# A Comparative Study of Multilayer Perceptrons and Support Vector Machines

**Abstract** - The objective of this research is to provide critical evaluations and comparisons of the performance of Multi-layer Perceptron (MLP) and Support Vector Machine (SVM) on Census income data. Different models for each of the algorithms were tested using different hyperparameters using a grid search approach. It was observed that regularization constants played a role in determining the best hyperparameters, but they were not the sole determinants. SVM outperformed MLP when the models were evaluated using ROC curves, validation curves and confusion matrix.

## 1. Introduction

The problem of income inequality has been a concern in the US for a couple of years. To further explore this problem, this paper uses the US Adult income dataset from the UCI machine-learning repository and aims to predict whether income exceeds $50K/year based on demographic information such as age, education, occupation, marital status, and race. For this purpose, two methods – MLP and SVM are used in this paper.

### 1.1. Multilayer Perceptron (MLP)

A multilayer perceptron (MLP) is a feedforward artificial neural network consisting of multiple layers of interconnected neurons. The advantage of using MLP is that

- It can handle non-linear relationships between the input features and the target variable.
- It can perform well even when the dataset has missing values or noisy data.

Along with the advantages, according to (Park and Lek) MLP has some disadvantages too:

- It is challenging to specify the number of neurons and hidden layers that will be needed for the optimal result.
- MLPs are like a black box, it is nearly impossible to see what happens inside.

### 1.2. Support Vector Machines (SVM)

SVM is a type of supervised learning and it works by finding the hyperplane that separates the two classes in the dataset with the maximum margin. The advantage of using SVM is that

- It is effective in high-dimensional spaces.
- It minimizes the probability of wrongly classifying test data. (Yue et al.)
- It produces a single unique solution when it finds the local minima as compared to NN. (Karamizadeh et al.)

Along with the advantages, SVM has some disadvantages too:

- It does not perform very well when the data set has noise. (Dhiraj)
- SVMs can be computationally expensive when they are trained on large datasets.

### 1.3. Hypothesis Statement

- SVM is likely to perform better in terms of accuracy, precision, and recall compared to MLP, due to its ability to handle high-dimensional data.
- The regularization parameter C for SVM and the number of hidden layers, neurons and the regularization parameter alpha for MLP will likely help in finding the optimal model.
- SVM is likely to train faster than MLP.

## 2. Dataset

The dataset used for this research is the adult income dataset from the UCI repository which contains demographic and income information of individuals from the 1994 census results extracted from the US Census Bureau. It contains 32561 instances and 15 features among which 14 are input features and 1 target variable “income”. The target variable has two classes:

- >$50,000: which represents individuals with income greater than $50k/year
- <=$50,000: which represents individuals with income less than or equal to $50k/year

There are missing values in the columns such as work-class, occupation etc. which are represented by “?”. These values were removed before feeding the data to the model which dropped down the number of instances to 32518. Further, the dataset has an imbalanced distribution of the target variable, with approximately 75% of the instances having an income less than or equal to $50k/year as shown in Figure 1.

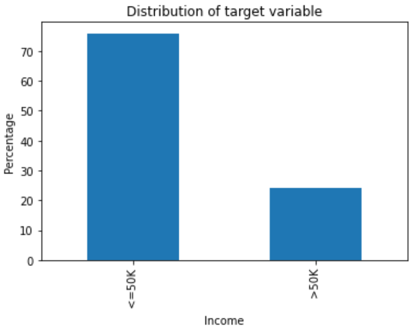


Figure 1. Class distribution of labels

### 2.1. Data Preparation and Analysis

Initially, the data contained 9 categorical variables including the target variable and 6 numeric variables. The EDUCATION column was dropped because it represents the same values as EDUCATION-NUM. The target column was converted to numeric using label encoding because the values can be represented in order. The other categorical columns were converted to numeric using one-hot encoding because there are a lot of different values in the columns and if label encoding was used, there could have been a chance that the model interprets the data of these columns to be in any order or rank.

To understand the relationship between variables, a Pearson’s correlation coefficient matrix was constructed as shown in Figure 2 which is a measure of the linear association between variables. It has a value between -1 and 1. From the figure, it was observed that FNLWGT had a negative correlation with the target variable which shows that there is a weak relationship between the two and the values of FNLWGT do not help in predicting the income. Therefore, the FNLWGT column was removed from the dataset.

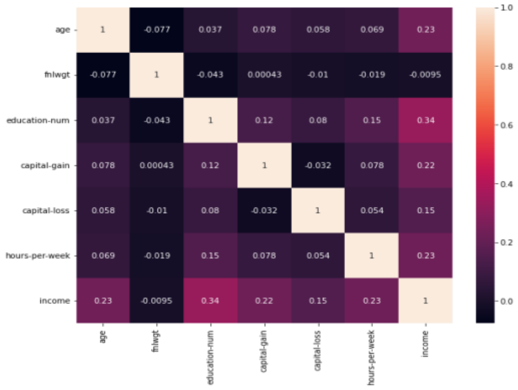


Figure 2. Correlation Matrix

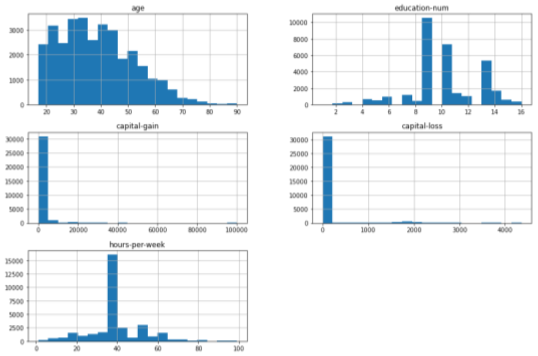


Figure 3. Histograms of numeric columns

Figure 3 represents the histograms for the numeric columns. The histogram of AGE shows that the majority of the people in the dataset are between 20 and 50 years old, with a peak around 35 years. There are fewer people in the dataset who are older than 50 years, indicating that the dataset may be skewed towards younger people, but the histogram does not represent if the younger people have income higher than 50k or not. The histogram of EDUCATION-NUM shows a bimodal distribution (Ted Hessing, Ramana), with peaks around 9 and 13. This variable represents the number of years of education completed by each person, and its distribution shows that there may be a majority of two groups of people in the dataset with education levels of HS GRAD(num=9) and BACHELORS(num=13). The histograms of CAPITAL-GAIN and CAPITAL-LOSS show a skewed distribution whereas the histogram for HOURS-PER-WEEK shows a bimodal distribution, with peaks around 40 and 50 hours/week. The columns CAPITAL-GAIN and CAPITAL-LOSS were removed because 90% of their values were equal to 0. The data were then scaled before being fed into the model. Since the dataset was imbalanced, random under-sampling was used to under-sample the majority class i.e. <=50K by randomly picking samples with or without replacement.

## 3. Methodology

The research's initial data processing phase used only visual display and analysis. This made it possible to show the links between variables in the datasets. In this phase of the research, the dataset will be modelled using SVM and MLP. For this, the dataset was split into a ratio of 80:20 i.e. 80% of the data was kept for training and 20% for testing. Further, for each model, 10% of the data was set aside for validation.

To choose the best model, a grid search approach was used to adjust the hyperparameters for both the MLP and SVM models. Each model was trained and validated using a 5-fold cv, which involves splitting the data into multiple folds and using each fold as the validation set while training the model on the remaining folds. This process is repeated for each fold so that every sample is used for both training and validation.

### 3.1. Architecture and Parameters for MLP

The MLP architecture used consists of multiple hidden layers with different sizes. The model used early stopping to prevent overfitting by stopping training when the validation score no longer improves. The problem with MLPs is that they can easily fall into local minima in the training process and if it does get stuck in a local minimum, they will not be able to find a better set of weights that could lead to better performance. To avoid this, the model used different solver combinations and learning rates which determine the step size taken during the gradient descent optimization that updates the weights of MLP. (Hafidz Zulkifli)
To find the optimal model, the model was put through a grid search that iterates over various hyperparameters. The hyperparameters that are tuned include the number of neurons in each hidden layer and the sizes of hidden layers, the regularization parameter alpha, the activation function, the solver used for weight optimization, the learning rate, and the momentum.

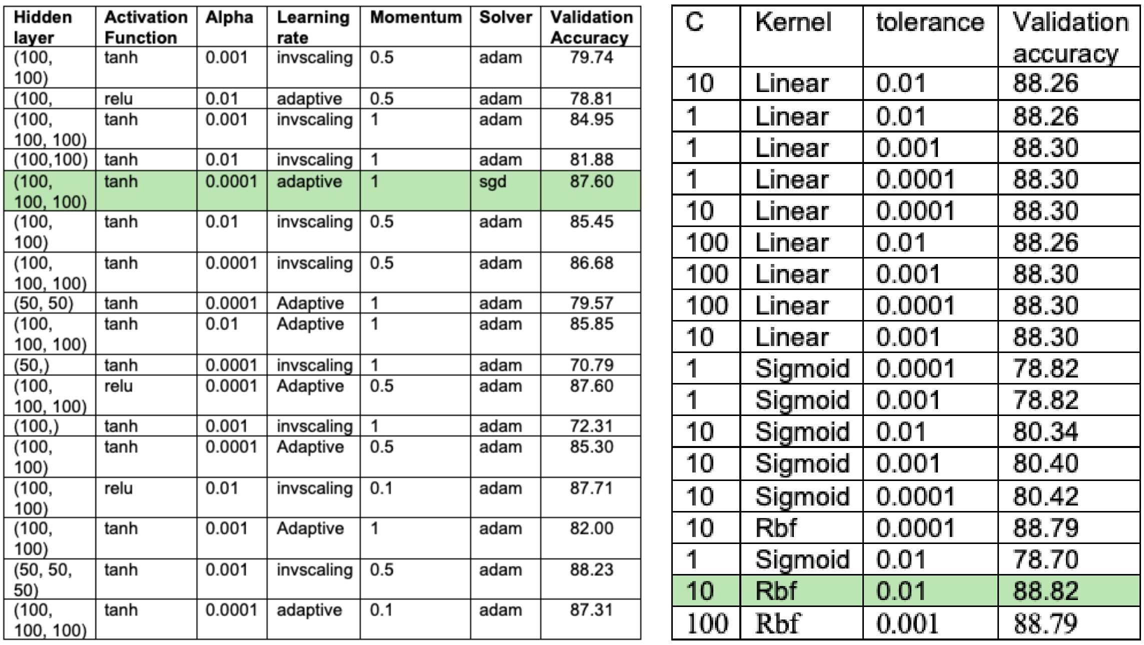
### 3.2. Architecture and Parameters for SVM

During the training phase, SVM tries to find the best hyperplane that can separate the data into different classes with maximum margin. One of the main problems with SVMs is their sensitivity to the choice of hyperparameters, which can significantly affect the model's performance. To address this, a grid search is used to tune the hyperparameters.
The hyperparameters that are tuned include C, which is the regularization parameter, the tolerance criteria which determines the convergence of the algorithm, and the kernel function which is one of the most important hyperparameters.

## 4. Results, Findings & Evaluation

### 4.1. Model Selection

For each combination of hyperparameters in the parameter grid, the model is trained and evaluated using cross-validation. While performing the grid search, accuracy was used as the scoring method. Table 1 shows the top 18 hyperparameter combinations obtained from grid search based on their performance during cross-validation.
For SVM, it can be observed that most of the hyperparameters have a linear kernel which exhibits very similar validation accuracies ranging from 88.27% to 88.30%. The RBF kernel shows the highest validation accuracy among all the kernel types, with the highest accuracy of 88.82% achieved with C=10 and tolerance=0.01. This shows that the RBF kernel might be the most suitable choice for this specific dataset, given the current hyperparameter search space. From the analysis above, the hypothesis statement is accepted which said C would be the key factor in determining the optimal model because all the combinations with C=10 have a higher accuracy.
For MLP it can be observed that models with more hidden layers and a larger number of neurons have achieved better validation accuracy. The majority of the best-performing models use the tanh activation function, which has a high validation accuracy range of 79.75% to 87.61%. This shows that the tanh activation function might be the most suitable choice for this specific dataset, given the current hyperparameter search space. It can also be seen that models with lower alpha values (0.0001 and 0.001) and high momentum values tend to perform better than those with higher regularization values (0.01), which suggests that less regularization might be better. This indicates that the choice of alpha has some impact on the model's performance, fulfilling the hypothesis statement. The selected hyperparameters on which the final models for SVM and MLP are trained are highlighted in Table 1.



| Hidden layer | Activation Function | Alpha | Learning rate | Momentum | Solver | Validation Accuracy |
|---|---|---|---|---|---|---|
| (100, 100) | tanh | 0.001 | invscaling | 0.5 | adam | 79.74 |
| (100, | relu | 0.01 | adaptive | 0.5 | adam | 78.81 |
| (100, 100, 100) | tanh | 0.001 | invscaling | 1 | adam | 84.95 |
| (100,100) | tanh | 0.01 | invscaling | 1 | adam | 81.88 |
| (100, 100, 100) | tanh | 0.0001 | adaptive | 1 | sgd | 87.60 |
| (100, 100) | tanh | 0.01 | invscaling | 0.5 | adam | 85.45 |
| (100, 100, 100) | tanh | 0.0001 | invscaling | 0.5 | adam | 86.68 |
| (50, 50) | tanh | 0.0001 | Adaptive | 1 | adam | 79.57 |
| (100, 100, 100) | tanh | 0.01 | Adaptive | 1 | adam | 85.85 |
| (50,) | tanh | 0.0001 | invscaling | 1 | adam | 70.79 |
| (100, 100, 100) | relu | 0.0001 | Adaptive | 0.5 | adam | 87.60 |
| (100,) | tanh | 0.001 | invscaling | 1 | adam | 72.31 |
| (100, 100) | tanh | 0.0001 | Adaptive | 0.5 | adam | 85.30 |
| (100, 100) | relu | 0.01 | invscaling | 0.1 | adam | 87.71 |
| (100, 100) | tanh | 0.001 | Adaptive | 1 | adam | 82.00 |
| (50, 50, 50) | tanh | 0.001 | invscaling | 0.5 | adam | 88.23 |
| (100, 100, 100) | tanh | 0.0001 | adaptive | 0.1 | adam | 87.31 |

| C | Kernel | tolerance | Validation accuracy |
|---|---|---|---|
| 10 | Linear | 0.01 | 88.26 |
| 1 | Linear | 0.01 | 88.26 |
| 1 | Linear | 0.001 | 88.30 |
| 1 | Linear | 0.0001 | 88.30 |
| 10 | Linear | 0.0001 | 88.30 |
| 100 | Linear | 0.01 | 88.26 |
| 100 | Linear | 0.001 | 88.30 |
| 100 | Linear | 0.0001 | 88.30 |
| 10 | Linear | 0.001 | 88.30 |
| 1 | Sigmoid | 0.0001 | 78.82 |
| 1 | Sigmoid | 0.001 | 78.82 |
| 10 | Sigmoid | 0.01 | 80.34 |
| 10 | Sigmoid | 0.001 | 80.40 |
| 10 | Sigmoid | 0.0001 | 80.42 |
| 10 | Rbf | 0.0001 | 88.79 |
| 1 | Sigmoid | 0.01 | 78.70 |
| 10 | Rbf | 0.01 | 88.82 |
| 100 | Rbf | 0.001 | 88.79 |

Table 1 - Hyperparameters Grid Search

### 4.2. Algorithm Comparison

The performance of SVM and MLP models based on the best hyperparameters obtained from grid search is evaluated using confusion matrix, roc curves, validation curves and loss curve for MLP for the test data. In the previous section, it was seen that the highest validation accuracy of MLP was 87.6% and SVM 88.8%, but when tested on unseen data the accuracy of SVM was almost the same but for MLP the accuracy dropped to 85.12%. So, one answer for MLP's performance is that the model may be overfitting on the training data and not generalizing well to unseen data.

Figure 4 shows the confusion matrix for both models and it can be seen that SVM performs better in terms of True Negatives (TN) for Class 0 and True Positives (TP) for Class 1 compared to MLP. However, SVM also has a higher number of False Positives (FP) for Class 0 and a lower number of False Negatives (FN) for Class 1. Overall, both models have a higher rate of classifying negatives(values for class 0) which means the imbalanced data problem still lies in the data.

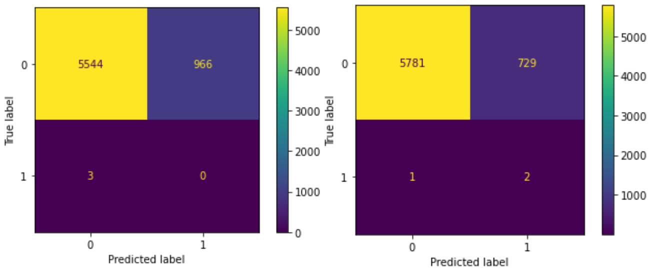


Figure 4. Confusion Matrix for MLP and SVM.

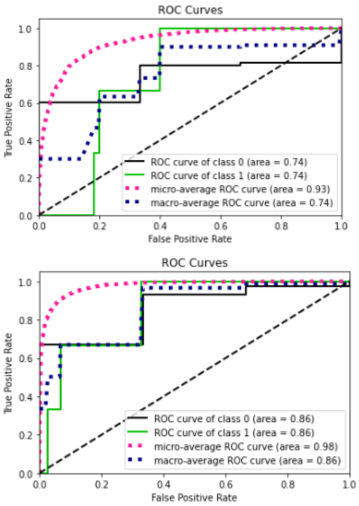


Figure 5. ROC curves for MLP and SVM

Figure 5 shows ROC curves for both models using the probabilities predicted on test data. The ROC curves show that the SVM has a higher area under the curve (AUC) for both classes 0 and 1 compared to MLP. Macro and micro-average ROC curves for SVM also have a much higher AUC than MLP which shows that SVM is better at performing well in individual classes.

The loss curve as shown in Figure 6 shows that MLP was not performing well initially, with a loss value higher than 0.7. However, as the iterations progressed, the model gradually improved, resulting in a significant decrease in the loss. After 125 iterations, the loss value started to decrease to less than 0.1, which suggests that the model had converged and was learning well.

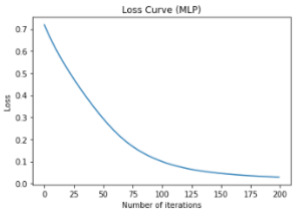


Figure 6. Loss Curve for MLP

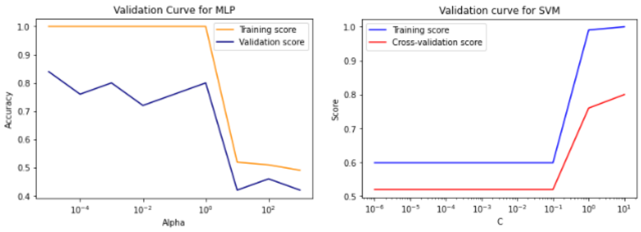


Figure 7. Validation curves for MLP and SVM

To check if the models were overfitting on training and validation data, validation curves were plotted on the values of regularizing constants C and ALPHA for both SVM and MLP as shown in Figure 7. In the case of MLP, the training score was constant at 100% accuracy for small values of alpha, indicating that the model was overfitting to the training data. As the value of alpha increased, the training score started to decrease, indicating that the model was generalizing better. The validation score started at 85% accuracy for small values of alpha, indicating that the model was performing reasonably well on the validation data. However, as the value of alpha increased, the validation score started to decrease, indicating that the model was starting to overfit. The sudden dip in the validation score, when the alpha value became greater than 10^0, suggests that the model was being too heavily regularized and was losing too much information during training. For SVM, the training score was constant at 60% but suddenly jumped to 90% accuracy when C became 10^-1. This suggests that the model needed a higher value of C to perform well on the training data. The validation score was 50% for low values of C (10^-6) and was constant but jumped to 75% when C became 10^-1. This indicates that 10^-1 is an optimal value for C.

## 5. Conclusion

In conclusion, based on the evaluation metrics, the SVM model appears to outperform the MLP model on the Adult income dataset. The highest accuracy of SVM achieved is 88.79% whereas for MLP it was 85.12%. The time taken for MLP to train was 283 seconds whereas it took only 4.8 seconds for SVM to train, which proved the hypothesis statement that SVM will take less time to train

For future work, other hyperparameters can be explored by other methods such as random search to evaluate different hyperparameter combinations. Some useful techniques can be used for class imbalance such as SMOTE because class imbalance was the main hurdle in this dataset. Different feature engineering techniques can be explored, such as feature selection or extraction, to improve model performance by reducing the noise or dimensionality of the dataset.

## 6. Reference

[1] S. Karamizadeh, S. M. Abdullah, M. Halimi, J. Shayan and M. j. Rajabi, "Advantage and drawback of support vector machine functionality," 2014 International Conference on Computer, Communications, and Control Technology (I4CT), Langkawi, Malaysia, 2014, pp. 63-65, doi: 10.1109/I4CT.2014.6914146.

[2] Chakrabarty, Navoneel, and Sanket Biswas. “A Statistical Approach to Adult Census Income Level Prediction.” IEEE Xplore, 1 Oct. 2018

[3] Dhiraj. “Top 4 Advantages and Disadvantages of Support Vector Machine or SVM.” Medium, 25 Sept. 2020

[4] Ding, Frances, et al. Retiring Adult: New Datasets for Fair Machine Learning.Hafidz Zulkifli. “Understanding Learning Rates and How It Improves Performance in Deep Learning.” Medium, Towards Data Science, 21 Jan. 2018

[5] Park, Y. -S., and S. Lek. “Chapter 7 - Artificial Neural Networks: Multilayer Perceptron for Ecological Modeling.” ScienceDirect, Elsevier, 1 Jan. 2016

[6] Ted Hessing, Ramana. “Bimodal Distribution.” Six Sigma Study Guide, 11 Apr. 2014

[7] Yue, Shihong, et al. “SVM Classification:Its Contents and Challenges.” Applied Mathematics-A Journal of Chinese Universities, vol. 18, no. 3, Sept. 2003, pp. 332–342,

## Appendix 1 – Glossary

- **Accuracy**- total correctly classified examples by the model divided by the total number of classified examples.
- **Confusion matrix**- a table that summarizes how successful the classification model is at predicting examples belonging to various classes.
- **Correlation**- a measure of how well two or more variables are related.
- **Hyperparameters**- parameters whose values control the learning process and determine the values of model parameters that a learning algorithm ends up learning.
- **Imbalanced data**-a lack of balance between the target variable's classes.
- **Regularization**- techniques that are used to calibrate machine learning models in order to minimize the adjusted loss function and prevent overfitting or underfitting.
- **ROC-curve-** a graph showing performance of a classification model at all classification thresholds.

## Appendix 2 – Implementation details

### Experiments:

- For MLP I used 500 iterations first but then it gave me a warning Stochastic Optimizer: Maximum iterations (500) reached and the optimization hasn't converged. I increase the max iterations to 2000 but still, this warning does not resolve.
- I used Sklearn's linear SVC for SVM first and the model performed well since it scales better with more samples but the accuracy was a bit low so I Switched to SVC. The training time significantly increased from 2 seconds to 10 seconds whereas the accuracy also increased to 86% from 75%.

### Negative results:

- I used one hot encoding for categorical variables because all the values in those variables were not in a specific order. But this resulted in my columns increasing from 10 to 83 which is very memory-consuming and is taking a lot of time to run.
- Before under-sampling, the accuracy score was around 98% but after it, the accuracy dropped to 81% for MLP which is less than before. But at least I am sure that the training set has equal data for both classes.
- Since I only did under-sampling on training data, a sample of data is used for training in which both the classes have equal representation but since the data used for training is less and test data is 6513 rows, the predictions made are poor.