COMP 9417

Machine Learning and Data Mining

Group Project:

Multitask Machine Learning (MML)

Internal Project

Lecturer: Dr G Mohammadi

April 2024

Group Name: Code Runner

Zishan Fu	z5490252
Weizhe Yang	z5444789
Will Ren	z5429870
Xieling Wang	z5522088

1. Introduction

We are the machine learning team at Predictive Solutions Inc. Our recent project focused on applying machine learning to a clinical trial dataset. The primary objective is to leverage this technology to aid medical researchers in predicting medical conditions based on the data from the trial.

The dataset provided by a medical institution for this project contains 1,000 observations and 111 features, covering a variety of data types including binary, categorical, and continuous values. The diversity of data types, anonymized features and limited data instances pose unique challenges as we need to do more work for data analysis and preprocessing. Also, there are multiple target variables to predict as opposed to the usual case in which we have a single target variable to predict.

After carefully analyzing and preprocessing the dataset, we explored both the single target prediction method and the multi-task learning approach.

2. Exploratory Data Analysis and Literature Review

2.1 Raw Data Visualization

We visualized the 111 features from the original dataset. Figure 2.1 below is an example and shows the distribution of some features. Also, we can see how many nan values or 0 value in each column clearly. For the 11 target variables, all are binary, with one representing approximately 25%.

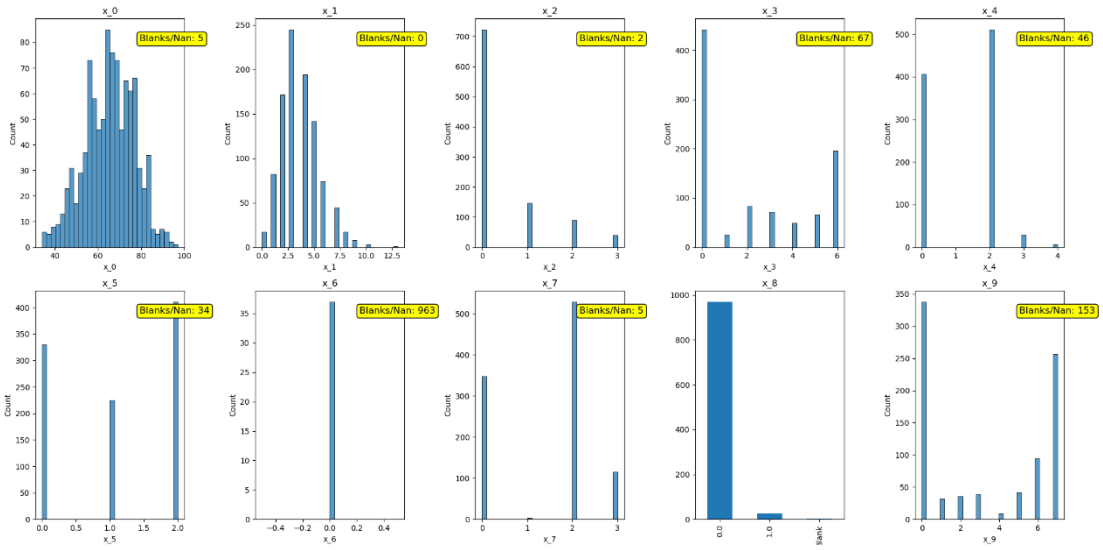


Figure 2.1 Example of Distribution of Features Data Preprocessing

2.2 Data Preprocessing

2.2.1 Drop Features with too Many Missing Values.

We can tell that some features have enormous amounts of nan/blank values. We defined that if a feature's 0 or nan values accounts for more than 998 of 1000. This feature will be removed from the dataset as it brings no information.

Features dropped: ['x_6', 'x_12', 'x_16', 'x_17', 'x_18', 'x_19', 'x_21', 'x_22', 'x_23', 'x_24', 'x_31', 'x_40', 'x_47', 'x_51', 'x_55', 'x_60', 'x_61', 'x_62', 'x_63', 'x_65', 'x_66', 'x_69', 'x_77', 'x_80', 'x_87']

2.2.2 Fill Missing Values

For features with missing values, we use KNN interpolation method. The KNN interpolation method can find the data in the data set that is most like the observed value of the missing value to fill in the missing data (Daberdaku et al., 2020). KNN has the advantages of high accuracy, insensitivity to outliers, and no data input assumptions, which makes it very suitable for the given dataset to use KNN interpolation for missing value filling. However, at the same time, KNN interpolation also has the problems of high computational complexity and high space complexity, but the data in this dataset is not too large, so KNN interpolation can be applied perfectly. Figure 2.2.2.1 below shows the distribution of the same part of different features in Figure 1, after deleting columns with too many missing values and filling missing data. Now dimension of X dataset is 1000*86.

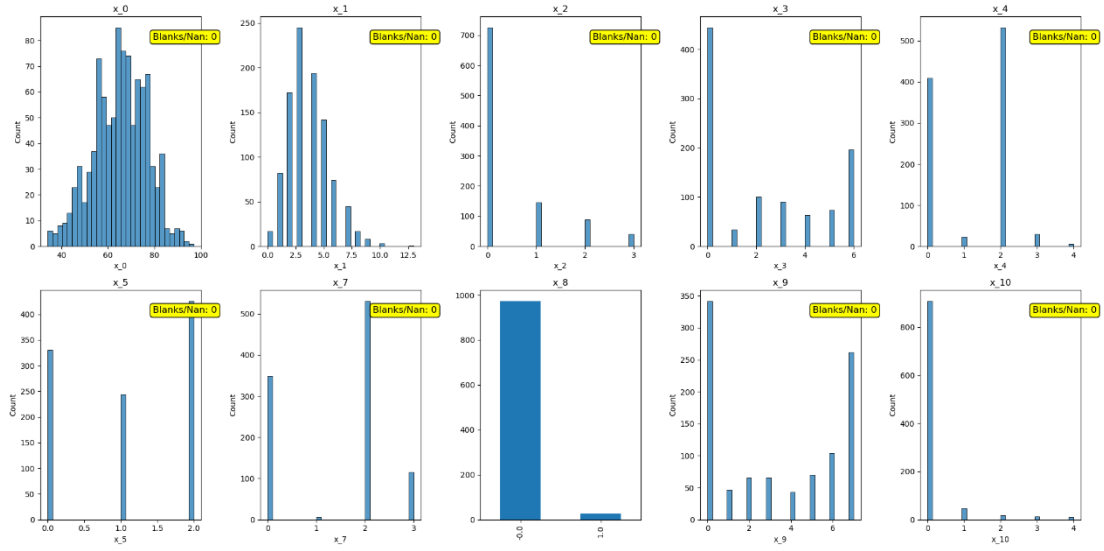


Figure 2.2.2.1 Distribution of identical parts of different features

2.3 Features Importance

In order to find which features that contribute most to the predictions of a model, we used RandomForest Regressor to calculate the importance scores for these features. Removing low importance features can reduce overfitting and improve model performance. After comparing, we chose the importance score of 0.01 as threshold.

2.4 Data Correlation Analysis

The correlation heatmap presented below(figure2.4.1 and figure 2.4.2) illustrates the relationships among the 34 features remaining post-preprocessing. It is evident that a significant number of features do not exhibit strong correlations. Similarly, the subsequent heatmap displays the correlation among the target labels, indicating a lack of strong correlation among them as well.

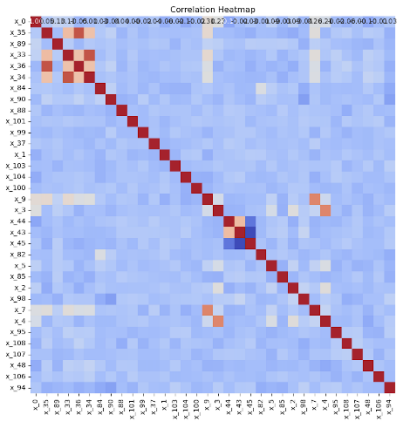


Figure 2.4.1 Correlation Heatmap

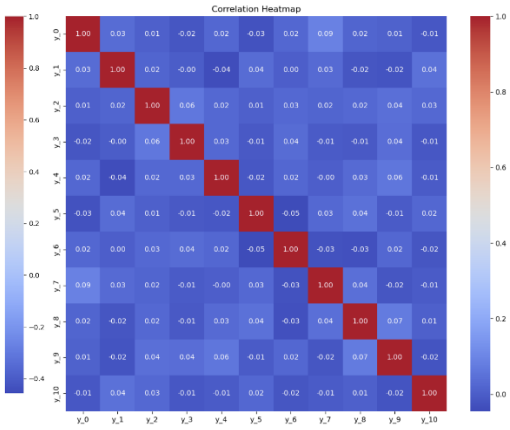


Figure 2.4.2 Correlation Heatmap

2.5 Literature Review

Definition of multi-task learning: Given m learning tasks $\{T_i\}_{i=1}^{m}$ where all the tasks or a subset of them are related but not identical, multi-task learning aims to help improve the learning of a model for T_i by using the knowledge contained in the m tasks. Types of MML include multi-task supervised learning, multi-task unsupervised learning, multi-task semi-supervised learning, multi-task active learning, multi-task reinforcement learning and multi-task online learning. MTL has been effectively applied in various domains including natural language processing, computer vision, and speech recognition.

Multi-task learning has gained significant attention in the field of healthcare prediction. (Qi et al. ,2010) introduced a semi-supervised multi-task framework for predicting protein-protein interactions, utilizing both labeled and partially labeled reference sets. (Zeng et al. ,2015) proposed a deep learning model, MIMT-CNN, for multi-instance multi-task learning, where multiple images are taken as inputs for solving tasks independently. (Vandenhende et al. ,2020) emphasized the importance of

considering task interactions at multiple scales in multi-task learning setups. (Aglietti et al., 2020) introduced a multi-task causal Gaussian process model, DAG-GP, for jointly learning causal effects of interventions on different subsets of variables in healthcare. Overall, the literature highlights the benefits of multi-task learning in healthcare prediction, showcasing advancements in model architectures, causal learning, risk factor analysis, fairness assessment, and active learning strategies. The proposed frameworks and models aim to improve prediction accuracy, generalization, and efficiency in healthcare applications.

3. Methodology

To get the best results, we have chosen two methods, and finally by comparing the metrics scores, then decide the optimal model. Two modes for model training: a) Single Task Learning Models and combine each labels together; b) multi-task learning with neural network.

3.1 Single Task Learning Models - Support Vector Machine (SVC), Adaboost, Random Forests

3.1.1 Justification of the methods chosen

Medical diagnostics is one of the most critical parts of healthcare, involving identifying diseases, predicting their progression, and formulating treatments. Traditional diagnostic methods rely on doctors' experience and expertise, which have some limitations, such as time consumption, labor costs, and diagnostic inaccuracies. As data grows, the field of medical diagnosis requires more efficient, accurate and reliable methods to process large amounts of medical data.

In the model selection process, we selected support vector machine (SVC), Adaboost and random forests as candidate models for this project. The three machine learning models chosen by our group are widely used in the medical field, and there are many experiments and studies that have proved the effectiveness of the application of these models in the medical field. Meanwhile, the generalization of these four models is relatively good, and they can accurately predict the unseen data. In terms of interpretability, their prediction results can be understood and interpreted intuitively, which is convenient for doctors to make further judgments. By analyzing the prediction results of the four models together, we can also arrive at more accurate results than those predicted by a single model.

a) Support Vector Machine (SVC)

Support vector machine is a binary classification machine learning algorithm mainly used to solve classification and regression problems. Its basic idea is to map data

points to the nearest category label by finding the maximum separation (Euclidean distance) between two categories in the data space.

Support vector machines can handle high-dimensional data and non-linear relationships and have good classification capabilities for complex data sets. Through the search for the maximum interval hyperplane, it can effectively handle classification boundary problems and has strong generalization ability.

b) Adaboost

AdaBoost is one of the most common machine learning algorithms today. It can be used for regression or classification algorithms. Compared with other machine learning algorithms, it overcomes the problem of overfitting and is usually sensitive to outliers and noisy data.

It determines the weight of each sample based on whether the classification of each sample in each training set is correct and the accuracy of the last overall classification. All the weighted samples are again sent to the next basic classifier for training. This process is repeated until a predetermined small enough error rate is reached or the pre-specified maximum number of iterations is reached. The classifiers obtained each time are fused to form the final decision classifier.

The Adaboost model is widely used in disease prediction and medical cost prediction.

Adaboost has two main hyperparameters: n_estimators and learning_rate. For n_estimators, it is usually set to an integer between 50-500, and the optimal number of basic learners is selected within the range. Increasing the number of learners can improve model performance, but also increases learning costs. For learning_rate, it is usually set between 0.01-0.1.

c) Random Forests

Random Forest, as the name suggests, Random means random sampling; Forest means that there is more than one tree, but a forest composed of a group of decision trees. Together, they use random sampling to train a group of decision trees to complete the classification task.

RF uses two random draws; one is a random draw of training samples; the other is a random draw of features. This is mainly to solve the problem of limited sample size (IBM, n.d.).

The core of RF is the application of the idea of changing from weak to strong. Since each decision tree is trained using only some variables and some samples, the individual classification accuracy may not be very high. But when a group of such

decision trees are combined to make judgments on the input data respectively, it can bring higher accuracy.

The random forest model is also widely used in the medical field, and it is often seen in the fields of disease prediction, medical imaging, and drug management (Mbonyinshuti et al., 2022).

The advantages of random forests are also obvious. It can handle relatively large data sets and has high parallelism and computational efficiency. By merging multiple decision trees, the risk of overfitting can be effectively reduced, and the generalization ability of the model can be improved. At the same time, it can capture the complex relationships between features and has good fitting ability for non-linear relationships that may exist in medical data.

There are two main hyperparameters of the decision forest, n_estimators and max_features. n_estimators usually range from 100-1000 integers, and max_features is usually tuned between 0.5 and 1. For n_estimators, the higher the number of decision trees, the stronger the performance of the model, but this will increase the learning cost, and for max_features a smaller proportion will increase the randomness of the model, which can reduce the risk of overfitting. Both can be tuned using grid search or cross-validation methods.

3.1.2 Further Preprocessing – Resample Minority Class for Training Dataset

As the positive class only accounts for about 25% of total labels and minority class is critical in medical diagnosis. We duplicated samples from the minority class to increase its size. By balancing the classes, models are less likely to overlook the minority class, potentially improving performance metrics such as accuracy, precision, recall, and F1-score for that class.

3.1.3 Hyper-parameter Tunning – Grid Search and Cross Validation

GridSearchCV was used to tune hyperparameters for these methods. It methodically goes through the combinations of parameters and uses cross-validation during this process. This will increase the likelihood of finding a more optimal set of parameters which can lead to better model performance. GridSearchCV can be computationally expensive for large datasets. Given the relatively small size of this clinical dataset, the utilization of GridSearchCV is deemed highly suitable in this context.

Classifier	Hyperparameters
RandomForest	n_estimators: [100, 200, 300]

	max_depth: [None, 10, 20]
	min_samples_split: [2, 5, 10]
SVC	C: [0.1, 1, 10]
	kernel: ['rbf']
AdaBoost	n_estimators: [50, 100, 200]
	learning_rate: [0.01, 0.1, 1]

3.2 Multi-Task Learning Model – Neural Network

3.2.1 Justification of the methods chosen

Multi-task learning (MTL) in neural networks is an exciting field of research that centers on learning multiple related tasks at once through a shared architecture. The idea behind MTL is that tasks frequently exhibit similar patterns, allowing a neural network to capture these shared characteristics to enhance learning efficiency and performance across tasks. These are the initial layers of the network where feature extraction occurs. The shared architecture enables the network to capture underlying representations that are beneficial across multiple tasks. By training on multiple tasks simultaneously, the network can generalize better on each task. MTL allows the network to learn features that are useful across multiple tasks, potentially reducing the amount of data needed for training each task separately. MTL has been successfully applied in various domains such as NLP, Computer Vision and Speech Recognition.

3.2.2 Further Preprocessing – Features Classification, Scaling and Encoding

a) Feature Classification

Neural networks require that all input data be numerical and scaled properly, so each type of feature is treated properly and equally. We first classify features into binary, categorical, and continuous types, then take corresponding feature engineering method to transform these features.

b) Scale Continuous Features

Neural networks commonly use gradient-based optimization methods such as SGD, Adam, etc., to minimize the loss function. When features have diverse value ranges, it can slow down or destabilize the training process because of significant variations in gradient updates for various weights. Scaling helps to ensure that all input features have an equal impact on the model's learning, avoiding any biases towards features with higher magnitudes. Properly scaled features help the learning algorithm to

converge more quickly towards the global minimum of the loss function.

c) Encode Categorical Features

When input categorical data directly, like using integers 1, 2, 3, it could confuse a model into thinking there's an order to the data when they're just labels. By using one-hot encoding, these labels get converted into a binary vector format. This helps the model grasp the information without implying any order or distance between the categories.

3.2.3 Hyper-parameter Tunning – Neural Network

Tune parameters with different values as shown below.

Parameters	Values
Layers	Add and reduce different layers with different number of nodes
Dropout and Batch Normalization	Different Dropout rate such as 0.3,0.5
Learning rate	0.001, 0.0001
Batch Size	8, 16, 32
Optimizer	SGD, Adam
Epochs	100, 200, 300

3.3 Evaluation Metrics

Metrics	Description
Accuracy	Accuracy is the simplest and most intuitive performance measure. It is the ratio of correctly predicted observations to the total observations. It can be misleading if data is imbalanced such as predicting rare positive cases in medical conditions.
Precision	Precision is the ratio of correctly predicted positive observations to the total predicted positives. Particularly useful in cases when the cost of a false positive is high such as in fraud detection.
Recall	Recall is the ratio of correctly predicted positive observations to all actual positive class. It is a measure of a classifier’s completeness. Essential in medical cases were catching all positive cases (diseases) is crucial even if it means enduring some false positives

	(misdiagnoses).
F1-score	The weighted average of precision and recall. Particularly useful when the classes are imbalanced.
Loss	Loss is a computation of error and is used by models to improve during training by minimizing this value. It is a measure of how far the predictions deviate from the actual outcomes. Loss here is the total cross entropy loss defined as the final target in this project specs by calculating the distance between predicted probability and target class.

We used a combination of these 5 metrics above to provide a comprehensive view of our models' performance, particularly in the case that the dataset is imbalanced in medical domain. Loss is the final indicator for assessing models' performance as required in project specifications.

4. Results

4.1 Results - Single Task Learning Models

Method	Label	Accuracies	Precision	Recall	F1	Loss
SVC	1	0.520	0.323	0.500	0.392	0.690
	2	0.225	0.214	0.977	0.351	0.705
	3	0.800	0.000	0.000	0.000	0.696
	4	0.665	0.262	0.421	0.323	0.676
	5	0.580	0.208	0.357	0.263	0.708
	6	0.595	0.326	0.549	0.409	0.671
	7	0.650	0.349	0.431	0.386	0.674
	8	0.740	0.350	0.350	0.350	0.685
	9	0.495	0.423	0.808	0.555	0.692
	10	0.535	0.325	0.765	0.456	0.663
	11	0.595	0.315	0.426	0.362	0.706
	Avg	**0.582**	**0.281**	**0.508**	**0.350**	**0.688**
Ada-Boost	1	0.675	0.444	0.194	0.270	0.690
	2	0.560	0.215	0.395	0.279	0.689
	3	0.475	0.200	0.564	0.295	0.690
	4	0.745	0.276	0.211	0.239	0.692
	5	0.345	0.220	0.833	0.348	0.689
	6	0.730	0.455	0.294	0.357	0.690
	7	0.720	0.381	0.157	0.222	0.688

	8	0.690	0.261	0.300	0.279	0.691
	9	0.465	0.393	0.679	0.498	0.694
	10	0.625	0.354	0.569	0.436	0.689
	11	0.630	0.344	0.407	0.373	0.693
	Avg	**0.605**	**0.322**	**0.418**	**0.327**	**0.690**
Random Forest	**1**	0.505	0.357	0.742	0.482	0.630
	2	0.235	0.217	0.977	0.354	0.566
	3	0.650	0.228	0.333	0.271	0.526
	4	0.785	0.273	0.079	0.122	0.529
	5	0.450	0.239	0.738	0.360	0.541
	6	0.620	0.338	0.510	0.406	0.587
	7	0.750	0.546	0.118	0.194	0.583
	8	0.475	0.240	0.750	0.364	0.536
	9	0.440	0.409	0.974	0.576	0.678
	10	0.360	0.278	0.941	0.429	0.575
	11	0.605	0.324	0.426	0.368	0.581
	Avg	**0.534**	**0.314**	**0.599**	**0.357**	**0.576**

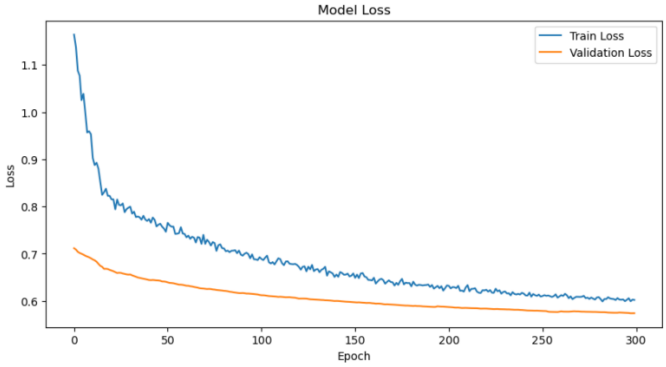
By comparison, we can conclude that Random Forest has the best performance with respect to Loss. For F1 score, Random Forest has similar result with SVC. So, we decided to use Random Forest as the optimal method for single task learning. The optimal parameters for Random Forest are as below.

Label	**Model**	**Optimal Parameters**			**Loss**
		max_depth	**min_samples_split**	**n_estimators**	
1	Random Forest	20	2	200	0.630
2	Random Forest	None	2	300	0.566
3	Random Forest	20	2	300	0.526
4	Random Forest	20	2	200	0.529
5	Random Forest	20	2	300	0.5410
6	Random Forest	None	5	100	0.5865

7	Random Forest	None	2	300	0.583
8	Random Forest	20	2	300	0.5360
9	Random Forest	20	2	100	0.678
10	Random Forest	None	2	300	0.5745
11	Random Forest	20	2	100	0.581

4.2 Results – Multi-Task Learning Neural Network Model

The plot below is the train loss and validation loss during model training. We can see that the final loss converged to about 0.6.



Metrics Result for Neural Network Model:

Hamming Loss	Precision	Recall	F1-Score	Train_loss	Test_loss
0.557	0.264	0.656	0.347	0.602	0.574

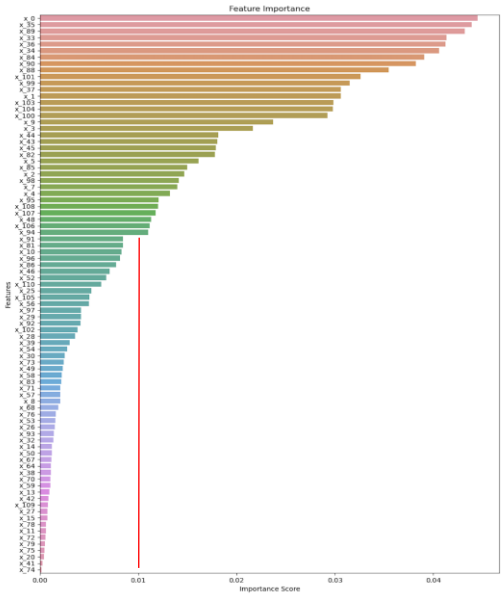
Parameters used for Neural Network Model:

Layers	Dropout	Learning rate	Batch Size	Optimizer	Epochs
4	0.5	0.0001	30	Adam	300

4.3 Analysis of Feature Importance

We used RandomForest Regressor to calculate the importance scores for features.

Removing low importance features can reduce overfitting and improve model performance. After comparing, we chose 34 features whose importance score is above 0.01.



5. Discussion

5.1 Comparison of Different Methods

Single-task	Accuracies	Precision	Recall	F1	Loss
SVC	0.582	0.281	0.508	0.350	0.688
Adaboost	0.605	0.322	0.418	0.327	0.690
Random Forest	0.534	0.314	0.599	0.357	0.576

Multi-task	Hamming Loss	Precision	Recall	F1-Score	Train loss	Test loss
Neural Network	0.557	0.264	0.656	0.347	0.602	0.574

For single-task methods, we can conclude that Random Forest has the best loss performance which is 0.576. And its F1 score 0.357 is very close to SVC's F1 score which is 0.350. And has the best Recall score which is 0.599 which is very crucial in medical conditions as we need to find all positive cases as we can. Overall, Random Forest method is the optimal method for single task learning. However, it may perform worse for some specific classes such as target label 4.

For multi-task, we used neural network algorithm. The final training results are shown

in Table 1. Due to the low degree of correlation between the various labels, the performance metrics of the training results are also not very ideal. Figure 4 visualizes the model loss, showing the changes in train loss and validation loss as the number of training iterations increases.

5.2 Future Improvements

a) The first is to collect more data. This dataset only has 1,000 entries, which is far from sufficient for predicting the onset and progression of specified conditions.
b) The dataset does not specify what each feature and label represent in the real world, which is abstract and inconvenient for researchers to understand.
c) During the data preprocessing stage, some extreme data were present. After collecting more data, further processing should be conducted to remove particularly deviant data.
d) More methods or fine-tuning should be done for some specific target classes which do not have good prediction results.

6. Conclusion

The medical machine learning project was based on a clinical dataset which has 111 features and 11 target labels to predict. The dataset size is 1000. Firstly, we visualized the data for all features and target labels to have a better understanding of the original dataset. Then preprocessed the dataset by removing redundant columns, filling missing values using the KNN interpolation method and removing lower importance features. To get the best results, we have chosen two modes for model training: a) single task learning methods including SVM, Adaboost and Random Forest; b) multi-task learning – Neural Network Method. GridSearch and Cross Validation were used in tuning parameters. The performance was evaluated considering five aspects: Accuracies, Precision, Recall, F and Loss, with F1 and Loss scores being the final and most important criterion for measurement.

By comparing F1 scores and Loss which are more crucial in medical domain, we found that Random Forest performed best, while multi-task learning – neural network performed second but close to Random Forest. Also, there is variance in prediction performance across all labels. For example, for label 4, we could also use other models such as Neural Network instead of Random Forest.

Reference

Aglietti, V., Damoulas, T., Álvarez, M. A., & González, J. (2020). Multi-task causal learning with Gaussian processes. *Neural Information Processing Systems*, *33*, 6293–6304. https://papers.nips.cc/paper/2020/file/45c166d697d65080d54501403b433256-Paper.pdf

Crawshaw, M. & Department of Computer Science, George Mason University. (2020). Multi-Task Learning with Deep Neural Networks: A Survey [Survey]. *arXiv, 2009.09796v1*, 1–10. https://arxiv.org/pdf/2009.09796.pdf

Daberdaku, S., Tavazzi, E., & Di Camillo, B. (2020). A combined interpolation and weighted K-Nearest neighbours approach for the imputation of longitudinal ICU laboratory data. *Journal of Healthcare Informatics Research*, *4*(2), 174–188. https://doi.org/10.1007/s41666-020-00069-1

Marquet, T., & Oswald, E. (2023). A comparison of Multi-task learning and Single-Task learning approaches. In *Lecture notes in computer science* (pp. 121–138). https://doi.org/10.1007/978-3-031-41181-6_7

Mbonyinshuti, F., Nkurunziza, J., Niyobuhungiro, J., & Kayitare, E. (2022). Application of random forest model to predict the demand of essential medicines for noncommunicable diseases management in public health facilities. ˜ *the œPan African Medical Journal*, *42*. https://doi.org/10.11604/pamj.2022.42.89.33833

Qi, Y., Taştan, Ö., Carbonell, J. G., Klein-Seetharaman, J., & Weston, J. (2010). Semi-supervised multi-task learning for predicting interactions between HIV-1 and human proteins. *Bioinformatics*, *26*(18), i645–i652. https://doi.org/10.1093/bioinformatics/btq394

Shukla, P. K., Verma, A., Abhishek, A., Verma, S., & Kumar, M. (2020). Interpreting SVM for medical images using Quadtree. *Multimedia Tools and Applications*, *79*(39–40), 29353–29373. https://doi.org/10.1007/s11042-020-09431-2

Vandenhende, S., Georgoulis, S., & Van Gool, L. (2020). MTI-NET: Multi-scale task Interaction Networks for multi-task learning. In *Lecture notes in computer science* (pp. 527–543). https://doi.org/10.1007/978-3-030-58548-8_31

What is Random Forest? | IBM. (n.d.). https://www.ibm.com/topics/random-forest

Zeng, T., & Ji, S. (2015). Deep Convolutional Neural Networks for Multi-instance Multi-task Learning. *2015 IEEE International Conference on Data Mining*. https://doi.org/10.1109/icdm.2015.92