Phase 4

Kel Gruber

November 26, 2023

1 Phase Goal

In this phase, I took the best model architecture from my Phase 3 models and evaluated the effects of each feature on the accuracy of the model. I want to determine if all features in the dataset are relevant and working to improve the accuracy of the model and if not which ones could be removed to build a better prediction model for this problem.

2 Introduction

To identify which features are important to the model and which features might be causing unnecessary noise I will begin by taking the best model I found in Phase 3 and create an individual model for each feature in my data set.

The best model from Phase 3 was the logistic regression model with one neuron. I decided to use this model because it was the most consistent and balanced of all my models. In the final evaluation, this model's accuracy on the training set and accuracy on the testing dataset were closest to each other indicating that the model had not become overfit during training or due to too large of architecture. It also had the best recall and f1-score of all of my models and the precision is similar to the recall score. This indicates that the balance and completeness of the true positives are good. Typically as precision goes up recall will go down in a model and this happened in the other models I tried.

3 Identifying Individual Feature Impact on the Model

To identify a feature's relative importance on the model's accuracy I set up a for loop that would create 26 models, one for each feature in my dataset. These models each take only one input feature to see the effect this feature has on the validation accuracy of the model. I then stored the feature and validation accuracy of that feature's model in a pandas dataframe. With this dataframe I could then graph and visualize the relationship between each input feature and the model's validation accuracy. Each model was given 60 epochs to train. I had attempted to give each model 100 epochs like in the Phase 3 models but this was too much for my jupyter notebook and the notebook would crash so I had to reduce the total to 60 epochs. The early stopping flag was also set to stop the training if the val-loss did not improve after 20 epochs to prevent overtraining and overfitting the individual models. The results of these models are graphed below.

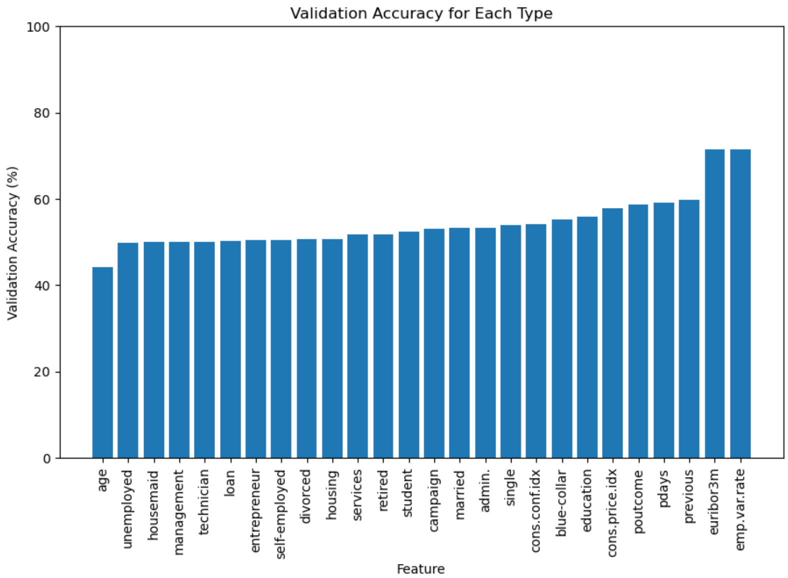

Figure 1: Validation accuracy of the models built with just one input feature to determine a feature's relative importance to the model.

4 Comparison of Reduced Features to the Original Model

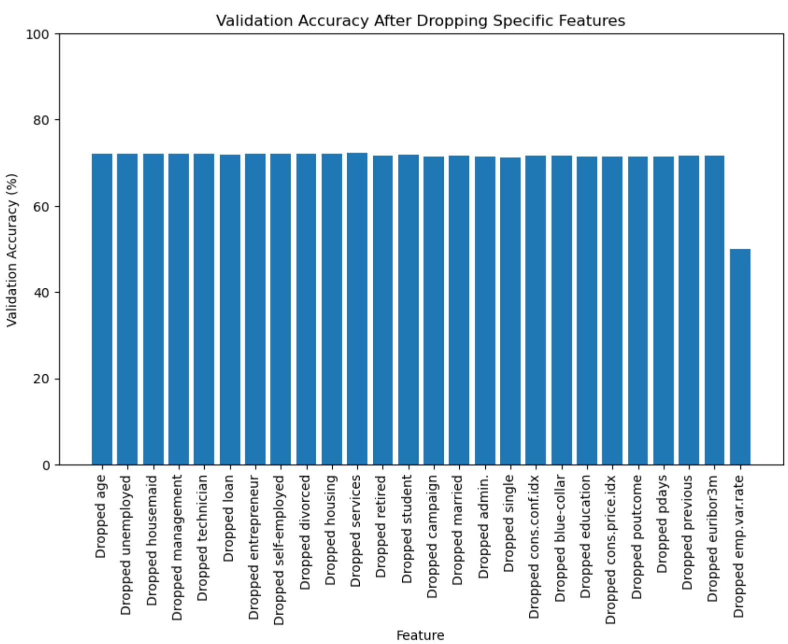

Figure 2: This graph illustrates the impact removing certain features has on the model's validation accuracy. The accuracy of the model is depicted after each iteration where features were removed one by one in order of least importance.

Looking at the results of dropping specific columns it does not appear that any particular column was adding severe noise or bias to the model because little changed in the overall accuracy of the model as the columns with low effect on the model's accuracy were removed from the dataset. Accuracy for the most part stayed about the same as the initial model built in Phase 3 which had a 71.6% validation accuracy. I thought there would have been a bigger difference once columns like age, the euribor and previous contacts features were removed. This is because age shown to not be a good predictor for the output variable with only 44% accuracy. While and the euribor and previous campaign features had higher validation accuracy with 71% and 59% respectively so I thought accuracy would have dropped more with their removal. In the end there was little change from iteration to iteration of the loop until the emp.var.rate feature was dropped which is when the model had no input features available to train from so the model simply had to guess and ended up with a 50% accuracy rate because of basic probability. I suspect the minimal impact is because as seen earlier in Phase 1 none of these features had a strong correlation with the output variable so it does not appear that any of them have a strong impact on the prediction model. I believe that if there were features with a stronger correlation to the output variable then this model would potentially perform better overall and we would have seen a greater change in this phase. At the same time this was a good exercise because I'd questioned in Phase 1 if I should have removed the poutcome and pdays features from the model to prevent noise and looking at the results here that removing them would have not drastically changed my overall results as initially thought.

5 Challenges

Challenges for this phase included writing code so that the models could be built in a loop and storing the model's accuracy information. It was also difficult to choose which model architecture was best to examine. I ended up choosing the same model structure from Phase 3 that appeared to best fit the problem to remain consistent and be able to compare the final results with the initial model. Please see the Phase 4 Jupyter Notebook for more details on the Phase 4 execution and results.

References

[Vid, 2023] (2023). Feature selection techniques in machine learning (updated 2023).

[Keldenich, 2021] Keldenich, T. (2021). Recall, precision, f1 score - simple metric explanation machine learning.

[Whiteboard, 2023] Whiteboard, L. W. (2023). 8 tips on how to choose neural network architecture.