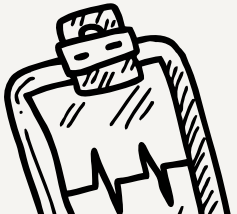
Comparative Analysis of Traditional Machine Learning Models for Pneumonia Detection

JACK BANSEN,
PATRICIA FELIZ,
SETHAN CUMMINGS,
NITISH MAINDOLIYA,
JAMIE HUANG
(SECTION 4 / GROUP 2)

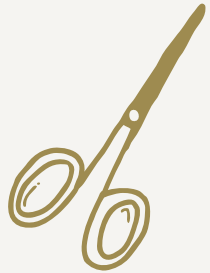
Problem Statement

Can we accurately detect pneumonia from chest X-ray images using traditional machine learning classifiers? How does preprocessing (e.g., scaling, PCA) affect each model's performance?



Dataset Overview

- **Source:** Kaggle – Chest X-Ray Images (Pneumonia) by Paul Timothy Mooney
- **Total Images**: 5,863 chest X-ray images
- **Categories**: Two classes – *Normal* and *Pneumonia*
- **Image Size**: Varies
- **Dataset Split (after initial downloading and preprocessing)**
  - **Training Set**: 2000 images (1000 per class)
  - **Test Set**: 500 images (250 per class)
  - **Image Size:** 256x256
- **Data Source**: Images sourced from Guangzhou Women and Children's Medical Center



Background

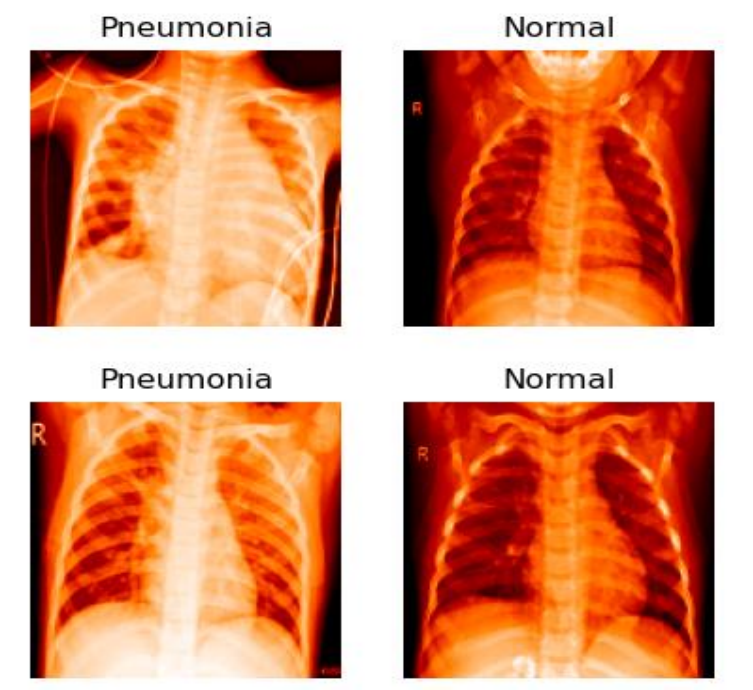


Figure: Heatmap of X-Ray Images

- Normal chest X-Rays appear **darker** as lungs are filled with air.
- Pneumonia chest X-Rays appear **whiter** with **cloudy or patchy opacities**.
- These strong visual differences in grayscale intensity and structure made this dataset a good candidate to use to test Traditional ML classifiers.

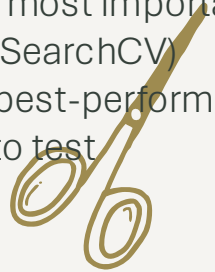
Methodology

1. Data loading
   - Folders of the form Pneumonia/Test, Pneumonia/Train, Normal/Test, Normal/Train
   - Loaded in using sklearn's load_files (labels were encoded)
2. Image processing pipeline
   - Images(256x256) => Grayscale => Images(128x128) => Flattened np.array
3. Data Preprocessing
   - Three variants were made
     - Raw: Original image **(16384 features)**
     - Scaled: StandardScalar **(16384 features)**
     - Scaled+PCA: StanadardScalar + PCA retaining 95% variance **(357 features)**
4. Model Training/Testing Pipeline
   - Data => RandomizedSearchCV => **(Perceptron, SVM, Logistic Regression, KNN)**
   - The training time was measured, and the best model was used to classify the test set

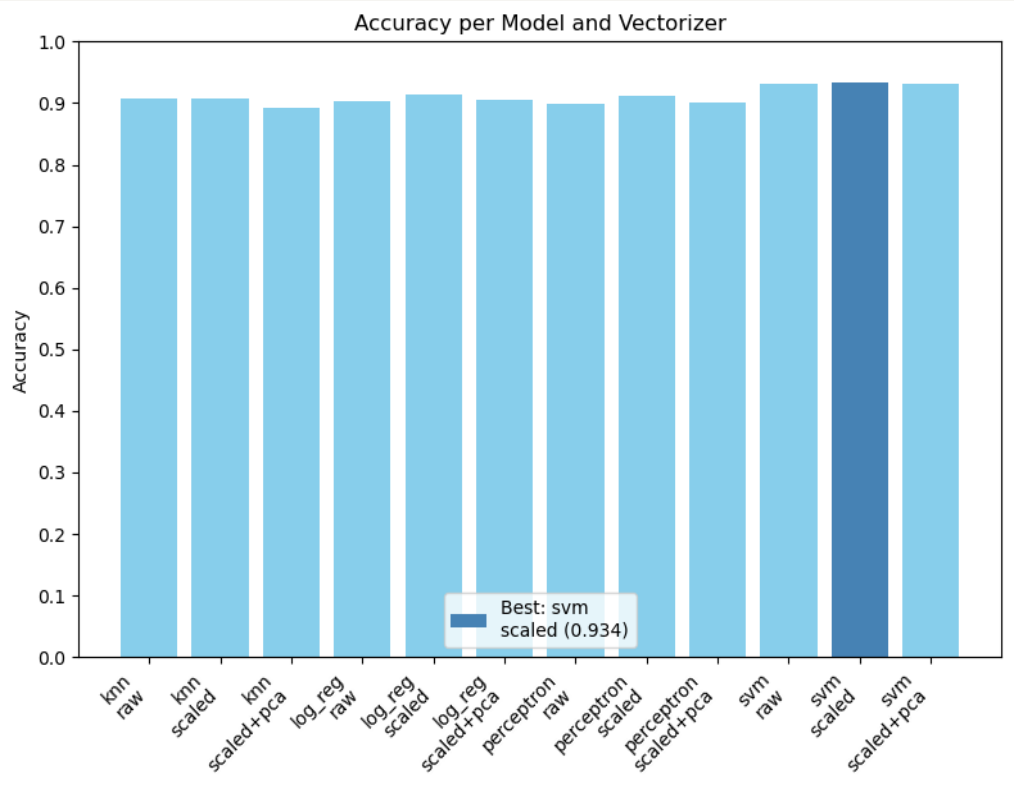
Model, Math and Compute

1. **Logistic Regression** (sklearn.linear_model.LogisticRegression)
   - Learns weights to model probability using the sigmoid function
2. **Support Vector Machine** (sklearn.svm.SVC)
   - Finds the optimal hyperplane that maximizes the margin between classes
3. **Perceptron** (sklearn.linear_model.Perceptron)
   - Updates weights based on misclassification using a rule
4. **K-Nearest Neighbors** (sklearn.neighbors.KNeighborsClassifier)
   - Predicts class by majority vote among k nearest neighbors based on distance metrics

5. **Principal Component Analysis** (sklearn.decomposition.PCA)
   - Used to reduce the number of features while keeping the most important information
6. **RandomizedSearchCV** (sklearn.model_selection.RandomizedSearchCV)
   - Tries out different combinations randomly and picks the best-performing ones
   - Better than GridSearchCV when there are many options to test

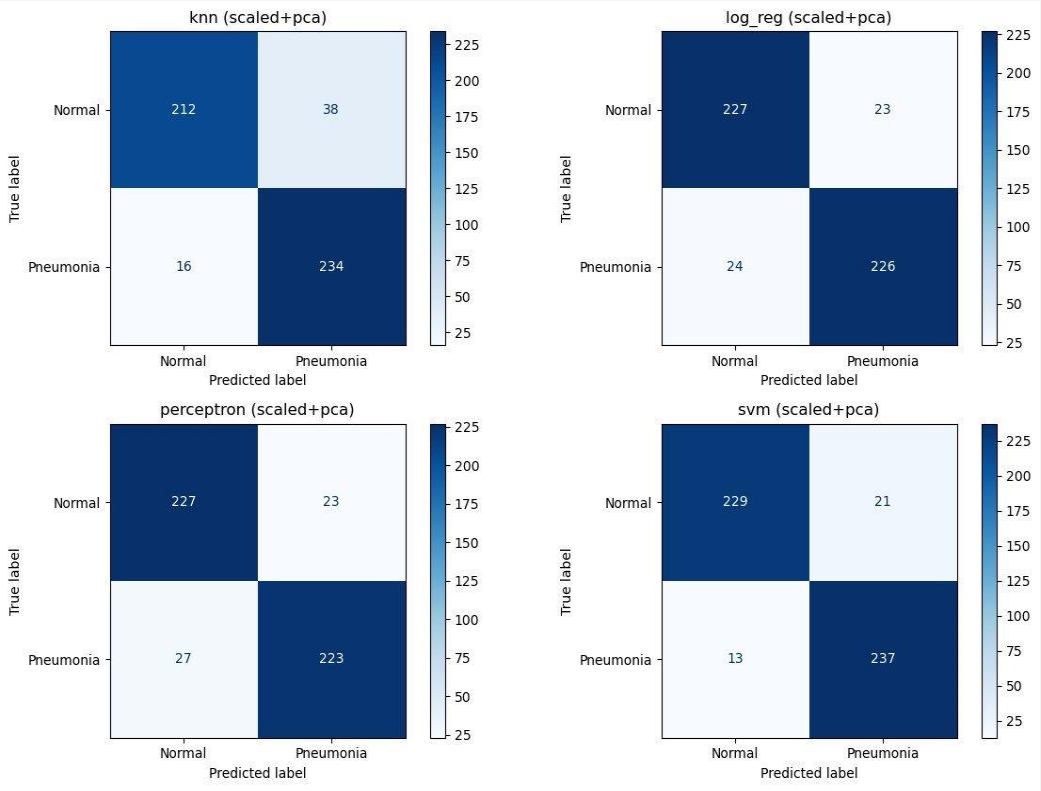


How accurate were we ?



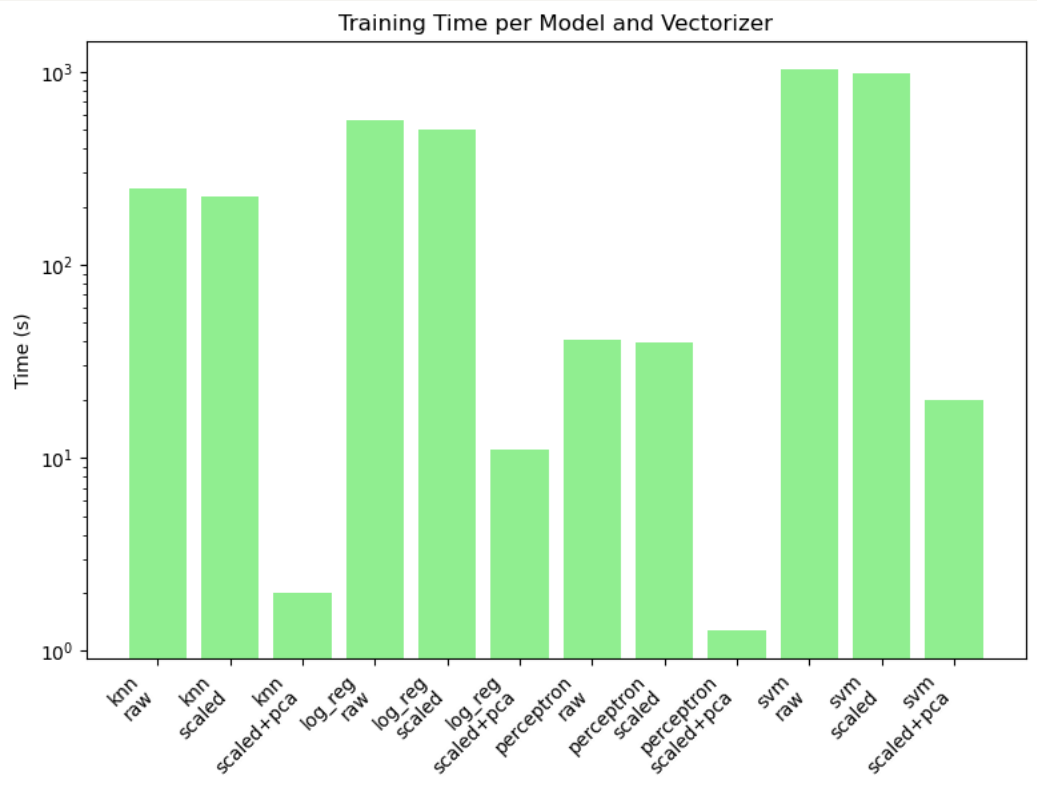
- All traditional classifiers were able to achieve accuracy **above 85%.**
- SVM is the best in general (scaled being the highest)
- **Note:** All data variants (raw, scaled, scaled+PCA) have almost equal accuracy.

How many misclassifications ?



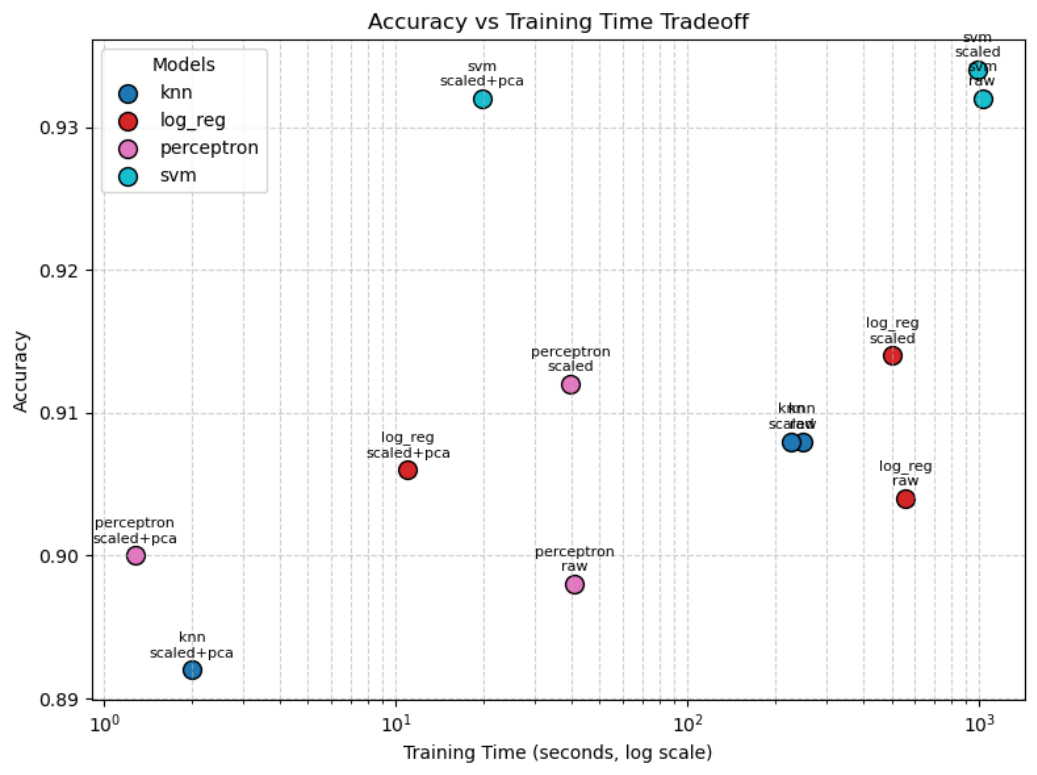
- Not many misclassification **(less than 60 out of 500),** indicating high precision (> 85%) values for all the models
- **Note:** SVM is the best in general with the fewest misclassifications

How fast ? (training time)



- SVM was the slowest model to train in general.
- Raw and scaled data both took almost equal time to train.
- **Note:** PCA data was the fastest amongst each classifier (significantly less features)

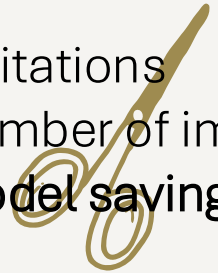
Accuracy/Training Time trade-off



- PCA data was in every case very close to the original/scaled data in terms of accuracy
- PCA data took a fraction of the time to compute (e.g. SVM: 1024s vs 19s)
- **Note:** Optimal params for SVM in regular/PCA data were the same.

Limitations/Difficulties/Solutions

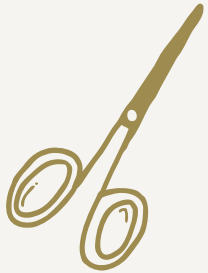
- Initial set of data was difficult to handle
  - Visual variation between classes was high.
  - Image quality & preprocessing were poor because there were no clear markers on the images
  - Without domain-specific knowledge, it was unclear which features to target.
  - Traditional ML models failed to classify reliably on raw image data.
- **Solution: Switched to a simpler binary dataset**
- Had difficulties with GitHub due to file size limitations
- Models took a lot of time to train due to the number of images.
- **Solution: Used efficient data management/model saving**



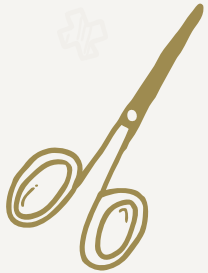
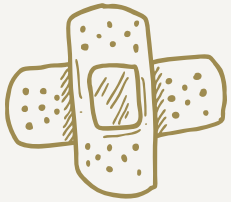
Conclusion

Traditional ML models (especially SVM) can be fast and effective for pneumonia detection when paired with scaling and dimensionality reduction.

While they lack the power of deep learning models for complex, multi-label classification tasks, they remain valuable for interpretable, resource-efficient pipelines.





Thank You