Exploring Machine Learning Techniques on Diverse Datasets

c
Atharva Gundawar(1230126029)
MS in Robotics and Autonomous Systems

Atharva Hude(1229854940)
MS in Robotics and Autonomous Systems

Sharan Patel(1230596772)
MS in Data Science, Analytics and Engineering

1. Introduction

1.1 Datasets Overview

The **UCI Adult dataset**, composed of census data, presents a classification challenge with attributes like age, education, and occupation, aimed at predicting income levels. It contains both continuous and categorical variables, often requiring preprocessing for handling missing values and class imbalance. The **Wisconsin Breast Cancer dataset** features numerical attributes derived from cell nuclei characteristics in breast mass images, focusing on binary classification into benign or malignant categories. This dataset is notable for its high dimensionality relative to the number of samples, necessitating careful feature selection and analysis. Finally, the **Fashion MNIST dataset** offers a collection of 70,000 28x28 grayscale images of fashion items, divided into 10 categories. It poses challenges in image processing and pattern recognition, demanding robust feature extraction and classification techniques in computer vision.

1.2 Model Classes for Classification

Logistic Regression and SVM are key for binary classification and high-dimensional data, respectively, with **SVM's** adaptability through kernel functions. Meanwhile, **PCA+k-NN** simplifies dimensionality reduction and classification, whereas **FNN** and **CNN** excel in capturing complex, non-linear relationships, particularly in image data.

1.3 Team Member's Contribution

·**Atharva Gundawar:**

Implementing all models on the UCI Adult dataset and writing the term paper.

·**Atharva Hude:**

Implementing all models on the Wisconsin Breast Cancer dataset and writing the term paper.

·**Sharan Patel:**

Implementing all models on the Fashion MNIST dataset and leading the result collation process.

2. UCI Adult Dataset

The UCI Adult Dataset, also known as the ”Census Income” dataset, is a collection of data derived from the 1994 US Census database. It contains over 30,000 instances, each with 14 attributes, including age, work class, education, marital status, occupation, relationship, race, sex, capital gain/loss, hours per week, and native country. The primary task associated with this dataset is to predict whether an individual earns more than $50,000 per year based on these attributes.

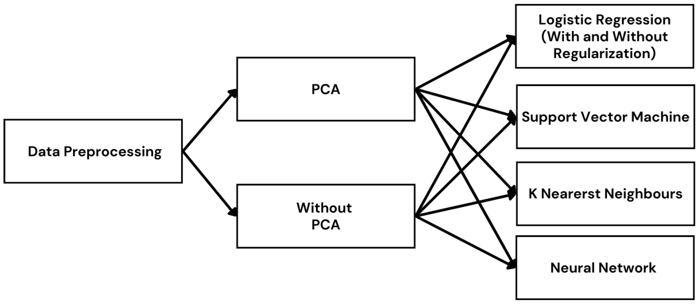

Figure 1: Flow of data

2.1 Preprocessing

The models are trained using data that has been meticulously preprocessed, with two approaches: one directly utilizing the preprocessed data and another employing Principal Component Analysis (PCA) for dimensionality reduction. This dual approach allows for a comparative analysis of model performance with and without PCA.

2.2 Logistic Regression with and without Regularization

2.2.1 hyperparameters used and values and chosen

These values test solvers (`liblinear`, `sag`), penalties (`l1`, `l2`, `elasticnet`), regularization strength (C from 0.00001 to 100000), intercept fitting (`True`, `False`), tolerance (0.0001, 0.001, 0.01), and iteration limit (1000).

2.2.2 Results

Best hyperparameters: {’C’: 100000.0, ’fit_intercept’: False, ’max_iter’: 1000, ’penalty’: ’l1’, ’solver’: ’liblinear’, ’tol’: 0.01} This was when we ran the model on without PCA treated data and with regularization.

2.3 SVM with and without kernels

2.3.1 hyperparameters used and values and chosen

The `param_grid_svm` tests SVM parameters: regularization (C at 0.1, 1, 10), kernel types (`linear`, `rbf`), and gamma values (`scale`, `auto`).

2.3.2 Results

Best parameters for SVM: {'svm__C': 1, 'svm__gamma': 'auto', 'svm__kernel': 'linear'} Best score for SVM: 0.8440574024950911. This was when we ran the model on without PCA treated data and with kernels.

2.4 K nearest neigbour

2.4.1 hyperparameters used and values and chosen

The param_grid tests KNN parameters: neighbors (1 to 30), weight functions (uniform, distance), and distance metrics (euclidean, manhattan, chebyshev, minkowski).

2.4.2 - Results

{'knn__metric': 'manhattan', 'knn__n_neighbors': 29, 'knn__weights': 'uniform'}
Accuracy: 0.84

2.5 Feedforward Neural Network

2.5.1 hyperparameters used and values and chosen

The hyperparameters for configuring a Feedforward Neural Network (FNN) model are designed to provide comprehensive control over the model's architecture and training process. num_layers determines the depth of the network with a range from 2 to 6 layers, while units_i allows customization of the number of units in each layer, offering choices from 32 to 512 with a step size of 32. activation enables the selection of activation functions ('relu,' 'tanh,' or 'sigmoid') for each layer. l2_reg specifies L2 regularization strength for weight regularization within each layer in a logarithmic range from 1e-5 to 1e-2. dropout_i allows fine-grained control over dropout rates for regularization (ranging from 0.1 to 0.5). Finally, learning_rate governs the learning rate employed by the Adam optimizer during training, adjustable within a logarithmic range from 1e-4 to 1e-2. These hyperparameters collectively empower the customization and optimization of FNN models for diverse machine learning tasks and datasets.

2.5.2 Results

Test Accuracy: 0.8553661704063416, Test Loss: 0.3448124825954437 Num_layers: 4
Refer Table 1 and 2 in the Appendix

3. Wisconsin Breast Cancer dataset

3.1 Logistic Regression with and without Regularization

3.1.1 hyperparameters used and values and chosen

The `param_grid` tests various hyperparameters: solvers ('liblinear', 'sag'), penalties ('l1', 'l2'), regularization strength ('C' across 50 values from 10^{-5} to 10^{5}), fitting the intercept ('True' only), and the number of iterations (fixed at '1000').

3.1.2 Results

Best hyperparameters for the model are: {'C': 0.04714866363457394, 'fit_intercept': True, 'max_iter': 1000, 'penalty': 'l2', 'solver': 'liblinear'}.

Accuracy: 0.99

3.2 SVM with and without kernels

3.2.1 hyperparameters used and values and chosen

The `param_grid_svm` tests SVM parameters: regularization ('C' at 0.1, 1, 10, 100), kernels ('linear', 'poly', 'rbf'), polynomial degrees (2, 3, 4 for the 'poly' kernel), and gamma values ('scale', 'auto').

3.2.2 Results

Best parameters for SVM: {'svm__C': 100, 'svm__degree': 2, 'svm__gamma': 'scale', 'svm__kernel': 'linear'}

Best score for SVM: 0.9736263736263735.

3.3 K nearest neigbour

3.3.1 hyperparameters used and values and chosen

The `param_grid` tests KNN parameters: neighbors (1 to 30), weight functions (`uniform`, `distance`), and distance metrics (`euclidean`, `manhattan`, `chebyshev`, `minkowski`).

3.3.2 - Results

Best hyperparameters: {'knn__metric': 'manhattan', 'knn__n_neighbors': 6, 'knn__weights': 'distance'}

Accuracy : 0.89

3.4 Feedforward Neural Network

3.4.1 - hyperparameters used and values and chosen

In the context of building Feedforward Neural Networks (FNN), an extensive set of hyperparameters was meticulously defined to tailor the architecture and training process. These hyperparameters include the number of hidden layers (ranging from 1 to 3), the number of neurons within each hidden layer (32 to 512 with step size 32), the choice of activation functions ('relu,' 'tanh,' or 'sigmoid'), L2 regularization strength (sampled logarithmically from 1e-5 to 1e-2), dropout rates for each hidden layer (ranging from 0.1 to 0.5 with step size 0.1), and the learning rate (sampled logarithmically from 1e-4 to 1e-2). This versatile selection of hyperparameters empowers the fine-tuning and optimization of FNN models to suit specific datasets and tasks, allowing for flexibility in network architecture and training dynamics.

3.4.2 Results

Test Accuracy: 0.9649122953414917, Test Loss: 0.4382508397102356

'num_layers': 3,

Refer Table 3 and 4 in the Appendix

4. Fashion MNIST

4.1 Logistic Regression with and without Regularization

4.1.1 hyperparameters used and values and chosen

The `param_grid` tests regularization strength (C at 0.01, 0.1, 1, 10, 100), penalty types (`None`, `l1`, `l2`), and the solver (`liblinear`).

4.1.2 Results

Best parameters: {'C': 1, 'penalty': 'l2', 'solver': 'liblinear'}
Test set accuracy: 0.8341

4.2 SVM with and without kernels

4.2.1 hyperparameters used and values and chosen

The `param_grid` tests regularization strength (C at 0.1, 1, 10), kernel type (`poly` only), and gamma values (`scale`, `auto`).

4.2.2 - Results

Best parameters: 'C': 10, 'gamma': 'scale', 'kernel': 'poly' Test set accuracy: 0.6575

4.3 K nearest neigbour

4.3.1 hyperparameters used and values and chosen

The `param_grid` tests the number of neighbors (10 to 20), weight functions (`uniform`, `distance`), and distance metrics (`euclidean`, `manhattan`).

4.3.2 Results

Best parameters for k-NN: {'metric': 'manhattan', 'n_neighbors': 10, 'weights': 'distance'}
Test set accuracy: 0.8636.

4.4 Convolutional Neural Network

4.4.1 hyperparameters used and values and chosen

In the design of Convolutional Neural Networks (CNNs), a comprehensive set of hyperparameters has been thoughtfully crafted with specific value ranges. These include configurations for the first convolutional layer, additional convolutional layers, and dense layers. For the first convolutional layer, the number of filters ranges from 32 to 256 with a step of 32, and the kernel size is chosen between 3 and 5.

4.4.2 - Results

Best val_accuracy: 0.8525000214576721

5. Value of Concepts Learned in Course

5.1 Application of Course Techniques

Throughout this project, several key techniques and concepts from the EEE549 course were instrumental in understanding and applying the models. The foundational theories behind Logistic Regression and SVM provided a clear perspective on linear models and their limitations, prompting the exploration of regularization and kernel methods. The principles of dimensionality reduction, central to our understanding of PCA, were crucial in preprocessing stages, particularly for handling high-dimensional data effectively.

5.2 Connections to Deep Learning

In the realm of deep learning, the course's emphasis on learning low-dimensional representations was directly applicable to our implementation of Convolutional Neural Networks (CNNs) and Feedforward Neural Networks (FNNs). The concepts of feature extraction and transformation in CNNs, as well as the significance of the last linear/logistic layer in these networks, were understood and applied, drawing parallels to simpler models studied in the course. This connection highlighted the importance of understanding basic machine learning principles, even when dealing with more complex models like deep neural networks.

5.3 Concluding Remarks

The comprehensive exploration of various machine learning models in this project, underpinned by the theoretical knowledge acquired in EEE549, not only fortified our understanding of these models but also provided practical insights into their applications and limitations. The course's balanced focus on both theoretical and practical aspects of machine learning was invaluable in guiding our approach to model selection, implementation, and evaluation, proving the relevance and applicability of the concepts learned in real-world scenarios.

6. Conclusions

6.1 Evaluation of Datasets

UCI Adult Dataset: *Pros:* Rich in real-world socio-economic attributes, providing a diverse range of features for classification tasks. *Cons:* Challenges include class imbalance and missing values, requiring careful preprocessing.

Wisconsin Breast Cancer Dataset: *Pros:* Medical relevance with clear binary classification objectives. *Cons:* High dimensionality relative to sample size poses challenges in feature selection and model overfitting.

Fashion MNIST Dataset: *Pros:* Offers a standard benchmark for image classification and is suitable for complex pattern recognition tasks. *Cons:* Grayscale images limit the complexity of patterns, potentially impacting the efficacy of advanced models.

6.2 Algorithm Performance and Challenges

Logistic Regression: Efficient for linearly separable data but struggles with complex, non-linear datasets.

SVM: Excels in high-dimensional spaces, especially with kernel tricks, but can be computationally intensive.

CNN: Outperforms in image recognition tasks but requires significant computational resources and data.

FNN: Versatile for various data types but prone to overfitting without proper regularization.

6.3 Comparative Analysis and Best Performing Algorithm

UCI Adult Dataset: FNN showed the most promising results, balancing accuracy and computational efficiency. *Wisconsin Breast Cancer Dataset:* Logistic Regression without PCA and with Regularization demonstrated superior performance, effectively handling the high-dimensional nature of the data. *Fashion MNIST Dataset:* KNN stood out in terms of accuracy and pattern recognition capabilities in image data.

6.4 Final Remarks

This comparative analysis underscores the importance of selecting appropriate models based on dataset characteristics and desired outcomes. Each algorithm has its strengths and weaknesses, and the best choice depends on the specific requirements of the dataset and task at hand. This project not only reinforced the theoretical knowledge gained in the course but also provided practical insights into the nuances of machine learning model selection and application.

6.4.1 Fucntion for hyper tunning (gridsearch)

The variable grid_search_no_reg was employed to perform hyperparameter tuning on a logistic regression model (lr_no_reg) without regularization. It systematically explored various hyperparameter combinations defined in param_grid_no_reg, leveraging 5-fold cross-validation for robustness and parallel processing for efficiency. This process aimed to fine-tune the logistic regression model for improved predictive performance.

Appendix

	With PCA		Without PCA	
	Regularization	No Regularization	Regularization	No Regularization
Precision 0	0.88	0.88	0.88	0.87
Precision 1	0.74	0.74	0.74	0.73
Recall 0	0.93	0.93	0.93	0.94
Recall 1	0.59	0.59	0.59	0.57
F1 0	0.90	0.90	0.90	0.90
F1 1	0.65	0.65	0.66	0.64
Accuracy	0.85	0.85	0.85	0.85

Table 1: UCI Adult Dataset (SVM).

	With PCA		Without PCA	
	Regularization	No Regularization	Regularization	No Regularization
Precision 0	0.87	0.88	0.88	0.88
Precision 1	0.74	0.73	0.73	0.73
Recall 0	0.93	0.93	0.93	0.93
Recall 1	0.58	0.59	0.60	0.60
F1 0	0.90	0.90	0.90	0.90
F1 1	0.65	0.65	0.66	0.66
Accuracy	0.85	0.85	0.85	0.85

Table 2: UCI Adult Dataset (Logistic Regression).

	With PCA		Without PCA	
	Regularization	No Regularization	Regularization	No Regularization
Precision 0	0.99	0.99	0.99	0.99
Precision 1	0.98	0.91	1.00	1.00
Recall 0	0.99	0.94	1.00	1.00
Recall 1	0.58	0.98	0.98	0.98
F1 0	0.99	0.96	0.99	0.99
F1 1	0.98	0.94	0.99	0.99
Accuracy	0.98	0.96	0.99	0.99

Table 3: Wisconsin Breast Cancer (Logistic Regression).

	With PCA		Without PCA	
	Regularization	No Regularization	Regularization	No Regularization
Precision 0	0.88	0.97	0.88	0.97
Precision 1	0.89	1.00	0.89	1.00
Recall 0	0.94	1.00	0.94	1.00
Recall 1	0.79	0.95	0.79	0.95
F1 0	0.91	0.99	0.91	0.99
F1 1	0.84	0.98	0.84	0.98
Accuracy	0.89	0.98	0.89	0.98

Table 4: Wisconsin Breast Cancer (SVM).

7. Refrences

References

[1] D. W. Hosmer Jr., S. Lemeshow, and R. X. Sturdivant, *Applied Logistic Regression*, 3rd ed. Wiley, 2013.

[2] C. Cortes and V. Vapnik, *Support-vector networks*, Machine Learning, vol. 20, no. 3, pp. 273–297, 1995.

[3] I. T. Jolliffe, *Principal Component Analysis*, Springer Series in Statistics, 2002.

[4] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*, 2nd ed. Springer, 2009.

[5] Y. LeCun, Y. Bengio, and G. Hinton, *Deep learning*, Nature, vol. 521, pp. 436–444, 2015.

[6] A. Krizhevsky, I. Sutskever, and G. E. Hinton, *Imagenet classification with deep convolutional neural networks*, in Advances in Neural Information Processing Systems, 2012, pp. 1097–1105.