Predicting Bank Churn with Artificial Intelligence

Jared Diehl

May 2021

Contents

1 Introduction

Churn, short for churn rate, is when customers stop doing business with an entity. [3] There can be several factors that affect churn, including age, gender, and salary. Europe houses some of the world's largest economies, such as France, Germany, and Spain. Therefore, a business may want to predict when their customers might leave, especially a bank.

This project aims to develop a machine model to predict which customers are more likely to leave their bank. The development process is separated into several phases: data cleaning, model selection and evaluation, and feature importance. The development of the model is documented in this report, along with visualizations to aid understanding. First, data cleaning is needed to prepare the data for the model to increase learning potential. Then several models will be tested. The first step is to find the model that overfits the data. Then a more efficient will be built and evaluated on the validation and test sets. Next is to use regularization techniques to reduce overfitting and make the model generalize. Lastly, is to evaluate it on the test set. Finally, feature importance is studied to show which features least significantly impact the model's predictions. This shows where data redundancies exist and which features can be removed. Thus, increasing training speed and reducing overall complexity. Google Colab was used to develop this project at a rapid speed across several months. This project will serve as an example of building, training, and evaluating a feed-forward neural networking model capable of predicting customer bank churn.

The objective is to build a feed-forward neural network model that solves a binary classification problem. This is also a supervised learning problem which means data collection, model creation, and model evaluation are done manually.

Link to Project: https://github.com/jaredible/CS4300-Project

2 Data Analysis and Preparation

To explore why customers leave their bank, I have chosen a churn modelling dataset from Kaggle relating to banks. This dataset is made for binary classification, and it is tabular. The author did not state how or when the data was collected.

2.1 Data Attributes

- Row Number
- Customer ID
- Surname
- Credit Score
- Geography
- Gender
- Age
- Tenure
- Balance
- Product Count
- Has Credit Card
- Is Active Member
- Estimated Salary
- Exited

2.2 Data Cleaning

Data cleaning refers to identifying and correcting errors in the dataset that may negatively impact a predictive model. [2] Fortunately, there are no missing values, so all 10,000 rows can be used. For columns, some are irrelevant for helping predict churn, so they will be removed. Some names were changed to provide better meaning and follow the convention of lowercase words separated by underscores.

2.3 Data Distribution and Visualization

Data visualization is the representation of data in a graph, chart, or other visual formats. It communicates the data with images. This is important because it allows trends and patterns to be more easily seen. [4] In Figure 1, a correlogram is used to visualize churning amongst the input features. This way, we can see which are correlated with each other.

Churned	2037
Unchurned	7963

Table 1: Data Distribution

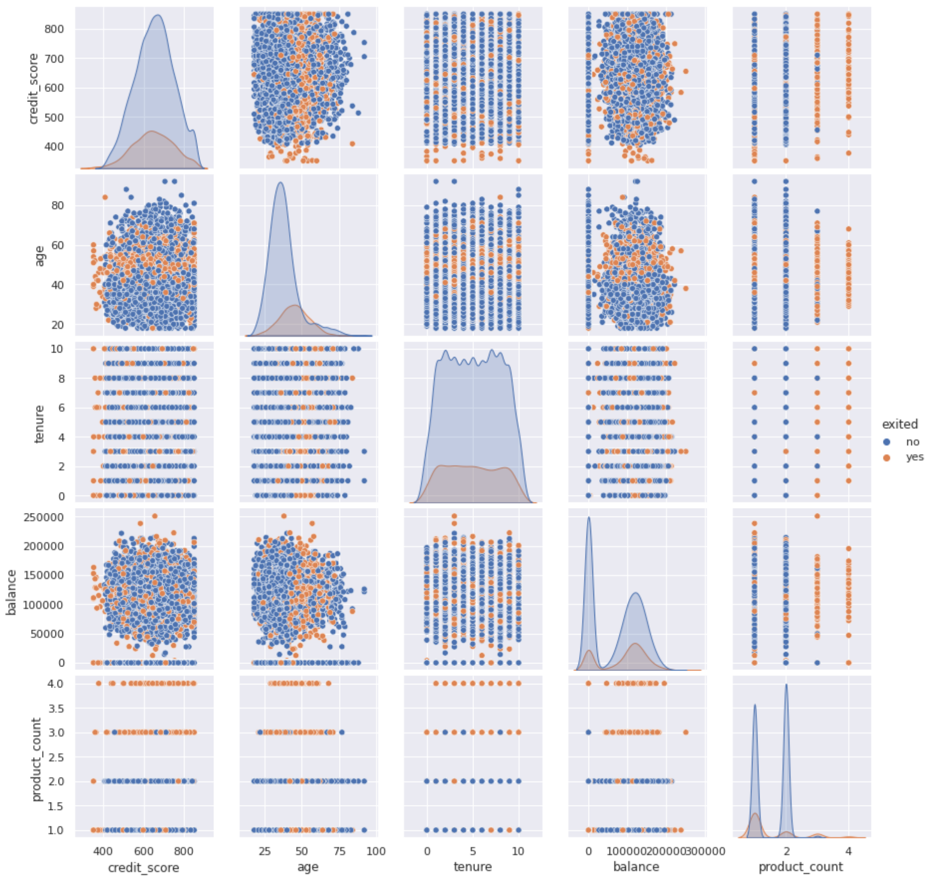


Figure 1: Feature Correlations Sample

Testing

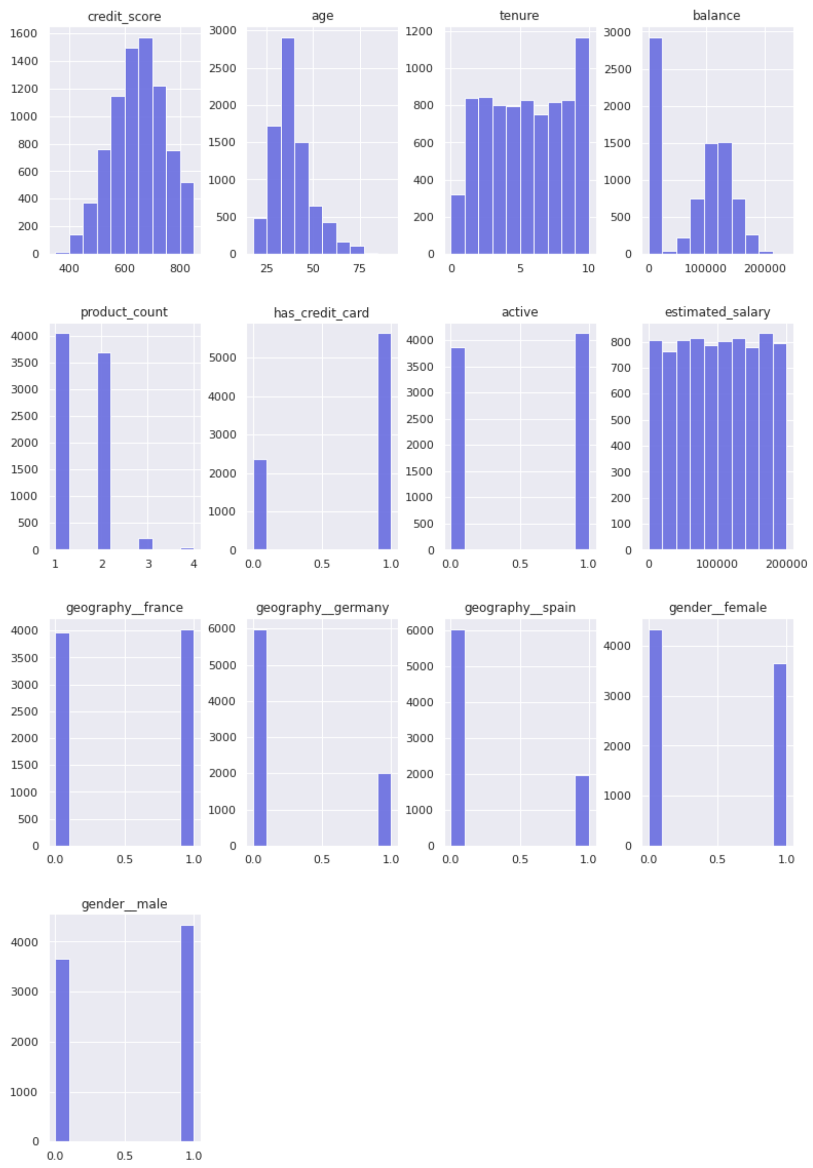


Figure 2: Input Distributions

2.4 Data Normalization

Normalization is a scaling technique in which values are shifted and rescaled so that they end up ranging between 0 and 1, which helps the model make better predictions faster. The data will be normalized using the mean normalization technique, as can be seen in (1).

$$X_{new} = \frac{X - X_{mean}}{X_{max} - X_{min}} \tag{1}$$

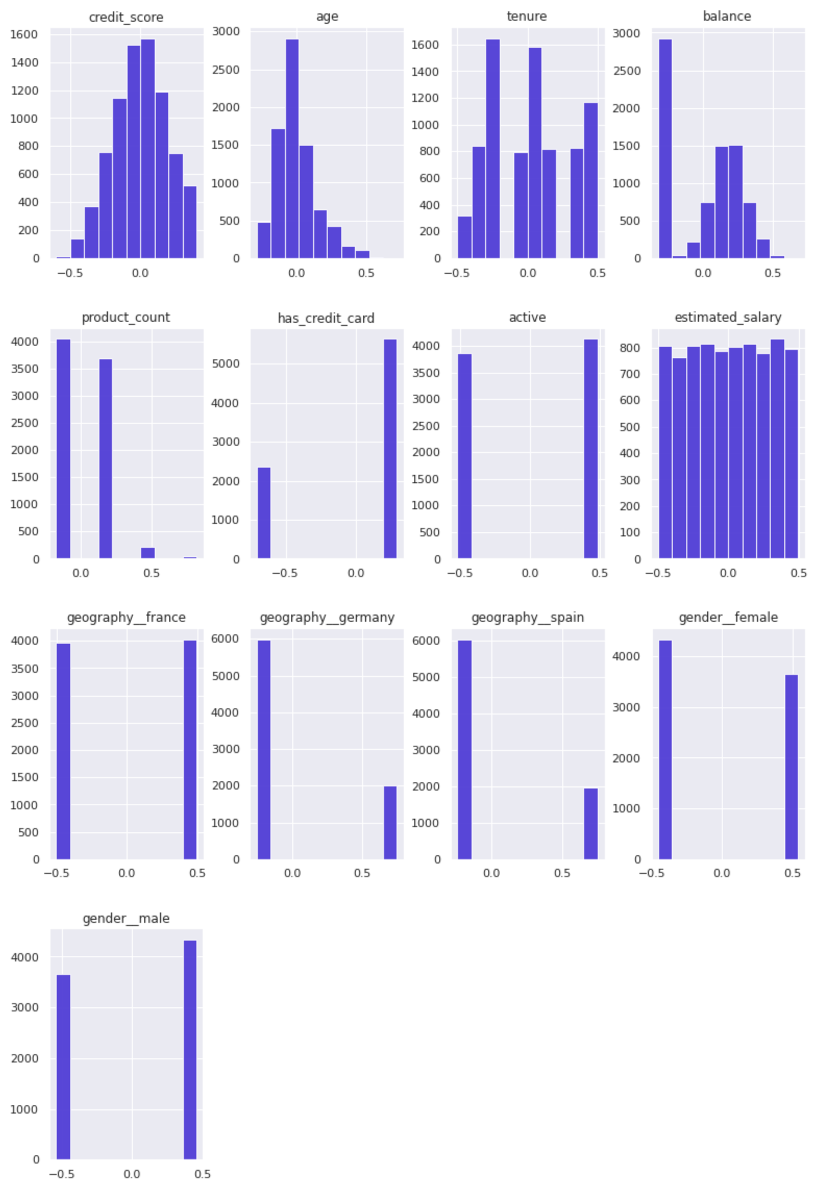


Figure 3: Normalized Input Distributions

3 Model Selection and Evaluation

Deep learning models are built using neural networks. A neural network takes in inputs, which are then processed in hidden layers using weights that are adjusted during training. Then the model spits out a prediction. The weights are adjusted to find patterns to make better predictions. [1] To find the best model, nine models were tested. Their learning curves can be seen in Table 2 below.

Model Evaluation is an integral part of the model development process. It helps to find the best model that represents our data and how well the chosen model will work in the future.

The first step to finding a good model is to build an overfitting model, then to regularize it. An overfit model will perform very well on the training data but perform very poorly on the validation data, as can be seen in Figure 11 below. To fix this, the overfit model will need to be regularized. A regularized model will perform about the same on both the training and validation data. This means the model is generalizing well, as can be seen in Figure 12 below.

Typically, training can take a great deal of time. This is because the model is being trained for too long. To stop this, an early-stopping technique can be used. That is if our model is not making better predictions, then we can stop the training early. This technique was used to reduce the total training times of these models. The models were trained with 2^{10} epochs and a batch size of 2^5 while using the early-stopping technique. Ultimately, this reduced the total training time substantially.

The goal of building and evaluating models is to find the one that makes the best predictions. This model is made of four layers: an input layer, two hidden layers, and an output layer. The first three layers use leaky-relu activation functions, and the last layer uses a sigmoid activation function. Dropout is employed between each of these layers.

	Accuracy	**MAE**	**Loss**	**Precision**	**Recall**	**F1**
Baseline 1	80.55%	-	42.32	51.54%	29.55%	37.56%
Baseline 2	85.80%	-	34.24	71.37%	47.22%	56.84%
Baseline 3	86.05%	-	34.11	73.31%	46.46%	56.88%
Linear Last Neuron	86.15%	21.42	10.38	72.45%	48.48%	58.09%
Linear All Neurons	81.05%	27.80	13.60	66.67%	8.59%	15.21%
Logistic Last Neuron	85.75%	-	33.63	69.89%	49.24%	57.78%
Logistic All Neurons	86.75%	-	33.5	77.64%	46.46%	58.14%
Overfit	82.60%	-	47.89	56.67%	51.52%	53.97%
Regularized	86.40%	-	33.08	74.22%	47.98%	58.28%

Table 2: Model Performance Statistics

3.1 Baseline Model 1

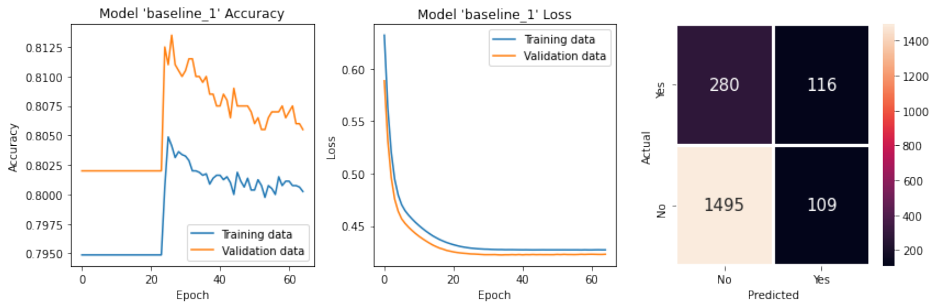


Figure 4: Baseline Model 1 Evaluation

3.2 Baseline Model 2

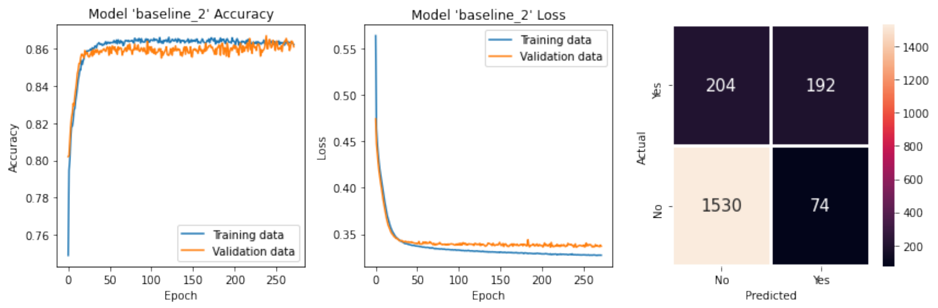


Figure 5: Baseline Model 2 Evaluation

3.3 Baseline Model 3

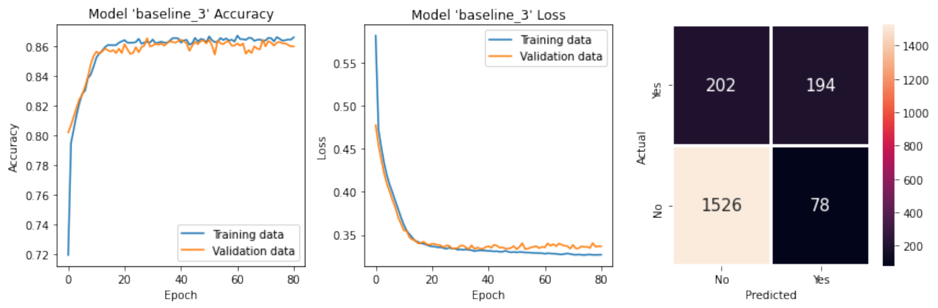


Figure 6: Baseline Model 3 Evaluation

3.4 Last Neuron Linear Model

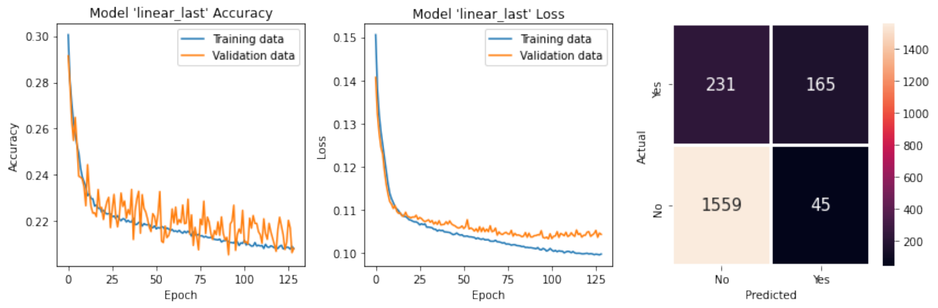


Figure 7: Last Neuron Linear Model Evaluation

3.5 All Neurons Linear Model

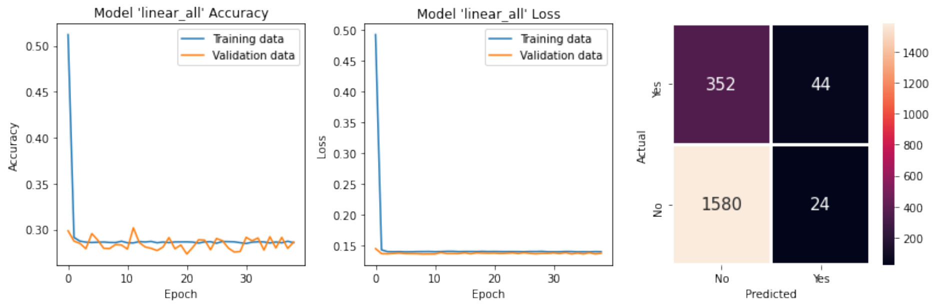


Figure 8: All Neurons Linear Model Evaluation

3.6 Last Neuron Logistic Model

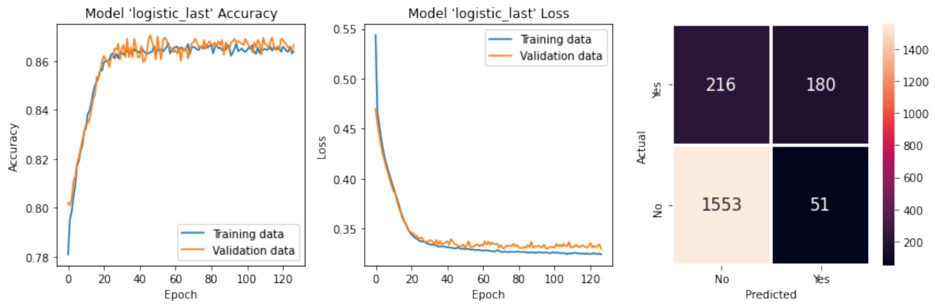


Figure 9: Last Neuron Logistic Model Evaluation

3.7 All Neurons Logistic Model

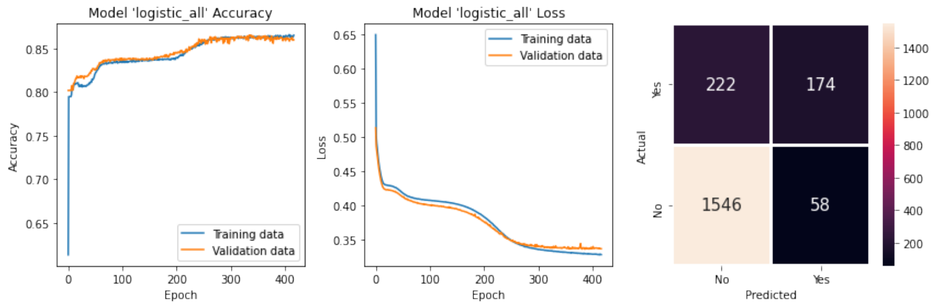


Figure 10: All Neurons Logistic Model Evaluation

3.8 Overfit Model

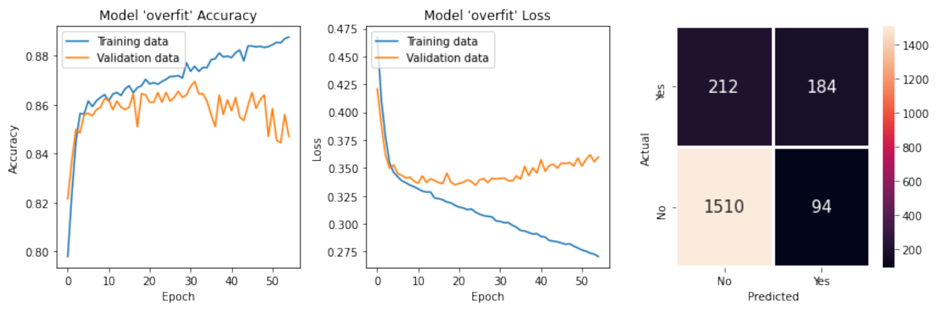


Figure 11: Overfit Model Evaluation

3.9 Regularized Model

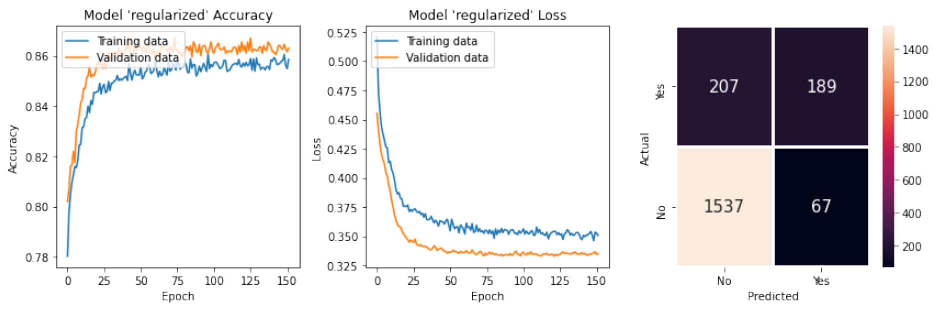
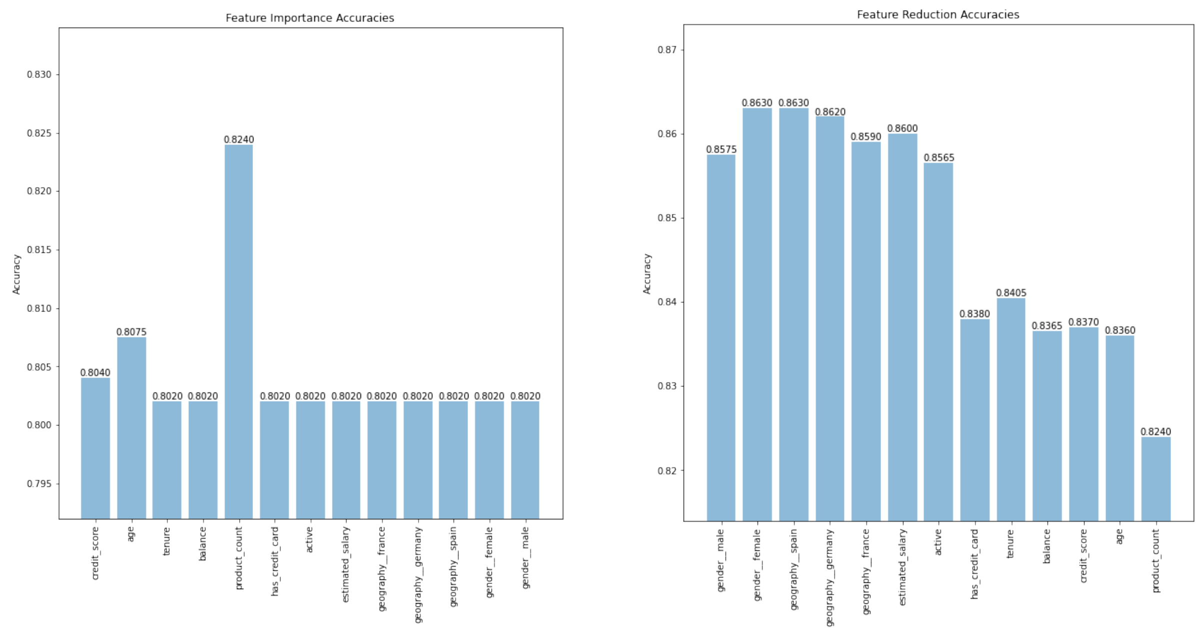


Figure 12: Regularized Model Evaluation

4 Feature Importance and Reduction

As of now, we have the best model and know how to train optimally. Here is where we study how important each feature is and observe how well the model does predicting when we remove less important features. We do this to see if we can remove features without significantly affecting prediction accuracy. This could show that we can build a smaller model, which can ultimately reduce model complexity and decrease training time.



(a) Feature Importance Evaluation

(b) Feature Reduction Evaluation

Figure 13: Feature Importance and Reduction Evaluation

4.1 Feature Significance

To study how significant each feature contributes to the model making predictions, the best model was trained on each feature individually. The model prediction accuracies can be seen in Figure 13a. The features with the highest accuracies have the higher significance, such as: `product_count`, `age`, `credit_score`, and `balance`.

4.2 Feature Reduction

Now, we need to study how removing features affects model accuracy. The model was trained by removing less significant features one at a time. The model prediction accuracies can be see in Figure 13b. The model performed near its optimal accuracy with the first 4 least significant features removed, which also confirms that the features were insignificant. Ultimately, it appears that `product_count` alone significantly determines churn.

5 Conclusion

The goal of this several-month-long project was met as an optimal feed-forward neural network model was achieved. This project now serves as a complete example of building, training, and evaluating a feed-forward neural networking model capable of predicting bank churn. The data collected was sufficient enough to find this model and test. With even more data, a more generalized model could be constructed. The learning curves and evaluations of the models prove this finding across all the development phases. Even though the data was imbalanced, the model still made quite accurate predictions. Using one-hot encoding for geography and gender shall have improved the models learning too. The hyperparameters were found that yielded the most optimal predictive model across the test set. With less skewed data, the model could also generalize better. Testing the baseline models showed how small the network needed to be to overfit the data. The results of the special linear and logistic models showed that logistic activations are preferred for this type of classification problem. Building an overfit model was also quick to find and evaluate on the test set with the help of the early-stopping technique. The effects of model regularization significantly reduced overfitting. The typical regularization techniques used were sufficient enough to improve training speed and reduce complexity. Overall, model regularization did significantly improve the development process. This project can help facilitate understanding of how feed-forward networks can be used for a binary classification problem and may serve as inspiration to go beyond just these types of models.

References

[1] Eijaz Allibhai. *Building A Deep Learning Model using Keras.* URL: https://towardsdatascience.com/building-a-deep-learning-model-using-keras-1548ca149d37. (accessed: 09.17.2018).

[2] Jason Brownlee. *How to Perform Data Cleaning for Machine Learning with Python.* URL: https://machinelearningmastery.com/basic-data-cleaning-for-machine-learning/. (accessed: 02.25.2021).

[3] Jake Frankenfield. *Churn Rate.* URL: https://www.investopedia.com/terms/c/churnrate.asp. (accessed: 02.25.2021).

[4] Import.io. *What is Data Visualization and Why Is It Important?* URL: https://www.import.io/post/what-is-data-visualization. (accessed: 02.25.2021).