Computer Science
School of Informatics, Computing, and Engineering
Indiana University, Bloomington, IN, USA
Datamining B565 Fall 2017

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

Jeevan Reddy Rachepalli[1]*, Abhishek Rapelli[2],Pavan Kumar Madineni[3]

Executive Summary Kaggle is an online platform for Datascience competitions. We did "spooky author identification" and "statoil/C-Core Iceberg Classifier Challenge" projects from it. Got an accuracy of predicting whether a given image is iceberg or not with an accuracy of 89 percent using Keras Deep Neural Network. Got a log loss of 0.8 using Modified Naïve bayes classifier for Author identification

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

Contents

1. Introduction

- **Kaggle** is an opensource internet platform for Data Science and analytics competitions in which students, statisticians, engineers, data miners etc., compete to produce the best models for predicting and describing the datasets uploaded by various corporate companies and users.

- This approach relies on the fact that there are countless strategies that can be applied to any predictive modelling task and it is impossible to know beforehand which technique or analyst will be most effective. There will be users or competitors will be ranked based on the performance of their model and its predictions.

- One can get to access to Kaggle competitions freely by agreeing upon certain terms and conditions of the platform. The website link is: http://kaggle.com/ . To get access to the datasets and be involved in the competition one has to signup with Kaggle by creating an account as per the website instructions.

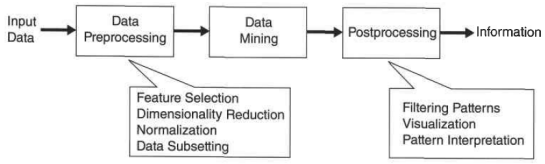
- **Data Mining:** Unlike data analytics, in which discovery goals are often not known or well defined at the outset, data mining efforts are usually driven by a specific absence of information that can't be satisfied through standard data queries or reports.

- Data mining yields information from which predictive models can be derived and then tested, leading to a greater understanding of the marketplace. It is an integral part of knowledge discovery in databases, which is the overall process of converting raw data into useful

information.

- This process consists of a series of transformation steps, from data preprocessing to postprocessing of data mining results.

- It uses a combination of human statistical skill and software that is programmed with pattern-recognition algorithms that detect anomalies. Thus, the term refers to both an information technology competency as well as a category of software technology.



- **Applications of Data Mining:** The application of data mining is broad. It can be used for everything from pharmaceutical research to modeling traffic patterns. However, a classic use case is to predict customer behaviors in-order to optimize sales and marketing activities.

- For example, retailers often use data mining to predict what purchases their customers might make next. They can then respond with targeted promotions to increase the sale.

1.1 Author Identification

- We are given a dataset consisting of text extracted from various books or writings of the given authors and these text data are labelled by the corresponding author name in the train dataset.

- Our goal is to create a model which can predict the probability that it is extracted from the books of each of the three authors viz., Edgar Allan Poe, HP Lovecraft and Mary Shelley for the test dataset. Our goal is also to have maximum accuracy in the predictions for the test dataset.

- Currently, many institutions and companies use aerial reconnaissance and shore-based support to monitor environmental conditions and assess risks from icebergs. However, in remote areas with particularly harsh weather, these methods are not feasible, and the only viable monitoring option is via satellite. In this competition, you're challenged to build an algorithm that automatically identifies if a remotely sensed target is a ship or iceberg.

- **id** a unique identifier for each sentence.

- **text** some text written by one of the authors

- **author** the author of the sentence (EAP: Edgar Allan Poe, HPL: HP Lovecraft; MWS: Mary Wollstonecraft Shelley).

- The train dataset consists of 19,580 rows with each row having 3 columns viz., the unique id, the text, and the author, which are the attributes. The 'author' attribute is categorical type having labels as EAP, HPL, MWS, which stand for Edgar Allan Poe, HP Lovecraft and Mary Shelley, respectively. The test data has 8,392 rows.

- An example of text from the works of the author, Edgar Allan Poe is shown below:

> This process, however, afforded me no means of ascertaining the dimensions of my dungeon; as I might make its circuit, and return to the point whence I set out, without being aware of the fact; so perfectly uniform seemed the wall.

The above text is extracted from the book "The complete tales and poems of Edgar Allan Poe" by Edgar Allan Poe.

- The goodness of the model is quantified based on the multi-class logarithmic loss (as instructed in the Kaggle). For each id, we calculate the log-loss from the predicted probabilities for the test dataset of each author.
The formula of log-loss calculation is shown below:

$$logloss = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{M} y_{ij}\log(p_{ij}),$$

1.2 Statoi/C-CORE Iceberg Classifier Challenge

- The given dataset consists of 1604 rows and 5 columns, which are 'ID of an image', 'Incident angle', 'band1', 'band2' and 'Is iceberg'. The problem is to predict the target variable 'Is iceberg' i.e., yes or no.

- This is a binary classification problem. If it is not an iceberg it has to be a ship. Our goal is to predict the target variable with maximum accuracy.

- **id** - the id of the image

- **band1, band2** - the flattened image data. Each band has seventy five by seventy five pixel values in the list, so the list has five thousand six hundred and twenty five elements. Note that these values are not the normal non-negative integers in image files since they have physical meanings - these are float numbers with unit being dB. Band 1 and Band 2 are signals characterized by radar backscatter produced from different polarizations at a particular incidence angle. The polarizations correspond to HH (transmit/receive horizontally) and HV (transmit horizontally and receive vertically).

- **incangle** the incidence angle of which the image was taken. Note that this field has missing data marked as "na", and those images with "na" incidence angles are all in the training data to prevent leakage.

- **isiceberg** the target variable, set to 1 if it is an iceberg, and 0 if it is a ship.

- The attributes 'band1', 'band2' consist of numerical floating values ranging from -35 to +1. The numeric values visualize a picture, which may be a ship or a iceberg. Our model has to predict this.

- The goodness of the model is quantified based on the multi-class logarithmic loss (as instructed in the Kaggle). For each id, we calculate the log-loss from the predicted probabilities for the test dataset of each author.
  The formula of log-loss calculation is shown below:

$$logloss = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{M} y_{ij}\log(p_{ij}),$$

2. Datamining

- **What is Data Mining?**

- Data mining is the practice of automatically searching large stores of data to discover patterns and trends that go beyond simple analysis.
- It uses sophisticated mathematical algorithms to segment the data and evaluate the probability of future events. Data mining is also known as Knowledge Discovery in Data (KDD). The key properties of data mining are:
  - Automatic discovery of patterns
  - Prediction of likely outcomes
  - Creation of actionable information
  - Focus on large data sets and databases
- Data mining can answer questions that cannot be addressed through simple query and reporting techniques. Data mining is accomplished by building models. A model uses an algorithm to act on a set of data. The notion of automatic discovery refers to the execution of data mining models.

- Data mining models can be used to mine the data on which they are built, but most types of models are generalizable to new data. The process of applying a model to new data is known as scoring.

- Many forms of data mining are predictive. For example, a model might predict income based on education and other demographic factors. Other forms of data mining identify natural groupings in the data. For example, a model might identify the segment of the population that has an income within a specified range, that has a good driving record, and that leases a new car on a yearly basis.

- **What does it yield?**

- Data mining is a powerful tool that can help you find patterns and relationships within your data. But data mining does not work by itself. It does not eliminate the need to know your business, to understand your data, or to understand analytical methods. Data mining discovers hidden information in your data, but it cannot tell you the value of the information to your organization.

- We might already be aware of important patterns as a result of working with your data over time. Data mining can confirm or qualify such empirical observations in addition to finding new patterns that may not be imme-

diately discernible through simple observation.

- It is important to remember that the predictive relationships discovered through data mining are not necessarily causes of an action or behavior.

- For example, data mining might determine that males with incomes between /50,000 and /65,000 who subscribe to certain magazines are likely to buy a given product. We can use this information to help you develop a marketing strategy. However, we should not assume that the population identified through data mining will buy the product because they belong to this population.

- Data mining does not automatically discover solutions without guidance. The patterns we find through data mining will be very different depending on how you formulate the problem.

- To obtain meaningful results, we must learn how to ask the right questions. For example, rather than trying to learn how to "improve the response to a direct mail solicitation," we might try to find the characteristics of people who have responded to your solicitations in the past.

- **What are the general steps?**

- The data mining process is often characterized as a multi-stage iterative process involving data selection, data cleaning, application of data mining algorithms, evaluation, and so forth. Here we adopt a somewhat different process-oriented view and break it down into five basic steps:

  - 1. **Exploring and Preprocessing:** The initial steps of exploring, visualizing, and querying the data, to gain insight into the data in an interactive manner. Preprocessing steps such variable selection, data focusing, and data validation can also be included in these initial steps.

  - 2. **Modeling:** The steps involved in (a) selecting the model representations that we seek to fit to the data (b) selecting the score functions that score different models with respect to the data, and (c) specifying the computational methods and algorithms to optimize the score function.

  - 3. **Mining:** The step of actually running a particular data mining algorithm on a particular data set.

  - 4. **Evaluating:** The step of critically evaluating the quality of the output of the data mining algorithm from Mining step, both the predictions of the model and the interpretation of the fitted model itself.

  - 5. **Deploying:** The step of putting a model from a data mining algorithm into routine predictive use.

- **What is clustering *vs.* classfication?**

- **1. Definition:**

  Clustering: Clustering is an unsupervised learning technique used to group similar instances on the basis of features.
  Classification: Classification is a supervised learning technique used to assign predefined tags to instances on the basis of features.

- **2. Supervision:**

  Clustering: Clustering is an unsupervised learning technique.
  Classification: Classification is a supervised learning technique.

- **3. Training set:** Clustering: A training set is not used in clustering.
  Classification: A training set is used to find similarities in classification.

- **4. Process:** Clustering: Statistical concepts are used, and datasets are split into subsets with similar features.
  Classification: Classification uses the algorithms to categorize the new data according to the observations of the training set.

- **5. Labels:** Clustering: There are no labels in clustering. Classification: There are labels for some points.

- **6. Aim:** Clustering: The aim of clustering is, grouping a set of objects in order to find whether there is any relationship between them.
  Classification: The aim of clustering is to find which class a new object belongs to from the set of predefined classes

- **What is a loss function?**

- Binary Cross Entropy or Loss function: If our prediction is y' and real value is y, then binary cross entropy is defined as

$$b(y,y') = -ylog(y') - (1-y)log(1-y')$$

2.1 Data Preprossing

- There are some missing values in the "incidence angle" column from the data. Replaced the missing value with the mean of the "incidence angle" . So that we do not drop many rows in the dataset.

- One observation is that most of the data is concentrated on the $[20:50, 20:50]$ matrix grid and the rest of the image has ocean as a background.

- Second observation is that there is not much difference if we take band1 data or band2 data.

- Reshape the data in each column of band1 and band2 and select the $[20:50, 20:50]$ matrix grid, and reshape it to an array and convert each value of the array into a attribute for building the model.
  We have $900*2+1(inc_angle) = 1801$ attributes and one value to predict i.e., isiceberg(Binary Variable).

2.2 Mining, Interpretation, and Action

- Data mining is a very broad area that integrates techniques from several fields including machine learning, statistics, pattern recognition, artificial intelligence, and database systems, for the analysis of large volumes of data. There have been a large number of data mining algorithms rooted in these fields to perform different data analysis tasks. The 10 algorithms identified by the IEEE International Conference on Data Mining (ICDM) are among the most influential algorithms for classification, clustering, statistical learning, association analysis, and link mining.

  The top 10 algorithms:

  1. C4.5 algorithm:

- The process of predicting qualitative responses is called classification. Predicting a qualitative response for an observation can be referred to as classifying that observation, since it involves assigning the observation to a category or class.

- Systems that construct classifiers are one of the commonly used tools in data mining. These systems take a collection of cases as input, each belonging to one of a small number of classes described by their values for a fixed set of attributes and results a classifier that can accurately predict the class to which a new case belongs.

- C4.5 generates classifiers in the form of decision trees and also in the form of ruleset forms which are relatively more comprehensible. C4.5 grows an initial tree using the divide and conquer approach.

- From a given set S of cases, if all the cases belong to the same class or S is small, then the tree is a leaf labeled with the most frequent class in S. Otherwise, it chooses a test based on a single attribute with two or more outcomes.

- Make this test the root of the tree with one branch for reach outcome of the test, partition S into corresponding subsets according to the outcome for each case and apply the same procedure recursive for each subset.

- The initial tree is then pruned to avoid over fitting. Pruning is carried out from the leaves to the root. The pruning algorithm is based on a pessimistic estimate of the error rate associated with a set of N cases, E of which do not belong to the most frequent class.
  The notable characteristics of C4.5 are as follows:

  – C4.5 uses information-based criteria and allows two or more outcomes.

  – C4.5 apportions the case probabilistically among the outcomes.

  – C4.5 prunes the trees using a single-pass algorithm which is derived from binomial confidence limits.

  – C4.5 forms rulesets from the initial unpruned decision trees. The principal disadvantage of rulesets over decision trees is the amount of CPU time and memory that they require.

  2. The k-means Algorithm:

- The k-means algorithm is a simple iterative method to partition a given dataset into a specified number of distinct, non-overlapping clusters, k. If n observations have to be partitioned into k different clusters, there are almost Kn ways of partitioning.

- This is a really huge number unless k and n are tiny. K-means is simple algorithm which provides an optimum solution to partition n observations into k different clusters. The steps followed are:

  – Randomly assign a number, from 1 to K, to each of the observations. These serve as initial cluster assignments for the observations. This also achieves the partitioning of data.

  - Iterate until the cluster assignments stop changing:
    - For each of the K clusters, compute the cluster centroid. The kth cluster centroid is the vector of the p feature means for the observations in the kth cluster.
    - Assign each observation to the cluster whose centroid is closest (where closest is defined using Euclidean distance).

- Clustering can be a very useful tool for data analysis in the unsupervised setting. However, there are a number of issues that arise in performing clustering. They are:
  - Should the observations or features be standardized in some way? : For instance, may be the data has to be normalized i.e. should be centered to have mean zero and standard deviation 1.
  - In the case of K-means in specific, choosing the value of K is a major concern.
  - Validating the clustering obtained? : If we have to ascertain whether the clusters that have been found represent the true subgroups in the data, of whether they are simply a result of clustering the noise.
  - A tempered approach to interpreting the results of K-means clustering: The result obtained from clustering should not be taken as the absolute truth about the data set. Instead, they should constitute a starting point for the development of a scientific hypothesis and further study, preferably on an independent data set.

3. Support Vector Machines

- Support Vector Machines offers one of the most robust and accurate methods among all well-known algorithms. It requires only a handful number of examples for training and is insensitive to the number of dimensions. The support vector machine is an extension of the support vector classifier that results from enlarging the feature space in a specific way using kernels where kernel is a function that quantifies the similarity of two observations.
- For a two-class classification task, SVM segregates the members of the two classes in the training data using the best classifier function. For a dataset which is linearly separable, the classification function corresponds to a separating hyperplane that passes through the middle of the two classes.
- In a pragmatics situation, there exits numerous such linear hyperplanes, but SVM guarantees that the best classifier function is fit by maximizing the margin between the two classes. Intuitively, the margin appears to be the amount of space or the separation between the two classes but geometrically it is the shortest distance between the closest data points to a point on the hyperplane.
- SVM Heavily insists on finding the maximum margin hyperplanes so that it offers the best generalization ability. It allows not only the best classification performance on the training data but also leaves much room for appropriate classification on the future data.
- There have been numerous proposals on extending SVMs to a general case where we have some arbitrary number of classes. It turns out that the concept of separating hyperplanes upon which SVMs are based do not lend itself naturally to more than two classes. The most popular SVM approaches are one-versus-one and one-versus-all approaches.

4. The Apriori Algorithm

- The role of Data Mining has been constantly changing with the rapid growth of e-commerce applications which leads to accumulation of vast quantity of data in a relatively short span of time.
- Data Mining today is also known as Knowledge Discovery in Databases has its role predominantly confined to finding anomalies, correlations, patterns and trends to predict outcomes.
- Apriori algorithm is a classical one in data mining that is used to find frequent itemsets from a transaction dataset and derive association rules.
- It is not a trivial task to find the frequent itemsets because of its combinatorial explosion. But once the frequent itemsets are obtained it is straightforward to generate association rules with confidence larger than or equal to a user specified minimum confidence.
- The key concept in Apriori algorithm is the anti-monotonicity of the support measure. It assumes that all subsets of a frequent item set must be frequent and similarly for any infrequent itemset, all its supersets must be infrequent too. The apriori algorithm works mainly in two steps. They are:

- **Step 1:** Apply minimum support to find all the frequent sets with k items in a database. This is the frequent itemset generation.

- **Step 2:** Rule generation: Create rules from each frequent itemset using the binary partition of frequent itemsets and look for the ones with high confidence. These rules are called candidate rules.

- This approach of extending a frequent itemset one at a time is called the bottom up approach.

- The pros and cons of the apriori algorithm are:

  - It is very easy to implement and understand.

  - It can be used on large itemsets.

  - Sometimes, it is a very tedious task to find the frequent itemsets due to its combinatorial explosion. So, it may need to find a large number of candidate rules which can be computationally expensive.

  - Since this algorithm mainly deals in browsing through the entire database, calculating support is expensive.

5. The Expectation-Maximization Algorithm

- The Expectation Maximization algorithm is an unsupervised clustering method, i.e. doesn't require a training phase.

- It follows an iterative method to find maximum likelihood or maximum estimates of parameters in statistical models where the model depends on unobserved latent variables.

- The expectation maximization alternated between performing an expectation step, which creates a function for the expectation of the log-likelihood evaluated using the current estimate for the parameters, and a maximization step which computes parameters maximizing the expected log-likelihood found on the expectation step.

- The EM algorithm is based on the principle of maximum likelihood of unobserved values and this clustering process can be based on probability models to predict the missing values.
- In general, the EM algorithm takes the data set, total number of clusters, accepted error to converge and the maximum number of iterations as input values.

- For each iteration, the Expectation step is executed first, which estimates of probability of each point belonging to each cluster, followed by the maximization step that re-estimates the parameter vector of the probability distribution of each class.

- The algorithm closes when the distribution parameters converge to the maximum number of iterations.

- It is generally concluded that from among the available classification algorithms, the EM clustering technique, despite the dependence on the number of clusters and on the initialization phase, is an efficient method that produces good results for several scenarios of data dispersion.

6. PageRank

- PageRank is considered one of the earliest techniques of link-based analysis to increase the performance of Information Retrieval on the Web; this technique is used by Google, a Web search engine developed at Stanford University. PageRank was developed in the process of developing an ambitious project of associating every existing web page with just one number, its PageRank, which expresses the importance, or rank of that web page in the whole web.

- The algorithm relies on the democratic nature of the Web by using its vast link structure as an indicator of an individual page's quality. It interprets a hyperlink from page x to page y as a vote, by page x, for page y.

- However, it looks at more than just the sheer number of votes, or links that a page receives. It also analyzes the page that casts the vote. Votes casted by pages that are themselves important weigh more heavily and help to make other pages more important. This is exactly the idea of rank prestige in social networks.

- The ideas that are used to derive the PageRank algorithm are based on rank prestige. The first idea is a hyperlink from a page pointing to another page is an implicit conveyance of authority to the target page. Thus, the more in-links that a particular page receives, the more prestige that page has. The pages that point to a particular page have their own prestige scores.

- A page with a higher prestige score pointing to that page is more important than a page with a lower prestige score pointing to the same page. In other words, a page is important if it is pointed to by other important pages.

- The PageRank algorithm outputs a probability distri-

bution used to represent the likelihood that a person randomly clicking on links will arrive at any particular page. PageRank can be calculated for collections of documents of any size. The distribution is evenly divided among all documents in the collection at the beginning of the computational process.

- The PageRank computations require several passes, called iterations, through the collection to adjust approximate PageRank values to more closely reflect the theoretical true value. A probability is expressed as a numeric value between 0 and 1. A 0.5 probability is commonly expressed as a 50 percent chance of something happening. Hence, a PageRank of 0.5 means there is a 50 percent chance that a person clicking on a random link will be directed to the document with the 0.5 PageRank.

7. AdaBoost Algorithm

- Many real-world applications are born with high dimensionality, i.e., with a large amount of input features. There are two paradigms that can help us to deal with such kind of data, i.e., dimension reduction and feature selection. Dimension reduction methods are usually based on mathematical projections, which attempt to transform the original features into an appropriate feature space.

- After dimension reduction, the original meaning of the features is usually lost. Feature selection methods directly select some original features to use, and therefore they can preserve the original meaning of the features, which is very desirable in many applications. AdaBoost could be very useful in feature selection, especially when considering that it has solid theoretical foundation.

- Boosting is an ensemble technique that attempts to create a strong classifier from a number of weak classifiers. This is done by building a model from the training data, then creating a second model that attempts to correct the errors from the first model. Models are added until the training set is predicted perfectly or a maximum number of models are added.

- AdaBoost is best used to boost the performance of decision trees on binary classification problems. These are models that achieve accuracy just above random chance on a classification problem. The most suited and therefore most common algorithm used with AdaBoost is decision trees with one level. Because these trees are so short and only contain one decision for classification, they are often called decision stumps.

- A weak classifier (decision stump) is prepared on the training data using the weighted samples. Only binary classification problems are supported, so each decision stump makes one decision on one input variable and outputs a +1.0 or -1.0 value for the first or second class value. Weak models are added sequentially, trained using the weighted training data. The process continues until a pre-set number of weak learners have been created or no further improvement can be made on the training dataset. Once completed, you are left with a pool of weak learners each with a stage value.

- Many empirical studies show that AdaBoost does not overfit the data i.e. the test error of AdaBoost often tends to decrease even after the training error is zero. It is argued that AdaBoost is able to increase the margins even after the training error is zero, and thus it does not overfit even after a large number of rounds.

- However, large margin does not necessarily mean better generalization, which seriously challenged the margin-based explanation. It is found that scientists considered minimum margin instead of average or median margin, which suggests that the margin-based explanation still has chance to survive. If this explanation succeeds, a strong connection between AdaBoost and SVM could be found. It is obvious that this topic is well worth studying.

8. kNN: k-Nearest Neighbor Classification

- K Nearest Neighbor (kNN) is also known as a lazy learning classifier. Decision tree and rule-based classifiers are designed to learn a model that maps the input attributes to the class label as soon as the training data becomes available, and thus they are known as eager learning classifiers.

- Unlike eager learning classifier, KNN does not construct a classification model from data, it performs classification by matching the test instance with K training examples and decides its class based on the similarity to K nearest neighbors.

- The basic idea of K-Nearest Neighbors is that the class of a test instance is determined by the class type of its nearest neighbors. The kNN classifier represents each example as a data point in a d-dimensional space, where d is the number of attribute. The algorithm computes its proximity to the rest of the data points in the training set, using an appropriate proximity measurement metric.

- The k-nearest neighbors of a given point z means the k points that are closest to point z. Then the data point z

is classified based on the class labels of its neighbors. If the neighbors have more than one label, the data point is assigned to the majority class of its nearest neighbors. Choosing a proper K value is of prime importance to the performance of K-Nearest Neighbor classifier.

- If k value is too large, the nearest neighbor classifier may misclassify the test instance because its list of nearest neighbors may include data points that are located far away from its neighborhood. On the other hand, if k value is too small, then the nearest neighbor classifier may be susceptible to overfitting because of noise in the training data set.

- The pros and cons of kNN classifier are:

  Pros:

  - kNN is simple to implement.
  - kNN executes quickly for small training data sets.
  - Performance asymptotically approaches the performance of the Bayes Classifier.
  - Don't need any prior knowledge about the structure of data in the training set.
  - No retraining is required if the new training pattern is added to the existing training set.

  Cons:

  - When the training set is large, it may take a lot of space.
  - For every test data, the distance should be computed between test data and all the training data. Thus a lot of time may be needed for the testing.

9. Naive Bayes classifier

- The Naive Bayes classifier is a simple probabilistic classifier which is based on Bayes theorem with strong and naïve independence assumptions.

- It is one of the most basic text classification techniques with various applications in email spam detection, personal email sorting, document categorization, sexually explicit content detection, language detection and sentiment detection. Despite the naïve design and oversimplified assumptions that this technique uses, Naive Bayes performs well in many complex real-world problems.

- Usually Naive Bayes is used when the multiple occurrences of the words matter a lot in the classification problem. The Naive Bayes algorithm calculates the probability of every state of each input column, given each possible state of the predictable column.

- The algorithm then performs parameter estimation using the method of maximum likelihood; in other words the naïve Bayes model works without using ay Bayesian methods.

- Naive Bayes is a kind of classifier which uses the Bayes Theorem. It predicts membership probabilities for each class such as the probability that given record or data point belongs to a particular class. The class with the highest probability is considered as the most likely class.

- Despite the name, Naive Bayes turns out to be excellent in certain applications. Text classification is one area where it really shines. Even though it is often outperformed by other techniques such as boosted trees, random forests, Max Entropy, Support Vector Machines etc, Naive Bayes classifier is very efficient since it is less computationally intensive (in both CPU and memory) and it requires a small amount of training data. Moreover, the training time with Naive Bayes is significantly smaller as opposed to alternative methods.

- Naive Bayes classifier is superior in terms of CPU and memory consumption and in several cases its performance is very close to more complicated and slower techniques.

10. The CART

- Classification and Regression Tree Analysis, CART, is a simple yet powerful analytic tool that helps determine the most important variables in a particular dataset, and can help craft a potent explanatory model. CART can statistically demonstrate which factors are particularly important in a model or relationship in terms of explanatory power and variance.

- This process is mathematically identical to certain familiar regression techniques, but presents the data in a way that is easily interpreted by those not well versed in statistical analysis. In this way, CART presents a sophisticated snapshot of the relationship of variables in the data and can be used as a first step in constructing an informative model or a final visualization of important associations.

- The statistical processes behind classification and regression in tree analysis are very similar. For a response variable which has classes, often 0–1 binary, we want to organize the dataset into groups by the response variable – classification.

- When our response variable is instead numeric or continuous we wish to use the data to predict the outcome, and will use regression trees in this situation. In traditional regression models, linear or polynomial, we develop a single equation (or model) to represent the entire data set.

- CART is an alternative approach to this, where the data space is partitioned into smaller sections where variable interactions are much clearer. CART analysis uses this recursive partitioning to create a tree where each node – T represents a cell of the partition.

- Each cell has attached to it a simplified model which applied to that cell only, and it is useful to draw an analogy here to conditional modeling, where as we move down the nodes, or leaves, of the tree we are conditioning on a certain variable. The final split or node is sometimes referred to as a leaf.

- The main elements of CART (and any decision tree algorithm) are:

  - Rules for splitting data at a node based on the value of one variable;

  - Stopping rules for deciding when a branch is terminal and can be split no more; and

  - Finally, a prediction for the target variable in each terminal node.

- The present research can offer several suggestions for researchers:

  - First, although many techniques can be successfully used to assess the true selection model, pruned CART and random forest analysis appear to perform particularly well.

  - Second, of all the machine learning techniques, pruned CART seems like a strong choice under the various selection models, sample sizes, and amounts of incomplete data.

  - Third, CART's overall strong performance depended somewhat on sample size. This method seems to be a particularly strong choice when dealing with larger samples

- **Does data mining tell us what to do?**

- Datamining is a very exhaustive process of collecting the data filtering the data, preprocess the data and keep the data ready for modelling it. After we build a model from our cleaned data, we can predict the value of the data that is unknown to our trained model. Yes, Datamining helps us in predicting an unknown value with a certain probability and also it gives us an intuition of what the problem (hypothesis) is and how to tackle it with a different situation in the future.

- **What are some new types of problems in datamining?**

- The main problem is with the preprocessing the data, it takes a lot of time and we need to take correct decisions while preprocessing the data. Collection of data is also somewhat a challenge when it comes to a sensitive data and when there is a bias in collecting our data. When modelling the data we need to take some assumptions which will be affecting the outcome of the predicting the data. We should not overfit or underfit a particular model.

- **What are some new types of problems in datamining?**

- Poor data quality such as noisy data, dirty data, missing values, inexact or incorrect values, inadequate data size and poor representation in data sampling.

- Integrating conflicting or redundant data from different sources and forms: multimedia files (audio, video and images), geo data, text, social, numeric, etc…

- Proliferation of security and privacy concerns by individuals, organisations and governments.

- Unavailability of data or difficult access to data.

- Efficiency and scalability of data mining algorithms to effectively extract the information from huge amount of data in databases.

- Dealing with huge datasets that require distributed approaches.

- Dealing with non-static, unbalanced and cost-sensitive data.

- Mining information from heterogeneous databases and

global information systems.

- Constant updation of models to handle data velocity or new incoming data.

- High cost of buying and maintaining powerful softwares, servers and storage hardwares that handle large amounts of data.

3. Author Idenfication: Full Problem Description

- The problem statement is to create a model which can predict the probability that it is extracted from the books of each of the three authors viz., Edgar Allan Poe, HP Lovecraft and Mary Shelley for the test dataset. Our goal is also to have maximum accuracy in the predictions for the test dataset.

- We are given a dataset consisting of text extracted from various books or writings of the given authors and these text data are labelled by the corresponding author name in the train dataset.

- **id** a unique identifier for each sentence.

- **text** some text written by one of the authors

- **author** the author of the sentence (EAP: Edgar Allan Poe, HPL: HP Lovecraft; MWS: Mary Wollstonecraft Shelley).

- Our target variable set of probabilities that the text was the work of the three given authors.

3.1 Data Analysis

- Downloaded the training and test data from the Kaggle and observed that the data does not have any missing values.

- Splitted the sentence of each author into words and stored in three different arrays (one for each author).

- Filtered out the special characters.

- Converted all words into lower letters of alphabets.
- Built a Pandas dataframe with columns as words and rows as authors. [word counts are stored in here]

- Provide summary statistics and relationships.

3.2 Methods

- **First Method:**
  Used Naïve-Bayes theorem for finding the probability that the given line is written by a particular author. Stored the results in a csv file and uploaded it in the Kaggle and found out that our position is in the 30th percentile.

- **Second Method:** Deleted all the stopping words from the words data frame. Followed the same procedure as the First method this time our position has improved to 37th percentile.

- **Third Method:** This time we have splitted the data into test and train from the training data itself. Log loss error is coming around 0.8.

- The major drawback of this method is that when many new words comes from any author's sentence this method will work poorly. We are trying to find a new way to tackle this problem.
- We have used Anacondas Python 3.5 as programmming tool. The packages used were Pandas, Numpy,Matplotlib, JSON, scikitlearn, Seaborn, Keras, Tensarflow and other python basic tools for programming purpose.
- Hardware used was intel-7 processor, 8 GB RAM, 1TB HDD, Dell Inspiron i5378-5743GRY model was used with a frquency of 2.4 GHz with an inbuilt intel graphics 620.

Modelling the data:

- Splitting the data into training and test in the ratio of 60:40 Doing the same preprocessing with the test data.

- By following Naïve bayes we need to find out the probability of each author i.e., the number of sentences written by the particular author. After that treated each word independently for finding the conditional probabilities for finding the posterior probabilities.

$$P(x_1,\ldots,x_n \mid c) = P(x_1 \mid c) \bullet P(x_2 \mid c) \bullet P(x_3 \mid c) \bullet \ldots \bullet P(x_n \mid c)$$

As there is a multiplication over here when a word is not present in our training set this causes zero value. So, we use modified Naïve bayes for this

$$\hat{P}(w_i \mid c) = \frac{count(w_i,c)+1}{\sum_{w \in V} \left(count(w,c)\right)+1)}$$

$$= \frac{count(w_i,c)+1}{\left(\sum_{w \in V} count(w,c)\right) + |V|}$$

- First calculate the sum of each column and storing that in a variable Observing the above formula, we can transform our data frame by adding 1 to each word In above formula —V— means total number of count of frequencies of all the words in the dataframe
- Initilize each list for each author go through each line of a test set (bag of words). Using modified Naïve bayes we calculate the probability that each line is written by one of the three authors. After that we get three lists with probabilities.
- Convert those lists into a dataframe with rows as id's of a particular sentence and columns as a probability that the line is written by that author.
- Divide each element with its row sum, doing so we get the row sum equal to one and get the calculated probabilities of each author. Actual probabilities are stored in our test set we can use pandas get-dummies function i.e., if the line is written by author the actual probability for that author is 1 and for other two authors it is zero.

- Log loss calculation for the test set (from Kaggle):
Log loss evaluated using multi-class logarithmic loss. Each id has one true class. For each id, you must submit a predicted probability for each author. The formula is then:

$$logloss = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{M} y_{ij}\log(p_{ij})$$

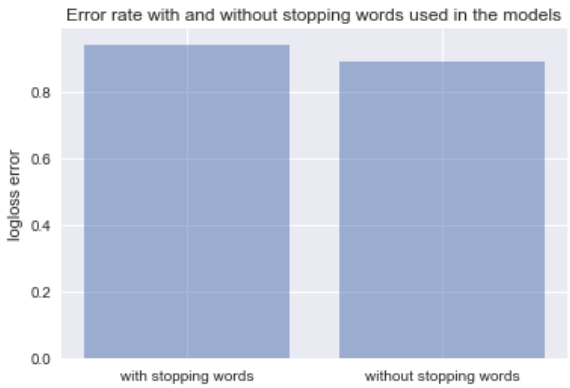
- where N is the number of observations in the test set, M is the number of class labels (3 classes), loglog is the natural logarithm, yijyij is 1 if observation ii belongs to class jj and 0 otherwise, and pijpij is the predicted probability that observation ii belongs to class jj

- The goodness of the model is quantified based on the multi-class logarithmic loss (as instructed in the Kaggle). For each id, we calculate the log-loss from the predicted probabilities for the test dataset of each author. The formula of log-loss calculation is shown below:

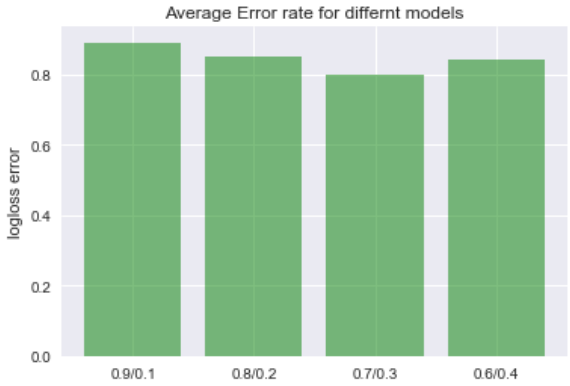
$logloss = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{M} y_{ij}\log(p_{ij}),$

- If we get a minimum value from this then our error is very less. This is how a goodness of our model is defined.

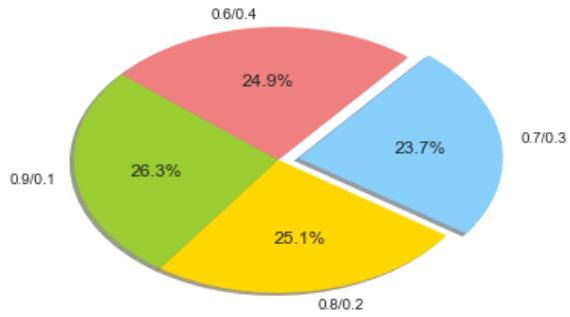
- Suppose for example if we get probability of an author belonging to particular class as 0.6 then $1 * log(1/0.6) + 0 + 0$ will be our error for that particular sentence.

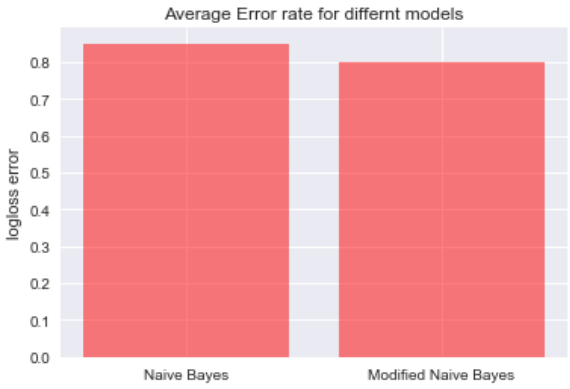


- If we remove stopping words the error rate is decreased.



- Error rate is minimum for a train/test split of 70/30.





- Logloss error is minimum if we use Modified Naïve bayes

Reference to Figure **??**.

3.3 Results

- The results obtained were the probabilities of the text belonging to the works of each f the three authors.
- Each of these probabilities were between 0 and 1 and also the sum of the probabilities of three authors for each text in each row is equal to 1.

- I dont consider my model to be fully sucessful because we have used the training data from the kaggle itslef and hence we did not have enough data to train our model and hence the model is deemed to be underfit.

- The results suggested the log loss of the model to be 0.8 approximately.
- The changelles were basically cleaning the data to pure form and doing natural language processing to create bag of words.

3.4 Summary and Future Work

- If we can get much more data to train our model, out model would not be underfitted and hence the accuracies or log loss could be minimized.

4. Iceberg: Full Problem Description

- This is a binary classification problem. If it is not an iceberg it has to be a ship. Our goal is to predict the target variable with maximum accuracy.

- The given dataset consists of 1604 rows and 5 columns, which are 'ID of an image', 'Incident angle', 'band1', 'band2' and 'Is iceberg'. The problem is to predict the target variable 'Is iceberg' i.e., yes or no.

- **id** - the id of the image

- **band1, band2** - the flattened image data. Each band has seventy five by seventy five pixel values in the list, so the list has five thousand six hundred and twenty five elements. Note that these values are not the normal non-negative integers in image files since they have physical meanings - these are float numbers with unit being dB. Band 1 and Band 2 are signals characterized by radar backscatter produced from different polarizations at a particular incidence angle. The polarizations correspond to HH (transmit/receive horizontally) and HV (transmit horizontally and receive vertically).

- **incangle** the incidence angle of which the image was taken. Note that this field has missing data marked as "na", and those images with "na" incidence angles are all in the training data to prevent leakage.

The following are the attributes and their description:

- **isiceberg** the target variable, set to 1 if it is an iceberg, and 0 if it is a ship.

- First we have downloaded the .7Z files and extracted the json files and loaded it into pandas dataframe and found that there are missing values in "inc-angle" column

- So, we replaced the missing("na") values with its mean.

First method:

- Used Keras module for building Convolutional Neural Network model for this image classification problem. Transformed each array of image(5625 array values) into a feature attribute and added "band-2" column. The output variable will be "is-iceberg".

- Splitted the data in the ratio of 60(train), test (40).

- Built a sequential model of neural net.

- Used different layers of neural nets like 3, 4 and 5 layers.

- Added different neuron counts in each layer with different activation functions.

- As it is a binary classifier we used "sigmoid" activation function in the last level.

- Compiled the model using “adam” optimizer.

- Evaluated the model on the train data. The accuracy level of prediction is coming around 51 percent.

Second method:

- Observed that only the image of the “iceberg” or “ship” is captured in the 30*30 [20:50,20:50] area of 75*75.

- Extracted that area pixels and used both” band-1” and ” band-2” images and “inc-angle”(1801 features).

- Followed the first method with different number of activation layers, different activation functions. This time got an accuracy level of 53 percent.

Third method:

- Used the same features from the second method and built a KNeighborsClassifier (n=3) model. Splitted the data in the ratio of 90(train), test (10).

- Fitted the model on training data and predicted the values of test data.

- Found out the confusion matrix and got an accuracy level of 59 percent.

- Used the 10-fold cross validation technique on the whole data and got an average accuracy of 79 percent.Right now, we are using Trial and error method for finding the optimal solution by varying the model and parameters.

Preprocessing the data:

- There are some missing values in the “incidence angle” column from the data. Replaced the missing value with the mean of the “incidence angle” . So that we do not drop many rows in the dataset. Some sample images from the data are as follows:

Some sample images from the data are as follows:

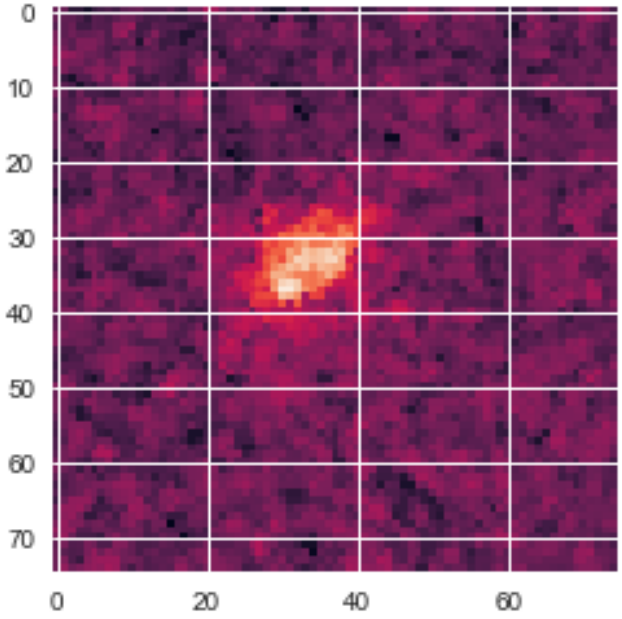
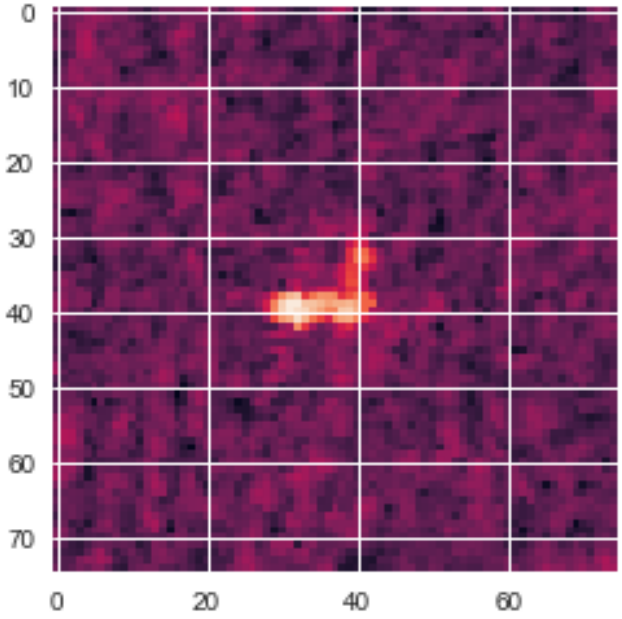
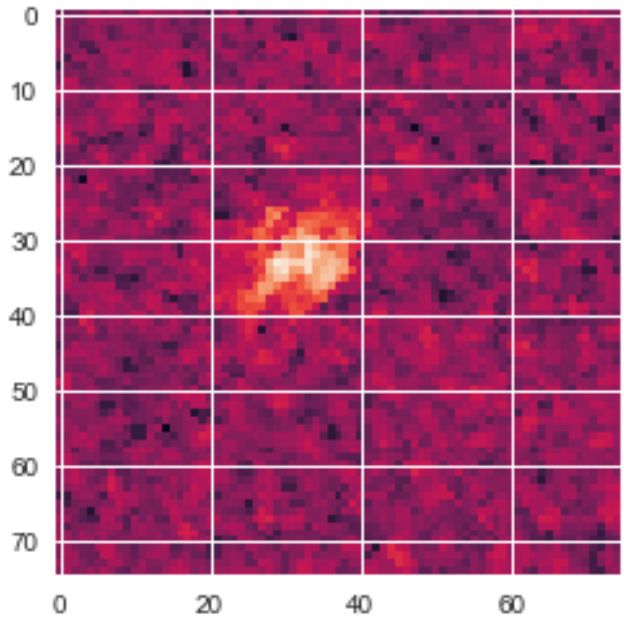


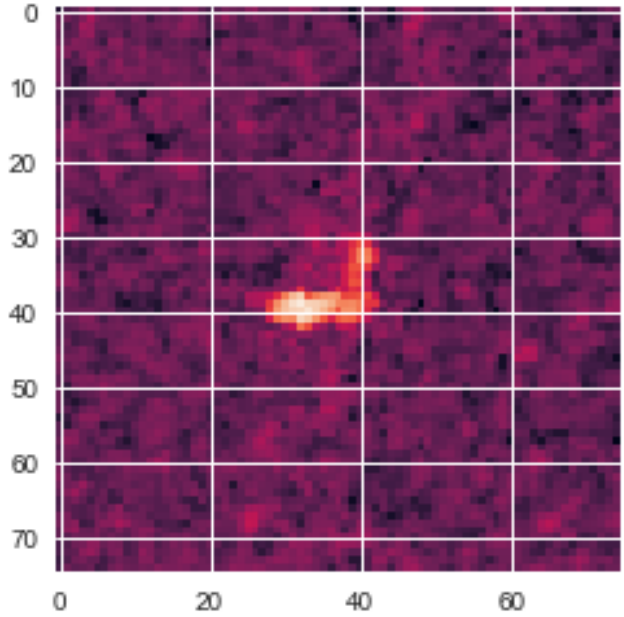
Image of iceberg for band1 values



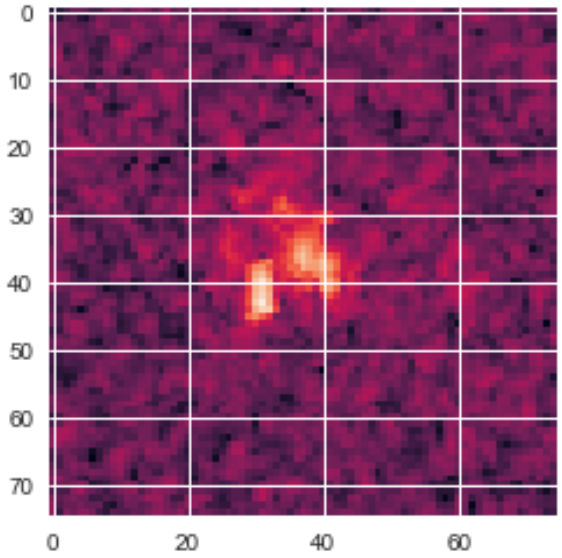
Sample image of iceberg observed for band1 values



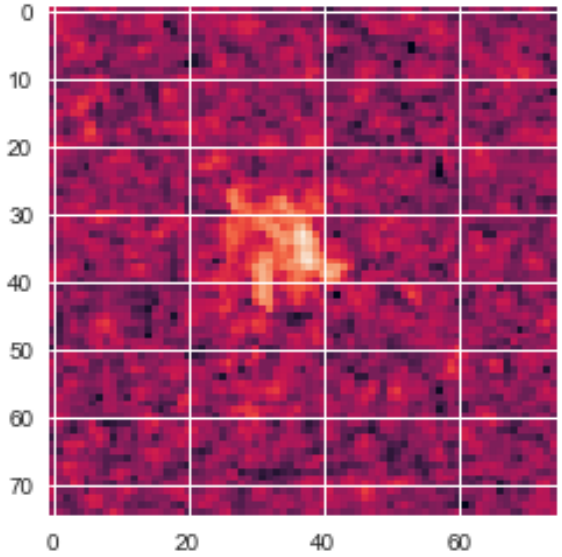
Sample image of iceberg observed for band 2 values



Sample image of ship observed for band 2



Sample image of iceberg observed for band1



Sample image of iceberg observed for band2

- One observation is that most of the data is concentrated on the [20:50,20:50] matrix grid and the rest of the image has ocean as a background. Second observation is that there is not much difference if we take band-1 data or band-2 data. Reshape the data in each column of band-1 and band-2 and select the [20:50,20:50] matrix grid, and reshape it to an array and convert each value of the array into a attribute for building the model We have 900*2 + 1(inc-angle)= 1801 attributes and one value to predict i.e., is-iceberg(Binary Variable).

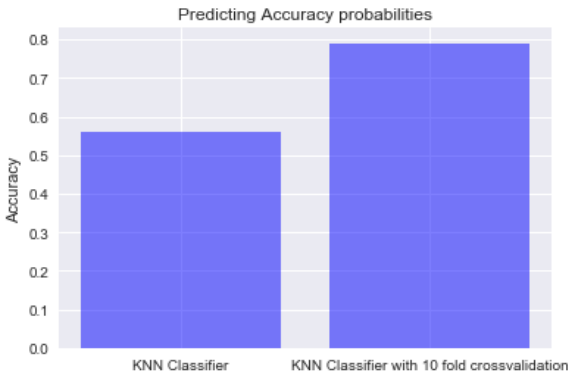
Implementing KNearest Neighbours classifier:

- Splitted the data into train and test in the ratio of 80/20

- Fitted the Scikit learn's KNearest Neighbours classifier model to fit the data on the training data.

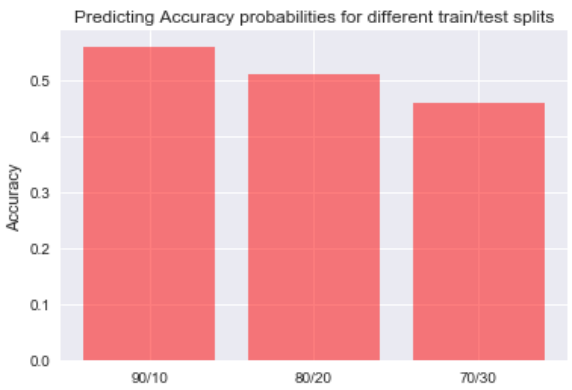
- Found out the confusion matrix and got an accuracy of 56 percent for prediction on test data

- Used Scikit learn's 10 fold cross validation score to find the mean accuracy and got an accuracy of 79 percent on the total data.

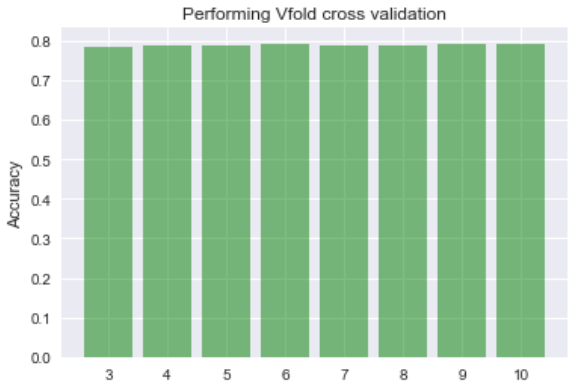


KNN predictions

- If we do Vfold cross validation the model predicting accuracy will improve



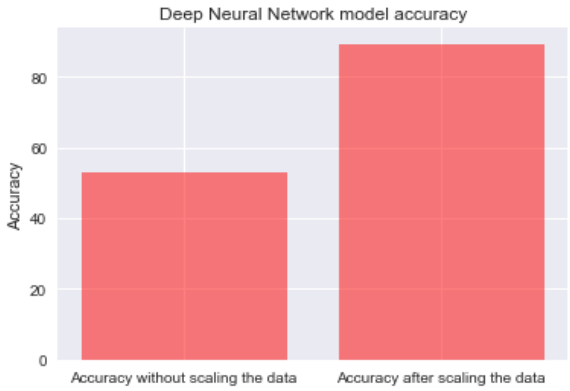
For this data the here maximum accuracy is achieved when we keep the train/test ratio to be 90/10.



- We can conclude that for this data set Vfold cross validation is independent of number of cross validations we have done.

Implementing TensorFlow (Keras Deep Neural network):

- Splitted the data into train and test in the ratio of 60/40



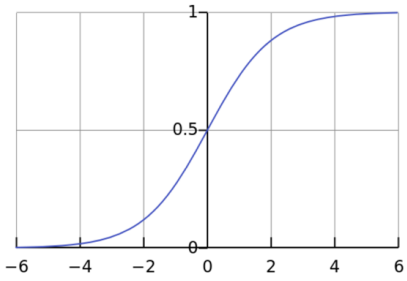
- Accuracy Improves way better than expected for deep neural nets if we scale the attributes.

- In neural networks activation functions work like an abstraction of the rate of action potential firing in a cell. The working of an activational function is that which decides whether a neuron should fire an input or not. The output is calculated as the sum of weighted input plus the bias and decided if it should be fires or not based on the outcome generated. The decision is taken by the activation function to fire or not.

- Some of the activation functions used by us are:

- 1. Sigmoid Function:

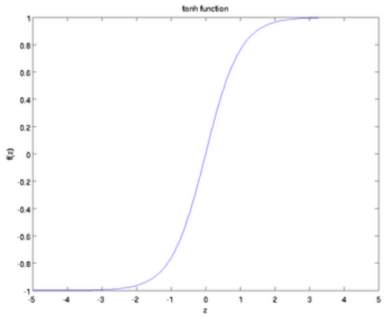
$$A=\frac{1}{1+e^{-x}}$$



- This function is like a step function with values ranging from (0,1). This is a non-linear function. The specialty of this function is that for input range -2 to 2 the change in output is significant that is the slope is steep for the curve showing the tendency to bring the values to either side of the curve making clear distinctions on prediction. Moreover, since the range is (0,1) unlike a linear function that has a range (-inf, inf), the activations are bound in a range. The problem with sigmoid is that it gives rise to the problem of "vanishing gradient" – the gradient is that small or vanished when the activations reach the horizontal part of the curve.

- 2. Tanh function :

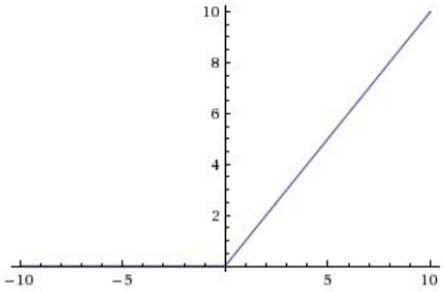
$$A=\frac{e^{x}-e^{-x}}{e^{x}+e^{-x}}$$



- Tanh function is similar to sigmoid function but the values range here form (-1,1), so there is no worry of activation blow ups. The major difference is that the gradient is stronger than sigmoid. Depending on the gradient strength the function is chosen between sigmoid and tanh

- 3. Relu function

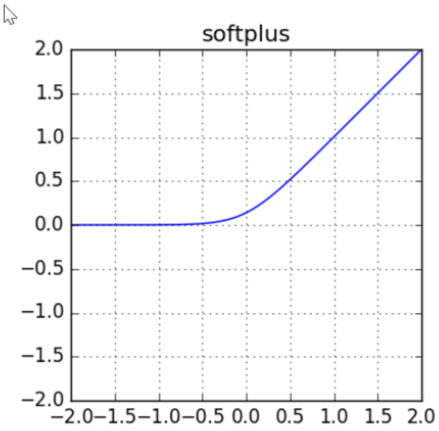
$$A=max(0,x)$$



- Relu function is a nonlinear function that ranges from [0,). Since it is not bound there is a chance of activation blow up. Any function can be approximated using Relu. Relu function reduce the activation costs. For activations in the region where gradient is 0, the neurons will stop responding to variations in error/ input.

- 4. SoftPlus function

$$A=log(1+e^{x})$$



- Softplus function is also a nonlinear function like Relu and has a range of (0,). The derivative of softplus is sigmoid function. This is essentially a smoother version of Relu and can be approximated by a max function commonly known as Rectified Linear Function

Adam Optimizer

- Adam is an acronym that is derived from Adaptive moment estimation. Adam optimization is straightforward and computationally efficient. It also uses less memory. It is used for problems that are have large data and parameters and is appropriate for non-stationary objectives. For problems with noisy and sparse gradients, this method is ideal.

Binary Cross Entropy

- The classic way of generating an error model in neural networks is called binary cross entropy. It is defined as, If our prediction is y' and real value is y, then binary cross entropy is defined as:
$b(y,y') = -ylog(y') - (1-y)log(1-y')$
Epoch and batch-size

- In neural network one epoch is onefoward and backward pass of all training examples and the batch size is the number of training examples in one forward/ backward pass.

4.0.1 Citations and Subsubsection

- Dan Jurafsky. "Text Classification and Naïve Bayes", https://web.stanford.edu/class/cs124/lec/naivebayes.pdf

- Francisco Iacobelli. "Text Classification and Naïve Bayes", https://www.youtube.com/watch?v= EGKeC2S44Rst=569s

- Baiju NT. " 12 common problems in Data Mining" http://bigdata-madesimple.com/12-common-problems-in-data-mining/

- Dr. Jason Brownlee. "Binary Classification Tutorial with the Keras Deep Learning Library" https://machinelearningmastery.com/binary-classification-

  tutorial-with-the-keras-deep-learning-library/

- Karlijin Willems."Keras Tutorial: Deep Learning in Python" https://www.datacamp.com/community/tutorials/deep-learning-python

Acknowledgments

The authors wishes to thank Prof. Mehmet Dalklic for numerous helpful discussions and comments. Also, thanking Marcin Malec, lead Associate Instructor, for his guidance and valuable suggestions. We also want to extend thanks to the Kaggle community for their varied analysis and approaches that inspired us to solve the problem.

References

- Naïve Bayes Bag of words classifier https://web.stanford.edu /class/cs124/lec/naivebayes.pdf

- Modified Naïve Bayes Bag of words classifier https://www.youtube.com/watch?v= EGKeC2S44Rst=569s Common problems in data mining http://bigdata-madesimple.com/12-common-problems-in-data-mining/
  Keras classification tutorial

- https://machinelearningmastery.com/binary-classification-tutorial-with-the-keras-deep-learning-library/

- Pandas Data Frame
  https://pandas.pydata.org/

- Scikit learn Python module http://scikit-learn.org/stable/
  Scikit learn Python module for classification models

- http://scikit-learn.org/stable/supervised-learning.htmlsupervised-learning

- Scikit learn Python module for preprocessing
  http://scikit-learn.org/stable/modules /preprocessing.htmlpreprocessing

- Tensor flow backend for keras
  https://www.tensorflow.org/

- Keras library for implementing neural net
  https://keras.io/

- Deep learning in python using keras
  https://www.datacamp.com/community/tutorials/deep-learning-python