Kaggle Competitions:
Spooky Author Identification
Statoil/C-CORE Iceberg Classifier Challenge

John Stein[1]*

Executive Summary Briefly explain the kaggle competition and datamining as a solution. Briefly explain each problem, the solution, and results.

Keywords
Keyword1 — Keyword2 — Keyword3

[1]*Computer Science, School of Informatics, Computing, and Engineering, Indiana University, Bloomington, IN, USA*
***Corresponding author**: jodstein@iu.edu

Contents

1. Introduction

- Briefly, but more completely than in the Executive Summary, explain the kaggle competition – how to get to it, *etc.*.
- Discuss datamining abstractly and how it fits as a solution to the kaggle competition.

- Briefly, but more completely than in the Executive Summary, explain the two problems in two different subsections.

Author Identification
What is the problem to be solved? What is the data? How is goodness quantified? This should not be too technical, but we can say, for example, given three authors $A = \{a_1, a_2, a_3\}$ and selections of their individiually corresponding works $S_{a_1}, S_{a_2}, S_{a_3}$, we are constructing a probability mass function f over A that is applied to a text t written by one of the authors, but unlabled:

$$f(A|t,S) = \{p_{a_1}, p_{a_2}, p_{a_3}\} \tag{1}$$

A text is simply a passage from one of the author's works:

> Once upon a midnight dreary, while I pondered, weak and weary, Over many a quaint and curious volume of forgotten lore—While I nodded, nearly napping, suddenly there came a tapping, As of some one gently rapping, rapping at my chamber door."'Tis some visitor," I muttered, "tapping at my chamber door— Only this and nothing more."

from Edgar Allen Poe's, *The Raven*. You must find out whether it's only prose – and how much of the original structure is maintained. Likely you will not take in text directly – so this must be described.

Statoi/C-CORE Iceberg Classifier Challenge
Use footnotes sparingly, but you must use at least two in this document[1].

---

[1]A footnote provides some ancillary information that's somewhat misplaced in the text, *i.e.*, the reader can skip it without missing any critical

2. Datamining

- What is datamining? Datamining seeks to inform decision making by answering questions about, exploring or discovering relationships in, or providing annotation of data.

- What does it yield? Datamining yields information that is not possible or practical to directly observe, based on data that is practical to observe. This information may be predicted trends in human behavior, perceived relationships or clusters within high-dimension data, or a prediction of future events. In all cases, the yielded information is deemed valuable to someone who is then enabled to make decisions based on that information.

- What are the general steps?

    - The first (and most difficult) step in datamining is developing the problem statement: What decision or action is the datamining effort required to inform or enable, and what must the datamining effort yield in order to satisfy that requirement?

    - The second step is data acquisition. Acquisition of data is non-trivial. It can be difficult, expensive, and in some cases even prohibited or restricted by law. Therefore, the miner must be intentional about pursuing the acquisition of data that is both needed (from the problem statement) and available, whether by manual observation, survey, or automated sensing/recording.

    - The third (and most time consuming) step in datamining is data pre-processing. The data must be analyzed, including identifying unknown, missing, or outlier records, examining attributes and their domains, and visualizing the data for first-order behaviors, trends, and distributions. The data must then be integrated and/or cleaned, including dealing with unknown, missing, or bad data, enriching the data from other sources, and/or transforming the data to work better in the algorithm.

    - The fourth step is to actually mine the data (i.e. answer questions, explore or discover relationships, or annotate). The methodology, assumptions, models, and parameters of how the mining is performed can depend on the objective, the data itself, and the miner themselves.

    - The fifth and final step in datamining is to interpret the output of the mining step and either validate that it meets the stated objective/requirement of the problem statement, or determine that changes to the pre-processing or mining steps are warranted and start again with a modified approach.

- What is clustering *vs.* classification? Classification refers to the task of predicting the label of some unobserved target attribute given a set of observed attributes for some record. The label is generally a single member of a finite list of possible labels. Clustering refers to the task of identifying or discovering relationships among data. In other words, are there properties or measures for which subsets of data can be interpreted as similar or dissimilar in a meaningful way? Clustering aims to discover these 'clumps' or similar data as well as the properties or measurements from which the similarity can be determined. It is useful to think of Classification as a supervised learning task because it is notionally possible to produce a training set with labeled records, and Clustering as an unsupervised learning task because the number and meaning of the discovered clusters is not known apriori (if they were, they could be labeled and it would be called Classification).

- What is a loss function? A loss function is a function which attempts to assign a real value to some undesirable property of an intermediate outcome during the mining step. This value (known as cost or loss) may represent error, variance, or bias of a model, clustering, or classifier. The actual value of loss is usually not meaningful. Instead, the goal is generally to minimize the loss over some parameter or variable during an optimization step.

Data Preprossing

- What are the steps, and what challenges does each present? The first step in data preprocessing is data analysis. Data analysis can include examining histograms, bar/pair-plots, frequency tables, etc. The goal is to get familiar with the data attributes and their domains, frequency of values, trends and distributions, and first-order behaviors. From the analysis, the two primary remaining tasks are selection and modification. Selection (i.e. sampling and attribute selection) refers to deciding which subset or records and/or which subset of attributes for each record to use for the mining task. Modification (i.e. aggregation, dimensionality reduction, discretization, and transformation) refers to changing existing, creating new, or transforming, attributes and their values.

    - Sampling refers to choosing a subset of the total data from which to develop the analysis technique - often because working on the entire dataset would result in more resource investment than it would improve the performance of the technique. The challenge with sampling is that, especially for high-dimensionality data, the data data can quickly become sparse, resulting in a greater potential for over-fitting due to the small data size and high dimensionality (degrees of freedom).

    - Attribute Selection refers to selecting which at-

---

information. On the other hand, a dedicated reader appreciates the extra information. A footnote might clarify information in an informal way. Although possible, putting mathematics in a footnote is a bit odd.

tributes to include in the analysis and which to exclude, based on some preliminary analysis. Manual attribute selection (or de-selection) is challenging because humans can only perceive date in two, three, or perhaps four dimensions. One can only reasonably expect to act on the most obvious of attribute properties without impacting the analysis in unexpected ways.

– Aggregation refers to manually grouping several attributes or attribute values together, preferably while preserving some higher-level relationship among those being grouped. Aggregation can be relatively straightforward in some cases, but in other cases it can be very difficult to group attributes or values in a way that recognizes the essential common property of its members.

– Dimensionality Reduction refers to the creation of fewer new attributes which are combinations of the many old attributes. Two common methods include Principal Component Analysis (PCA) and Neural Networks. The primary challenge with dimensionality reduction is that the meaning of the original attributes are all but lost as the data are mapped to the new, reduced feature space.

– Discretization refers to the assignment of values from a continous-valued attributes into one of finite-many bins.

– Transformation refers to the mapping of old attribute values to new attribute values - generally reversible.

Finally, data with unknown, erroneous, or missing values must be handled. Some of these handling techniques may be implemented in the mining algorithm(s) themselves, but some may be performed prior to the mining step. Unknown/missing data handling is more challenging for some algorithms (i.e. regression) than it is for others (i.e. decision trees). Some algorithms need each record to have good data, or they will simply not work. Other algorithms can be made to handle unknown/missing data in a reliable way and achieve good performance. Still other algorithms can operate on records having missing data with no extra effort. The primary challenge, especially for high-dimensionality data, is that the impact of the handling technique can be different for every attribute, and must be understood and chosen carefully.

• What is the general load (time, space, $) for preprocessing? Computationally, preprocessing may consume $O(n)$ or $O(nlogm)$ time where n is the number of records to process and m may be search steps if matching, inclusion, or sorting operations are required. Preprocessing may consume anywhere from constant space to $O(n)$ space, if the records are processed one at a time and stored back into their original location, or copies

are made. Practically speaking, preprocessing can consume upwards of 80% of the labor hours, and therefore 80% of the funding, of the overall datamining effort. That is because it is the most human-intensive step of the process. Automation of the preprocessing step, although possible, is of limited benefit if the datamining problems vary significantly in size, scope, data representation, purpose, etc. The ability to predict the preprocessing impacts of data across problem spaces is indeed an analytics problem unto itself.

Mining, Interpretation, and Action
• Briefly discuss the top 10 algorithms.
• Does datamining tell us what to do?
• What are some new types of problems in datamining?

Top Ten Algorithms in Datamining   Below is a brief overview of the top 10 algorithms in Datamining. [1]

• **C4.5:** C4.5 is an algorithm that can be used for decisions trees or rulesets. For trees, tt essentially builds a decision tree by recursively splitting the data on attributes until the data is fully explained by the tree. It then prunes the tree by calculating a binomial error estimate at each of the leaves, and comparing that with the error that would be introduced by collapsing to the parent node, and repeating until complete. For rulesets, it similarly generates rules representing every complete path through the notional unpruned tree (i.e. the rules generated fully explain the data). Then, C4.5 discards conditions in the generated rules to minimize the binomial error estimate using hill-climing techniques. Finally, in both cases, a default rule is created in case data do not satisfy any of the remaining rules.

• **K-Means:** K-Means naively assigns K centroids to the space shared by the data, and then assigns each data point to the centroid that is closest (according to some distance function). Next, the algorithm computes new locations for the existing centroids based on that data points that are assigned to them. Then the data points are reassigned, if they are now closer to a different centroid. The process continues until convergence, which is guaranteed, and the centroids are taken to be the resulting clusters.

• **Support Vector Machines:** SVMs attempt to find some hyperplan that fully seperates the data into their proper classes, so that the hyperplane can be used to estimate classes for future data. It chooses the hyperplane with the maximum distance from the nearest data. Many times the data is not linearly separable. Two primary approaches have been adopted in such cases: (a) the introduction of a slack variable can be used to assign a penalty to any hyperplane that incorrectly classifies some of the training data, and the task becomes to minimize that penalty, and (b) the data can be kernelized, or

mapped onto a new space, which is linearly separable using traditional SVM.

- **Apriori:** The Apriori algorithm was developed to efficiently find frequent sets. It relies on the truth that if an item is not frequent in some set, then any superset containing that item is also not frequent in that set. Starting with singletons, candidate frequent subsets are generated from frequent subsets of lower cardinality, and then eliminated, thereby eliminating the need to check for frequency on subsets that could not be frequent.

- **Expectation Maximization:** EM is a more general case of K-Means. For any probabilistic model, the parameters for that model are initialized to some starting value, and then the model is then used to cluster the training data. In many cases, the clustering is performed via soft-assignment, meaning each data point is assigned a probability of belonging to each cluster. Based on the new clustering, parameters for the model are recalculated to best fit the data assigned to each. Then the data are reassigned, and the process repeats until convergence (to some threshold) is achieved.

- **Page Rank:** Page Rank is intended to find relative importance of hub nodes in a complex network, commonly applied to web pages on the Internet. Essentially each page is assigned a rank value that is the sum of all contributions of pages that link to it. Each incoming contribution is the contributing page's rank value divided by the number of outgoing links it has. In this way, the highest page ranks are achieved when a page has many links to it that come from pages with high rank and few outgoing links.

- **Adaboost:** Adaboost is a type of ensemble learning algorithm that uses voting to create strong learners from multiple weak learners. Essentially, the algorithm picks a single (best) learner by which to classify training (labled) data. It then assigns a weight to each training sample, lower for those that were correctly classified and higher for those that were not. It then selects a new (best) feature and similarly redistributes the weight to misclassified examples. After the last weak learner is used, each learner is then weighted according to the weighted error they would have using the final weights on the misclassified examples in each learner. The final weighted weak learners represents the new stronger ensemble learner.

- **K Nearest Neighbors:** KNN is a relatively simple algorithm that finds the K nearest neighbors to each data points and either clusters that data point according to whichever cluster was most frequent in the neighbors, or clusters that data point according to a weighted (by distance) vote from the nearest neighbors.

- **Naive Bayes:** Naive Bayes classifiers essentially assume that all the features belonging to a dataset are independent of each other, given the class. Essentially, joint probability distributions are created (rather efficiently using the chain rule, due to the independence assumption) to provide a purely probabilistic estimate of class, based on the training data.

- **Classification and Regression Trees (CART):** CART is an umbrella term encompassing several variations on building trees for use with categorical or real-valued data. In general, they work by building binary classification trees using intelligent selection criteria for choosing attributes to split on (such as information gain or the Gini impurity), and then perform pruning of the resulting tree (or stopping criteria) by minimizing a loss which contains a complexity penalty element and a misclassification (or error, in the real-valued case) element.

3. Author Idenfication: Full Problem Description

- Define problem
- Formally describe problem–inputs, outputs, training and testing method

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

$$\cos^3 \theta = \frac{1}{4}\cos\theta + \frac{3}{4}\cos 3\theta \qquad (2)$$

Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit ultricies tellus, ac venenatis arcu wisi vel nisl. Vestibulum diam. Aliquam pellentesque, augue quis sagittis posuere, turpis lacus congue quam, in hendrerit risus eros eget felis. Maecenas eget erat in sapien mattis porttitor. Vestibulum porttitor. Nulla facilisi. Sed a turpis eu lacus commodo facilisis. Morbi fringilla, wisi in dignissim interdum, justo lectus sagittis dui, et vehicula libero dui cursus dui. Mauris tempor ligula sed lacus. Duis cursus enim ut augue. Cras ac magna. Cras nulla. Nulla egestas. Curabitur a leo. Quisque egestas wisi eget nunc. Nam feugiat lacus vel est. Curabitur consectetuer.

Data Analysis
- Describe the data in full detail–from its raw form to the transformation

- Provide summary statistics and relationships

Suspendisse vel felis. Ut lorem lorem, interdum eu, tincidunt sit amet, laoreet vitae, arcu. Aenean faucibus pede eu ante. Praesent enim elit, rutrum at, molestie non, nonummy vel, nisl. Ut lectus eros, malesuada sit amet, fermentum eu, sodales cursus, magna. Donec eu purus. Quisque vehicula, urna sed ultricies auctor, pede lorem egestas dui, et convallis elit erat sed nulla. Donec luctus. Curabitur et nunc. Aliquam dolor odio, commodo pretium, ultricies non, pharetra in, velit. Integer arcu est, nonummy in, fermentum faucibus, egestas vel, odio.

Methods

- Discuss the algorithm you've chosen, *e.g.*, why you chose it
- Provide some background material on your method that shows you are well-acquianted with it
- Challenges to your method
- What software and hardware did you use, packages, *etc.*
- Final structure of data after preprocessing
- Present training and testing as some combination of text and visualizations

Algorithm 1 shows an extended *k*-means.

Results

- Dispassionately describe your results both quantified and qualified
- Do you deem this successful
- What do the results suggest
- What were challenges

Remember, we're interested in the journey, so simply because an approach failed doesn't mean failure if you discuss the failure!

Summary and Future Work

- Briefly summarize project and outcome
- What would you do differently in the future?

Suspendisse vitae elit. Aliquam arcu neque, ornare in, ullamcorper quis, commodo eu, libero. Fusce sagittis erat at erat tristique mollis. Maecenas sapien libero, molestie et, lobortis in, sodales eget, dui. Morbi ultrices rutrum lorem. Nam elementum ullamcorper leo. Morbi dui. Aliquam sagittis. Nunc placerat. Pellentesque tristique sodales est. Maecenas imperdiet lacinia velit. Cras non urna. Morbi eros pede, suscipit ac, varius vel, egestas non, eros. Praesent malesuada, diam id pretium elementum, eros sem dictum tortor, vel consectetuer odio sem sed wisi.

4. Iceberg: Full Problem Description

Statoil, an energy company with oil and natural gas operations based largely in the Norwegian Continental Shelf region, has an operational need to track the presence of icebergs that may

---

Algorithm 1 This is a caption for the algorithm. *k*-means* over Δ

1:   **INPUT** data Δ, blocks k, distance $\mathbf{d} : \Delta^2 \to \Re_{\geq 0}$
2:   **OUTPUT** centroids $C_1, \dots, C_l$
3:   %% assume that a centroid is a pair (v, X)
4:   %% $v \in \Re^m$ and (a possibly empty) $X \subseteq \Delta$
5:   %% heap $H \subseteq \Delta$
6:   randomly construct k centroids $C^0 = \{C_1^0, C_2^0, \dots, C_k^0\}$
7:   $i \leftarrow 0$
8:   %% Δ_{HE} represents HE data, $\Delta = \Delta_{HE} + \Delta_{LE}$
9:   $\Delta_{HE} \leftarrow \Delta$
10: **repeat**
11:    **for** $\mathbf{x} \in \Delta_{HE}$ **do**
12:      **for** $C_j^i \in \mathsf{C}^i$ **do**
13:       %% assign data to centroid/heap that is nearest
14:       %% $\sigma \Rightarrow d$
15:       $C_j^i.H.insert(\mathbf{x}, d)$, where $\min\{\mathbf{d}(\mathbf{x}, C_j^i.v)\}$
16:      **end for**
17:    **end for**
18:    $\Delta' \leftarrow \emptyset$
19:    **for** $C_j^i \in \mathsf{C}^i$ **do**
20:      %% recalculate centroid as average of over $C.H$
21:      $C_j^{i+1}.v \leftarrow \sum_{\mathbf{x} \in C_j^i.X}(\mathbf{x}/|C_j^i.X|)$
22:      $\Delta' \leftarrow C_j^i.H.flush(\sigma)$
23:      $\mathsf{C}^{i+1} \overset{\cup}{\leftarrow} \{C_j^{i+1}\}$
24:    **end for**
25:    $i \leftarrow i + 1$
26:    $\Delta_{HE} \leftarrow \Delta'$
27: **until** threshold on C^{i-1}

---

present a threat to the safety and efficiency of those operations. [2] The company has therefore posed the following iceberg classification problem to the data science community. Statoil partners with C-CORE (a Canadian R&D corporation) to obtain access to Synthetic Aperture Radar (SAR) imagery data from the Sentinel-1 satellite constellation, and the challenge is to classify the object in each image as an iceberg or non-iceberg (presumably a ship, which is not likely to pose a threat to operations). A labeled training set (1604 records) and an unlabeled test set (8424 records) is provided, and the task is to classify each image as containing an iceberg (is_iceberg=1) or not (is_iceberg=0). [3]

Data Analysis

Each record contains three components (besides an identifier):

- band_1 - 75×75 pixel image representing horizontally-polarized backscatter intensity from the horizontally-polarized C-band radar transmission

- band_2 - 75×75 pixel image representing vertically-polarized backscatter intensity from the horizontally-polarized C-band radar transmission

- inc_angle - angle of incidence (between the transmission path and Earth-normal at the target object)

Initial Analysis   Some simple statistical analysis on the image data reveals that maximum and mean values in both bands are meaningful contributors to the classification task, as shown in Figure 1.
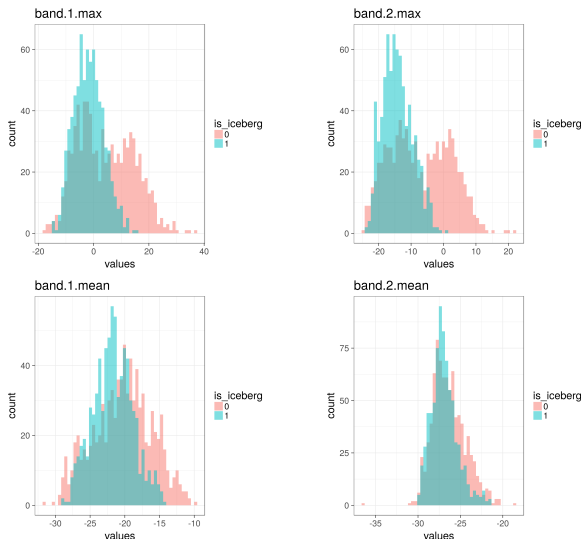

Figure 1. Band 1 and 2 Max and Mean

Further Analysis   The overlap areas in the histograms represent those records that will cause difficulty for a classification model. Additional features are required for good performance. The following functionality was developed to enable visual analysis as well as experimentation with new useful features:
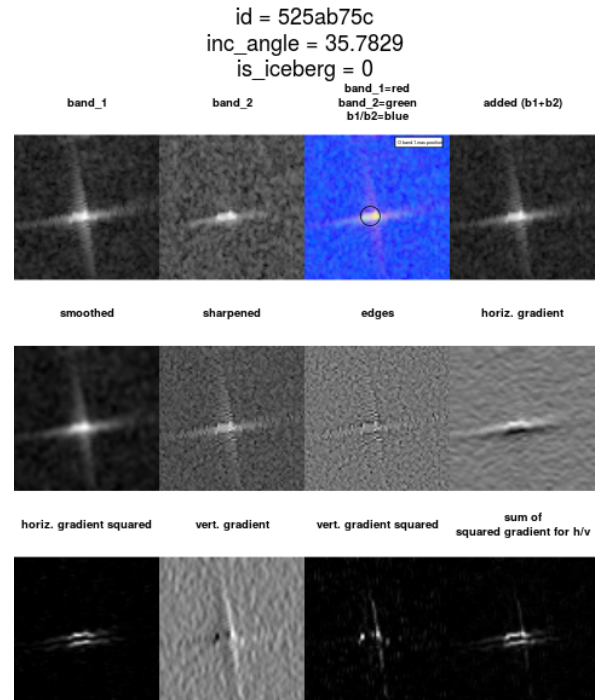

Figure 2. Image Filtering

- Simple convolution matrix filtering was implemented to achieve some rudimentary image processing. Specifically, 3×3 kernels were used for smoothing, sharpening, edge detection, and gradient. [4] These filters were used to automatically produce images during visual analysis to aid in new feature development. Examples are shown in Figure 2.

- A simple center-finding function was implemented to calculate the sum of a sliding window over the image, and the position which yields the largest sum is taken to be centered on the object of interest (OoI). Several of the features are calculated solely from a fixed-size mini-matrix centered on that position. That mini-matrix is herein referred to as the 'target'. The intent was yield feature values that are more influenced by target pixels than by non-target pixels.

Hypotheses   From visual analysis and research in the domain area [5] [6] [7], several hypotheses were developed which drove new feature experimentation.

- Ships are relatively sharp and angular, whereas icebergs are smooth and irregular, so differences in surface backscatter intensities (mean, max, variance) may yield class information.

- Volume backscatter properties of ice and metal [7] are different, so comparisons between band 1 and band 2 may yield some class information.

- Icebergs (being made of frozen water) may have some observable characteristics that are similar to the surrounding water, whereas ships may not. Therefore, comparison of the OoI to the background seawater may yield class information.

- The fast majority of iceberg volume is below the ocean's surface, and the angle of the ice as it protrudes from the water's surface is likely to be off-normal, whereas that of ships is likely to be near-normal. Therefore, gradient and edge-detection approaches near the object-to-ocean transition area may yield class information.

- Incident angle is likely to have an effect on both the intensity and polarity of the return signal. [8]

Methods

Model Selection Two primary classification methodologies were considered for this classification task: Convolution Neural Networks (CNN) and Logistic Regression (LgR). CNN methods have been shown to exhibit good performance on image recognition tasks. [9] [10] Two challenges with Neural Networks are that (a) they can be arbitrarily complex and so may require large data sets to achieve high accuracy and low variance, and (b) the hidden nature of the internal neurons pose challenges to applying useful domain knowledge to the classification problem. They work particularly well in applications where they learn telling features of a visible object within the hidden layers. In the context of this task, the size, shape, and orientation of the OoI is inconsistent and difficult to normalize, which would be highly desirable for a CNN approach. [11] LgR can be thought of as a special case of a Neural Network which has a single neuron that uses a logit function as its output. The Since the single neuron is at the outer layer, its inputs and outputs are known. This makes it straightforward to apply some domain knowledge to the raw data in order to provide inputs (features) that are thought to contain class information. Additionally, the low complexity of the model makes it ideal for smaller training sets, such as the one provided by Kaggle. [12] [13] LrG was chosen for this classification task. The Logistic Regression Algorithm is shown in Algorithm 1. For this project, the authors used R's built-in `glm` function.

Logistic Regression Essentially, LgR uses an initial guess of feature weights to transform the data feature vector into a value between 0 and 1, representing the probabilities that each record belongs to class 1. A likelihood function and its derivative (with respect to the weights) are calculated, representing the curve of the likelihood that the weights are optimal over the domain of W. The derivative use used to influence the magnitude and direction for which the weight vector should be adjusted. The user-defined value α is also used to influence the step size of these adjustments. This processes is repeated until the gradient is sufficiently small, representing the peak (or valley) of the likelihood function, indicating an optimal weight vector has been reached.

Algorithm 2 Logistic Regression

1: **INPUT:** data $X_{n\times(d-1)}$, class labels $C_{n\times1}$
2: **OUTPUT:** weights $W_{1\times d}$
3: %% Define gradient descent parameters
4: $\varepsilon \leftarrow$ user threshold
5: $\alpha \leftarrow$ user step size
6: %% Add ones column for offset
7: $X \leftarrow \left[X_{n\times(d-1)} \vec{1}_{n\times1} \right]$
8: %% Initialize weights
9: $W \leftarrow \vec{0}_{d\times1}$
10: **while** $|\nabla_W L(W)| > \varepsilon$ **do**
11: %% Class probability as function of weights
12: $y_i(W) \leftarrow \frac{1}{1+e^{-x_i W}} \forall x_i \in X$
13: %% Likelihood as function of weights
14: $L(W) \leftarrow -\sum_{i=1}^n c_i log(y_i) + (1-c_i)log(1-y_i)$
15: %% Gradient of likelihood wrt weight vectors
16: $\nabla_W L(W) \leftarrow \sum_{i=1}^n x_i(y_i - c_i)$
17: %% Update weights
18: $W \leftarrow W + \alpha \nabla_W L(W)$
19: **end while**
20: **return** W

Features The chosen modeling approach enabled experimentation of new potential features, which became the next focus. Table 1 summarizes, in some detail, a few of the more interesting features that were explored based on the previously mentioned visual analysis and domain research. Additional features are briefly described below.

- **band.1/2.tar.var** Variation of image pixels in the target for each band. The seed for this idea originated from Bentes [6] who recognized variance among other signal characteristics as harmful in a Neural Network approach and therefore something to be reduced. The intent is to see whether there is class information found within the variance of the return signal from the target.

- **band.1/2.tb.mean.dif** The difference of means of the target and border regions of the image pixels for each band. The intent is to see if there is class information derivable from return signal intensity from the target as compared with the background.

- **band.1/2.tar.gvs.mean** The target in each band was filtered using a vertical gradient filter and then squared, and the mean of the result was taken for each band. The intent is to see if there is class information in the object-to-ocean interface area that would be captured by the gradient filter.

- **band.1/2.tar.ghs.mean** The target in each band was put through a horizontal gradient filter and then squared, and the mean of the result was taken for each band.

Figure 3. Feature Correlation

- **band.1/2.tar.mean** A mean was taken of the target pixels for each band.

- **tar.gvs.dif** The target in each band was filtered using a vertical gradient filter, squared, and meaned. The result from band 2 was subtracted from that of band 1.

- **tar.ghs.dif** The target in each band was filtered using a horizontal gradient filter, squared, and meaned. The results from band 2 was subtracted from that of band 1.

Challenges   Some challenges and drawbacks to the chosen modelling approach include the following.

- Structure Simplicity: LgR is capable of creating arbitarily complex models based on the number input features. However, as a statistical tool, it lacks some of the structural features of other models such as decision trees and neural nets. Because of this, the LgR model alone is not capable of branching on specific features.

- Feature Selection: During experimentation with new and existing feature, simple histogram analysis revealed many of the candidate features as containing some class information (i.e. good separability of classes over the feature). However, pairs of features are highly correlated and others were not (See figure 3). It is difficult to choose which features to include manually.

Implementation   Given the manual nature of the feature development and experimentation approach chosen, it was important to automate updates to the analysis, training, and validation approach to the greatest extent possible. Automation of these aspect enables rapid identification and incorporation

of improvements to the feature set. To this end, all database interaction, image processing, and feature calculation was centralized to a single code source. An analysis script was run after each change to the feature set to quickly visualize some characteristics of new features. In addition to automation, V-fold cross validation was utilized as the primary means of model performance prediction. It should be noted that test data used in cross-validation was present during all data analysis tasks, which may introduce bias as a result in the model selection process. [14] Finally, a simple greedy algorithm was chosen to select which features to use during the training process. Essentially, starting from an empty feature set, the feature which (when added to the set) results in the lowest Log Loss on the test data during V-fold cross validation is added. A subset of the (now ordered) feature list which results in the lowest total Log Loss on the test data is then selected for use. The author notes this algorithm is not guaranteed to find the optimal set of features on which to train. In fact it was observed that some of the later Kaggle submissions (which represented a superset of features as compared with past submissions) achieved worse performance! This is because the optimal subset of features was never encountered during the greedy selection process.

Algorithm 3 Feature Selection

1:   **Def:** $vFoldXVal(\Delta[features]) \rightarrow LogLoss_{test}$
2:   **INPUT:** feature list F, labeled data Δ
3:   **OUTPUT:** selected feature list F_S
4:   $F_S \leftarrow \emptyset$
5:   $F_S.last \leftarrow$ NA
6:   $F_S.lossmin \leftarrow \infty$
7:   **while** $F \neq \emptyset$ **do**
8:      $f_{best} \leftarrow$ NA
9:      $f_{best}.loss \leftarrow \infty$
10:     **for** f in F **do**
11:        $loss \leftarrow vFoldXVal(\Delta[\{F_S\} \cup \{f\}])$
12:        **if** $loss < f_{best}.loss$ **then**
13:           $f_{best} \leftarrow f$
14:           $f_{best}.loss \leftarrow loss$
15:           **if** $loss < F_S.lossmin$ **then**
16:               $F_S.last \leftarrow f$
17:               $F_S.lossmin \leftarrow loss$
18:           **end if**
19:        **end if**
20:     **end for**
21:     $F_S \leftarrow \{F_S\} \cup \{f_{best}\}$
22:     $F \leftarrow F \setminus \{f_{best}\}$
23:   **end while**
24:   $F_S \leftarrow F_S[1 : F_S.last]$
25:   **return** F_S

Results

Processing Performance   Feature calculation, model training and feature selection, and classification were performed
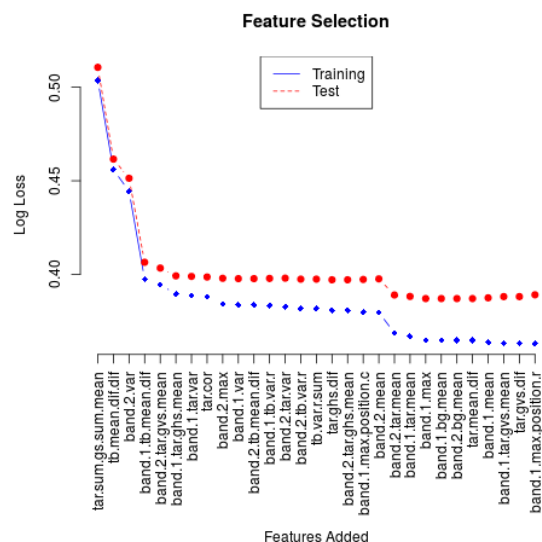
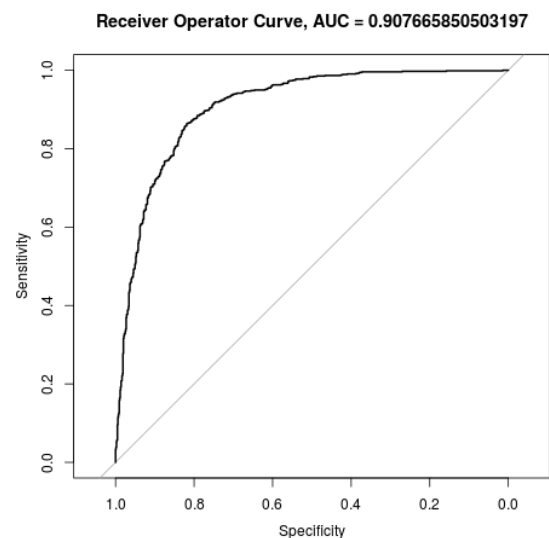**Figure 4.** Performance During Feature Selection

Figure 5. Logistic Regression ROC Results
Sensitivity represents icebergs correctly classified as icebergs, and specificity represents non-icebergs incorrectly classified as icebergs.

on a personal laptop, the specifications of which are found in Table 2. Processing times are shown in Table 3.

Table 2. Machine Specifications

Processor:	Intel Core i5-5200 @ 2.2 GHz x 4
Memory:	8 GiB
Operating System:	Ubuntu 16.04 LTS
Applications:	R and MySQL
R Packages:	RMySQL, rjson, grid, ggplot2, corrplot

Table 3. Processing Time (mm:ss)

Feature Calculation:	02:22
Feature Selection:	01:26
Model Training:	00:00
Test Data Classification:	13:04
Total:	16:52

Feature Selection Performance   One may assume that the feature selection ordering may well correlate well with those chosen by visual examination of the histograms, but that is not the case. Since some of the features inform on some of the others, once a feature is selected and incorporated into the model, the addition of a feature that is highly correlated to one already in the set would result in little performance improvement. The performance of the LgR model as a function of features added is shown in Figure 4. Overall, the performance on test data tracks relatively well to performance on training data, suggesting that overfitting is not occuring to a great extent for the first dozen or so features.

Predictive Performance   The best performance achieved as of the writing of this paper is **0.3711**, which is about the 18th

percentile as of the submission of this paper. [2]. The Receiver Operator Characteristics (ROC) plot is shown in Figure 5.

Summary and Future Work

Summary   The performance of the classifier resulting from this effort was not exemplary. The methodoloy chosen for feature experimentation and calculation was manual and labor intensive, and frankly the problem is not within the author's domain of experience. The author hypothesizes that the high level of mutual information shared by many of the custom features ultimately limited learning performance of the model. This hypothesis is supported by both Figure 3 and Figure 4. The author considers the project a success, despite the mediocre results achieved. The lessons learned about how to plan and structure a Datamining project as well as some of the practical aspects of implementing algorithms dealing with real data will serve the author well.

Future Work   Some potential future improvements to this approach would be to adopt an entropy-based feature selection algorithm rather than the simple greedy algorithm described herein - that would enables the selection of features according to the information they contain about class, derived from their probability density functions. Further, one could revisit the Neural Network approach which would allow for nonlinearity and higher complexity. Neural Networks could incorporate both customer features (as described herein) as well as image-oriented features such as the image processing filters discussed herein or alternative features such as Histogram of Oriented Gradients.

---

[2]Kaggle's scoring script uses Log Loss, which is consistent with the evaluation methods used herein

Citations and Subsubsection

Word Definition

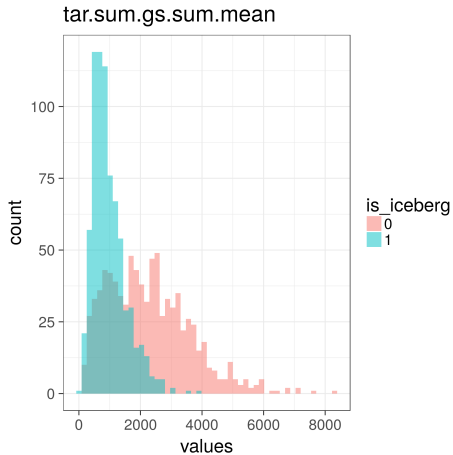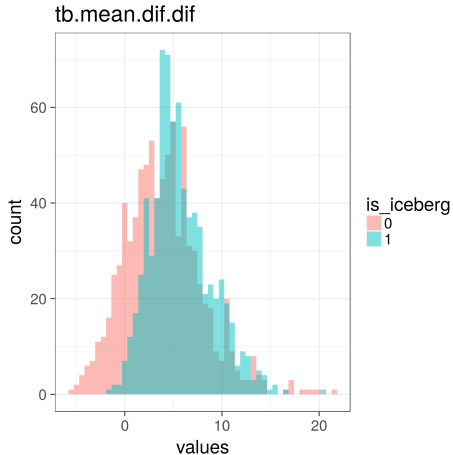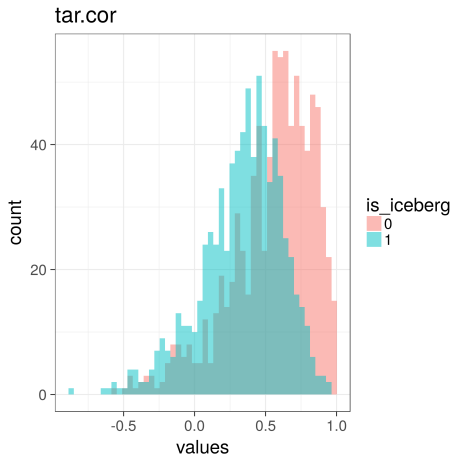**Concept** Explanation

Idea Text

References

[1] Xindong Wu, Vipin Kumar, J. Ross Quinlan, Joydeep Ghosh, Qiang Yang, Hiroshi Motoda, Geoffrey J. McLachlan, Angus Ng, Bing Liu, Philip S. Yu, Zhi-Hua Zhou, Michael Steinbach, David J. Hand, and Dan Steinberg. Top 10 algorithms in data mining. *Knowl. Inf. Syst.*, 14(1):1–37, December 2007.

[2] Statoil ASA. What we do.

[3] Kaggle Inc. Statoil/c-core iceberg classifier challenge.

[4] Gianfranco de Grandi Jong-Sen Lee, Mitchell Grunes. Polarimetric sar speckle filtering and its implication for classification. In IEEE, editor, *IEEE Transactions on Geoscience and Remote Sensing*, volume 37.

[5] Wolfgang Dierking Christine Wesche. Iceberg signatures and detection in sar images in two test regions of the weddell sea, antarctica. 58(208).

[6] Carlos Bentes. Ship-iceberg discrimination with convolutional neural networks in high resolution sar images.

[7] Carl Howell. *Iceberg and Ship Detection and Classification in Single, Dual and Quad Polarized Synthetic Aperture Radar*.

[8] Juha Karvonen Marko Makynen. Incidence angle dependence of first-year sea ice backscattering coefficient in sentinel-1 sar imagery over the kara sea. 55(11).

[9] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012.

[10] Hierarchical neural networks for image interpretation.

[11] Geoffrey Hinton Yann LeCun, Yoshua Bengio. Deep learning. 521:436–444.

[12] J.A. Nelder P. McCullagh. *Generalized Linear Models*. 2nd edition.

[13] A comparison of prediction accuracy, complexity, and training time of thirty-three old and new classification algorithms.

[14] Nicola Talbot Gavin Cawley. On over-fitting in model selection and subsequent selection bias in performance evaluation. 11:2079–2107.

Table 1. Additional Features

Feature	Description	Histogram
tar.sum.gs.sum.mean	The target mini-matrix for each band is added together. The result is filtered by vertical gradient and horizontal gradient kernel in parallel. Those two results are then squared and added together, and the mean of the result is taken to be the new feature. The intent of this feature is to let both bands contribute to squared gradient result. Two theories are at work here: (a) The edges of ships are vertical and so one might expect a sharper ship-to-water transition, resulting in a higher gradient, and (b) since icebergs are more similar in texture and composition to water than are ships, that may also contibute to a sharper contrast between ship and water, again resulting in a higher gradient.	
tb.mean.dif.dif	Visual analysis revealed most images to be rather centered on the OoI. A fixed-size border mask (think picture frame) was created to provide the subset of pixels presumed to reside in the background. For each band, the background mean was subtracted from the target mean. The restult for band 2 was then subtracted from the result for band 1, and the final result was taken to be the feature. The intent was essentially to see if the difference in target-to-background contrast between bands may contribute to classification. Put another way, the depolarizing properties of ice may be similar to that of the surrounding water, creating a consistently lower contrast image in band 2 as compared with band 1 for icerbergs.	
tar.cor	The band 1 and band 2 images are first smoothed to reduce extreme values caused by noise or speckle and zero-adjusted. Next, all of the pixel values that are greater than 0.7 of the maximum value are kept, the others discarded. These pixels are then used as a mask to extract all of the pixels in the raw image which are presumed to belong to the OoI. Finally, a correlation is calculated between these pixels between band 1 and band 2, and the resulting value is taken to be the feature. The intent is to capture how well band 2 correlates to band 1 over the OoI. The theory is that depolarization effect would affect correlation and be different for icebergs than for ships.	