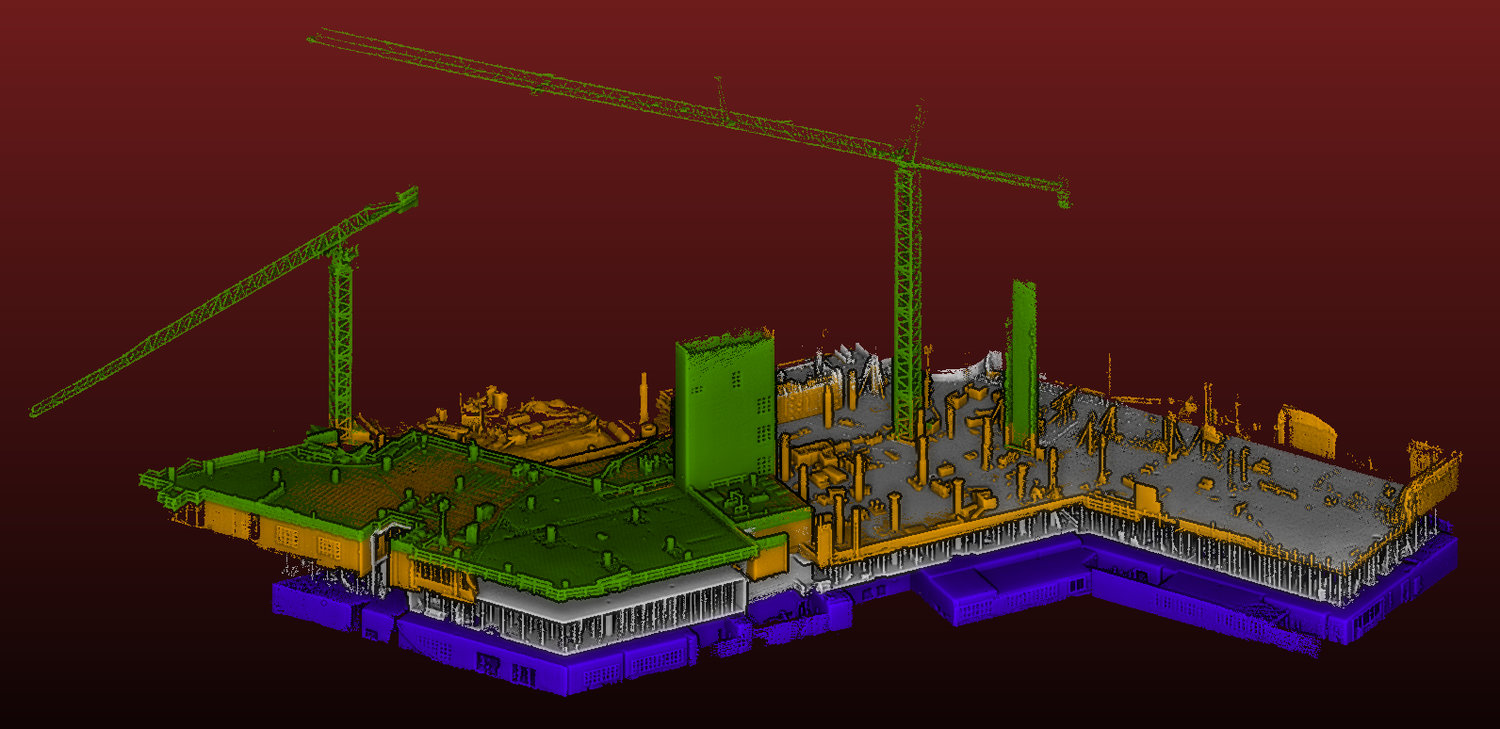


Figure 2: 3D Survey of a building under construction. The data is a 3 dimensional point cloud. In a point cloud, the building is represented by a dense cloud of (x,y,z) points, possibly with color or brightness information (Ref. kaarta.com/gallery)

In this section, look to provide a high-level overview of the project in layman’s terms. Questions to ask yourself when writing this section:

* *Has an overview of the project been provided, such as the problem domain, project origin, and related datasets or input data?*
* *Has enough background information been given so that an uninformed reader would understand the problem domain and following problem statement?*

**Problem Statement**

The purpose of this project is to create a supervised learning classifier to identify and locate an object within a 3D point cloud. Because this project uses supervised learning, a hand labeled data set must be created. The overall process used is the following:

1. Create point clouds of a prototypical scene
2. Hand label a training set
3. Define a set of features to train on
4. Investigate the data to ensure the defined features have the discriminating power necessary to separate the target object from others
5. Using cross validation, split the data in to a test and train set
6. Evaluate a variety of supervised learning algorithms,
7. Select the best performing algorithm
8. Tune the selected algorithm using grid search to optimize its parameters
9. Test the optimized algorithm on other point clouds to see if it can detect the object in untrained data

In this section, you will want to clearly define the problem that you are trying to solve, including the strategy (outline of tasks) you will use to achieve the desired solution. You should also thoroughly discuss what the intended solution will be for this problem. Questions to ask yourself when writing this section:

* *Is the problem statement clearly defined? Will the reader understand what you are expecting to solve?*
* *Have you thoroughly discussed how you will attempt to solve the problem?*
* *Is an anticipated solution clearly defined? Will the reader understand what results you are looking for?*

**Metrics**

To measure the performance of the classifier, the F1 score is used. The classifier we are going to create is binary. A set of features either represents the target object or it does not. The F1 score measures the accuracy of a test with binary results. With a test, there are four possible outcomes: true positive, false positive, true negative and false negative. Given the rate of true positives, true negatives, etc. the precision, recall and F1 score of a test are defined as:

Figure 3: Possible test results are True positive, false positive, false negative and true negative

True Positive

Test Result

False Negative

True Result

True

False

False Positive

True Negative

True

False

The F1 score has a maximum value of 1 if the test results match the true results and a minimum value of 0 if the test is always wrong.

In this section, you will need to clearly define the metrics or calculations you will use to measure performance of a model or result in your project. These calculations and metrics should be justified based on the characteristics of the problem and problem domain. Questions to ask yourself when writing this section:

* *Are the metrics you’ve chosen to measure the performance of your models clearly discussed and defined?*
* *Have you provided reasonable justification for the metrics chosen based on the problem and solution?*

**II. Analysis**

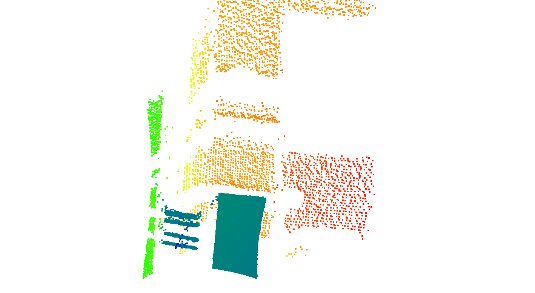
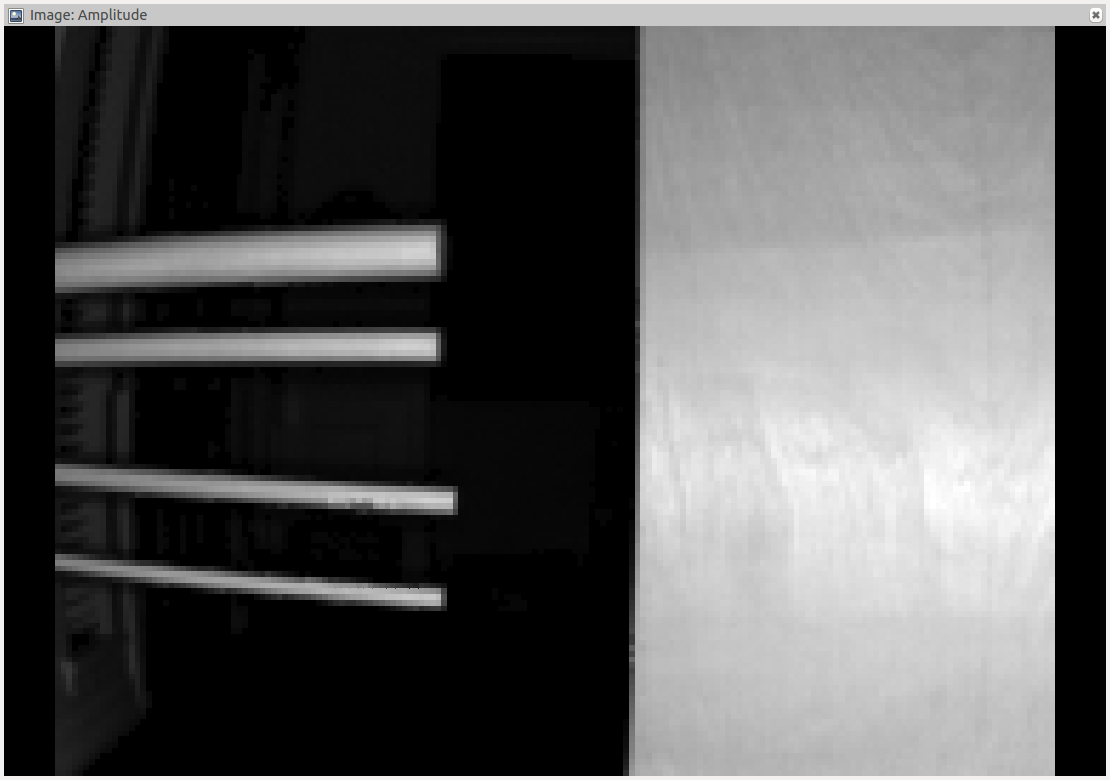
*(approx. 2-4 pages)*

**Data Exploration**

The purpose of this project is to create a classifier which can identify pipe-like structures within a 3D point cloud. Towards this end, I created a prototypical scene with a series of pipes, surrounded by walls and other objects [Figure 4]. The 3D scene was imaged by an IFM 03D303 flash lidar from several different angles. The flash lidar generates a 3D point cloud of the scene. Each pixel in the 3D point cloud has a value (x ,y, z, intensity) [Figure 5]. This raw data cannot be used by the classifier because a single point does not contain enough information to describe what it is a point of. Instead, the features describing a point must come from a region of the point cloud. Given the size of object being classified and the resolution of the flash lidar, a 6cm cube is an appropriate volume. However, because points within the point cloud are not evenly distributed, the size of the cube must be adapted especially in areas of low point density (Zakhor, 2011). In the case where insufficient points are located in a 6cm cube, the size of the cube is increased until a minimum number of points (15) are included. This is to ensure that a sufficient number of points are used to generate accurate statistics of the volume.

Figure 4: 3D Scene with pipes, walls and other objects

Figure 5: Flash Lidar Output. (a) Intensity image, (b), (c) point cloud as viewed from different angles. Points colored by distance from the sensor



(a)

(b)

(c)

Once a representative group of points is gathered, features describing the local geometry are calculated. The features used are common in analysis of point clouds. They use the eigen values and vectors of the covariance matrix in the region around a point. Given the eigen values of the covariance matrix, the geometric features are , These geometric features represent point-ness, surface-ness and linear-ness of the region. In addition, the algorithm contains directional features using the local tangent and normal vectors. The tangent and normal vectors are estimated using the eigen vectors of the largest and smallest eigen values. The sine and cosine of these vectors with respect to the horizontal plane are used, giving a total of 4 directional features. To estimate the confidence in these features, the features are scaled according to the strengths of their corresponding eigen values: . The complete feature vector concatenates the 3 geometric features and 4 directional features for a resulting 7D feature vector.

To train the classifier, a dataset was collected and segmented by hand. The points corresponding to the pipes were identified and labeled as class 1. The points corresponding to everything else are labeled as class 0. For the training set, there are 14,794 points in class 0 and 1,427 points in class 1. The statistics for both classes are shown in the following table:

Table 1: Class 0 (not pipe) feature statistics

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Pointness (x10-4)** | **Surfaceness (x10-4)** | **Linearness (x10-4)** | **Cos tangent** | **Sin tangent** | **Cos normal** | **Sin normal** |
| **Mean** | 1.09 | 6.14 | 4.71 | 0.222 | 0.15 | 0.865 | 0.011 |
| **Std** | 3.02 | 13.9 | 28.1 | 0.303 | 0.34 | 0.248 | 0.200 |
| **Min** | 0.00368 | 0.14 | 0.00843 | 0.0004 | -0.99 | 0.00186 | -0.999 |
| **25%** | 0.0772 | 4.43 | 0.392 | 0.0392 | 0.01 | 0.888 | -0.047 |
| **50%** | 0.0938 | 4.96 | 0.737 | 0.0837 | 0.05 | 0.993 | 0.064 |
| **75%** | 0.591 | 5.81 | 4.29 | 0.217 | 0.15 | 0.997 | 0.112 |
| **Max** | 200 | 1207 | 1641 | 1.00 | 0.99 | 1.00 | 0.989 |

Table 2: Class 1 (pipe) feature statistics

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Pointness (x10-4)** | **Surfaceness (x10-4)** | **Linearness (x10-4)** | **Cos tangent** | **Sin tangent** | **Cos normal** | **Sin normal** |
| **Mean** | 1.40 | 1.93 | 7.20 | 0.894 | 0.011 | 0.299 | -0.000697 |
| **Std** | 0.600 | 2.07 | 3.29 | 0.255 | 0.153 | 0.314 | 0.102 |
| **Min** | 0.270 | 0.13 | 0.24 | 0.00576 | -0.930 | 0.0114 | -0.859 |
| **25%** | 0.600 | 0.64 | 4.64 | 0.998 | -0.0284 | 0.0720 | -0.00349 |
| **50%** | 0.940 | 1.23 | 7.39 | 0.999 | 0.00119 | 0.189 | 0.00188 |
| **75%** | 1.37 | 2.01 | 10.4 | 0.999 | 0.0309 | 0.357 | 0.0117 |
| **Max** | 4.16 | 10.02 | 12.5 | 1.00 | 0.980 | 1.00 | 0.578 |

Table 3: Class 0 (not object of interest) Feature Statistics.

Comparing the data between classes, class 1 does seem to occupy a distinct region. In particular, because the pipes are very linear in shape, one would expect the “Linearness” feature to stand out, and it does. For class 0, the mean is 4.71 and std is 28.1. For class 1, the mean is 7.20 and std is 3.29. For class 1, the mean linearness is larger and standard deviation is smaller This indicates that the class 1 linearness forms a tight group higher in value than the class 0 linearness values.

The tangent data should also clearly identify the pipe objects because they were oriented horizontally. This shows up in the statistics. For a horizontal line, the cosine of the tanget should be approximately 1 and the sine approximately 0. This holds true in the data. The cosine for class 1 has a mean of 0.894 and std 0.255, indicating horizontall. The sine data for class 1 has mean 0.011 and std 0.153, indicating horizontal. Both of these values are tightly grouped and distinct when compared to the class 0 data. The cosine of the tanget for class 0 has mean 0.222 and std 0.303. This data tends to be lower in value than the class 1 data. The sine of the tangent for class 0 has mean 0.15 and std 0.34. This data tends to be higher in value and more spread out than the class 1 data.

The distinctiveness of the class 1 data holds across the other features as well. For pointness, class 0 Q3 < class 1 Q1. For surfaceness, class 1 Q3 < class 0 Q1. For cos normal, class 1 Q3 < class 0 Q1.

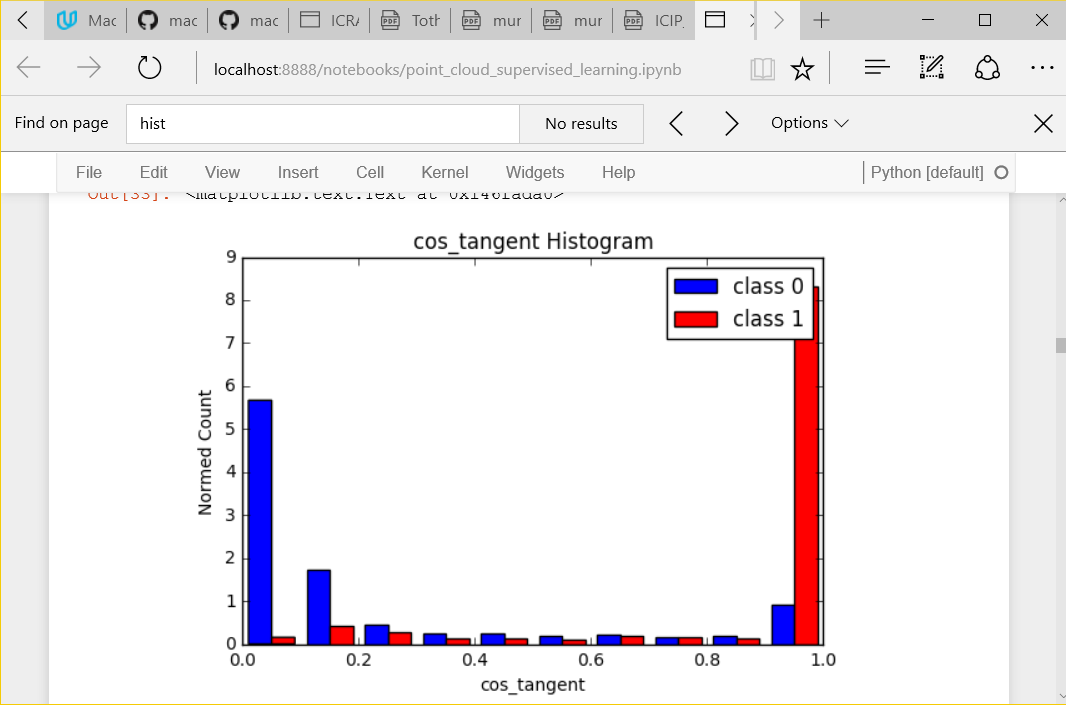
In this section, you will be expected to analyze the data you are using for the problem. This data can either be in the form of a dataset (or datasets), input data (or input files), or even an environment. The type of data should be thoroughly described and, if possible, have basic statistics and information presented (such as discussion of input features or defining characteristics about the input or environment). Any abnormalities or interesting qualities about the data that may need to be addressed have been identified (such as features that need to be transformed or the possibility of outliers). Questions to ask yourself when writing this section:

* *If a dataset is present for this problem, have you thoroughly discussed certain features about the dataset? Has a data sample been provided to the reader?*
* *If a dataset is present for this problem, are statistics about the dataset calculated and reported? Have any relevant results from this calculation been discussed?*
* *If a dataset is* ***not*** *present for this problem, has discussion been made about the input space or input data for your problem?*
* *Are there any abnormalities or characteristics about the input space or dataset that need to be addressed? (categorical variables, missing values, outliers, etc.)*

**Exploratory Visualization**

To verify that the class 0 and 1 objects form distinct clusters, histograms were made of the linearness and cosine tangent features. The results are shown in [Figure 6] and [Figure 7]. As expected, the class 0 and 1 distributions are distinct and agree with the data from the previous tables. For the cosine tangent feature, the class 1 data is clustered primarily around a value of 1.0, indicating horizontalness. The class 0 data is primarily clustered around 0. This is probably due to the fact that most of the other point cloud data is from vertical walls. For the linearness feature, the class 1 values tend to be larger than the class 0 values. This is as expected because the pipes are very linear in appearance. Again, the class 0 points will be primarily from planar surfaces, so the linearness feature should be lower.

Figure 6: Histogram of the cosine tangent feature for class 0 and 1.



In addition 2 dimensional scatter plots were created for different features [Figure 8]. In the plot of surfaceness vs linearness, the class 1 points are clustered in the upper left hand corner, indicating high linearness and low surfaceness. The class 0 points are less linear and more surface-like. In the plot of linearness vs cos\_tangent, the class 1 points are mainly to the right indicating horizontal linearity. The class 0 points are primarily on the left. Again, the class 1 objects are clustered together and are somewhat distinct from the class 0 objects.

Figure 7: Histogram of linearness feature for class 0 and 1.

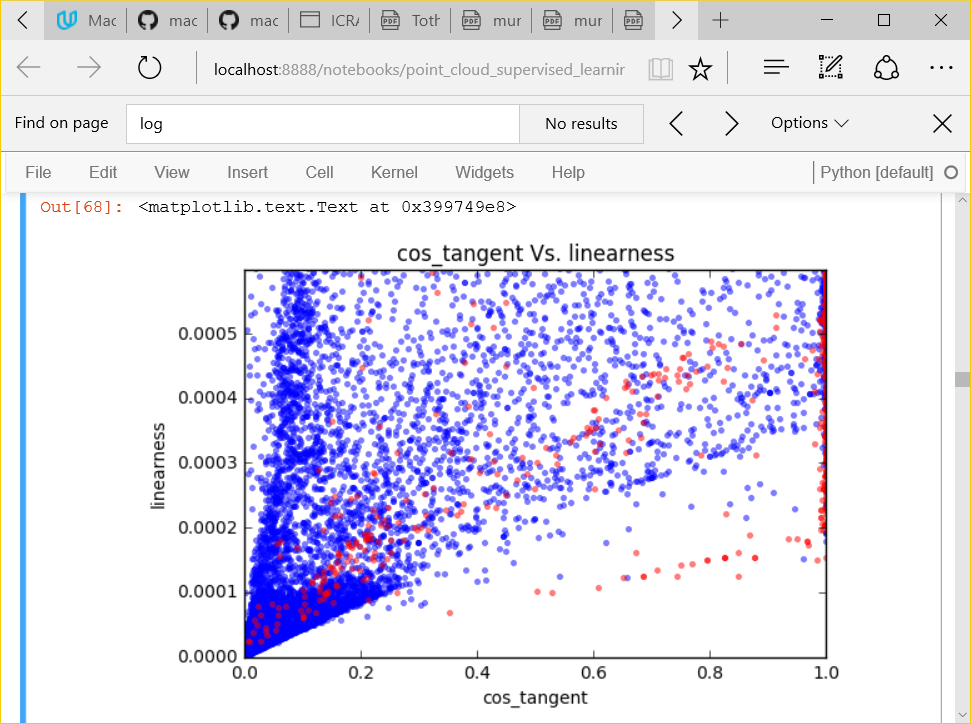
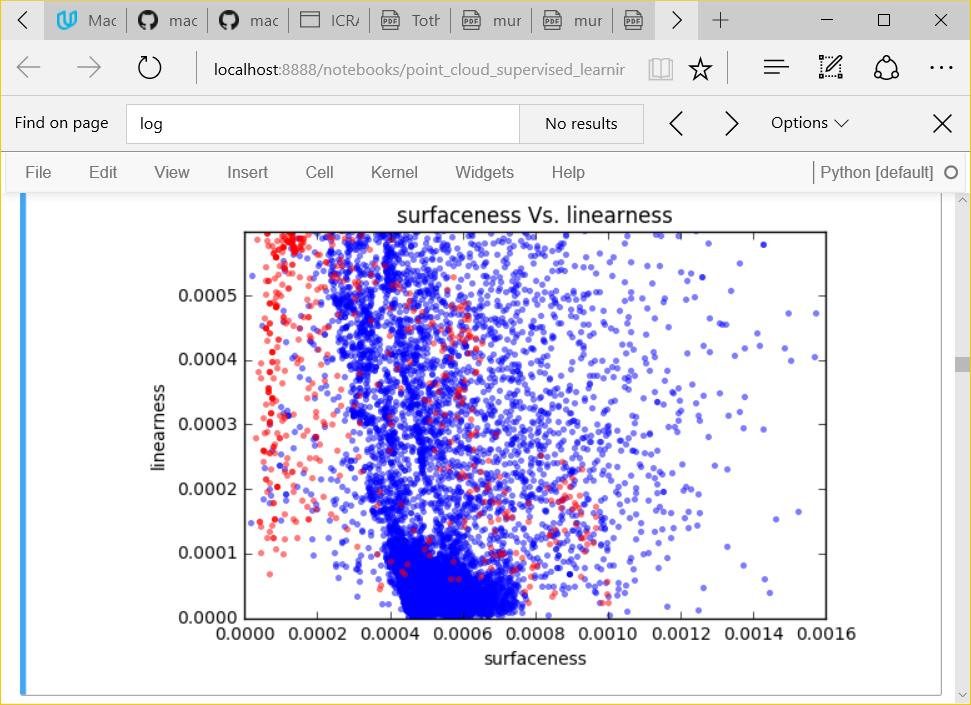
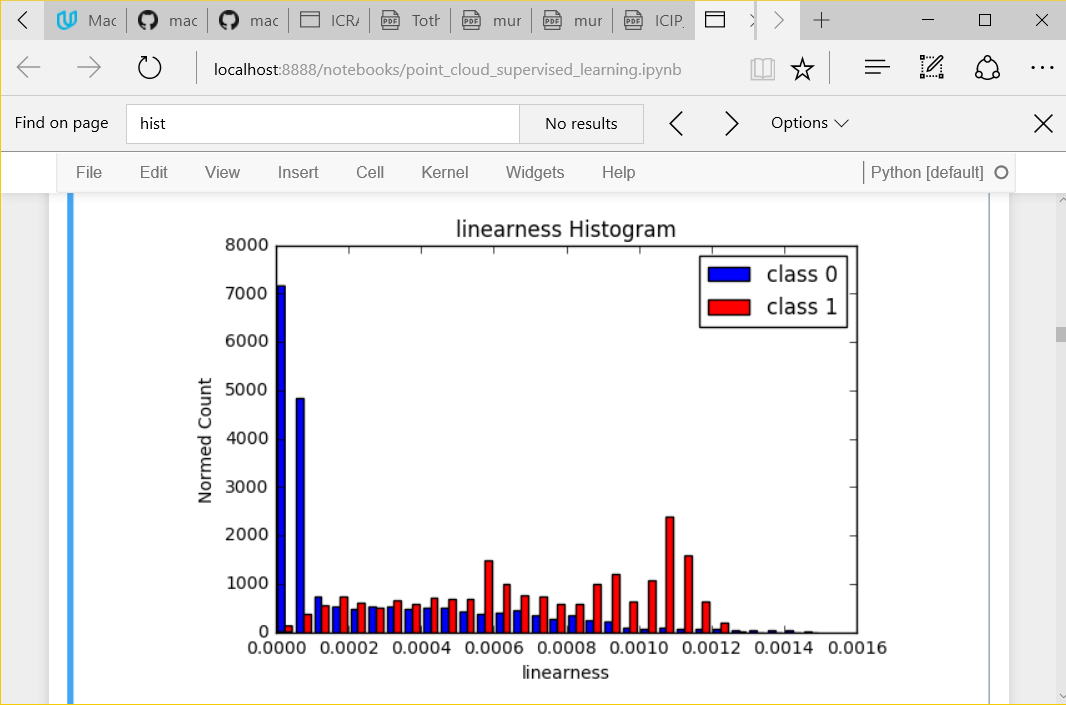


Figure 8: Scatter plots showing the class 0 (blue) and class 1 (red) feature distributions. The two classes mostly fall into two distinct clusters

In this section, you will need to provide some form of visualization that summarizes or extracts a relevant characteristic or feature about the data. The visualization should adequately support the data being used. Discuss why this visualization was chosen and how it is relevant. Questions to ask yourself when writing this section:

* *Have you visualized a relevant characteristic or feature about the dataset or input data?*
* *Is the visualization thoroughly analyzed and discussed?*
* *If a plot is provided, are the axes, title, and datum clearly defined?*

**Algorithms and Techniques**

In the sciKit Learn documentation (scikit-learn, 2016), in choosing the right estimator, it is suggested for problems with < 100K samples to use the linear SVC and if this fails to move onto K Nearest neighbors and Ensemble classifiers. In addition, after analyzing the data, there was no obvious boundary separating the different classes, so a classifier that could handle complex decision boundaries may perform better. For this project, I evaluated the following supervised learning algorithms: Support Vector Classifier, Ada Boost, K Nearest Neighbors and Gaussian Naïve Bayes.

Looking at the data we are using for training, there are 14,794 points in class 0 and 1,427 points in class 1. The feature vector used has 7 dimensions. With only 15,000 training points, the 7 dimensional feature space will be sparsely populated. For each classifier, a training subset of calculated features and hand labels will be fit. To evaluate each classifier, a subset of test data features will be used to predict a class ID. The predicted class ID will be compared to the hand labeled class ID and scored according to its F1 score.

The first classifier evaluated was the Support Vector Classifier. Support vector classifiers have been used in analyzing the solvency of companies by banks / lenders (Laura Auria, 2008). Balance sheet data from the companies is used as features and the classifier is used to predict the insolvency of a business within 3 years.

The strength of the support vector classifier lies in its ability to use kernels to create non-linear boundaries between classes.

The second classifier evaluated was AdaBoost. In industry, AdaBoost has been used in predicting energy consumption in the steel industry (Rui Hu, 2013). The authors are trying to predict the energy consumption in the steel industry. Because energy consumption in the steel industry is tied to current global demand for steel but also government incentives for steel production, traditional modeling methods fail. The authors use a neural network based Ada Boost algorithm.

The strength of the AdaBoost classifier is that it is one of the best "out of the box" classifiers, and tends to be less susceptible to over fitting. It is also computationally efficient. A disadvantage of AdaBoost is that it can be susceptible to outliers. The algorithm can spend too much time trying to fit a data point that is really just noise.

AdaBoost is a good candidate for this problem due to its versatility and computational efficiency.

K nearest neighbors has been used in fault detection in industrial plants (Fellipe do Prado Arruda e. a., 2014). The strength of the K nearest neighbor is that the training time is efficient. The algorithm simply needs to store all of the training data. The algorithm can also learn arbitrarily complex classification boundaries. However, a weakness is the classification time is costly because the algorithm needs to find the k nearest neighbors by sifting through the data. Also, because all of the training data needs to be stored, the memory requirements increase as the size of the learning set increases.

K nearest neighbors is a good candidate for this problem because of its ability to handle arbitrarily complex classification boundaries.

Naive Bayes classifier has been used in text classification (Andrew McCallum, 1998). Here, an algorithm is trying to classify a document based on an analysis of the text. For example, it may classify a document as being about tennis, physics, sports, etc. The naive bayes classifier is trained using a dictionary of words found to be highly discriminative of these categories.

The strength of the Naive Bayes classifier is that it can is very fast to learn and classify data. It has also been shown to work on a variety of machine learning problems. The speed at which it can learn and classify data comes from the assumption that probabilities are conditionally independent within the model. This is both a strength and weakness. For classification problems where things are conditionally dependent, the Naive Bayes algorithm will fail to correctly classify items. For example, a Naive Bayes classifier cannot learn XOR(x1, x2). Naive Bayes is a good candidate for this problem due to its computational efficiency.

In this section, you will need to discuss the algorithms and techniques you intend to use for solving the problem. You should justify the use of each one based on the characteristics of the problem and the problem domain. Questions to ask yourself when writing this section:

* *Are the algorithms you will use, including any default variables/parameters in the project clearly defined?*
* *Are the techniques to be used thoroughly discussed and justified?*
* *Is it made clear how the input data or datasets will be handled by the algorithms and techniques chosen?*

**Benchmark**

As mentioned before, each classifier will be ranked according to its F1 score on the testing data. Each classifier will be fit using a training subset of the hand labeled data. The classifier will then be ranked according to its F1 score on its predictions on the hand labeled test data. The training and testing data are independent subsets of the hand labeled data.

In this section, you will need to provide a clearly defined benchmark result or threshold for comparing across performances obtained by your solution. The reasoning behind the benchmark (in the case where it is not an established result) should be discussed. Questions to ask yourself when writing this section:

* *Has some result or value been provided that acts as a benchmark for measuring performance?*
* *Is it clear how this result or value was obtained (whether by data or by hypothesis)?*

**III. Methodology**

*(approx. 3-5 pages)*

**Data Preprocessing**

Fo

In this section, all of your preprocessing steps will need to be clearly documented, if any were necessary. From the previous section, any of the abnormalities or characteristics that you identified about the dataset will be addressed and corrected here. Questions to ask yourself when writing this section:

* *If the algorithms chosen require preprocessing steps like feature selection or feature transformations, have they been properly documented?*
* *Based on the* ***Data Exploration*** *section, if there were abnormalities or characteristics that needed to be addressed, have they been properly corrected?*
* *If no preprocessing is needed, has it been made clear why?*

**Implementation**

Foo

In this section, the process for which metrics, algorithms, and techniques that you implemented for the given data will need to be clearly documented. It should be abundantly clear how the implementation was carried out, and discussion should be made regarding any complications that occurred during this process. Questions to ask yourself when writing this section:

* *Is it made clear how the algorithms and techniques were implemented with the given datasets or input data?*
* *Were there any complications with the original metrics or techniques that required changing prior to acquiring a solution?*
* *Was there any part of the coding process (e.g., writing complicated functions) that should be documented?*

**Refinement**

Foo

In this section, you will need to discuss the process of improvement you made upon the algorithms and techniques you used in your implementation. For example, adjusting parameters for certain models to acquire improved solutions would fall under the refinement category. Your initial and final solutions should be reported, as well as any significant intermediate results as necessary. Questions to ask yourself when writing this section:

* *Has an initial solution been found and clearly reported?*
* *Is the process of improvement clearly documented, such as what techniques were used?*
* *Are intermediate and final solutions clearly reported as the process is improved?*

**IV. Results**

*(approx. 2-3 pages)*

**Model Evaluation and Validation**

Foo

In this section, the final model and any supporting qualities should be evaluated in detail. It should be clear how the final model was derived and why this model was chosen. In addition, some type of analysis should be used to validate the robustness of this model and its solution, such as manipulating the input data or environment to see how the model’s solution is affected (this is called sensitivity analysis). Questions to ask yourself when writing this section:

* *Is the final model reasonable and aligning with solution expectations? Are the final parameters of the model appropriate?*
* *Has the final model been tested with various inputs to evaluate whether the model generalizes well to unseen data?*
* *Is the model robust enough for the problem? Do small perturbations (changes) in training data or the input space greatly affect the results?*
* *Can results found from the model be trusted?*

**Justification**

Foo

In this section, your model’s final solution and its results should be compared to the benchmark you established earlier in the project using some type of statistical analysis. You should also justify whether these results and the solution are significant enough to have solved the problem posed in the project. Questions to ask yourself when writing this section:

* *Are the final results found stronger than the benchmark result reported earlier?*
* *Have you thoroughly analyzed and discussed the final solution?*
* *Is the final solution significant enough to have solved the problem?*

**V. Conclusion**

*(approx. 1-2 pages)*

**Free-Form Visualization**

Foo

In this section, you will need to provide some form of visualization that emphasizes an important quality about the project. It is much more free-form, but should reasonably support a significant result or characteristic about the problem that you want to discuss. Questions to ask yourself when writing this section:

* *Have you visualized a relevant or important quality about the problem, dataset, input data, or results?*
* *Is the visualization thoroughly analyzed and discussed?*
* *If a plot is provided, are the axes, title, and datum clearly defined?*

**Reflection**

Foo

In this section, you will summarize the entire end-to-end problem solution and discuss one or two particular aspects of the project you found interesting or difficult. You are expected to reflect on the project as a whole to show that you have a firm understanding of the entire process employed in your work. Questions to ask yourself when writing this section:

* *Have you thoroughly summarized the entire process you used for this project?*
* *Were there any interesting aspects of the project?*
* *Were there any difficult aspects of the project?*
* *Does the final model and solution fit your expectations for the problem, and should it be used in a general setting to solve these types of problems?*

**Improvement**

Foo

In this section, you will need to provide discussion as to how one aspect of the implementation you designed could be improved. As an example, consider ways your implementation can be made more general, and what would need to be modified. You do not need to make this improvement, but the potential solutions resulting from these changes are considered and compared/contrasted to your current solution. Questions to ask yourself when writing this section:

* *Are there further improvements that could be made on the algorithms or techniques you used in this project?*
* *Were there algorithms or techniques you researched that you did not know how to implement, but would consider using if you knew how?*
* *If you used your final solution as the new benchmark, do you think an even better solution exists?*

**Before submitting, ask yourself. . .**

* Does the project report you’ve written follow a well-organized structure similar to that of the project template?
* Is each section (particularly **Analysis** and **Methodology**) written in a clear, concise and specific fashion? Are there any ambiguous terms or phrases that need clarification?
* Would the intended audience of your project be able to understand your analysis, methods, and results?
* Have you properly proof-read your project report to assure there are minimal grammatical and spelling mistakes?
* Are all the resources used for this project correctly cited and referenced?
* Is the code that implements your solution easily readable and properly commented?
* Does the code execute without error and produce results similar to those reported?

# References

Fellipe do Prado Arruda, e. a. (2014). Fault Detection in Industrial Plant Using K-Nearest Neighbors with Random Subspace Method. *Proceedings on the International Conference on Artificial Intellicgence (ICAI). .* WorldComp.

Fellipe do Prado Arruda, V. d. (n.d.).

Laura Auria, R. A. (2008). Support Vector Machines (SVM) as a Technique for Solvency Analysis. *Discussion Papers of DIW Berlin 811* (p. 16). Berlin: DIW Berlin, German Institute for Economic Research.

Rui Hu, Q. (2013). Prediction of Energy Consumption in Steel Enterprises based on BP Adaboost Algorithm. *of the Sixth International Conference on Management Science and Engineering Management. Lecture Notes in Electrical Engineering, vol 185* (pp. 411-419). London: Springer.

scikit-learn. (2016). *Choosing the Right Estimator*. Retrieved from SciKit Learn: http://scikit-learn.org/stable/tutorial/machine\_learning\_map/index.html

Zakhor, X. S. (2011). Fast approximation for geometric classification of LiDAR returns. *18th IEEE International Conference on Image Processing* (pp. 2925-2928). Brussels: IEEE.