Transactions on Computational and Scientific Methods | Vo. 4, No. 12, 2024
ISSN: 2998-8780
https://pspress.org/index.php/tcsm
Pinnacle Science Press

Performance Evaluation and Insights into Supervised Learning Models for Single-Cell RNA-Seq Data Classification

Dorian Zhang[1], Mengting Jiao[2]
University of Oregon, Eugene, USA[1], University of Oregon, Eugene, USA[2]
dorian.zhang@uoregon.edu[1], mengtingj1313@gmail.com[2]

Abstract:

Single-cell RNA-sequencing (scRNA-seq) technology enables precise measurement of gene expression at the single-cell level, offering insights into cell subpopulations that bulk RNA sequencing cannot provide. However, effective classification of scRNA-seq data remains a challenge due to its high-dimensional, batch-variable, and complex nature. In this study, we empirically evaluate the performance of four supervised learning models—decision trees (DT), random forests (RF), boosting, and logistic regression (LR)—on scRNA-seq data. While decision tree-based methods have traditionally shown strong performance in gene expression analysis, our results reveal that logistic regression outperformed the other models in terms of accuracy. This suggests that LR provides a robust and interpretable solution for cell-type classification in scRNA-seq data. Despite its effectiveness, the model's performance is limited by the available training data and diversity of cell types. Future research should address these limitations through expanded datasets, further empirical evaluations, and integration of advanced ensemble techniques for improved classification performance.

Keywords:

Cell type classification; Machine learning; Computational biology; Supervised learning; Comparison.

1. Introduction

Single-cell RNA-sequencing technology have been rapidly developed in recent years, as a technique which can measure the transcriptome and gene expression level of individual cell, scRNA-seq can reveal many potential properties of cell subpopulations which could not be accomplished in bulk RNA sequencing [1]. From the count of publications in PubMed (Figure 1), the publication of the scRNA research is increasing dramatically, indicating the remarkable attention worldwide.

The focus of recent work is on the cell characterization and differentiation within each population being compared. Up to now, the work primarily depended on unsupervised methods or known markers. Known markers, in biological cases, are the specific genes which would be highly expressed in certain types of cells [2]. While the application of markers is useful, it may not be available for several cell types [3]. Although unsupervised methods are useful to solve the analysis of unlabeled dataset, some experts have shown that supervised methods can be much easier to interpret and much more accurate [4]. However, the lack of the empirical evaluation among multiple supervised models applied in single-cell RNA sequencing data will make the supervised learning methods less credible in the biology.

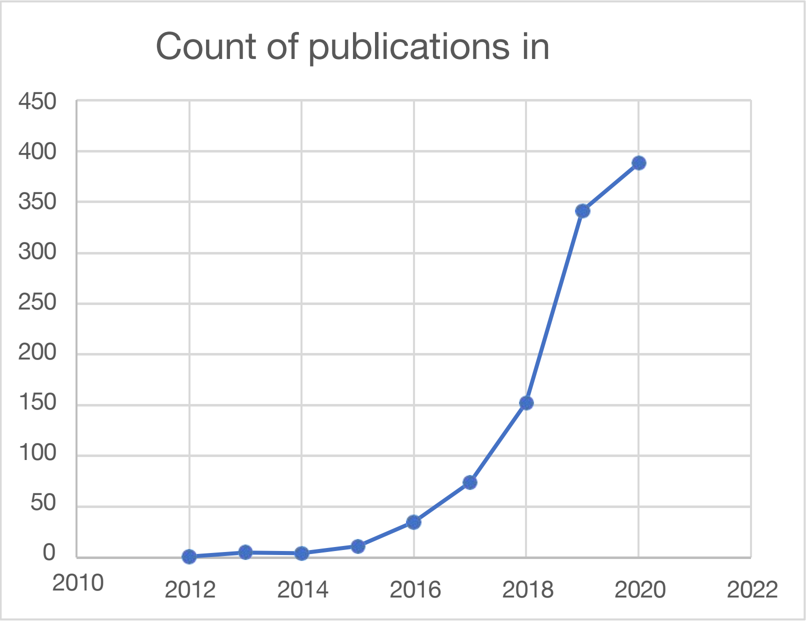

Figure 1. The count of publications about the topic Single-cell RNA-Seq in PubMed

Basically, the analysis of scRNA-seq data is a supervised, high dimensional, multiple classification problems. For the biological gene data, expression level of a single cell type under different experimental situation (termed as batch effect) may be highly variable which is either because of the difference from the sequencing platform or the variance induced by biological dynamic [5]. Below we will show various methods trying to overcome the problems, try to restore the real situation as much as possible, and solve the overfitting problem. Inspired by a paper published in 2006 [6], which is about the empirical comparison based on 11 binary classification problems, decision trees or the extensions of decision trees are assumed to be the most effective in the prediction of gene expression cell types. However, in terms of the final accuracy score, the optimal result of our project is not from DT, but from logistic regression.

2. Related Work

Supervised learning models have been extensively explored in various domains for classification tasks, which is directly relevant to the challenges posed by scRNA-seq data classification. Shen et al. [7] demonstrated the application of semi-supervised methods to enhance image classification under limited labeled data scenarios, which aligns with the challenges of limited training data in scRNA-seq. Xiao [8] emphasized the robustness of self-supervised learning for few-shot classification, offering insights into handling diverse cell types in limited data contexts. Dimensionality reduction and feature extraction, critical for high-dimensional datasets like scRNA-seq, were effectively addressed by Liang et al. [9] through an automated data mining framework using autoencoders. Feng et al. [10] further highlighted the potential of GAN-based approaches to enhance feature extraction in few-shot learning, which can be adapted to reduce batch effects and noise in scRNA-seq data. Optimization strategies for improving learning efficiency were explored by Qi et al. [11] and Chen et al. [12] in large-scale language models, providing valuable methodologies that could be leveraged to optimize supervised models for scRNA-seq classification. Additionally, Wang et al. [13] introduced adaptive optimization in spatiotemporal prediction, which has implications for enhancing the performance of supervised learning in scRNA-seq. In the context of time-series and sequential data analysis, Sun et al. [14] applied transformer models to complex time-series problems, while Sun et al. [15] integrated CNN-LSTM architectures for spatiotemporal prediction, both offering techniques that could be adapted for batch effect normalization in scRNA-seq data. Graph-based methods also hold promise for scRNA-seq analysis, as demonstrated by Du et al. [16], who used graph neural networks for relationship reasoning in knowledge graphs, a methodology potentially adaptable for gene expression interdependency studies. Furthermore, Yu et al. [17] tackled anomaly detection in anti-money laundering systems, providing transferable concepts for outlier detection in scRNA-seq data. Other innovative applications, such as Duan et al. [18] and Yao et al. [19], focused on deep learning-based UI generation and hierarchical graph neural networks for stock type prediction, respectively, demonstrating the versatility of supervised models across diverse fields and their potential adaptability to scRNA-seq classification.

3. Methods

3.1 Data source and pre-processing

Data from mouse gene dataset of interest which contains 20,499 genes with normalized gene expression levels - RPKM values for each cell and 24244 total samples were used (See Table 1) [3]. The data contains all_data.h5, together with separate train & test dataset. Classifier for cell types, in real experiment, would be used to predict the cells from new experiments. Therefore, to restore the actual situation, labels that are in the testing dataset can also be found in the training data. Meanwhile, compared with the train samples, test samples come from different experiments were selected.

Table 1. The dataset

	Train dataset	Test dataset
Samples	21389	2855
Unique cell types	46	21
Features (genes)	20499	20499

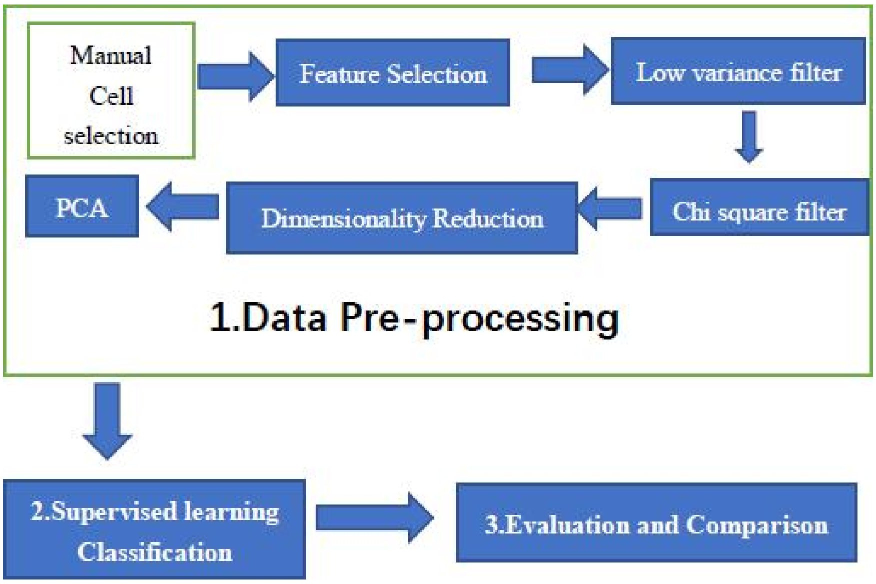

Figure 2. Workflow

3.1.1 Cell selection

This project select the cell types based on the test cell types manually as our real training labels to minimize the noise given by the train dataset redundancy.

3.1.2 Feature selection

Uninformative feature caused by irrelevance, correlation, and redundancy can impede the performance of classification model. Additionally, because of the technical variation which is mainly caused by the difference of sequencing platform and uninformative biological variation induced mainly by the experiment batch effect, low variance filter is applied based on scikit-learn package [20] to select highly variable genes based on the RPKM value (See Figure 2). Meanwhile, chi square function was applied to measure the relation between features and labels.

Table 2. Example of gene dataset for chi square computing

	Gene 1	Gene 2
Cell 1	RPKM value	RPKM value
Cell 2	RPKM value	RPKM value

Chi Square Formula:

$$\chi_e^2 = \sum \frac{(O_i - E_i)^2}{E_i}$$

In the Chi Square Formula, 'O' is the observed value, 'E' means the expected value and 'I' is the 'i'th position in the contingency table. The chi-squared statistic is a single number that tells how much difference exists between your observed rpkm values and the rpkm values you would expect if there were no relationship at all in the dataset. P-value was used here to explain the result of Chi square for different groups. The Select K-Best package was used here to select the top related features [21]. After the feature selection, highly variable genes and highly specific cell-type-related features would be left. Selection could facilitate downstream applications like DT-based classification and save the computational costs.

3.1.3 Principal component analysis (PCA)

After feature selection, it is still important to apply dimension reduction as too much noise still exists. The core concept of PCA is to map all features to K dimensions. Therefore, PCA for dimensionality reduction was applied and adjusted to the optimal dimension leading to best accuracy.

3.2 Supervised learning methods

3.2.1 Decision Tree

Decision tree is a basic model which was used as a controlled trial in the comparisons. The max-depth and pruning were not determined in the experiment. In our model, Gini impurity is used as the criterion.

3.2.2 Random Forest

In random forests, each DT in the model is built from a set of samples drawn with replacement from the training set, i.e., 632bootstrap, which means roughly 63% of the original data are selected. The input is the entire original training dataset. Cross validation is implemented to determine the best number for trees and the max-depth of the tree in the model. In this model, Gini impurity is used as the criterion.

3.2.3 Ada-Boosting

Ada-Boosting uses a set of week classifier, in our case, small DTs, to operate on repeatedly modified versions of the data. The predictions from all of them are then combined through a weighted majority vote to produce the final prediction. Therefore, there are two sets of weights: weights for DT and weights for data. Initially, those weights are all set to 1/N. We train the first weak classifier and focusing on the mistakenly classified cells by allocating new weight for each data point. Each subsequent weak learner is thereby forced to concentrate on the examples that are missed by the previous ones in the sequence. For a multi-class classification problem, the previously described two steps: learning and allocating new weight, will iterates until the training error reduce to zero.

3.2.4 Logistic Regression

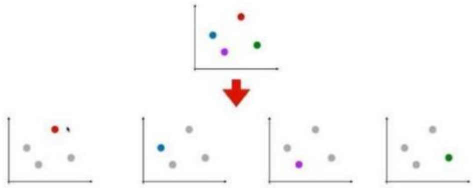

Figure 3 One versus rest [8]

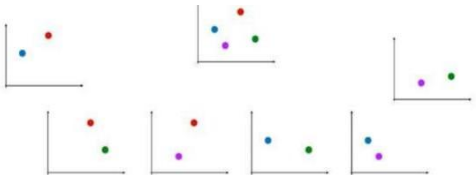

Figure 4. One versus one [8]

This model used One vs One (OvO) instead of One vs Rest (OvR) in the logistic regression model. One vs Rest (OvR) treat a multi-class(n) problem as n binary problem. Each binary problem the model select one set of data points as one class, and all the other point as the other class. Therefore, the algorithm takes O(nT), if the binary classification takes O(T).

3.3 Evaluation of classifications

Different standards are used to evaluate these models as they can judge these models from different aspects. It is possible to build a confusion matrix and calculated the accuracy score, recall score, precision score and f1_score based on it. Accuracy score is based on the whole data, and the other standards are based on each cell type. In addition, it normalized this confusion matrix to see the accuracy of specific cell types clearly.

However, in evaluation of biology classification models, the question of how similar two cell types are is quite important because a rigid (binary) distinction between cell types is not appropriate since “neuron”, “hippocampus”, and “brain” are all related cell types, and a model that groups these cell types together should not be penalized as much as a model that groups completely unrelated cell types together [3]. Therefore, the evaluation can be improved by adding some weights. A similarity matrix is downloaded [3]. In general, numbers are changed in the confusion matrix by timing them with the weight which equals to (1-similarity number), acquiring a new accuracy score by our weighed confusion matrix.

3.4 Software packages

Low variance filter, Chi square filter, PCA, Decision tree, Random forest, Ada-boosting, Logistic regression, and confusion matrix are all coded in Python based on scikit-learn package [21].

4. Results

4.1 Accuracy score of the four models

4.1.1 Decision Tree

In the Decision Tree Model, we both tried model without PCA or with PCA from 40 to 100. Finally, 34% is set as the baseline for comparison between decision tree-based models.

Table 3. Accuracy score of Decision Tree

Chi Square Filter	Lower Variance Filter	PCA	Accuracy score
None	15	100	0.212
None	15	50	0.194
None	15	45	0.262
None	15	40	0.273

None	None	None	0.349

Table 4. Accuracy score of Random Forest

Chi Square Filter	Lower Variance Filter	PCA	Accuracy score
None	15	200	0.321
None	15	100	0.300
None	15	50	0.429
None	15	45	0.430
None	15	40	0.451

4.1.2 Random Forest

In the Random Forest Model, the result is better, achieving 45% overall accuracy. The training data for each tree comes from bootstrap. The best result comes from 40-dimension 150 trees with maximum depth of 30.

The hyperparameters are determined using cross validation, for example, to determine the number of trees in the random forest, we perform 5- fold cross validation on 100 trees, 150 trees 200 trees and so on. The high cross validation mean accuracy comes from 150 trees, which is 93.2%.

4.1.3 Ada-Boosting

Table 5. Accuracy score of Ada-Boosting

Dimension after PCA	N_estimator	Max_depth	Accuracy on training set	Accuracy on test set
50	100	3	0.30	0.18
50	300	3	0.42	0.18
50	1000	3	0.47	0.21
50	100	10	0.89	0.26
50	300	10	0.94	0.32
50	800	10	0.95	0.33
40	300	10	0.94	0.38
40	400	10	0.95	0.39
40	850	7	0.85	0.37
40	1400	7	0.90	0.39

Some of the most representative parameter values are selected, shown in Table 5. From the last two rows in the table, it is obvious that even if the model almost doubles the max iteration number(n_estimator), the overall training error improve only slightly. It becomes incredibly time-consuming. By adjusting the Decision Tree's max depth, result is improved, but still, the highest success rate cannot even compete with random forest classifier's worst result. Therefore, the conclusion is that DT-based boosting algorithm is not suitable to be directly applied in a multi-class biology classification problem.

4.1.4 Logistic Regression

Table 6. Accuracy score of Logistic Regression

Solver	Multiclass	Max_iter	Accuracy on training set	Accuracy on test set
		Before Cell Selection		
Sag	Multiclass	100	0.78	0.5
Sag	Multiclass	200	0.81	0.54
Sag	Multiclass	500	0.84	0.54
Sag	Multiclass	1000	0.86	0.52
Sag	Multiclass	2000	0.88	0.49
		After Cell Selection		
Sag	Multiclass	100	0.88	0.55
Sag	Multiclass	200	0.9	0.57
Sag	Multiclass	500	0.9	0.56
Sag	Multiclass	1000	0.93	0.55
Sag	Multiclass	2000	0.93	0.51

Logistic Regression works best. It is the most time-efficient and has the best overall accuracy. As described in the method, the parameters chosen are for One vs One (OVO) multiclass classification, which is more time consuming than One vs Rest (OVR), but at the same time more accurate. If adding the cell selection, the result is even better, achieving 57% success rate.

4.2 Confusion matrix on random forest and logistic regression

Besides accuracy score, confusion matrix, precision, recall, and f1-score (Methods) are also used to evaluate our models of this project. The best two model are compared in terms of overall accuracy. In the figure 5 and figure 6, the deeper the color of each intersection grid is, the more cases there are that X label is predicted to be Y label. From the figure 5 and figure 6, it can be seen that there are more deep color grids in the diagonal in the Logistic Random than in the Random Forest, which means that more labels in the Logistic Regression were predicted correctly.

In fact, it is obvious that the number of cell-types each time the confusion matrix presents depends on the union of cells from both test data, which has 21 types of cells, and the predicted cell dataset. In that case, the confusion matrix, which is output from each model, or from the same model but at different times, will vary in the number of label-types. However, that does not affect our conclusion one thing. That is because the only correct case of classification is when a labeled cell is predicted to be itself, which is denoted by the diagonal grids, and those deep color grids in the diagonal only come from the 21 cell types to be predicted. Therefore, more labels to be predicted correctly means more labels to be predicted correctly within those 21 types of cells.

Meanwhile, there are more deep color grids in the lower left corner of Random Forest, which means that Random Forest assigns more labels to the wrong kinds.

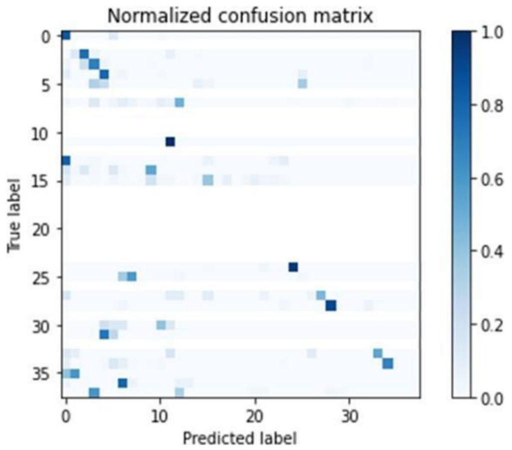

Figure 5. Random Forest

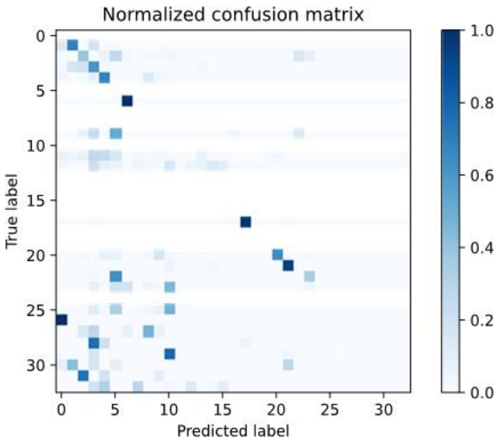

Figure 6. Logistic Regression

4.3 Precision, recall, f1-score on random forest, and logistic regression

Among all the test cells, the precision score, recall score and f1-score for them are tested. The number of each type of cells in the test data are also provided, so that a more intuitive insight into the relationship between number of cells and its prediction result can be provided. It is already known that f1-score is the harmonic average value of precision score and recall score (Methods).

Table 7. Random Forest

	precision	recall	f1-score	support
CL:0000137 osteocyte	0.98	0.97	0.98	108
CL:0000235 macrophage	0.85	0.93	0.89	42
UBERON:0000966 retina	0.99	0.65	0.79	250
CL:0002319 neural cell	0.58	1	0.74	81
UBERON:0001003 skin epidermis	0.83	0.63	0.72	678
CL:0002321 embryonic cell	0.6	0.69	0.64	173
CL:0002322 embryonic stem cell	0.5	0.6	0.55	358
UBERON:0000044 dorsal root ganglion	0.3	0.37	0.33	123

CL:0000037 hematopoietic stem cell	0.59	0.06	0.11	162
UBERON:0001851 cortex	0.16	0.04	0.07	266
UBERON:0001264 pancreas	1	0.01	0.02	162
CL:0000353 blastoderm cell	0	0	0	10
CL:0000540 neuron	0	0	0	133
CL:0000746 cardiac muscle cell	0	0	0	11
UBERON:0000115 lung epithelium	0	0	0	78
UBERON:0000922 embryo	0	0	0	60
UBERON:0000955 brain	0	0	0	38
UBERON:0001898 hypothalamus	0	0	0	29
UBERON:0001954 Ammon's horn	0	0	0	15
UBERON:0002048 lung	0	0	0	58
UBERON:0002107 liver	0	0	0	20
accuracy			0.43	2855

Table 8. Logistic Regression

	precision	recall	f1-score	support
CL:0000137 osteocyte	1	0.98	0.99	108
CL:0000235 macrophage	0.89	0.98	0.93	42
UBERON:0001003 skin epidermis	0.79	0.89	0.83	678
UBERON:0000955 brain	0.96	0.71	0.82	38
UBERON:0000966 retina	0.96	0.69	0.8	250
UBERON:0002107 liver	0.93	0.7	0.8	20
CL:0000037 hematopoietic stem cell	0.66	0.68	0.67	162
CL:0002321 embryonic cell	0.68	0.64	0.66	173
CL:0002319 neural cell	0.41	1	0.58	81
UBERON:0000044 dorsal root ganglion	0.47	0.76	0.58	123
CL:0002322 embryonic stem cell	0.43	0.62	0.51	358
UBERON:0001851 cortex	0.22	0.15	0.18	266
CL:0000540 neuron	0.21	0.06	0.09	133
UBERON:0001264 pancreas	1	0.01	0.02	162
accuracy			0.57	2855

	precision	recall	f1-score	support
CL:0000137 osteocyte	1	0.96	0.98	108
CL:0000235 macrophage	0.9	0.9	0.9	42
UBERON:0001003 skin epidermis	0.79	0.87	0.83	678
CL:0002319 neural cell	0.68	1	0.81	81
UBERON:0000966 retina	0.99	0.67	0.8	250
UBERON:0000044 dorsal root ganglion	0.74	0.79	0.76	123
UBERON:0000955 brain	1	0.53	0.69	38
CL:0002321 embryonic cell	0.6	0.76	0.67	173
CL:0002322 embryonic stem cell	0.62	0.67	0.64	358
UBERON:0002107 liver	1	0.45	0.62	20
CL:0000037 hematopoietic stem cell	0.86	0.38	0.53	162
UBERON:0001851 cortex	0.13	0.05	0.07	266
CL:0000540 neuron	0.14	0.02	0.04	133
CL:0000353 blastoderm cell	0	0	0	10
CL:0000746 cardiac muscle cell	0	0	0	11
UBERON:0000115 lung epithelium	0	0	0	78
UBERON:0000922 embryo	0	0	0	60
UBERON:0001264 pancreas	0	0	0	162
UBERON:0001898 hypothalamus	0	0	0	29
UBERON:0001954 Ammon's horn	0	0	0	15

UBERON:0002048 lung	0	0	0	58
accuracy			0.54	2855

The primary concern is to compare Random Forest and Logistic Regression. As can be seen from Table 7 and Table 8, all values that are greater than 0.9 are bolded. From the f1-score column, osteocyte cells were always classified the most accurately, and macrophage cells the second. Also, f1-score of Logistic Regression is usually larger than Random Forest, which means that Logistic Regression has a better classification ability. In conclusion, Logistic regression behaves better than random forest.

4.4 How pre-processing impact our results

Both feature selection and dimensionality reduction tools showed great impact on our models. Different combinations of these methods are compared to pre-process our data. In the Decision Tree Model, it is found that Decision Tree without PCA (Principal Component Analysis) performed much better than with PCA (Principal Component Analysis). Presumably, the model was relatively simple, therefore, it will be unable to effectively differentiate between cells when there were relatively fewer features for it to learn. In the Ada-Boosting Model, PCA (Principal Component Analysis) seemed to have very little impact on the accuracy score. However, feature selection had a very large impact on the program efficiency. Considering its low classification accuracy results, the best two classifiers in the experiment- Random Forest and Logistic Regression are compared. Compared with the Logistic Regression Model, which though has the best accuracy score, it is found that pre-processing had a larger impact on our Random Forest Model, especially when combined, which improved the accuracy score of Random Forest by 10% on average, Logistic Regression by 5% (See Table 9). From above, it is obviously that when these models were getting relatively more complicated compared with Decision Tree, the pre-processing could have a larger impact on the accuracy of the models.

Table 9. Pre-processing impact on the accuracy score

Feature selection types	Accuracy increase for RF	Accuracy increase for LR
Chi square filter (PCA 40)	2%	2%
Low variance filter (PCA 40)	5%	3%
Two-combined (PCA 40)	10%	5%

4.5 How similarity between cells revises our results

The similarity coefficients are used to revise the accuracy of the two best models. For the Random Forest Classifier, the accuracy score before applying similarity coefficients was 0.435, and after was 0.461, which increased by 2.6%. For Logistic Regression Model (the best model), the accuracy score before was 0.571, and after was 0.598, which increased by 2.7%. (See Table 10)

In conclusion, the accuracy score are revised according to Similarity Coefficients.

Table 10. Revision of the accuracy score using similarity coefficients between cells

Classifier	Accuracy before	Accuracy after	Accuracy increase
Random Forest	0.435	0.461	2.6%
Logistic Regression	0.571	0.598	2.7%

5. Discussion

The purposes of this paper are comparing the performance of different DT-based models, as well as trying different ensemble methods to address this multi-class classification problem. To improve the accuracy of the high dimension, multiclass, batch-related classification problem, we used the popular models, mostly ensemble methods, to implement on our model and finally the overall accuracy improved from baseline Decision Tree 34.9% to Logistic Regression 57.1% before revision, and 59.8%

after revision.

In all the models that have been tried, the Logistic Regression Model performed best, and the Random forest one the second. The Ada- Boosting model seemed to be the most time consuming and having the lowest accuracy. One possible reason why the Logistic Regression outperformed all the other Decision Tree-based models is that Logistic Regression uses a modified version of "divide and conquer". It breaks down multiclass problem into several binary classification problems, and then combines the result. This approach makes the algorithm more time efficient, and the voting technique also improves the Logistic Regression Model.

At the pre-processing stage, cell type selection was applied by extracting some of the cells which are selected based on the test cell dataset from train dataset, in order to reduce the computational difficulty and increase the accuracy of our supervised models. For the sake of the simplicity and manually filtering noise, the accuracy of our classifiers was raised by 10% on average. However, it is obvious that the use of manually selection to reduce the train data set will not fit the real-world application. Therefore, it is much more confident to classify the cell type in the situation that the researchers have the basic prediction and range of the potential cells. We also used two kinds of Feature Selection tools called Chi Square Filter and lower Variance Filter, and PCA (Methods), which all showed great impact on the models, not only increasing the accuracy score, but also improving the efficiency of models to a large extent, especially for large-scale scRNA-seq datasets whose computational time is long and memory-consuming. Furthermore, when testing the performances of each model with each PCA dimensionality, there was usually a 3% fluctuation, so in the project the average of the three tests as the final accuracy score.

With regard to Decision Tree Model, as can be seen in table3, it is found that Decision Tree without dimensionality reduction performed much better. This is an interesting finding. One possible reason is that the PCA method does not use labels, so the purpose of PCA is for reconstruction rather than classification.

6. Future work

For the best model in our research, the accuracy raised by the feature selection is not dramatic, therefore, other feature selection method such as the selection based on highly expressed gene should be tried in the future. Also, we can also take the advantage of the idea of divide-and-conquer to design an algorithm combined with Ada-Boost and One vs Rest (OVR), instead of just using the existing package, converting the multiclass classification problem into several binary classification problem, to check if the accuracy can be further improved. In addition, we can take into consideration to build a working software or a website server, therefore, it is necessary to consider designing adaptive hyperparameters. For example, the number of trees in random forest can be adaptively updated to the optimal value after new data is added to the training set. Furthermore, we need to generalize our model by discarding cell selection, because in real world, we usually cannot guarantee to extract labels every time we have a new test set. Since our project showed that cell selection did work, in the future, instead of cell selection, we can try some other methods to detect irrelevant cells in the training data to improve the result.

7. Conclusion

Analysis of scRNA data is essential for cell classification studies. Supervised learning models like decision trees, random forests, boosting and logistic regression all performed well in cell classification, yet logistic regression behaved the best in terms of the accuracy among four evaluated models. Our evaluation has potential to better understand the use of supervised classification models in scRNA data. However, because of some limitations such as the shortage of cell types in the training data, the logistic regression can only be predicted to be one of the most effective models for scRNA data. Apart from the detail optimization to the workflow, future work could do more empirical evaluation and try more complex combination with basic classifiers on different scRNA data.

References

[1] ALQUICIRA-HERNANDEZ, J., SATHE, A., JI, H. P., NGUYEN, Q. & POWELL, J. E. 2019. scPred:

accurate supervised method for cell-type classification from single-cell RNA-seq data. Genome Biology, 20, 264.

[2] EL AMRANI, K., ALANIS-LOBATO, G., MAH, N., KURTZ, A. & ANDRADE-NAVARRO, M. A. 2019. Detection of condition-specific marker genes from RNA-seq data with MGFR. PeerJ, 7, e6970-e6970.

[3] ALAVI, A., RUFFALO, M., PARVANGADA, A., HUANG, Z. & BAR-JOSEPH, Z. 2018. A web server for comparative analysis of single-cell RNA-seq data. Nature Communications, 9, 4768.

[4] LIN, C., JAIN, S., KIM, H. & BAR-JOSEPH, Z. 2017. Using neural networks for reducing the dimensions of single-cell RNA-Seq data. Nucleic Acids Research, 45, e156-e156.

[5] TRAN, H. T. N., ANG, K. S., CHEVRIER, M., ZHANG, X., LEE, N. Y. S., GOH, M. & CHEN, J. 2020. A benchmark of batch-effect correction methods for single-cell RNA sequencing data. Genome Biology, 21, 12.

[6] CARUANA, R. & NICULESCU-MIZIL, A. 2006. An empirical comparison of supervised learning algorithms. Proceedings of the 23rd international conference on Machine learning. Pittsburgh, Pennsylvania, USA: Association for Computing Machinery.

[7] A. Shen, M. Dai, J. Hu, Y. Liang, S. Wang, and J. Du, "Leveraging Semi-Supervised Learning to Enhance Data Mining for Image Classification under Limited Labeled Data," arXiv preprint arXiv:2411.18622, 2024.

[8] Y. Xiao, "Self-Supervised Learning in Deep Networks: A Pathway to Robust Few-Shot Classification," arXiv preprint arXiv:2411.12151, 2024.

[9] Y. Liang, X. Li, X. Huang, Z. Zhang, and Y. Yao, "An Automated Data Mining Framework Using Autoencoders for Feature Extraction and Dimensionality Reduction," arXiv preprint arXiv:2412.02211, 2024.

[10] Y. Feng, A. Shen, J. Hu, Y. Liang, S. Wang, and J. Du, "Enhancing Few-Shot Learning with Integrated Data and GAN Model Approaches," arXiv preprint arXiv:2411.16567, 2024.

[11] Z. Qi, J. Chen, S. Wang, B. Liu, H. Zheng, and C. Wang, "Optimizing Multi-Task Learning for Enhanced Performance in Large Language Models," arXiv preprint arXiv:2412.06249, 2024.

[12] J. Chen, B. Liu, X. Liao, J. Gao, H. Zheng, and Y. Li, "Adaptive Optimization for Enhanced Efficiency in Large-Scale Language Model Training," arXiv preprint arXiv:2412.04718, 2024.

[13] X. Wang, X. Li, L. Wang, T. Ruan, and P. Li, "Adaptive Cache Management for Complex Storage Systems Using CNN-LSTM-Based Spatiotemporal Prediction," arXiv preprint arXiv:2411.12161, 2024.

[14] W. Sun, Z. Xu, W. Zhang, K. Ma, Y. Wu, and M. Sun, "Advanced Risk Prediction and Stability Assessment of Banks Using Time Series Transformer Models," arXiv preprint arXiv:2412.03606, 2024.

[15] Q. Sun, T. Zhang, S. Gao, L. Yang, and F. Shao, "Optimizing Gesture Recognition for Seamless UI Interaction Using Convolutional Neural Networks," arXiv preprint arXiv:2411.15598, 2024.

[16] J. Du, G. Liu, J. Gao, X. Liao, J. Hu, and L. Wu, "Graph Neural Network-Based Entity Extraction and Relationship Reasoning in Complex Knowledge Graphs," arXiv preprint arXiv:2411.15195, 2024.

[17] Q. Yu, Z. Xu, and Z. Ke, "Deep Learning for Cross-Border Transaction Anomaly Detection in Anti-Money Laundering Systems," arXiv preprint arXiv:2412.07027, 2024.

[18] S. Duan, R. Zhang, M. Chen, Z. Wang, and S. Wang, "Efficient and Aesthetic UI Design with a Deep Learning-Based Interface Generation Tree Algorithm," arXiv preprint arXiv:2410.17586, 2024.

[19] J. Yao, Y. Dong, J. Wang, B. Wang, H. Zheng, and H. Qin, "Stock Type Prediction Model Based on Hierarchical Graph Neural Network," arXiv preprint arXiv:2412.06862, 2024.

[20] PEDREGOSA, F., VAROQUAUX, G., GRAMFORT, A., MICHEL, V., THIRION, B., GRISEL, O., BLONDEL, M., PRETTENHOFER, P., WEISS, R., DUBOURG, V., VANDERPLAS, J., PASSOS, A., COURNAPEAU, D., BRUCHER, M., PERROT, M. & DUCHESNAY, É. 2011. Scikit-learn: Machine Learning in Python. J. Mach. Learn. Res., 12, 2825–2830.

[21] Machine Learning: logistic regression (OvR and OvO) (2018) https:// blog.csdn.net/ ab1213456/ article/ details/ 102214443.