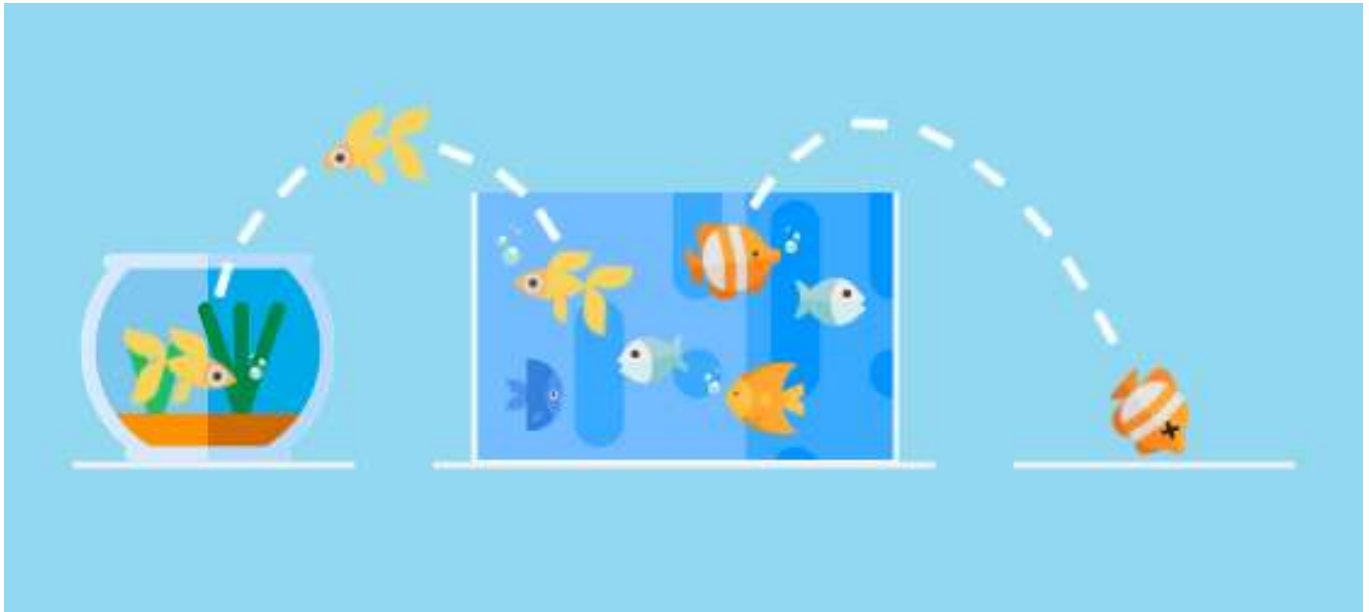


Customer Churn Rate

FINAL REPORT



Deep Learning Challenge

Haruna Faith 3010880

Talia Santos 3294579

Table of Contents

1. Introduction	2
2. Goal	2
3. Exploratory Data Analysis	2
3.1 Using Tableau - Faith	3
3.2 Using Python - Talia	10
4. Deep Learning Modelling with ANN	18
4.1 Individual Experiments - Faith	18
4.2 Individual Experiments - Talia	23
4.3 Joint Experiments	28
5. Ethical Consideration	35
6. Conclusion	36
7. Recommendations	36
References	37
Websites	37
Our Notebooks	37

1. Introduction

The churn rate is the percentage of subscribers to a service who discontinue their subscriptions to the service within a given time period. In this case, for a banking company to expand its clientele its growth rate, as measured by the number of new customers, must exceed its churn rate. This is a Classification Problem in which we need to classify a customer based on their Credit Score, Gender, Age, Tenure, Balance, Salary, among other things, to determine whether they will leave the bank or not.

To better understand what could be behind a client leaving the bank, we have explored the dataset and made visualizations which give us a better understanding of the data and correlations. We have also done individual experimentation and modeling to gather the best methods. Lastly, we have combined the efforts and further experimented on some other aspects we believed were of significant importance.

2. Goal

The general project goal is to create a model which can predict how likely it is for the customers to leave the bank (close their account) in the near future, therefore calculating the churn rate.

The goal of this document is to report on the findings we have gathered throughout our analysis of the dataset. We wanted to find out what features might have a higher impact on a client leaving or not. We experimented with different layer configurations, activations, optimizers and feature selection differences to investigate the performance. The performance was evaluated with the use of training and validation accuracy, as well as training and validation loss.

3. Exploratory Data Analysis

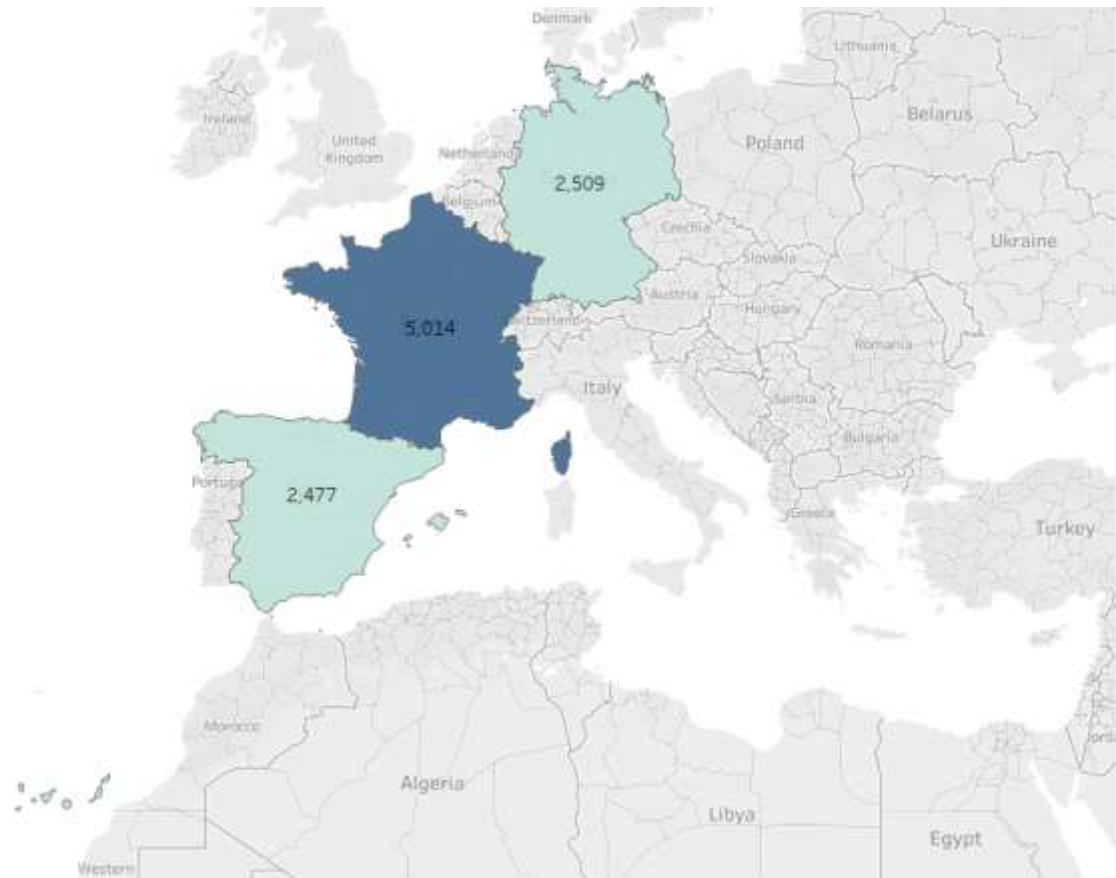
The data is in CSV format and has 10 000 records. There are 14 columns, which are described below:

- RowNumber - Number of the row
- CustomerId - Unique IDs for bank customer identification
- Surname - Customer's last name
- CreditScore - Customer's credit score, between 350 and 850. The higher the score, the better a borrower looks to potential lenders
- Geography - Customer geographical location (country)
- Gender - Customer's gender (male or female)
- Age - Customer's age
- Tenure - Number of years for which the customer has been with the bank
- Balance - Bank balance of the customer in Dollars
- NumOfProducts - Number of bank products the customer is utilising
- HasCrCard - Binary Flag for whether the customer holds a credit card with the bank or not
- IsActiveMember - Binary Flag for whether the customer is an active member with the bank or not
- EstimatedSalary - Estimated salary of the customer in Dollars
- Exited - Binary flag 1 if the customer closed account with bank and 0 if the customers retained

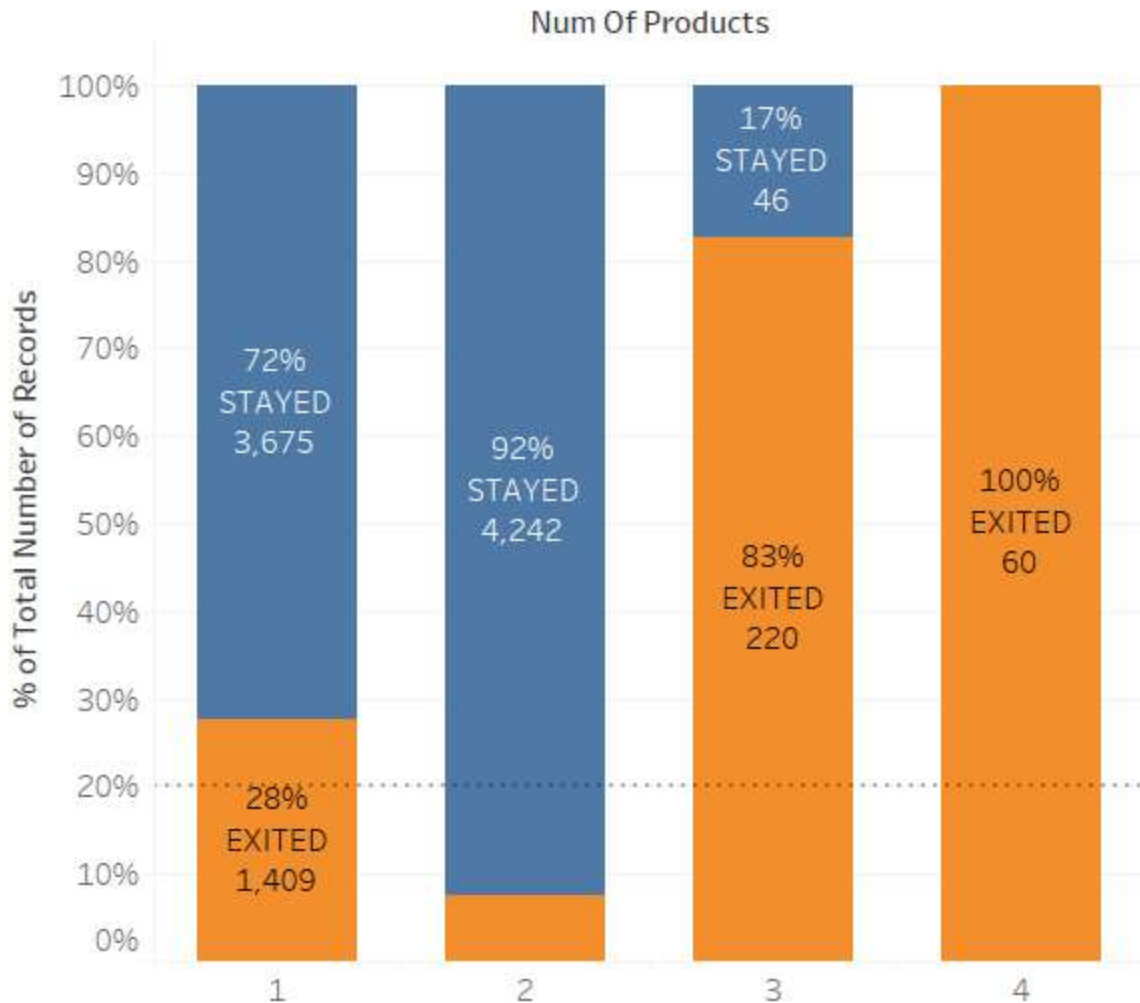
The analysis was done both in Tableau and Python.

3.1 Using Tableau - Faith

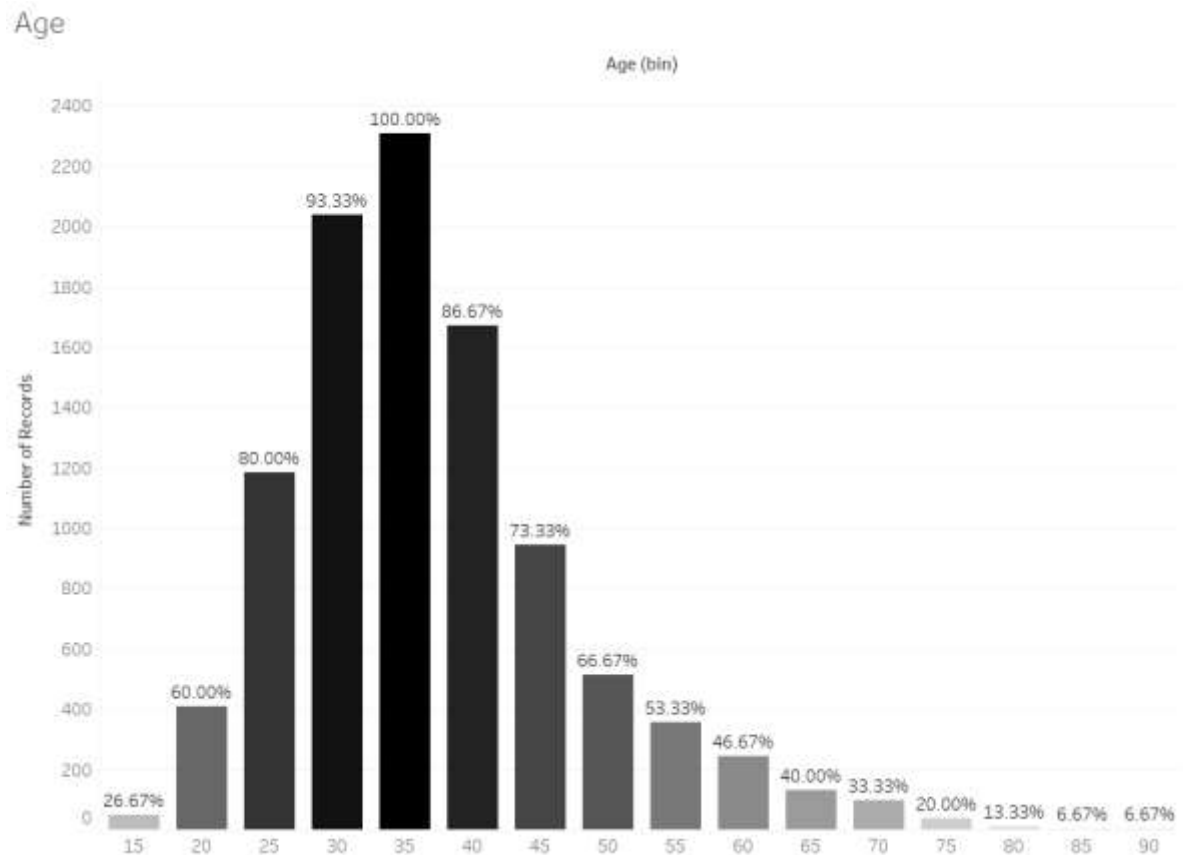
Number of Customers per Country



From the above map, we can see that the bank has clients in Spain, France and Germany. France is where most of the clients can be seen. Spain and Germany have similar amounts of clients and each of them individually constitute around half as much as those we can see in France.



The graph above shows the number of products. looking at the graph you can see that the more the products, customers are likely to exit the bank. it is also important to note that there are anomalies in the last two categories. This is why the count is displayed to show that it's not entirely true that people with three or four products are likely to leave the bank but to also note that the last two categories have few number people in those categories.



The graph above is a right skewed distribution, it shows the percentage of customers and the age range using a bin size of 5. Most of the customers that stayed are between age 25 categories and age 45 categories.

Insight derived from this graph is that more young people stayed in the bank than older people.



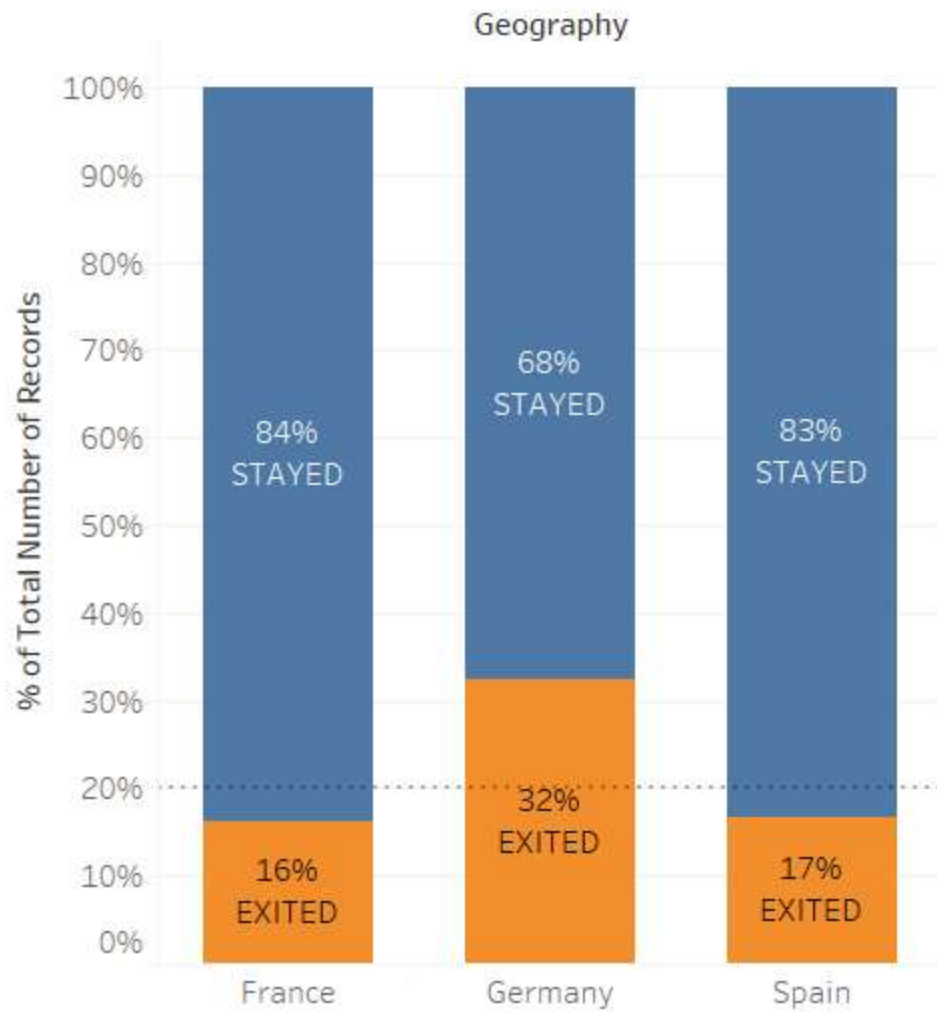
The graph above shows the percentage of male and female that stayed or exited the bank. Comparing them visually we can say that women are more likely to leave the bank.



From the above graph we can see the percentage of people who are active members or not which have either stayed or left the bank. From the values we can see that more people leave when they are not active members, than those who are active members.



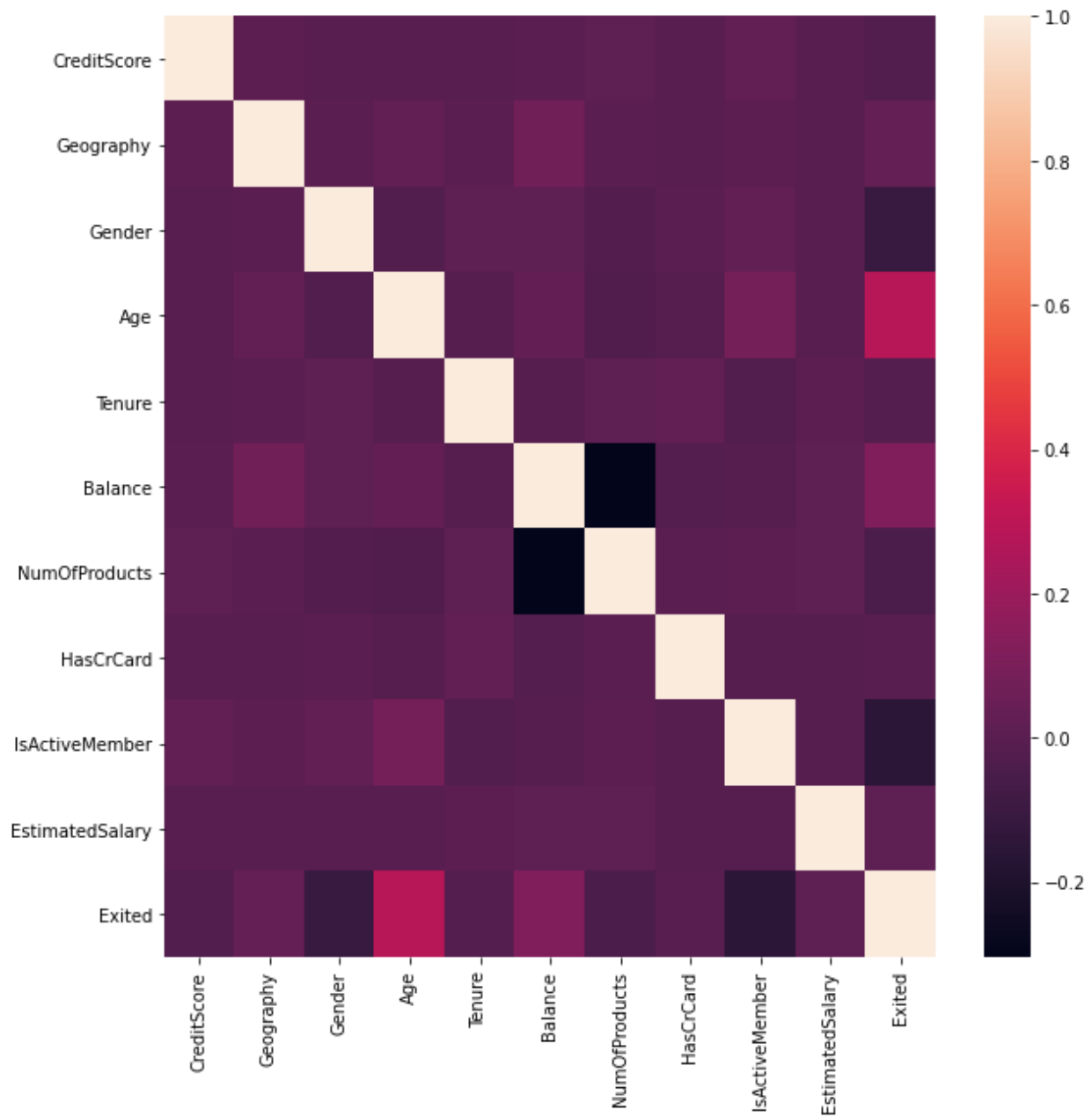
From the above graph we can see the percentage of people that have a credit card or not which have either stayed or left the bank. We can see the difference is not significant enough to conclude any trend. People leave/stay just as much, no matter if they have a credit card or not.



The graph above shows people in germany are exiting the bank at a rapid rate. 32 percent of customers in Germany left over the period of the observation. Also, you can see that france and spain are below average.

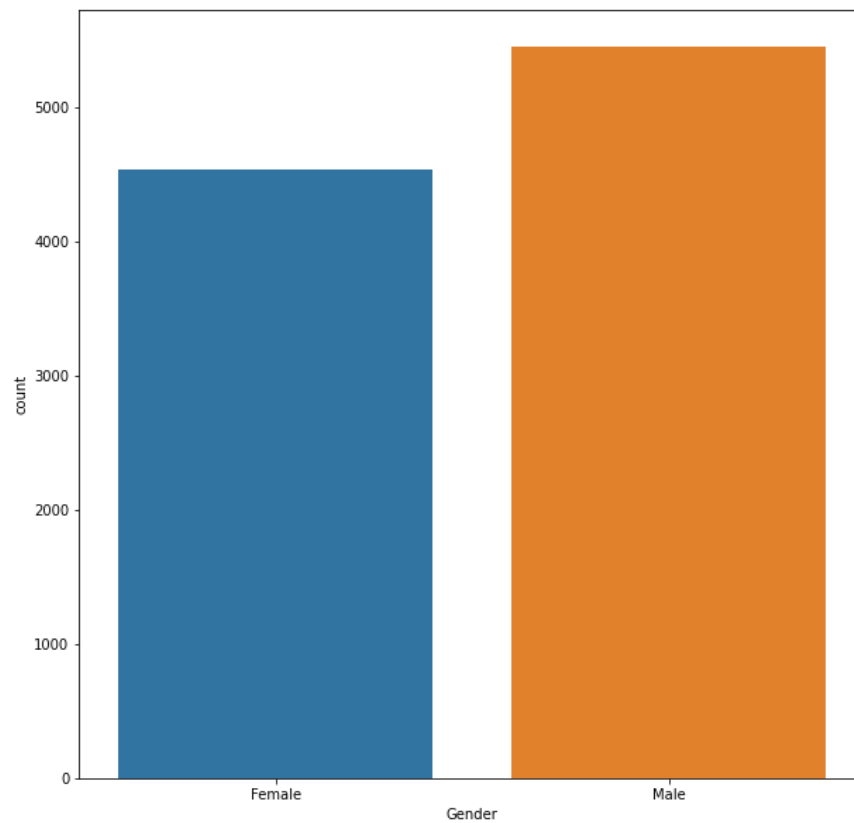
3.2 Using Python - Talia

Correlation Heatmap



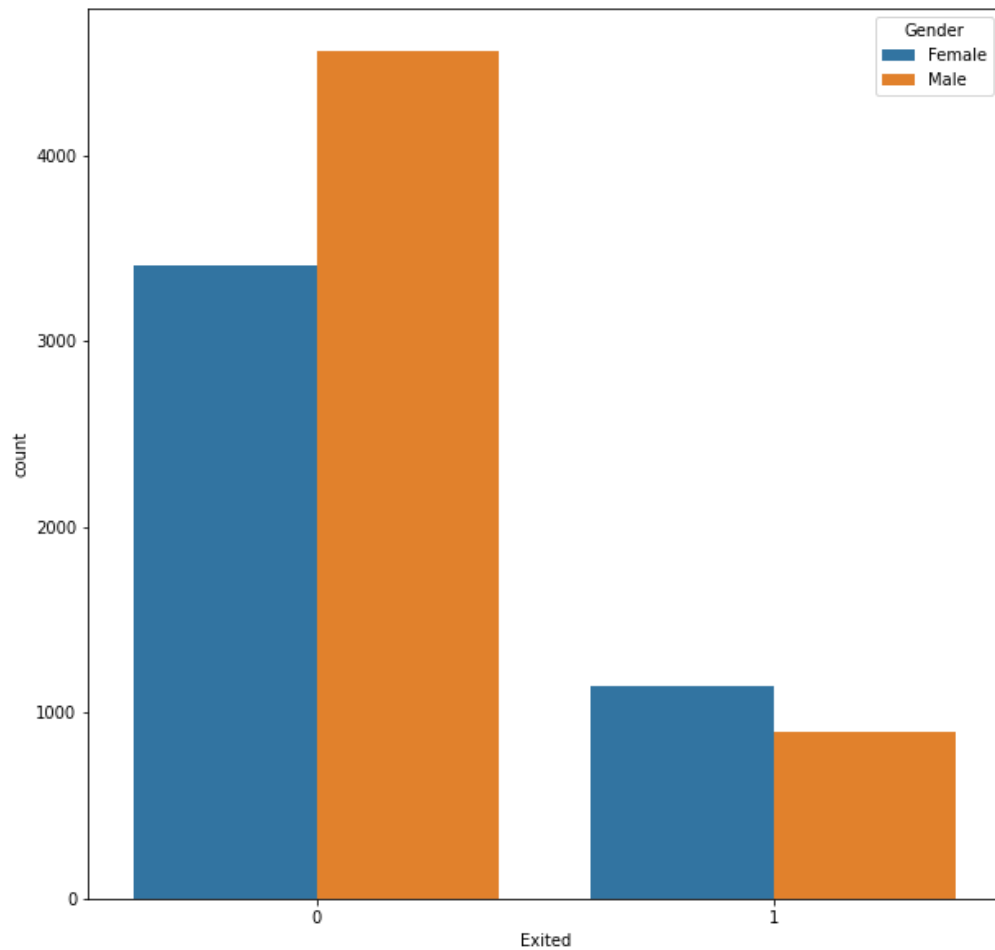
We can see from the heatmap that “Exited” has some negative correlations to “Gender” and “IsActiveMember”, and positive ones with “Age” and “Balance”. We can also see that “NumOfProducts” has a very strong negative correlation to “Balance”.

Gender distribution



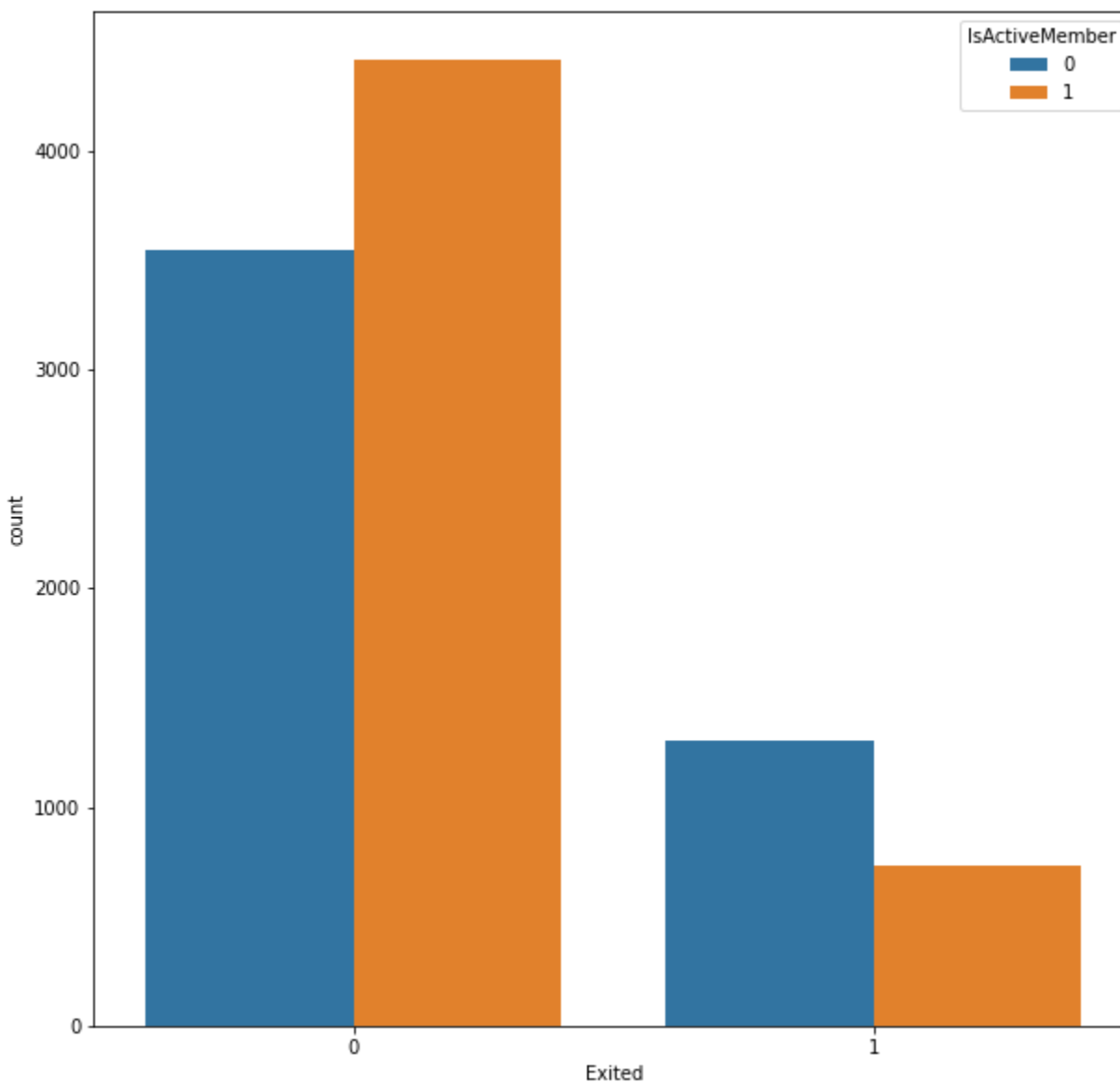
We can see that there are globally, more men as clients than women. This difference is significant enough, but not a huge discrepancy between the genders.

How many people stay or exit by gender



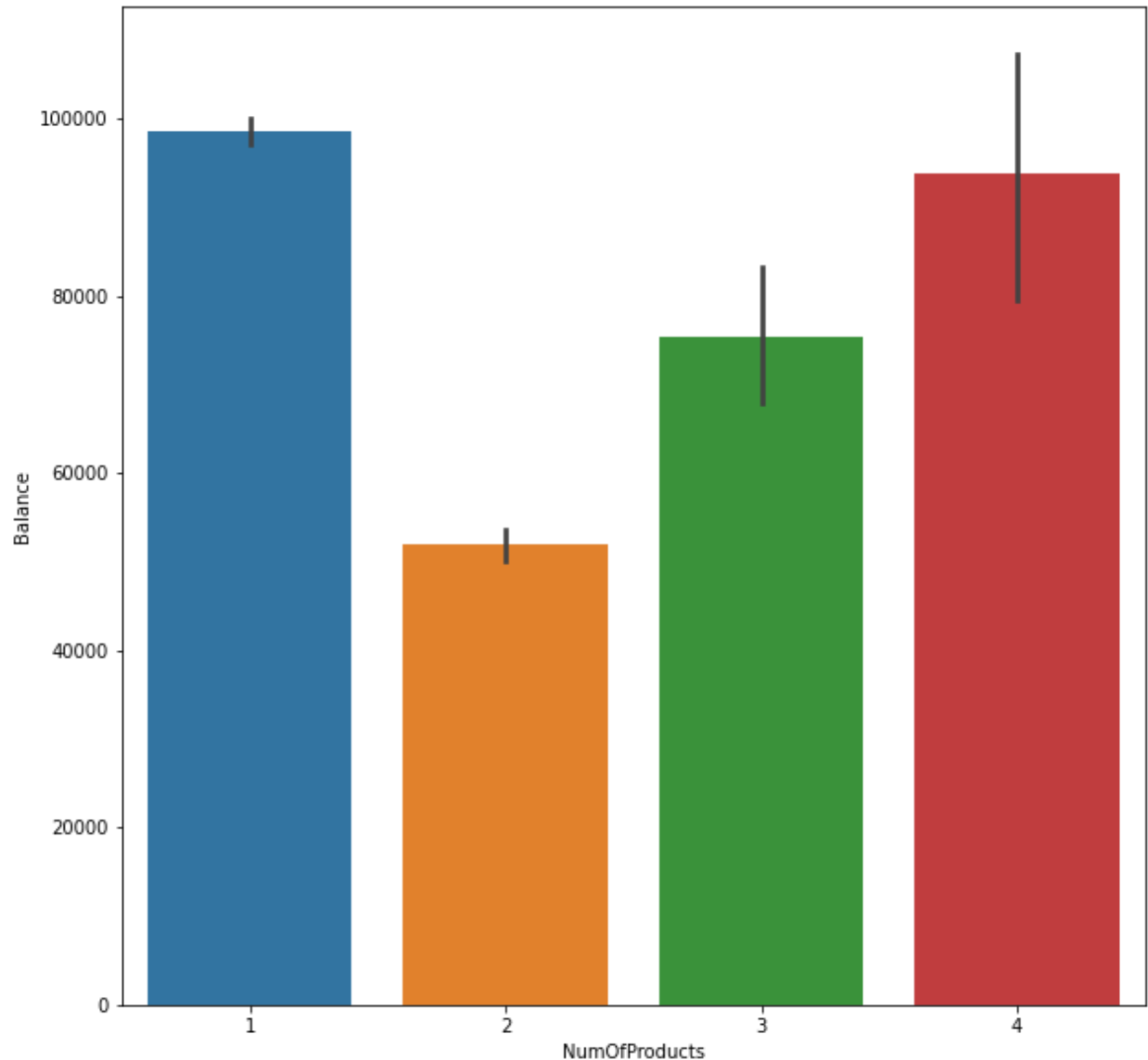
We can see that men tend to stay with the bank more than the women, but we also have to keep in mind that there are more men than women. Still, women seem to tend to leave a bit more than men, when looking at sheer numbers.

How many people exit or stay when they are active members or not



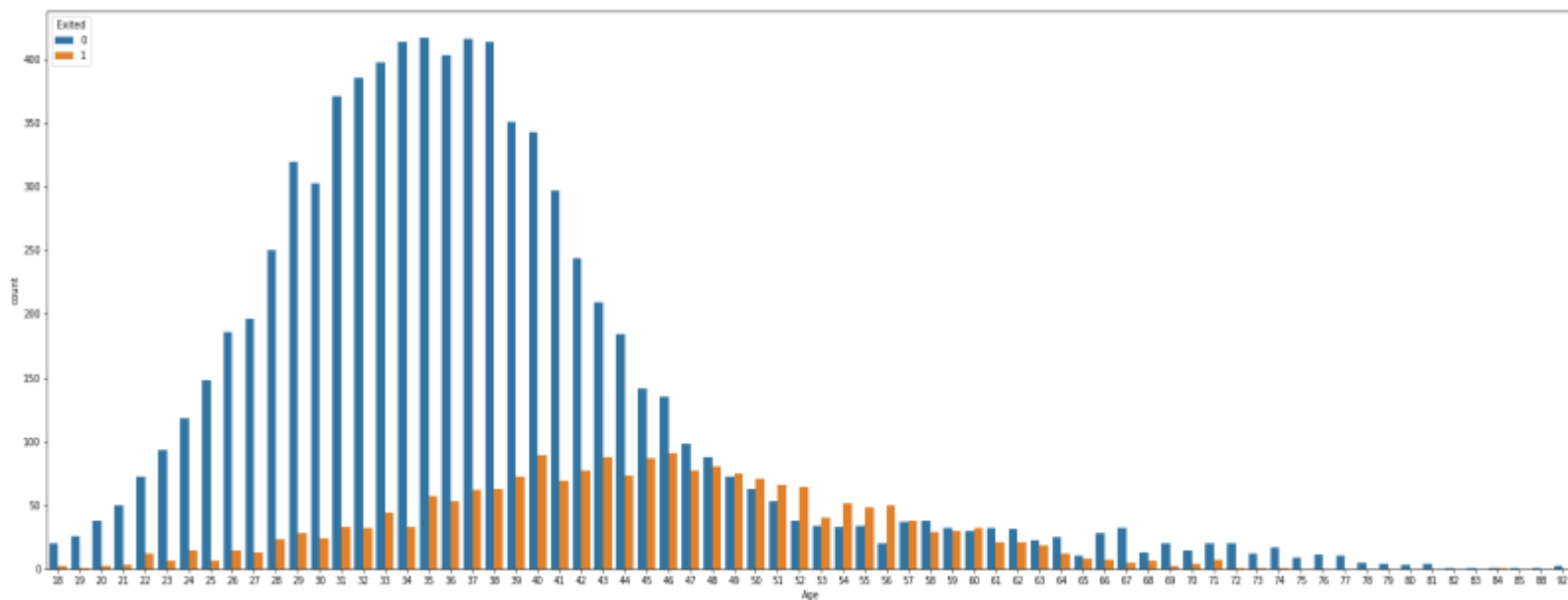
From the graph above, we can see that, in general, the people that leave the most are those who are not active members. From those that stay, active members are higher in number.

Balance and Number of products



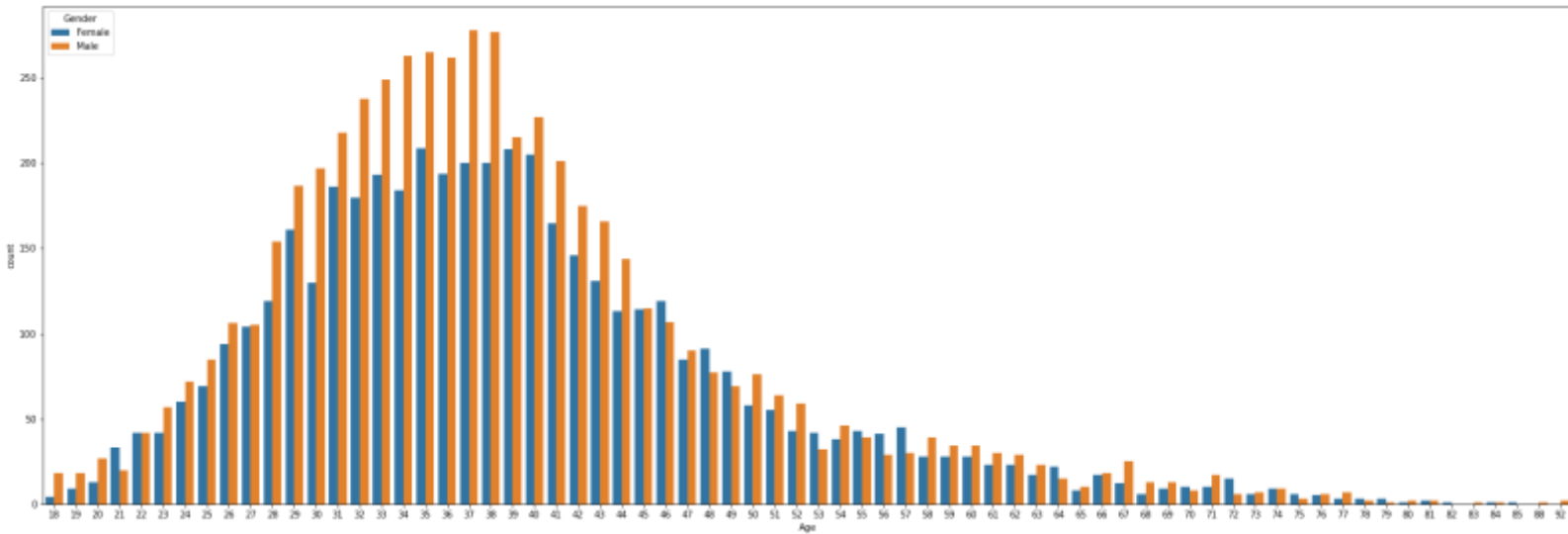
When comparing the number of products with the balance, it is hard to determine the reason behind the before seen strong negative correlation. This visualization might just not be suited for the correlation between these two features. From what we can see in the above graph, the highest balance is associated with having 1 product. After the 1 product, it goes down dramatically and then seems to follow an upward trend, where the higher the products, the higher the balance. The error bars seem to indicate that the highest uncertainty around the estimate is in the 3 products and 4 products.

How many people stay or exit by age



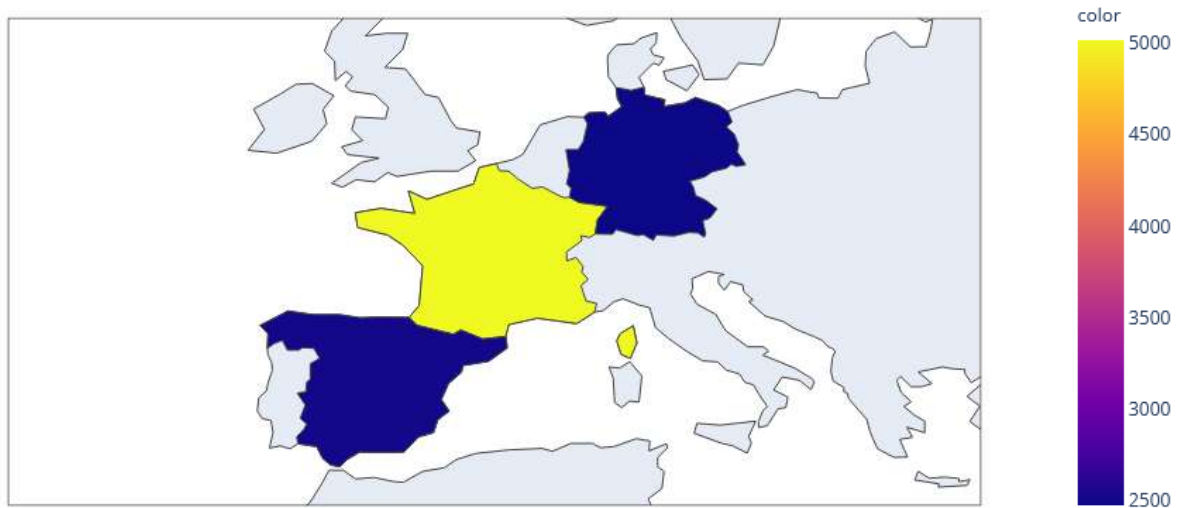
In the graph above we can see how the people that stay (blue bars) and those that leave (orange bars) are distributed by age. We see that the bulk of people that stay are around between 30-44. We can also see that, of those that leave, that the age where people start to tend to leave more than stay, seems to be sitting around 45-65. The very young (below 20) and very old (above 65) do not seem to leave much. Of the people that do stay, the distribution has a right skew and follows a normal distribution. The distribution of those that leave is a centered normal distribution with low variability compared to the distribution of those that stay, much lower slope.

Age distribution by gender



The above graph shows the age distribution by gender. We can see they are very similar but there are more men than women, which we have verified before in a previous graph, so it is not surprising. Both distributions (male and female), follow the same trend. Both have a right skew and peak around between the 30-40 years of age. Both follow a normal distribution.

Number of Customers per Country



This map is identical to the one done in Tableau. Since the one in Tableau is much more visually appealing and descriptive. This map is just to represent the work done in python, but the conclusions are the exact same as the ones described before in the Tableau map.

4. Deep Learning Modelling with ANN

4.1 Individual Experiments - Faith

First, the data was checked over to see what format the columns values had

```
[ ] df = pd.read_csv("../content/Churn_Modelling.csv")
df.head(10)
```

	RowNumber	CustomerId	Surname	Creditscore	Geography	Gender	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSalary	Exited
0	1	15634602	Hargrave	619	France	Female	42	2	0.00	1	1	1	101348.88	1
1	2	15647311	Hill	608	Spain	Female	41	1	83807.66	1	0	1	112542.58	0
2	3	15619904	Onio	502	France	Female	42	8	159660.80	3	1	0	113631.57	1
3	4	15701354	Boni	699	France	Female	39	1	0.00	2	0	0	93826.63	0
4	5	15737888	Mitchell	800	Spain	Female	43	2	125510.82	1	1	1	79084.10	0

checking how many persons that stayed and exited. After checking how many persons stayed and exited, I discovered the number of rows with stayed is three times as large as the number of rows with exited. Thus, there is a bias in the dataset. This bias may influence the results after training. To solve these issues,

```
[ ] len(df[df['Exited'] == 0])#stayed
```

7963

```
[ ] len(df[df['Exited'] == 1])#exited
```

2037

```
[ ] exited_df= len(df[df['Exited'] == 1])#exited
```

The next step below, I created a training set with 50% stayed and 50% exited. To avoid some data imbalance when running the model.

```
[ ] exited_df = (df[df['Exited'] == 1]).sample(2037)
len(exited_df[exited_df['Exited'] == 1])
```

2037

```
[ ] stayed_df = (df[df['Exited'] == 0]).sample(2037)
len(stayed_df[stayed_df['Exited'] == 0])
```

2037

After balancing the datasets, I decided to combine them again into one table. As seen below, the exited datasets are at the top while the stayed datasets are at the bottom. Again this could cause some bias during training.

```
combined_df = pd.concat([exited_df, stayed_df], axis=0)
combined_df.head()
```

	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSalary	Exited
3877	9876	15572182	Onwuamaeze	505	Germany	Female	33	3	106505.77	3	1	0	45445.78	1
4153	4154	15664001	Riedle	695	Germany	Female	53	8	95231.91	1	0	0	70140.80	1
588	595	15637476	Alexandrova	683	Germany	Female	57	5	152446.69	1	0	0	9221.78	1
5067	5058	15605386	Wang	753	Germany	Female	45	3	111512.73	3	1	0	159576.75	1
1066	1057	15688963	Ingram	731	France	Female	52	10	0.00	1	1	1	24996.75	1

To solve the issues above, I made a random state to help mix the data together.

```
[ ] randomize = combined_df.sample(frac=1).reset_index(drop=True)
[ ] randomize.head()
```

	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSalary	Exited
0	8004	15595426	Madukwe	603	Spain	Male	57	6	105000.85	2	1	1	87412.24	1
1	7945	15681476	Foveaux	520	France	Female	39	1	73493.17	1	0	1	109626.13	1
2	4900	15695852	Hsu	809	France	Female	32	9	152122.84	1	1	1	54277.45	1
3	5577	15635964	Eve	568	Germany	Male	55	4	120100.41	1	1	0	107363.16	1
4	2348	15745700	Richter	589	Germany	Male	55	7	119961.46	1	1	0	65156.83	1

To further prepare the dataset for training, I applied label encoding to the CustomerId, Surname, Geography and Gender to get categories instead. I also applied one-hot encoding to the exited. Then, I used the MinMaxScaler because this estimator scales and translates each feature individually to get the values between zero and one.

```

1. from sklearn.preprocessing import LabelEncoder
X[['CustomerId', 'Surname', 'Geography', 'Gender']] = X[['CustomerId', 'Surname', 'Geography', 'Gender']].apply(LabelEncoder().fit_transform)

2. from sklearn import preprocessing
min_max_scaler = preprocessing.MinMaxScaler()
x_scale = min_max_scaler.fit_transform(X) #To enable the neural network make a better prediction
x_scale

array([[0.72704841, 0.44389665, 0.03563365, ..., 1., 0., 0.],
       [0.2361483 , 0.32850479, 0.6598755 , ..., 1., 1., 0.],
       [0.15763153, 0.32850479, 0.6598755 , ..., 1., 1., 0.],
       [0.30356216, 0.32850479, 0.6598755 , ..., 1., 1., 0.],
       [0.3964793 , 0.40373188, 0.80531975, ..., 1., 1., 0.],
       [0.27890253, 0.32850479, 0.6598755 , ..., 1., 1., 0.],
       ...,
       [0.19113623, 0.09894427, 0.81494058, ..., 1., 0., 0.],
       [0.02978623, 0.32850479, 0.6598755 , ..., 1., 1., 0.],
       [0.49118824, 0.51043457, 0.99151104, ..., 1., 1., 0.],
       [0.24581929, 0.32850479, 0.6598755 , ..., 1., 1., 0.],
       [0.03330646, 0.65971029, 0.49122807, ..., 1., 0., 0.],
       [0.88906112]])

```

After the preprocessing, I split the data into train and test sets.

```

X_train, X_test, y_train, y_test = train_test_split(x_scale, y, test_size=0.2, random_state = 0)

```

Here, I created a sequential model by passing a list of layer instances to the constructor. also specifying the input shape to help the model know what input shape it should expect. Relu function was chosen because it does not activate all the neurons at the same time. This activation was considered to avoid dead neurons during the backpropagation process, the weights and biases.

According to Keras documentation, Softmax converts a real vector to a vector of categorical probabilities. The elements of the output vector are in range (0, 1) and sum to 1. Each vector is handled independently. The axis argument sets which axis of the input the function is applied along. Softmax is often used as the activation for the last layer of a classification network because the result could be interpreted as a probability distribution.

```

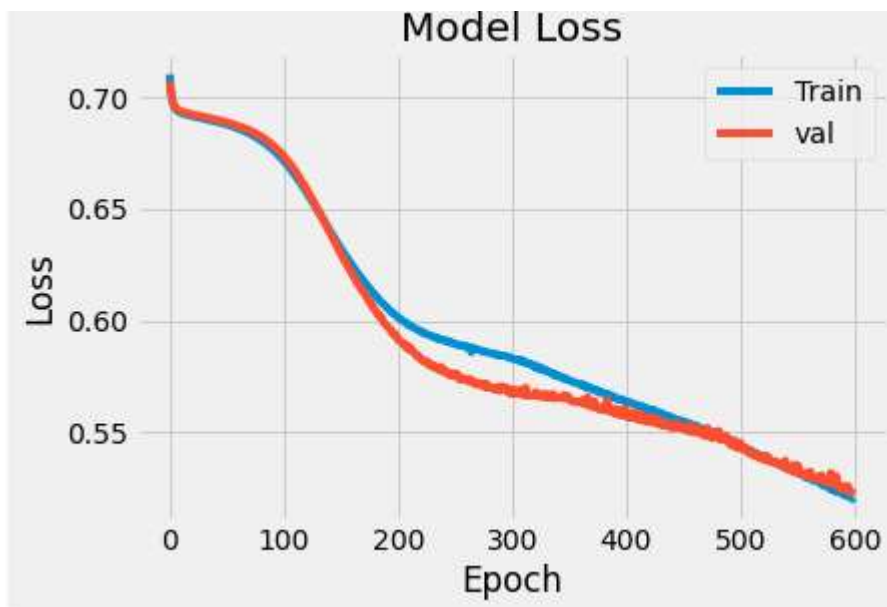
[102] #Building the model
model = Sequential([
    Dense(12, activation='relu', input_shape= (13,)),
    Dense(15, activation='relu'),
    Dense(2, activation='softmax')
])

```

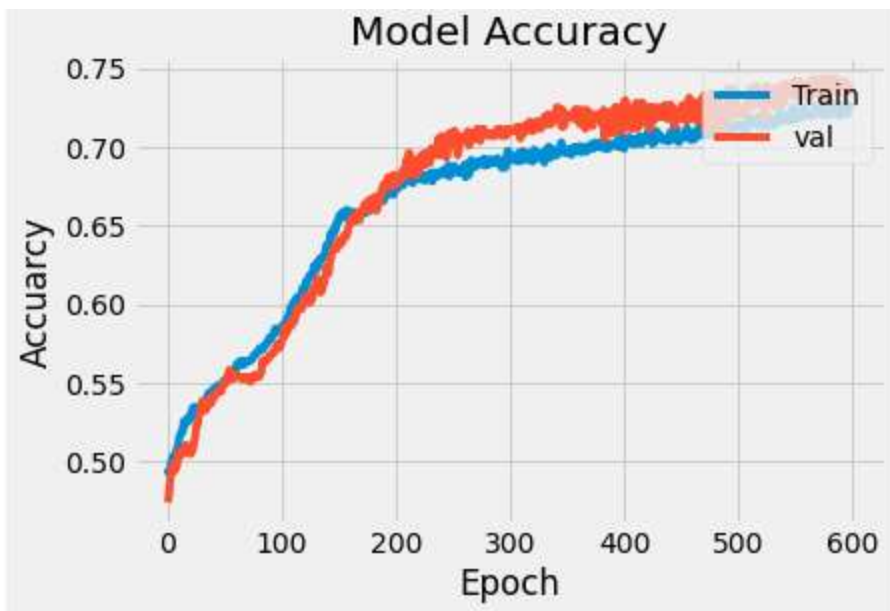
Compiling the model, gradient descent optimizer was used because it's computational efficient and it produces a stable error gradient and a stable convergence. Binary cross- entropy loss because it sets up a binary classification problem with two claC

```
model.compile(  
    optimizer = 'sgd',  
    loss = 'binary_crossentropy',  
    metrics = ['accuracy']  
)
```

After fitting the model with 600 epoch and a validation split of 20 percent, The model accuracy was about 73% as seen below. The loss curve during training and val lost here indicate a low learning rate . For the model accuracy, Here it's shows that there is a little overfitting in the training accuracy. This is a good sign, it shows the model was able to produce a good result after training for a long time.



rate.



4.2 Individual Experiments - Talia

First, the data was checked over to see what format the column values had.

RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSalary	Exited	
0	1	15634602	Hargrave	519	France	Female	42	2	0.00	1	1	1	101348.88	1
1	2	15647311	Hill	606	Spain	Female	41	1	83807.86	1	0	1	112542.58	0
2	3	15619304	Onio	502	France	Female	42	8	159660.80	3	1	0	113931.57	1
3	4	15701354	Boni	699	France	Female	39	1	0.00	2	0	0	93826.63	0
4	5	15737888	Mitchel	850	Spain	Female	43	2	125510.82	1	1	1	79084.10	0

As can be seen, the first three columns are not very interesting to be used as features. Some of the columns have string values. All this data needs to be prepared and normalized.

The amount of exited and not exited was also checked, but at this point the decision was to try out the model without making considerable changes to the data, such as downsampling for example.

CustomerId	
Exited	
0	7963
1	2037

Then, the data needed to be prepared. First, all string columns need to be encoded with integer values.

RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSalary	Exited
0	1	15634602	Hargrave	0	0	42	2	0.00	1	1	1	101348.88	1
1	2	15647311	Hill	2	0	41	1	83807.86	1	0	1	112542.58	0
2	3	15619304	Onio	0	0	42	8	159660.80	3	1	0	113931.57	1
3	4	15701354	Boni	0	0	39	1	0.00	2	0	0	93826.63	0
4	5	15737888	Mitchell	2	0	43	2	125510.82	1	1	1	79084.10	0

Now, we can exclude the columns we do not want and the dataset can be split into train and test. The values are then standardized.


```

X = df_copy.iloc[:, 3:13].values
y = df_copy.iloc[:, 13].values

# Splitting the dataset into the Training set and Test set
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2, random_state = 0)

# Feature Scaling
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
X_train = sc.fit_transform(X_train)
X_test = sc.transform(X_test)

```

The network configuration chosen was rather simple and can be seen below:

Model: "sequential"

Layer (type)	Output Shape	Param #
dense (Dense)	(None, 10)	110
dropout (Dropout)	(None, 10)	0
dense_1 (Dense)	(None, 10)	110
dropout_1 (Dropout)	(None, 10)	0
dense_2 (Dense)	(None, 1)	11
Total params: 231		
Trainable params: 231		
Non-trainable params: 0		

The idea was to have a small network with one input layer, one hidden layer and a logit output layer. A few drop out layers were also added since, according to Keras documentation, these help overcome potential overfitting. Swish was chosen as the activation function for this first model. This is an activation function researched by Google that had better performance than ReLu.

The idea to choose logits is because our output is either 0 or 1, so one neuron theoretically would be enough to determine this outcome.

To help with the learning rate, a callback was used to help adjust the learning rate when a metric is not improving. The model was compiled with the Adam algorithm as the optimizer of choice, which is a stochastic gradient descent method that is based on adaptive estimation of first-order and second-order moments, also being the most common algorithm to train neural networks with. For the loss function, binary cross-entropy was used, which determines the loss

between the predicted labels and true ones, with "from_logits=True" passed to it, since logits are being used.

The test set was used as the validation, which at the time seemed to make sense but later found out that best practice is to never let the model see the test set until evaluating or predicting.

The accuracy of the model on the test set was 86.25%. To see how accuracy and loss progressed throughout training, two graphs were plotted respectively.

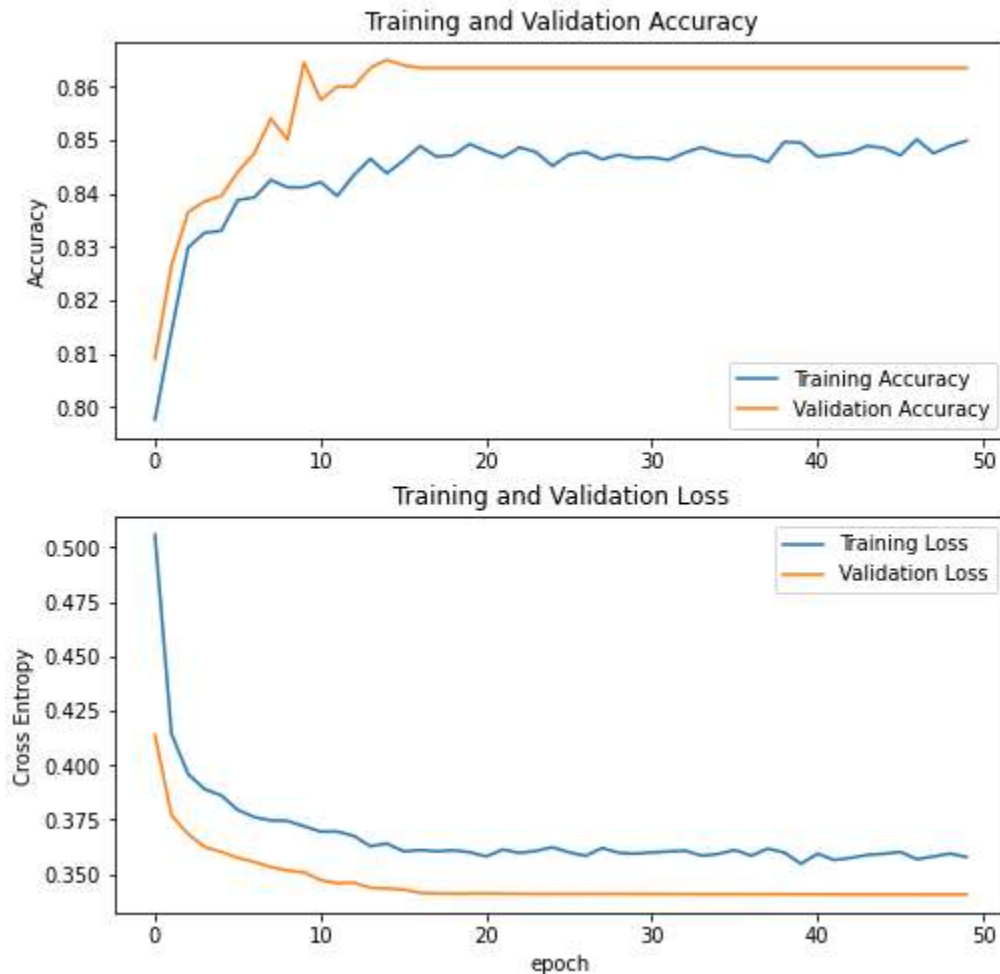


As can be seen from the graphs above, there is a big discrepancy between the lines for validation and training. This could be due to the test set being used as validation, misrepresentation of the data between the train and validation or even due to the fact that there is a known data imbalance.

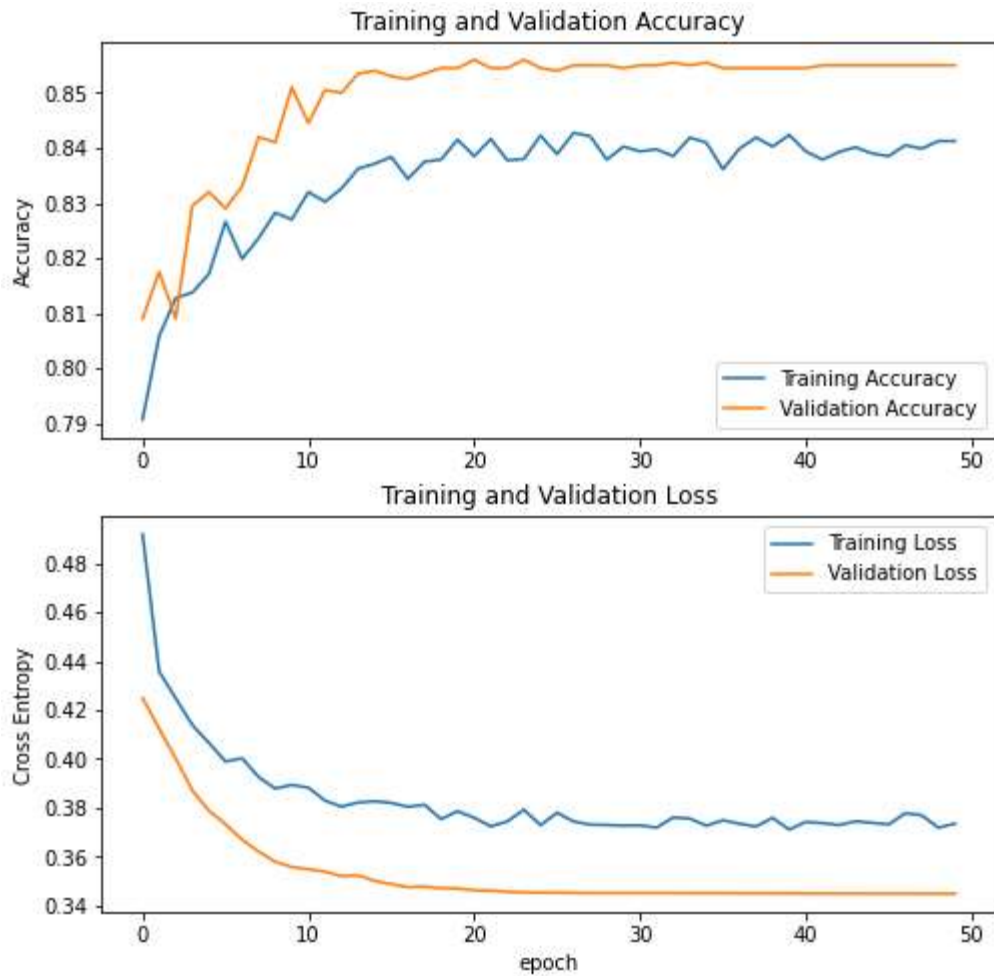
To further experiment and see what the effect of different activation functions have in the hidden layers, Tanh (hyperbolic tangent activation function) and GeLu (Gaussian error linear unit) were

also tried out without any other changes. GeLu is part of tf-nightly and is a function that is also reported to perform better than ReLu.

The model that ran with GeLu had an accuracy on the test set of 86.35%. The graphs of the accuracy and loss can be seen below:



The model that ran with Tanh had an accuracy on the test set of 85.5%, the worst out of the three activation functions but not by a big margin. The graphs of the accuracy and loss can be seen below:



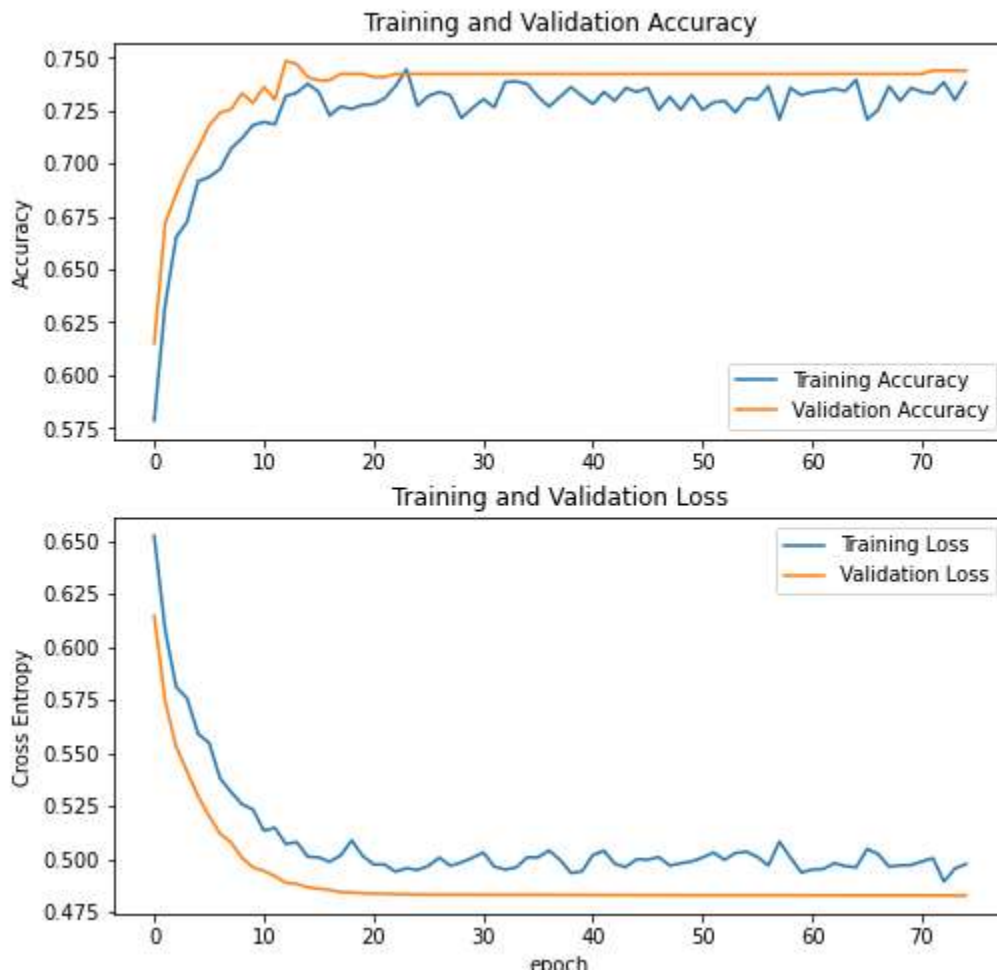
All graphs for the models with the different functions are similar and manifest the same issues, which means further preparation of the data was needed, as well as a proper validation set.

4.3 Joint Experiments

For the joined efforts, we decided to take the best of the individual experiments and do some further investigating as well. To prepare the data, we combined the methods from encoding the string data and standardizing, but also dealt with the data imbalance by using random downsampling of the records of the stayed (value = 0), which were much higher than those that exited. The data was randomized to ensure that there would be no issues when splitting. Then the data was split into train, validation and test sets.

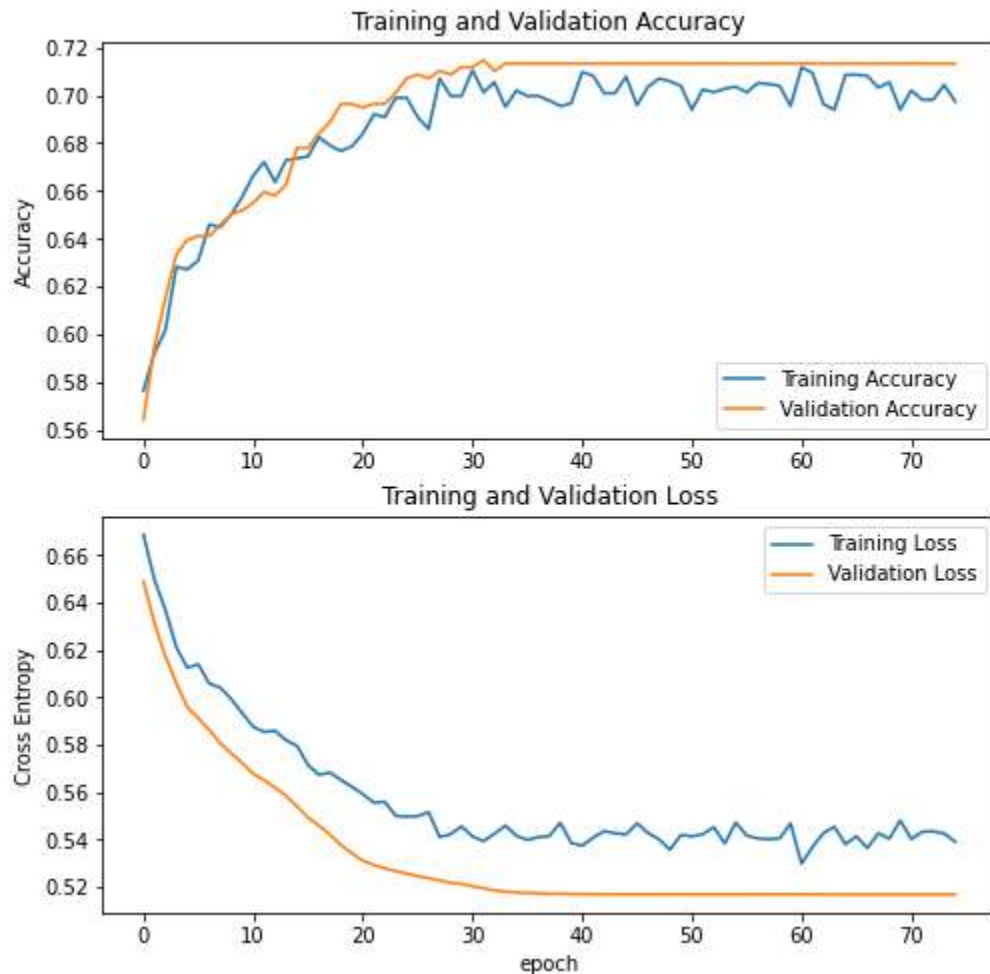
We decided to use one model structure but compare the performance when using different optimizers and different output layers. We also decided to go with Swish, since it had the best performance out of our individual experiments, and to make use of the learning rate adjuster callback. The models were trained with a batch size of 10 and for 75 epochs.

The first model used Adam as the optimizer. The accuracy on the test set was 75.83% and the graphs for the accuracy and loss can be seen below:



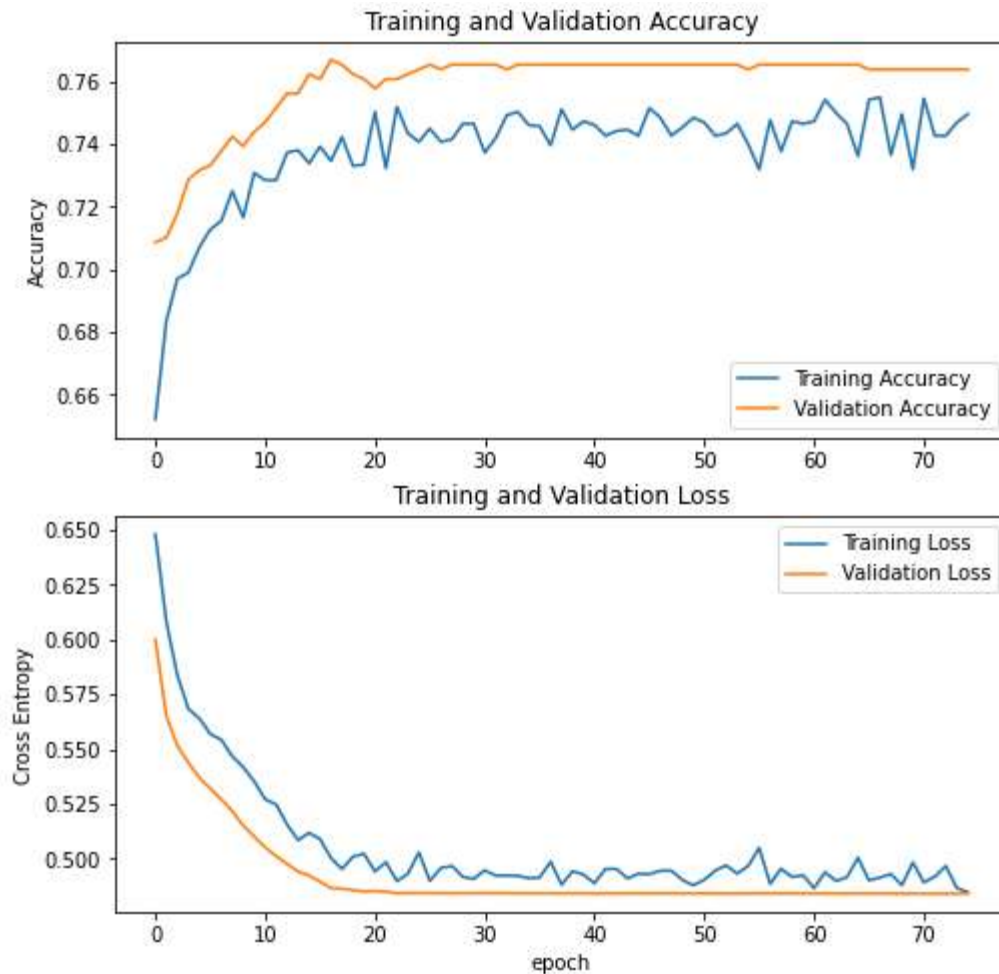
We can see that the validation accuracy is higher than the training accuracy, this could indicate that the validation set is “easier” to predict than the training set, or a deeper data imbalance issue.

The next model used SGD as the optimizer. The accuracy on the test set was 71.9% and the graphs for the accuracy and loss can be seen below:



We can see that the overall accuracy is lower than when using Adam as the optimizer. So knowing that Adam performs better, we decided to try a different output layer, one with two neurons and using Softmax as the activation function. The labels needed to be one-hot encoded so the output would match.

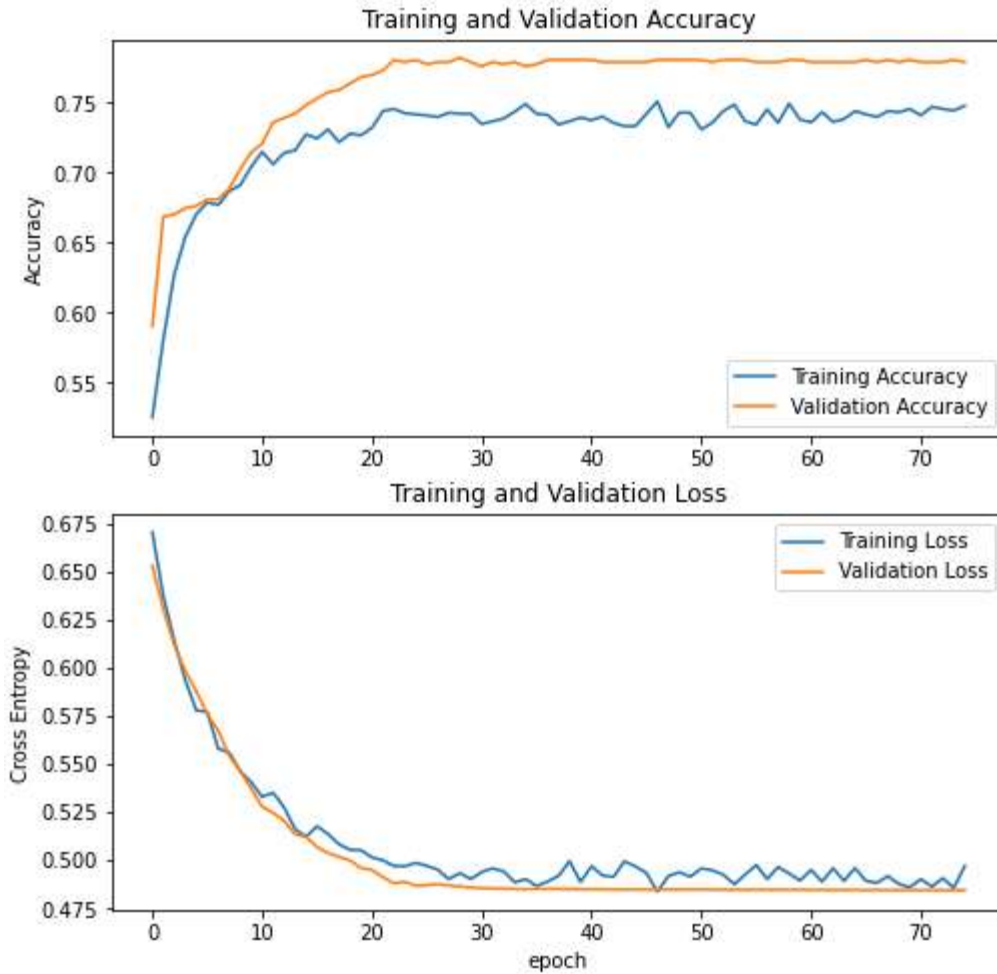
The accuracy on the test set was 74.48% and the accuracy and loss graphs can be seen below:



We can see that again the graphs look similar to the previous.

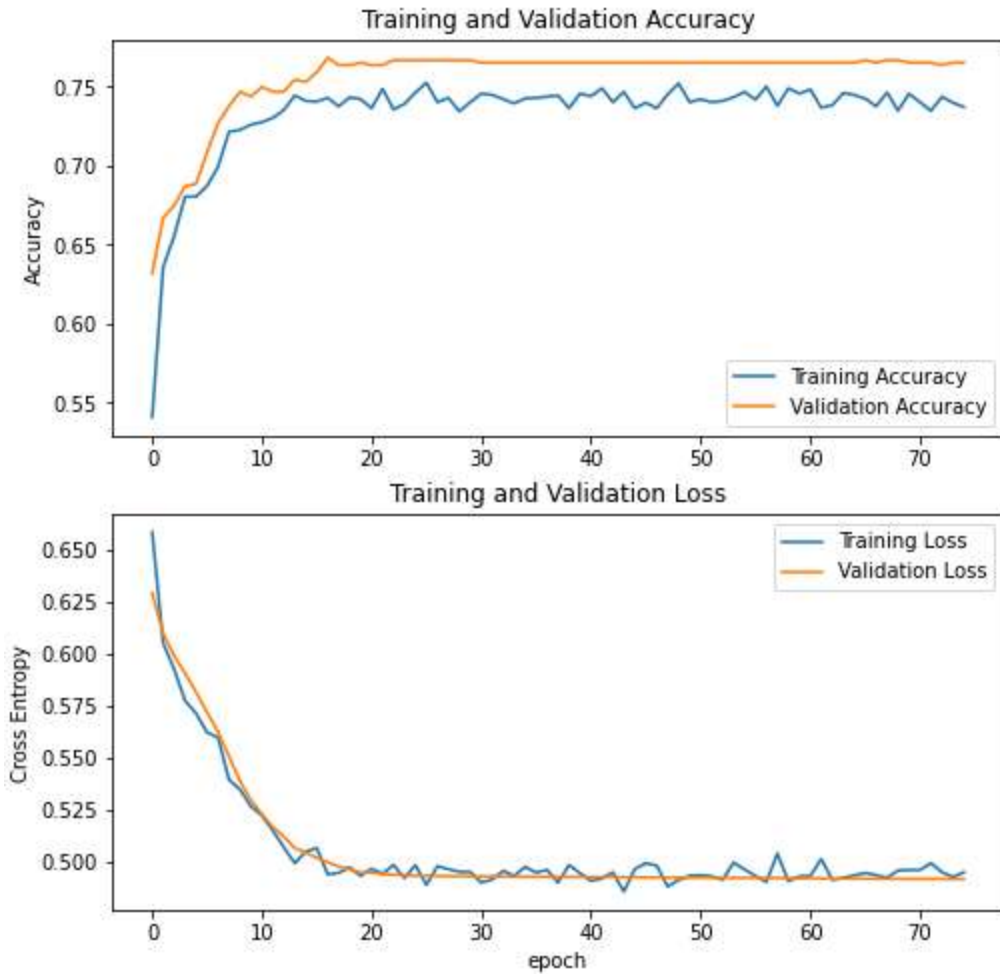
Because features such as Gender, Age and Geography can introduce bias in models, as well as can be discriminating when applied to real life situations, posing ethical concerns, we decided to take the model with Adam optimizer and the logits output layer, which was the combination with the best accuracy overall, and experiment training without each of these features, first individually and then all three together.

For the model trained without the Gender feature, the accuracy on the test set was 74.36%, and the accuracy and loss graphs are as seen below:



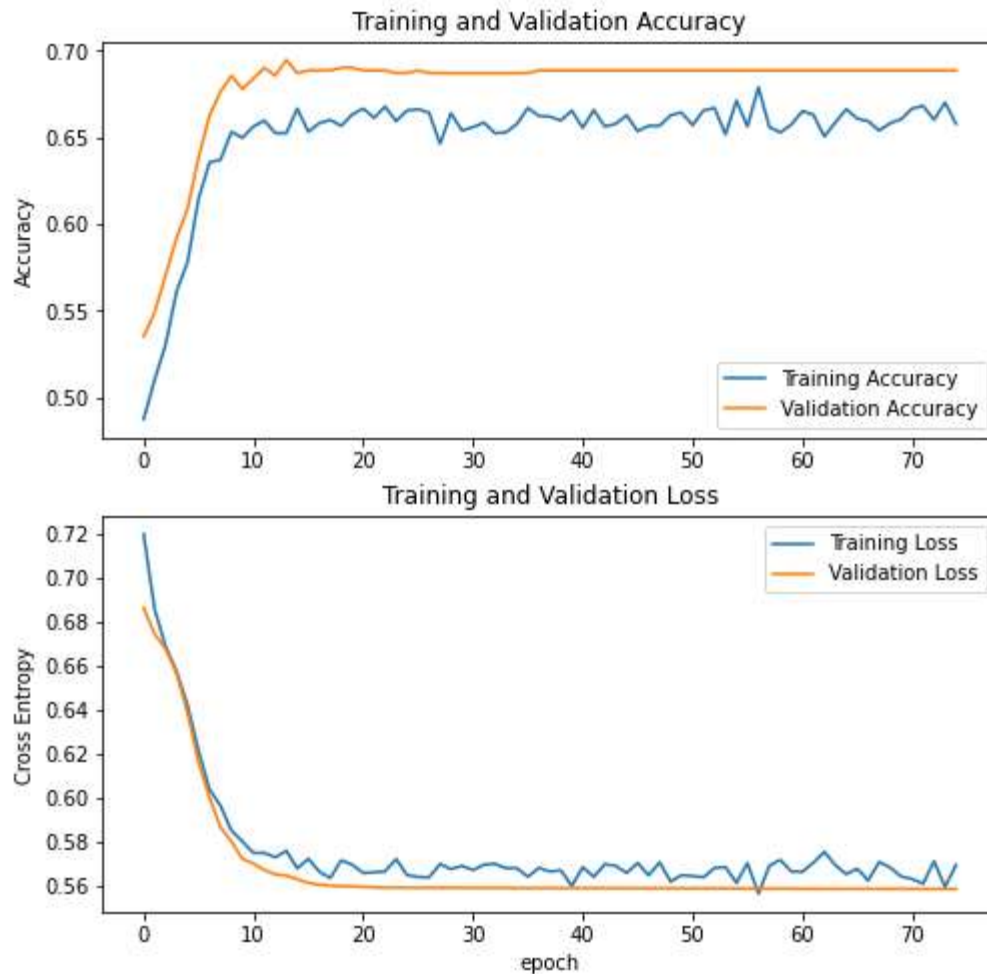
We can see that the accuracy did lower from the original 75.83% but not considerably so, which could mean that potentially, the model could be used without this feature and still perform decently.

For the model trained without the Geography feature, the accuracy on the test set was 73.37%, and the accuracy and loss graphs are as seen below:



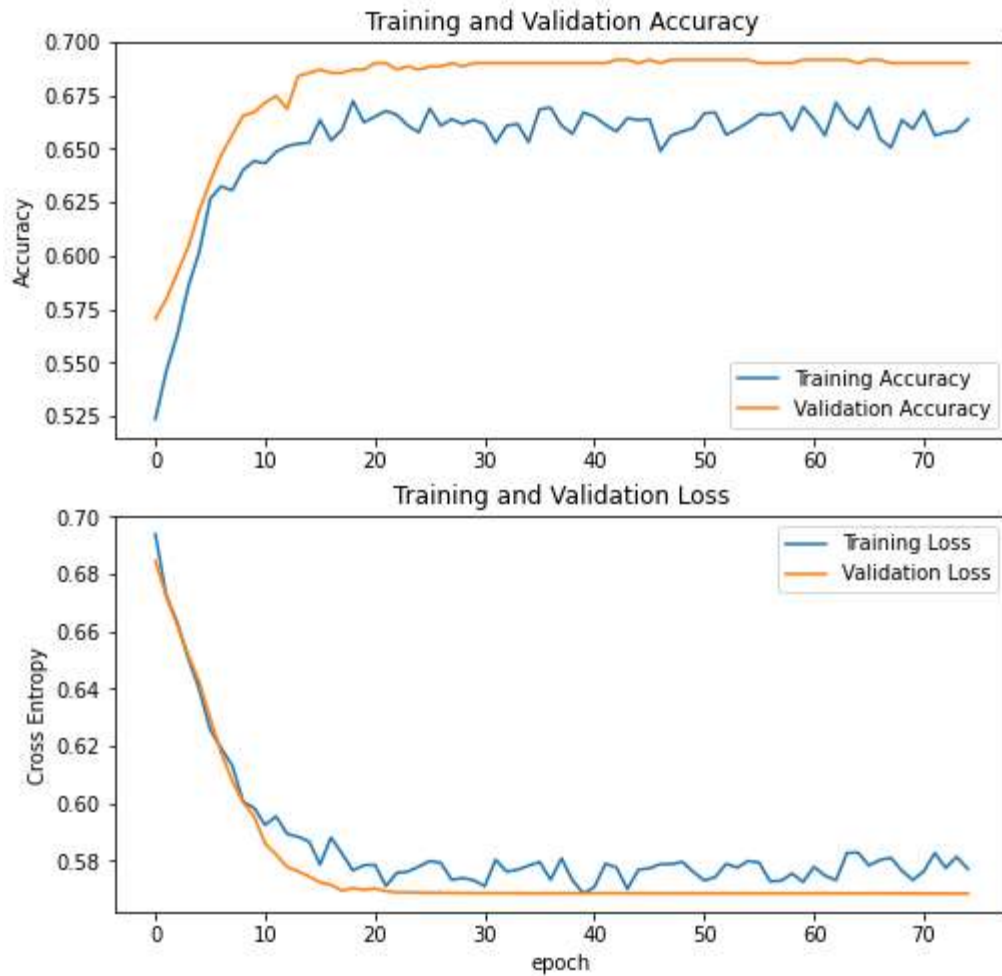
Again we see that the accuracy lowers, but not by a large margin. This could also be because of the three countries used in Geography, France had double the amount of records than either Spain or Germany, but overall there is potentially the possibility of dropping this feature without major consequence to the model's performance.

For the model trained without the Age feature, the accuracy on the test set was 69.69%, and the accuracy and loss graphs are as seen below:



In this case, age does seem to have a significant impact on the models predictive performance. The significant reduction in the accuracy potentially tells us that this feature might not be possible to be dropped if we want a more accurate model.







Finally, when dropping all three features, the accuracy on the test set was 69.33%, and the accuracy and loss graphs are as seen below:



The accuracy is not so much different from the one where we had dropped just Age, which could mean that in fact Age is the feature that has the most impact on the performance.

5. Ethical Consideration

In this chapter you will read about the ethical part of this challenge. To do this we have made use of Technology Impact Cycle Tool (TICT) framework's Quick Scan template.

 <p>NAME: Customer Churn Rate Predictor Model</p> <p>DATE: 18/12/2020</p> <p>DESCRIPTION OF TECHNOLOGY: Machine learning model that can predict the customer churn rate for a bank based on features such as tenure, credit score, member activity.</p>		 <p>SUSTAINABILITY: It is not known at this time what the sustainability impact would be from this model.</p>
 <p>DATA: The data has some imbalances when it comes to the number of records for each feature category and also has features that could introduce bias and ethical concerns, such as Age, Geography and Gender. This data imbalance is taken into account for the model.</p>	 <p>TRANSPARENCY: Because this model potentially uses sensitive data pertaining to bank services and human data we will consider basic inherent values in humans. This includes truth, honesty, loyalty. We want to make sure users know what their data is used for and how it is being used.</p>	 <p>HUMAN VALUES: With the use of this model and the data, the bank's perception of users based on things like their age and gender combined with their credit score, among other things, could alter the bank's perception of its clients in a way that does not entirely correspond to the real person's identity and values.</p>
 <p>HATEFUL AND CRIMINAL ACTORS : With this information, if it were to fall into the wrong hands, people could be targeted for their balance values or credit score, as well as the amount of accounts and services being used. It could also be used to track down individual people based on the information stored. The continuous hacking of personal data for different motives is always a huge concern. The bank should mitigate this as much as possible and keep itself updated since hackers always find new tactics to improve according to their selfish motives</p>	 <p>STAKEHOLDERS: Based on the goal, which is to create a model that can predict how likely it is for the customers to leave the bank (close their account) in the near future, therefore calculating the churn rate. The data acquired for this analysis will be beneficial to the company. The features of the dataset is adequate enough to be able to achieve the desired goals.</p>	 <p>IMPACT ON SOCIETY: Looking at the EDA, there is a lot of insight derived through exploratory data analysis, it was possible to gain a common understanding of the data. An example of the insights, Female customers are more likely to churn vs. male customers, but the difference is minimal. Banks can use this information to specifically target their customers who might be in "danger" of leaving.</p>
 <p>FUTURE: The technique of implementing the project and how the data will be used makes it progressive in the sense that it is possible to be able to add data later on after analysis to continuously train the model. The structure/features of the dataset can be easily connected to new sets of data for continuous training and modelling.</p>	 <p>PRIVACY: the bank collects a lot of personal data, specifically tenure, credit score, the age and gender, as well as geography, which could be used against users in the wrong hands. For this reason, it will comply with EU GDPR laws, as well as dutch laws to ensure privacy.</p>	 <p>INCLUSIVITY: To avoid data bias, the model might leave out features such as Gender and Geography to avoid these concerns. The possibility of not making use of Age as a feature could also happen, but this feature seems to be more closely related to the probability of a client leaving, which is crucial information for a bank, so this could be a concern in terms of bias in the future. A bank could, for example, reject the offer of a specific service or credit loan based on this.</p>

6. Conclusion

We can conclude from the Exploratory Data Analysis performed that the executives of the bank can use the insights derived from exploring the datasets to make a decision and it's also important for them to look into the anomalies that were found with the number of products.

Some features were not explored visually because, when looking at a correlation heatmap, there was no significant correlation to the "Exited" feature.

We can also conclude that, although equally as possible in execution, the visualizations done in Tableau are overall visually more appealing and descriptive than those done in Python and are less complicated in implementation.

In regards to the modelling, we can conclude based on our experiments, that using logits for this problems performed slightly better, as well as with the use of the Adam algorithm as optimizer and Swish for the activation function in the hidden and input layers.

We can also conclude that data imbalance dramatically affects model accuracy but also introduces potential biases, since training on more records of a given class will make the model more likely to predict said class, which might not correspond to reality.

We can also conclude that even though all three features that could pose ethical concerns - Gender, Age and Geography - have an impact on the model's performance, Age has the most impact and therefore would be very hard to exclude from a final model. Gender and Geography could potentially be dropped if ethical and bias concerns were to arise. Age could still pose an ethical issue since the distribution is not equal, therefore having more records of some ages over others, which could make the model biased towards specific ages.

7. Recommendations

It would be interesting to try out these experiments on a much larger dataset, something of Big Data proportions for example, but also a dataset which has a lot more diversity of features. It would also be interesting to try and balance out the features to make sure that the amount of records used for training sits within a similar scope, to avoid data imbalance issues such as biases.

Lastly, it would be interesting to see if there were other methods we did not investigate which could have brought the accuracy higher. Preferably, a model should have a performance that exceeds "gambling behaviour". This behaviour is proportional to the number of classes. Considering we have either 0 or 1, 50% or less would constitute "gambling behaviour". Our

highest accuracy sits at around 75.83% (including all features), but it would be interesting to see if there was some way to bring it up to at least 90% without introducing biases in the process. It could, however, simply mean that the features present are simply not representative enough to determine the outcomes accurately, hence the need for more and varied data.

References

Websites

1. <https://www.kaggle.com/shubh0799/churn-modelling>
2. <https://scikit-learn.org/stable/modules/classes.html#module-sklearn.preprocessing>
3. <https://keras.io/>
4. <https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.html>
5. <https://missinglink.ai/guides/neural-network-concepts/7-types-neural-network-activation-functions-right/>
6. https://www.tensorflow.org/api_docs/python/tf/keras/activations/gelu
7. <https://machinelearningmastery.com/how-to-control-the-speed-and-stability-of-training-neural-networks-with-gradient-descent-batch-size/>
8. <https://towardsdatascience.com/useful-plots-to-diagnose-your-neural-network-521907fa2f45>
9. <https://machinelearningmastery.com/learning-curves-for-diagnosing-machine-learning-model-performance/>
10. <https://thenextweb.com/contributors/2018/10/27/4-human-caused-biases-machine-learning/>
11. <https://towardsdatascience.com/biases-in-machine-learning-61186da78591>
12. <https://www.tict.io/>

Our Notebooks

- https://colab.research.google.com/drive/1hFM6IRS8yh12i_vLVrQyKbAEe0UBQZNH?usp=sharing - Faith
- https://colab.research.google.com/drive/15KOK-X9_QvckzTAZgWwcJnQBvHtr_J5l?usp=sharing - Talia
- <https://colab.research.google.com/drive/1Y6uxzzzokqWicASVoYhkF-0bUpFX137m?usp=sharing> - Faith & Talia