1. **Abstract**

Pneumonia accounts for over 15% of all deaths of children under 5 years old internationally. In 2015, 920,000 children under the age of 5 died from the disease. In the United States, pneumonia accounts for over 500,000 visits to emergency departments and over 50,000 deaths in 2015, keeping the ailment on the list of top 10 causes of death in the country. Our train and test set contains 26684 and 3000 images respectively. Our goal was create binary classifier to predict “Pneumonia” containing 0 and 1 classes.

We transformed images into arrays and applied normalization to build the machine and deep learning models. We applied different machine learning models like Logistic Regression, Naïve Bayes, KNN and Decision Tree and One deep Convolutional Neural Network. We applied various evaluation techniques like accuracy, precision score, recall score and f1 score on both train and test datasets respectively in which we got to know that CNN was giving the best results having least Lag Score of 0.014, having train accuracy of 77.74% and test accuracy of 77.73%.

1. **PROBLEM DESCRIPTION**

Our goal was create binary classifier to predict “Pneumonia” containing 0 and 1 classes.

1. **DATA EXPLORATION**

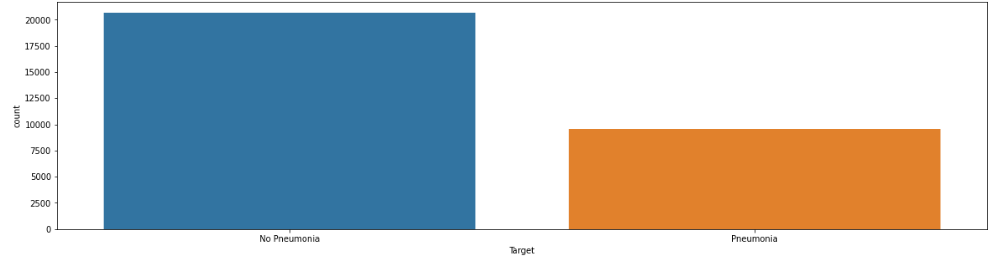
A preliminary exploration of the dataset was conducted to understand its characteristics better.

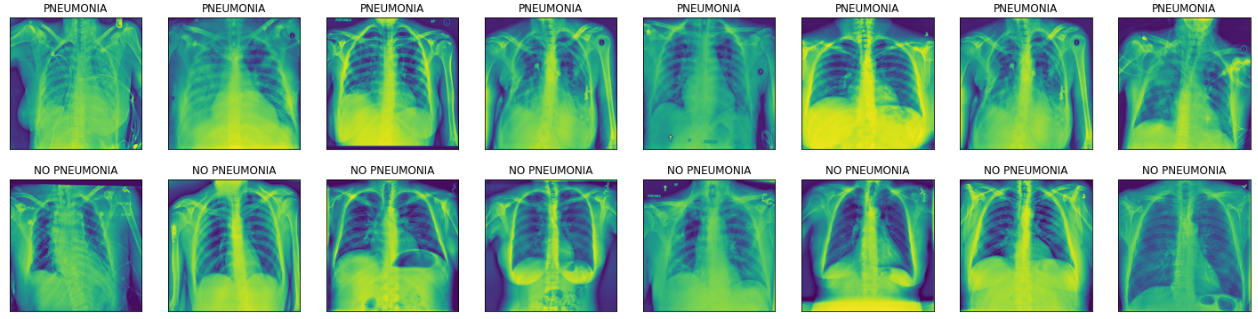
* 1. **Number of Images**

The train set contains 26684 images.

* 1. **Classes Distribution**

Before Augmentation:

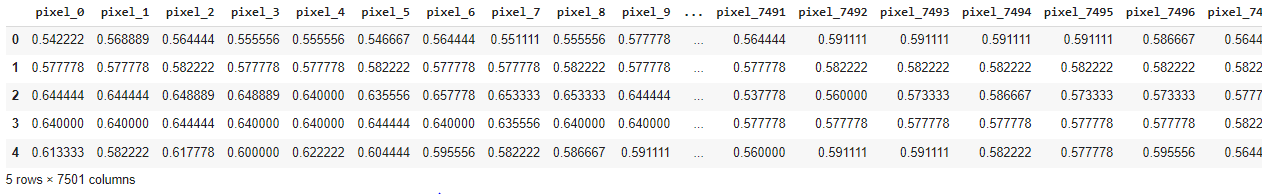




Augmentation: In the process of image augmentation techniques we applied the following steps to each train image:

* We resized the image to a same size.
* Applied 90 and 180 rotation.
* Applied blur.

Flatten Image:



1. **DATA PREPROCESSING**

**4.1 Feature Scaling/Normalization**

Feature scaling through Standardization (Z-score Normalization) is an essential preprocessing requirement for many machine learning algorithms, which helps them interpret features with different magnitudes and distributions equally. The data was split into features and target.

1. **DATA MINING TECHNIQUES/ALGORITHMS USED**

The supervised/predictive learning was used since the desired result of the model is known through the target variable (Class) with labels 1 and 2. Since the target variable has discrete and binary class labels, this was a binary classification problem. Classification algorithms such as Logistic Regression, Naïve Bayes, KNN and Decision Tree and One deep Convolutional Neural Network were built using standardized dataset, and their accuracies and results were compared.

**Train/Test Split**

The features and target were split into mutually exclusive train and test sets for standardized dataset using holdout with stratified sampling method by reserving 2/3 for training (model construction) and 1/3 for testing (prediction, accuracy estimation).

**5.1 Support Vector Machine**

Support Vector Machine (SVM) is a supervised machine learning algorithm that analyzes the data where each item is plotted as a point in n-dimensional space for classification. SVM finds the points closest to a line from both the classes called the support vectors and computes the distance between the line and the support vectors called the margin. SVM performs classification by finding a linear hyperplane (decision boundary) that linearly separates our data into two classes (placed and not placed). SVM is based on the margin maximization principle, hence finds the hyperplane that best maximizes the margin between the two classes. SVM was chosen since the number of dimensions in our case is considerable compared to the number of samples. SVM algorithm offers kernel functions such as Linear, Polynomial, Radial Basis Function (RBF), and Sigmoid to perform its processing. “Linear” function was chosen to build/fit the SVM model using the training set, and the model was then used to predict new values from the test set.

**5.2 Decision Tree**

The decision tree is a supervised machine learning algorithm used for classification and is a method for decision making over time with uncertainty. Decision trees classify data by sorting them down from the root to one of the leaf nodes, where the leaf represents the classification to the data. The process is recursive and is repeated for every subtree. Hunt’s algorithm is usually used to build decision trees. It starts by assuming that all data instances belong to the majority class. The best attribute is chosen to be the root of the tree based on how well the attribute splits the data into different groups using a node purity metric. Leaf nodes are created when a decision from a node split leads to data instances of only one class. Internal nodes are made using the best remaining attribute when a decision leads to data instances of two or more classes. The decision tree classifier was built/fit using the training set and used to predict new values from the test set. Inside the classifier, the node purity measure was specified as “entropy” for the information gain.

**5.3 Naïve Bayes**

Naïve Bayes classifier is a supervised machine learning classification algorithm based upon Bayes’ Theorem, which uses the entire dataset to classify a new data instance. It utilizes the most basic probability knowledge and assumes that each attribute makes an equal and independent contribution to the class value. In other words, the presence of one feature has nothing to do with that of any other, which is a naïve assumption about data in real life. Naïve Bayes uses prior information to compute the probability of related events to predict classification outcomes. Naïve Bayes has been chosen for its simplicity, accuracy, speed, ease of implementation and since it also requires less training data as in our case. The Naïve Bayes model was built/fit using the training set and used to predict new values from the test set.

**5.4 K-Nearest Neighbor**

K-Nearest Neighbor (KNN) is a non-parametric, lazy learning algorithm and is considered among the simplest supervised machine learning algorithms. KNN has no explicit training phase and does not build the model explicitly. KNN is based on feature similarity and relies on the assumption that similar things exist in close proximity. KNN computes distances to other training records (using a distance metric like Euclidean), identifies k-nearest neighbors and determines the class label of the unknown record by taking the majority vote (mode) of the class labels of k-nearest neighbors. KNN has been chosen for its simplicity, ease of implementation and relatively high accuracy. KNN models were built/fit for different k values (from 1 to 9) using the training set and used to predict new values from the test set.

**5.5 Logistics Regression:**

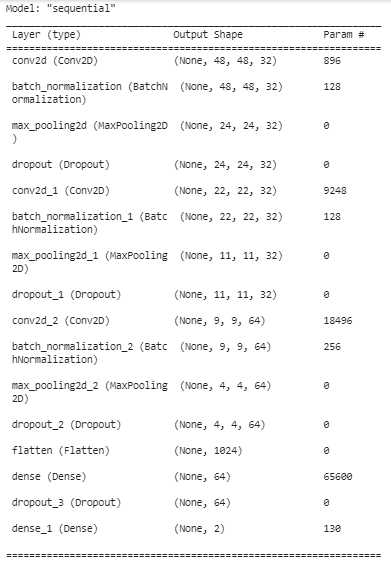
Logistic regression is basically a supervised classification algorithm. In a classification problem, the target variable(or output), y, can take only discrete values for a given set of features(or inputs), X. Contrary to popular belief, logistic regression IS a regression model. The model builds a regression model to predict the probability that a given data entry belongs to the category numbered as “1”. Just like Linear regression assumes that the data follows a linear function, Logistic regression models the data using the sigmoid function. Logistic regression becomes a classification technique only when a decision threshold is brought into the picture. The setting of the threshold value is a very important aspect of Logistic regression and is dependent on the classification problem itself.

**5.6 Convolutional Neural Network:**

A Convolutional Neural Network (CNN) is a Profound Learning calculation that can take in an information picture, dole out significance (learnable loads and predispositions) to different perspectives/objects in the picture, and have the option to separate one from the other. The pre-handling expected in a ConvNet is a lot of lower when contrasted with other order calculations. While in crude techniques channels are hand-designed, with enough preparation, ConvNets can become familiar with these channels/qualities.

The architecture of a ConvNet is analogous to that of the connectivity pattern of Neurons in the Human Brain and was inspired by the organization of the Visual Cortex. Individual neurons respond to stimuli only in a restricted region of the visual field known as the Receptive Field. A collection of such fields overlap to cover the entire visual area.

CNN Architecture:



**Results:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Classifier** | **Train Accuracy** | **Test Accuracy** | **Precision Score** | **Recall Score** | **F1 Score** |
| **Logistic Regression** | 77.750 | 77.450 | 0.318 | 0.010 | 0.020 |
| **Decision Tree** | 77.786 | 77.667 | 0.364 | 0.003 | 0.006 |
| **Naïve Bayes** | 71.886 | 72.217 | 0.284 | 0.162 | 0.207 |
| **KNN (Best k=2)** | 88.464 | 78.367 | 0.567 | 0.123 | 0.203 |
| **CNN** | 77.735 | 77.749 | 0.667 | 0.003 | 0.006 |

**Conclusion:**

We used many different types of classifier for our model to get the best results as possible as we trained on classifiers like SVM, decision tree, Naïve Bayes, KNN and Logistics Regression.

Least accuracy 72.216% is achieved by Naïve Bayes and greatest test accuracy of 78.366% were achieved by KNN (N = 2). We calculated the model accuracy on both train and test datasets respectively and calculated the “Lag Score” (Train Score – Test Score), in which we got to know that CNN was giving the best results having least Lag Score of 0.014, having train accuracy of 77.735% and test accuracy of 77.749%.