Nanjing University of Posts & Telecommunications

Report for Big Project 2

(2021 1st Semester)

Course	Software Engineering
Work	Big Project 2
Date	2021.11
School	School of Computer Science
Teacher	Zheng Liu

List of BigTeam:

Name:	ID:	College:	Major:
Alladin Al-Gannam	F18030115	Overseas Education	Computer Science
Anna Akulova	F19030132	Overseas Education	Computer Science

IMPLEMENTING CUSTOMER CLASSIFICATION WITH PYTHON.

OVERVIEW: PRODUCT DESCRIPTION AND PROBLEM STATEMENT.

With the rise of information technology, more and more people are adopting the use of e-commerce, especially recently. All this leads to a huge number of clients who need to be classified to optimize business and trade. To solve this problem, we can use the latest advances in artificial intelligence. With deep learning, we can afford to perform highly accurate calculations in the fields of probability theory and e-commerce.

Originally, we were thinking of using a readily available dataset and running a deep learning model. While this would have been a decent start, We did not think it would be a real case as compared to what we would encounter as a consultant. Alladin brainstormed ideas on how we might help a client that is looking to improve its business model. This made us think of Alladin's brother, Khaled Al-Gannam, who owns a small insurance company, Syria Insurance Service (SIS), that focuses on providing commercial and personal insurance programs to its clients.

SIS is an independent insurance company with an in-depth knowledge of multiple insurance products and carriers. They proactively provide service to their policyholders and present them to their clients. After speaking with Khaled, about the course and the final project, we decided to look at improving customer experience and create a Neural Network (NN) to see whether it would be a viable model compared to other more traditional algorithms and methods.

The goal of this project is to first, validate that a NN model is more powerful in accuracy than other models and two, how we can leverage this information to mitigate customers from leaving and reclaim customers that have left SIS.

DATA: EVALUATE DATA AND CONDUCT EXPLORATORY DATA ANALYSIS.

The dataset received was of SIS customers. As with insurance companies, their data is usually stored in a system that was not made for analysis but rather for accounting purposes. We ended up getting access to the data of 713 customers.

UI: HOW WILL LOOK OUR PROGRAM.

We decided to do our program in two parts: the web server to interpret the graphic result and the python code of our neural network to work with the dataset. As the web server backend, we will use Jupyter Notebook engine, which will allow us to create GUI environment for our python code of the program. The Jupyter Notebook extends the console-based approach to interactive computing in a qualitatively new direction, providing a web-based application suitable for capturing the whole computation process: developing, documenting, and executing code, as well as communicating the results.
The Jupyter Notebook combines two components:

A web application: a browser-based tool for interactive authoring of documents which combine explanatory text, mathematics, computations and their rich media output.

Notebook documents: a representation of all content visible in the web application, including inputs and outputs of the computations, explanatory text, mathematics, images, and rich media representations of objects.

Main UI features of our web application:

- In-browser editing for code, with automatic syntax highlighting, indentation, and tab completion/introspection.
- The ability to execute code from the browser, with the results of computations attached to the code which generated them.
- Displaying the result of computation using rich media representations, such as HTML, LaTeX, PNG, SVG, etc. For example, publication-quality figures rendered by the matplotlib library, can be included inline.
- In-browser editing for rich text using the Markdown markup language, which can provide commentary for the code, is not limited to plain text.
- The ability to easily include mathematical notation within markdown cells using LaTeX, and rendered natively by MathJax.

USE CASES: WHY OUR PROGRAM IS IMPORTANT.
The main use case of our software is the customer classification without regarding of company type. You can simply load the CSV file with the information of your company and the program will do it's work.

Provided Features.

Line of Business/ Non-Premium	Is Multi-Entity	Policy Status	Active Customer	Customer Number	Type of Business	New Business/ Renewal Group	Total Cost
Policy Effective Data	Policy Expiration Data	Customer Address 1	Customer Address 2	Customer City	Customer State	Customer Zip Code	Cancel Date
Cancel Reason	Writing Company	Policy Number	Invoice Month	Invoice Year	First Written Date	Invoice GL Month	

Selected And Newly Created Features.

Customer ID	Gender	Referrer	State	Paid Full Premium Before	Became Cust	Duration As Cust	Accident/ Health (P) Duration
Accident/ Health (P) Amount	Builders Risk (P) Duration	Builders Risk (P) Amount	Dwelling Fire Duration	Dwelling Fire Amount	Earthquake (P) Duration	Earthquake (P) Amount	Flood Duration
Flood Amount	Homeowners Duration	Homeowners Amount	Life (P) Duration	Life (P) Amount	Motorcycle Duration	Motorcycle Amount	Private Passenger Auto Duration
Private Passenger Auto Amount	Umbrella (P) Duration	Umbrella (P) Amount	Total Duration	Total Amount	Still Customer		

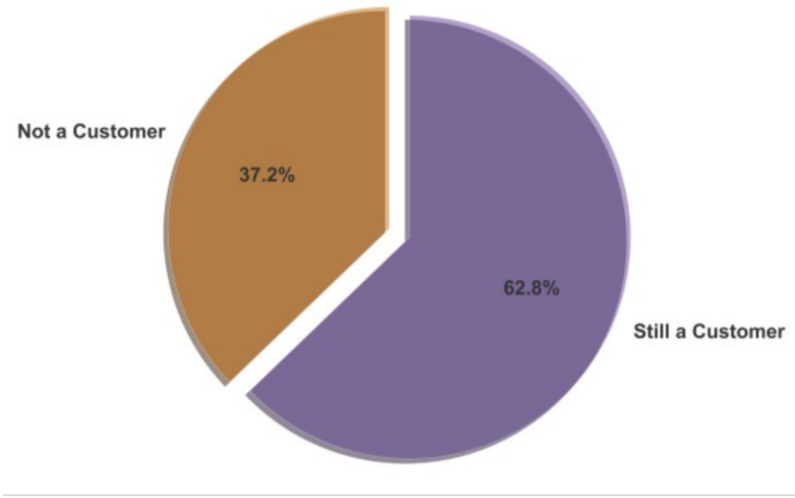
We created new features from the dataset that was provided and formatted the data, so each observation is associated with that customer.

EXPLORATORY DATA ANALYSIS (EDA).

After understanding the features and getting the data formatted in a proper manner, able to conduct EDA. Most of the EDA figures, as well as, Histograms, Correlation Plot, Mean, Standard Deviations, Minimum, Maximum and other summary statistic as part of EDA are provided in the Appendix.

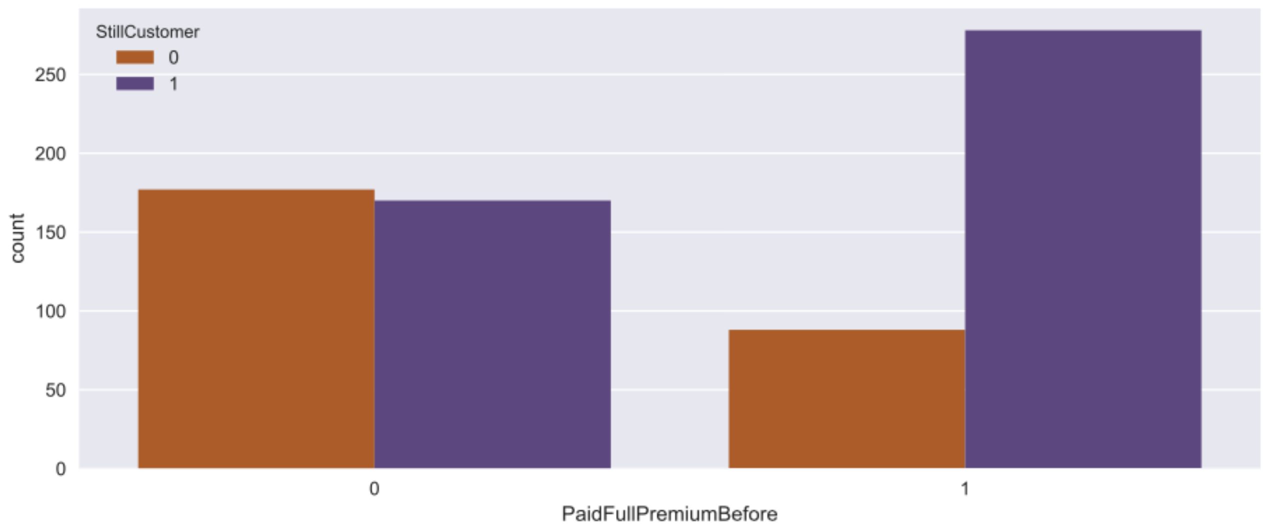
The split of response (target) variable: StillCustomer (0: Not a Customer, 1: Still a Customer). Out of 713 observations, 62.8% (448) are still a customer and 37.2% (265) are no longer a customer.

Proportionate of Customers.



That there is a split among customers who are no longer active and whether they have ever paid full their premium.

Comparing Customers with having Paid Full Premium Before.



Duration of a customer and total value derived from a customer are quite important when looking at ways to improve customer experience and ultimately increase revenue. Hence, us a Kernel Density Estimation (KDE) plot to estimate the Probability Density Function (PDF) of durations in months compared to whether the customer is still active.

KDE between Customers and Duration (months) of Customer.

KDE looking at the gender differences between the customers. So we see negligible differences.

KDE between Gender and Duration (months) of Customer.

BASELINE: SUPERVISED LEARNING.

Unlike other deep learning models that involve computer vision, time series analysis and such, our focus is a classification model with structured data. We created dummy variables for various features, such as, State, and transformed data to normality using Yeo-Johnson transformation.

Comparing 15 different supervised learning models using our dataset.

Index	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa
0	Logistic Regression	0.736	0.803	0.831	0.768	0.797	0.418
1	Extreme Gradient Boosting	0.728	0.803	0.816	0.769	0.790	0.403
2	Ridge Classifier	0.727	0.000	0.842	0.753	0.794	0.390
3	CatBoost Classifier	0.722	0.794	0.809	0.764	0.784	0.394
4	Linear Discriminant Analysis	0.722	0.798	0.816	0.760	0.785	0.391
5	Random Forest Classifier	0.721	0.773	0.767	0.784	0.774	0.408
6	Gradient Boosting Classifier	0.718	0.797	0.812	0.758	0.782	0.381
7	Ada Boost Classifier	0.712	0.776	0.802	0.757	0.777	0.368
8	K Neighbors Classifier	0.709	0.747	0.819	0.747	0.779	0.352
9	Light Gradient Boosting Machine	0.708	0.773	0.791	0.756	0.771	0.364
10	SVM - Linear Kernel	0.699	0.000	0.833	0.731	0.774	0.320
11	Extra Trees Classifier	0.693	0.736	0.772	0.749	0.759	0.335
12	Decision Tree Classifier	0.677	0.645	0.769	0.731	0.748	0.295
13	Naive Bayes	0.616	0.747	0.649	0.758	0.646	0.201
14	Quadratic Discriminant Analysis	0.598	0.661	0.689	0.733	0.634	0.132

BASELINE: SINGLE-LAYER NUERAL NETWORK MODEL.

We created a single-layer artificial neural network model. This model has an input layer with 41 inputs and 1,050 weights. It then outputs to 25 with 520 weights. The output layer is our response feature that has 21 weight parameters. The activation function for the hidden layer is (ReLU). The optimizer is Adam. The ReLU activation function will output the input directly if is positive, otherwise, it will output zero. It has become a popular and often a default activation function for many types of neural networks. We do not change any default parameters for the Adam optimizer (e.g., learning rate, decay rate, etc.) in this single-layer model.

Single-layer Artificial Neural Network (ANN).

```
Model: "sequential_3"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_layer (Dense)          (None, 25)                1050
_________________________________________________________________
hidden_layer (Dense)         (None, 20)                520
_________________________________________________________________
output_layer (Dense)         (None, 1)                 21
=================================================================
Total params: 1,591
Trainable params: 1,591
Non-trainable params: 0
_________________________________________________________________
```

The accuracy of this single-layer ANN is 75%.

OPTIMIZE THE NUERAL NETWORK MODEL.

The grid or random search across the variables (i.e., number of hidden layers, dropout rate, learning rate, input layer nodes, batch size, activation functions, learning rate decay) but it may not yield the best results and may take long for the model to generalize.

- **Brief description of terms:**

1. Hidden layer(s) are between the input and output layers. That perform the function of applying weights to the inputs and directing them through an activation function.
2. Learning rate controls the rate at which our model responds to each error when the weights are updated.
3. Dropout rate prevents overfitting and improves the overall generalization for the model. The dropout rate is between 0 and 0.5.
4. A batch size controls the number of training samples.
5. Epoch refers to the number of passes through the entire training dataset. Also employ a call back function to stop the training.
6. Back-propagation is the essence of training a neural network model. The idea is to fine-tune the weights of the neural network based on the Loss (error rate) that is provided from the previous epoch. By tuning the weights after each epoch, the model tries to increase its generalization.
7. Activation functions determine the output of a neural network. The function is attached to each neuron of the network and determines whether it should be activated based on each neuron’s input that is relevant for the model’s prediction. The activation functions also normalize the output of each neuron between 1 and 0 or between -1 and 1.

- Exponential Linear Unit (ELU) converge cost to zero faster. Different to other activation functions, ELU has an extra alpha constant that is a positive number.
- Sigmoid takes the real value as input and outputs a value between 0 and 1. It is non-linear, continuously differentiable, monotonic, and has a fixed output range.
- Rectified Linear Units (ReLU) is not linear and provides the same benefits as Sigmoid.

- Tanh flattens a real-valued number to the range between -1 and 1. It is non-linear, its output is zero-centered.
- Scaled Exponential Linear Unit (SELU) has an internal normalization component and is faster than external normalization, which means the network converges faster.

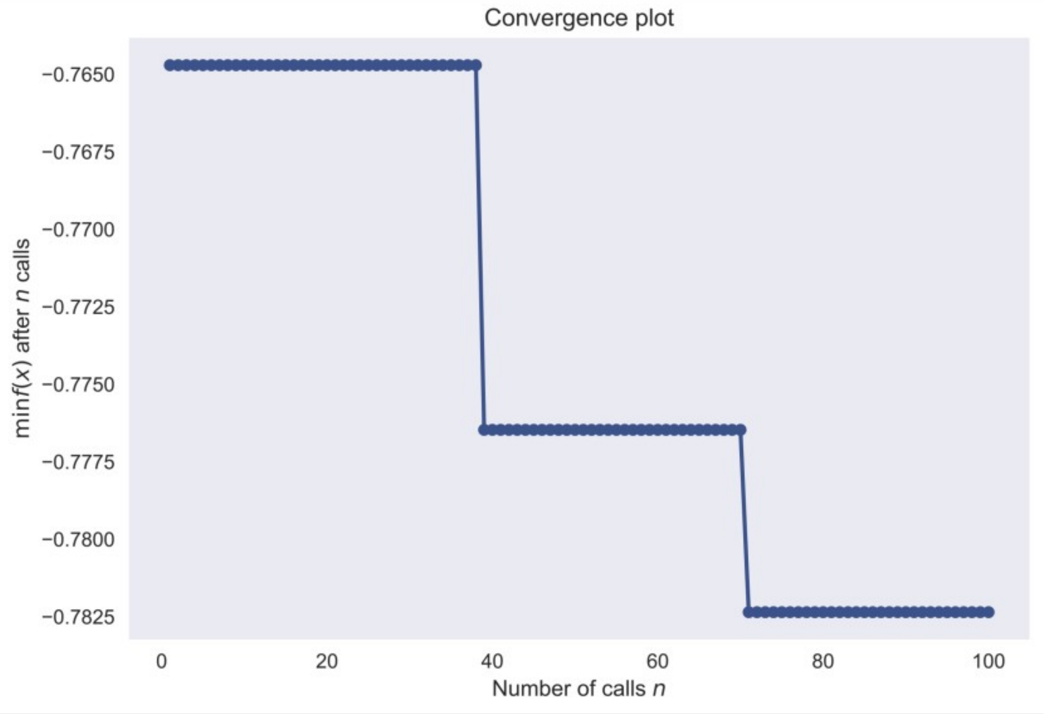
We can leverage Bayesian optimization method which allows us to approximate the function using a Gaussian process. The notion is the function values are assumed to follow a multivariate gaussian and of the function values are given by the Gaussian Process between the parameters. The next set of parameters to evaluate are selected by the acquisition function. We execut iterations (n_calls) using the range of the following parameters.

Parameters and Ranges with Bayesian Optimization using Gaussian Processes:

- Learning rate between 0.01 and 0.0001
- Number of Dense Layers between 1 and 5
- Number of Input Nodes between 1 and 512
- Number of Dense Nodes between 1 and 28
- Activation Functions of ReLU, Sigmoid, ELU, SELU and Tanh
- Batch size between 1 and 256
- Adam Decay rate between 0.1 and 0.000001
- Dropout rate between 0 and 0.5

The convergence plot for the iterations is shown below and a more detailed plot for each parameter is shown in the Appendix.

Convergence Plot.



The top 5 models based on accuracy are listed below and a complete listing of the parameters and results are in the Appendix.
I selected the second model (index: 60), as the first model had an execution time that was much greater than the second model.

Top 5 Optimized Models.

Index	Learning rate	Hidden layers	Input layer nodes	Hidden layers nodes	Activation function	Batch size	Adam learning rate decay	Dropout rate	Accuracy	Time (sec)
99	0.010000	5	512	28	elu	1	0.007296	0.000000	78.235292	43.58
60	0.010000	4	481	13	elu	150	0.006926	0.042091	78.235292	3.37
29	0.003240	5	143	15	relu	19	0.001240	0.068106	78.235292	4.59
75	0.010000	1	445	13	relu	208	0.003714	0.000000	78.235292	1.65
10	0.007348	3	277	15	elu	76	0.003821	0.029607	78.235292	2.9

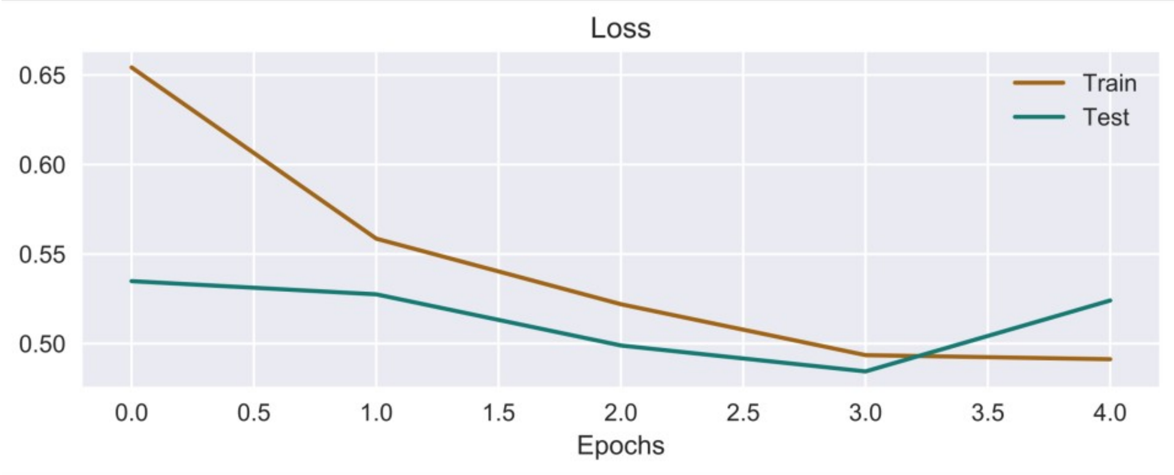
Our selected model (index: 60) is compiled with 4 hidden layers, a learning rate of 0.01, input layer nodes of 481, hidden layer nodes of 13, batch size of 150, learning rate decay of 0.0069, dropout rate of 0.04209 and an activation function of elu.

Optimized Neural Network.

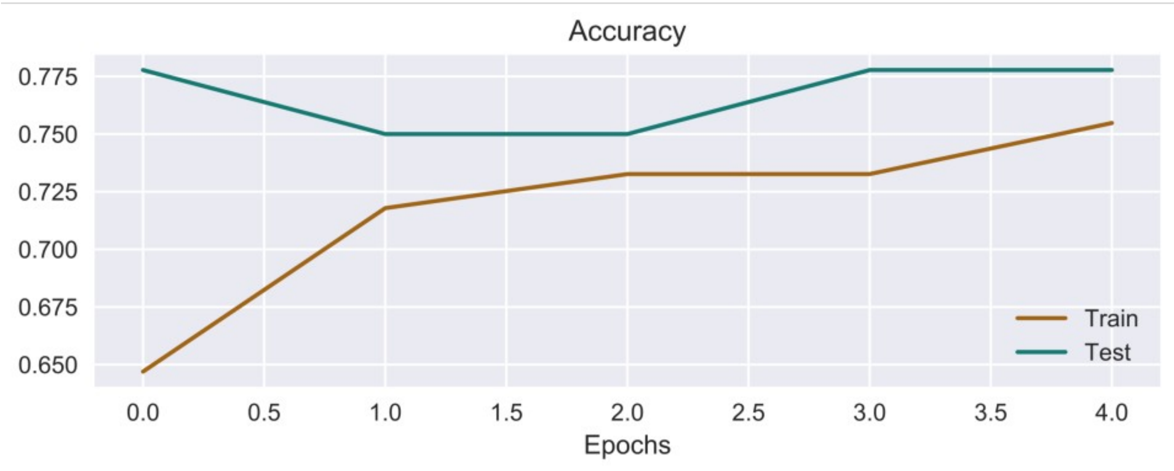
```
Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense_1 (Dense)              (None, 481)               20202
_________________________________________________________________
dropout_1 (Dropout)          (None, 481)               0
_________________________________________________________________
layer_dense_1 (Dense)        (None, 13)                6266
_________________________________________________________________
dropout_2 (Dropout)          (None, 13)                0
_________________________________________________________________
layer_dense_2 (Dense)        (None, 13)                182
_________________________________________________________________
dropout_3 (Dropout)          (None, 13)                0
_________________________________________________________________
layer_dense_3 (Dense)        (None, 13)                182
_________________________________________________________________
dropout_4 (Dropout)          (None, 13)                0
_________________________________________________________________
layer_dense_4 (Dense)        (None, 13)                182
_________________________________________________________________
dropout_5 (Dropout)          (None, 13)                0
_________________________________________________________________
dense_2 (Dense)              (None, 1)                 14
=================================================================
Total params: 27,028
Trainable params: 27,028
Non-trainable params: 0
_________________________________________________________________
```

EXECUTE AND INTERPRET THE OPTIMIZED MODEL.
Unseen holdout dataset consists of 36 (5%) of the original 713 observations. We executed the optimized model and employed an early stop mechanism.

Loss Results by Epoch for Unseen Holdout Dataset.



Accuracy Results by Epoch for Unseen Holdout Dataset.

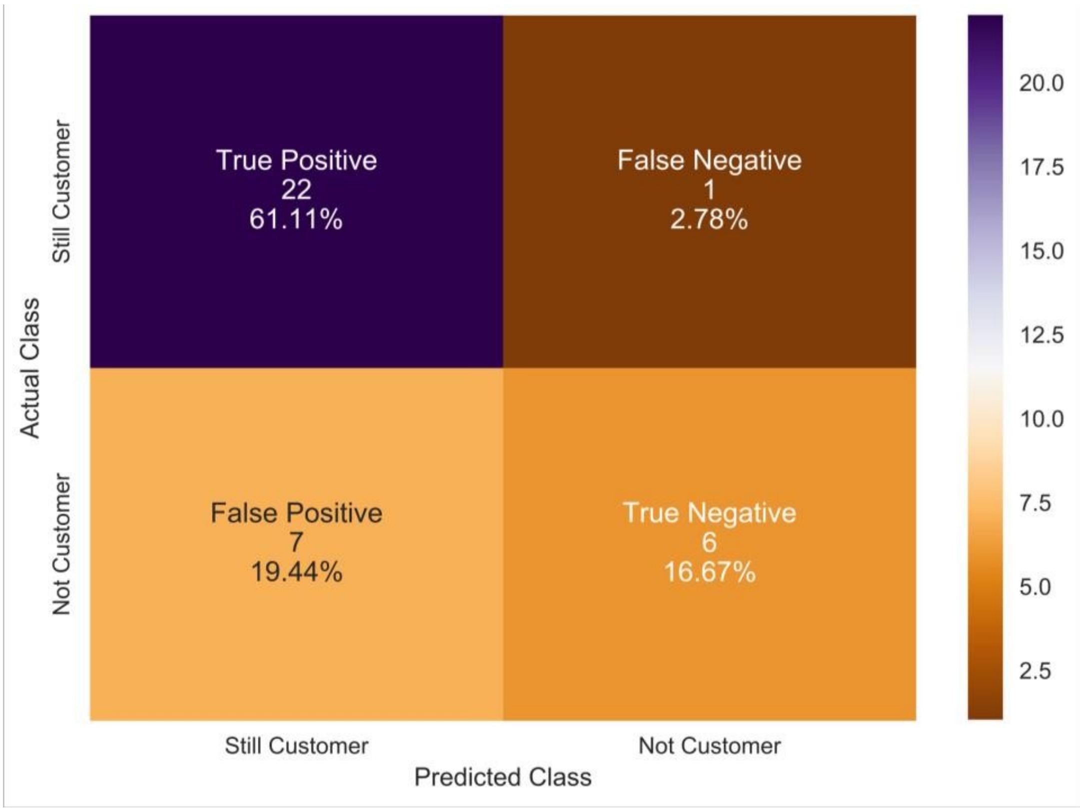


The accuracy results from this optimized model using the unseen holdout dataset is: 77.8%. Below is the summary of our accuracy results from the 3 models:

- Supervised Learning (Logistic Regression): 73.6%
- Baseline Single-Layered NN: 75%
- Bayesian Optimized Deep Learning NN: 77.8%

The Bayesian optimized deep learning model is much better than the previous models and should be used on the client's dataset.
Another metric that should look at is the F1 score for selected model. F1 score is the harmonic mean of Precision and Recall. It provides a better measure of the incorrectly classified cases than the accuracy metric. The F1 score for the optimized model 84.6% (Precision: 75.9% and Recall: 95.7%). The F1 score is much higher than Logistic Regression model (79.7%). The figure below shows the confusion matrix of results.

Confusion Matrix of Bayesian Optimized Deep Learning Model.



In the matrix is the low precision (high number of False Positives).

POSITIVE IMPACT TO COMPANY REVENUE.

Retaining existing customers is big but bringing the customer back to the company is even bigger. Optimizing and executing models it is helps the company.

CONCLUSION.

In this course work on creating software for customer classification, I learned (gained) new useful skills and knowledge in such areas as Python programming, deep learning and neural networks, working with popular frameworks, as well as analytical thinking in the field of marketing and office work. In the course of this project, I was able to practically design a working concept for customer classification and analysis, using the method of referring to the latest digital technologies in the field of Artificial Intelligence (AI).

This product meets the objectives of my course work and is proven to work.

APPENDIX.

1. Exploratory Data Analysis: Data

Dataset info

Number of variables	30
Number of observations	713
Missing cells	0 (0.0%)
Duplicate rows	0 (0.0%)
Total size in memory	167.2 KiB
Average record size in memory	240.2 B

Variables types

Numeric	15
Categorical	7
Boolean	2
Date	0
URL	0
Text (Unique)	0
Rejected	6
Unsupported	0

Accident/Health_(P)...
Categorical

Distinct count	2
Unique (%)	0.3%
Missing (%)	0.0%
Missing (n)	0

0	712
90.32	1

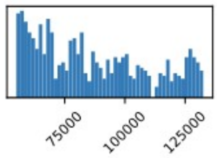
Builders_Risk_(P)_A...
Categorical

Distinct count	3
Unique (%)	0.4%
Missing (%)	0.0%
Missing (n)	0

0	711
400	1
2153	1

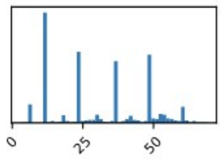
Customer_ID
Numeric

Distinct count	713	Mean	86390.00421
Unique (%)	100.0%	Minimum	54928
Missing (%)	0.0%	Maximum	132548
Missing (n)	0	Zeros (%)	0.0%
Infinite (%)	0.0%		
Infinite (n)	0		



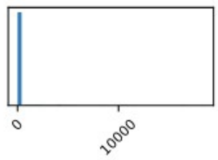
DurationAsCust
Numeric

Distinct count	60	Mean	30.89481066
Unique (%)	8.4%	Minimum	3
Missing (%)	0.0%	Maximum	69
Missing (n)	0	Zeros (%)	0.0%
Infinite (%)	0.0%		
Infinite (n)	0		



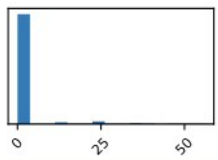
Dwelling_Fire_Amo...
Numeric

Distinct count	47	Mean	281.9679804
Unique (%)	6.6%	Minimum	0
Missing (%)	0.0%	Maximum	18478.92
Missing (n)	0	Zeros (%)	93.4%
Infinite (%)	0.0%		
Infinite (n)	0		



Dwelling_Fire_Durat...
Numeric

Distinct count	15	Mean	1.786816269
Unique (%)	2.1%	Minimum	0
Missing (%)	0.0%	Maximum	56
Missing (n)	0	Zeros (%)	93.4%
Infinite (%)	0.0%		
Infinite (n)	0		



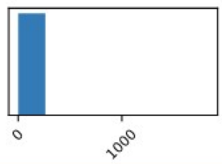
Earthquake_(P)_Am...
Categorical

Distinct count	2
Unique (%)	0.3%
Missing (%)	0.0%
Missing (n)	0

0	712
613.41	1

Flood_Amount
Numeric

Distinct count	7	Mean	10.52920056
Unique (%)	1.0%	Minimum	0
Missing (%)	0.0%	Maximum	1832
Missing (n)	0	Zeros (%)	99.2%
Infinite (%)	0.0%		
Infinite (n)	0		



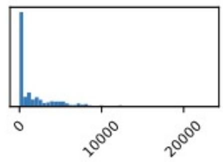
Gender
Categorical

Distinct count	2
Unique (%)	0.3%
Missing (%)	0.0%
Missing (n)	0

M	480
F	233

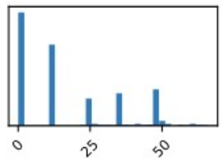
Homeowners_Amount
Numeric

Distinct count	431	Mean	2025.001515
Unique (%)	60.4%	Minimum	0
Missing (%)	0.0%	Maximum	23141.95
Missing (n)	0	Zeros (%)	36.2%
Infinite (%)	0.0%		
Infinite (n)	0		



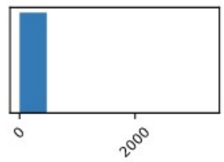
Homeowners_Durat...
Numeric

Distinct count	31	Mean	17.71248247
Unique (%)	4.3%	Minimum	0
Missing (%)	0.0%	Maximum	66
Missing (n)	0	Zeros (%)	36.3%
Infinite (%)	0.0%		
Infinite (n)	0		



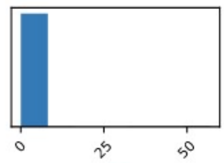
Life_(P)_Amount
Numeric

Distinct count	7	Mean	7.332847125
Unique (%)	1.0%	Minimum	0
Missing (%)	0.0%	Maximum	3295
Missing (n)	0	Zeros (%)	99.2%
Infinite (%)	0.0%		
Infinite (n)	0		



Life_(P)_Duration
Numeric

Distinct count	7	Mean	0.3646563815
Unique (%)	1.0%	Minimum	0
Missing (%)	0.0%	Maximum	57
Missing (n)	0	Zeros (%)	99.2%
Infinite (%)	0.0%		
Infinite (n)	0		



Motorcycle_Amount
Categorical

Distinct count	2
Unique (%)	0.3%
Missing (%)	0.0%
Missing (n)	0

0	712
466	1

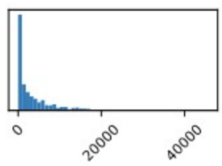
PaidFullPremiumBe...
Boolean

Distinct count	2
Unique (%)	0.3%
Missing (%)	0.0%
Missing (n)	0

1	366
0	347

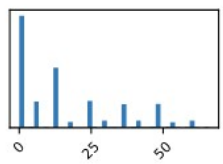
Private_Passenger_...
Numeric

Distinct count	460	Mean	3829.242342
Unique (%)	64.5%	Minimum	0
Missing (%)	0.0%	Maximum	45661.82
Missing (n)	0	Zeros (%)	35.3%
Infinite (%)	0.0%		
Infinite (n)	0		



Private_Passenger_...
Numeric

Distinct count	39	Mean	16.70967742
Unique (%)	5.5%	Minimum	0
Missing (%)	0.0%	Maximum	66
Missing (n)	0	Zeros (%)	35.3%
Infinite (%)	0.0%		
Infinite (n)	0		



Referrer
Categorical

Distinct count	5
Unique (%)	0.7%
Missing (%)	0.0%
Missing (n)	0

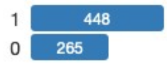
None	683
SC	11
Sam	10
Other values (2)	9

State
Categorical

Distinct count	9
Unique (%)	1.3%
Missing (%)	0.0%
Missing (n)	0

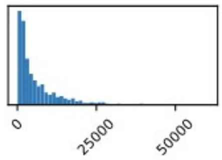
StillCustomer
Boolean

Distinct count	2
Unique (%)	0.3%
Missing (%)	0.0%
Missing (n)	0



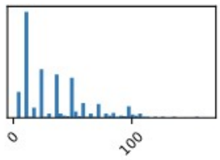
Total_Amount
Numeric

Distinct count	695	Mean	6171.380659
Unique (%)	97.5%	Minimum	81
Missing (%)	0.0%	Maximum	60330.92
Missing (n)	0	Zeros (%)	0.0%
Infinite (%)	0.0%		
Infinite (n)	0		



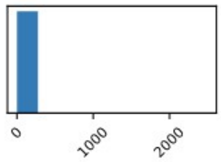
Total_Duration
Numeric

Distinct count	78	Mean	37.20757363
Unique (%)	10.9%	Minimum	3
Missing (%)	0.0%	Maximum	163
Missing (n)	0	Zeros (%)	0.0%
Infinite (%)	0.0%		
Infinite (n)	0		



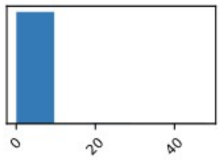
Umbrella_(P)_Amount
Numeric

Distinct count	9	Mean	12.085554
Unique (%)	1.3%	Minimum	0
Missing (%)	0.0%	Maximum	2490
Missing (n)	0	Zeros (%)	98.9%
Infinite (%)	0.0%		
Infinite (n)	0		



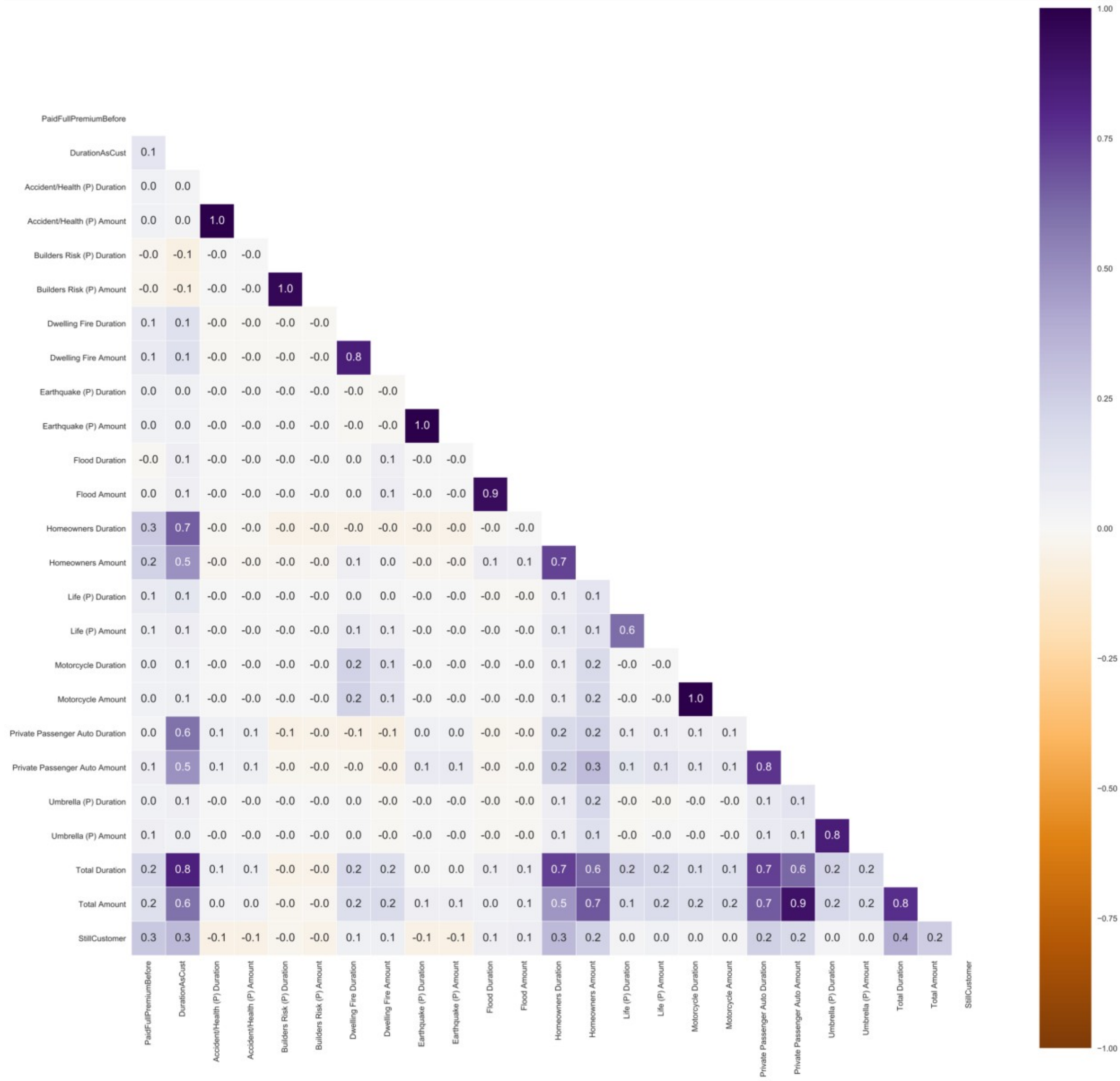
Umbrella_(P)_Durati…
Numeric

Distinct count	5	Mean	0.3197755961
Unique (%)	0.7%	Minimum	0
Missing (%)	0.0%	Maximum	48
Missing (n)	0	Zeros (%)	98.9%
Infinite (%)	0.0%		
Infinite (n)	0		

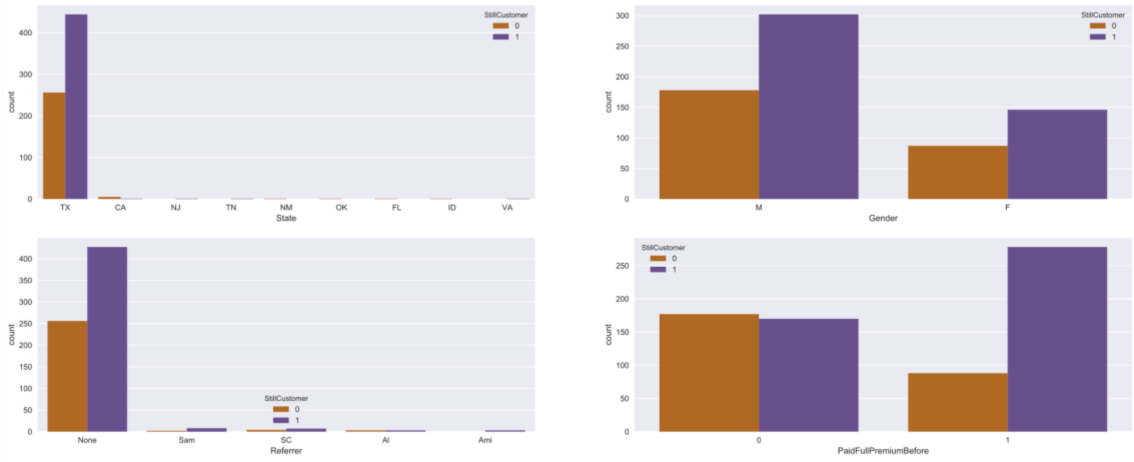


2. Exploratory Data Analysis: Visuals.

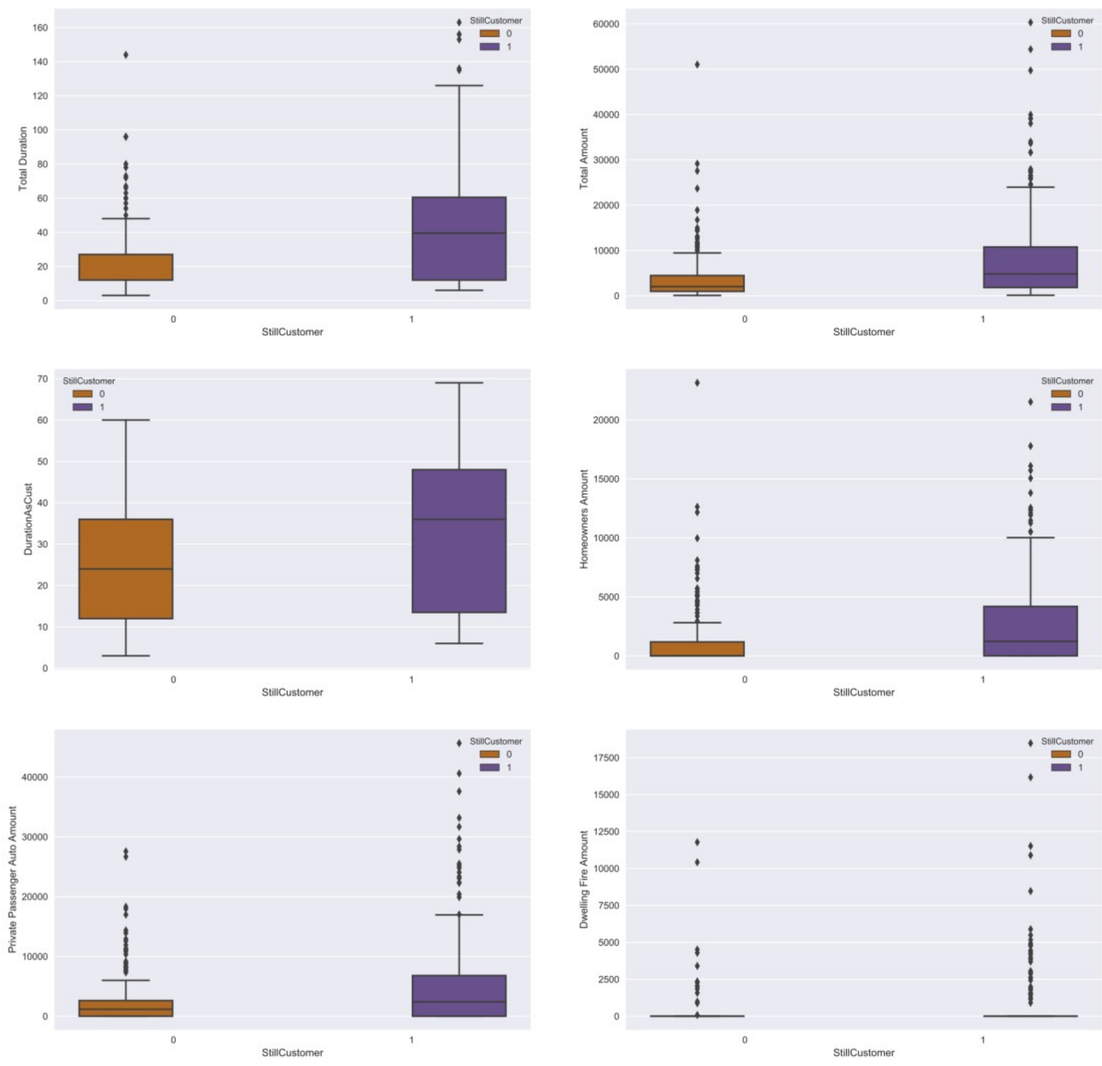
- Correlation Matrix of the Dataset.



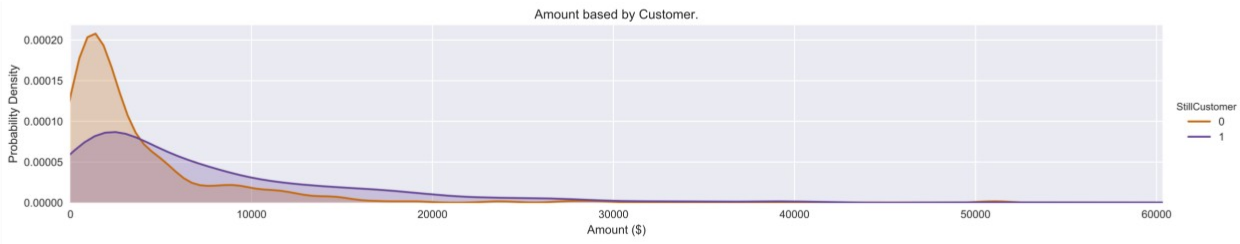
- Plots of Features (State, Gender, Referrer and Paid Full Premium Before).



- Box Plots of Numeric Features (Total Duration, Total Amount, Duration As Cust and Homeowner Amount).



- KDE Plot of Still Customer and Amount.



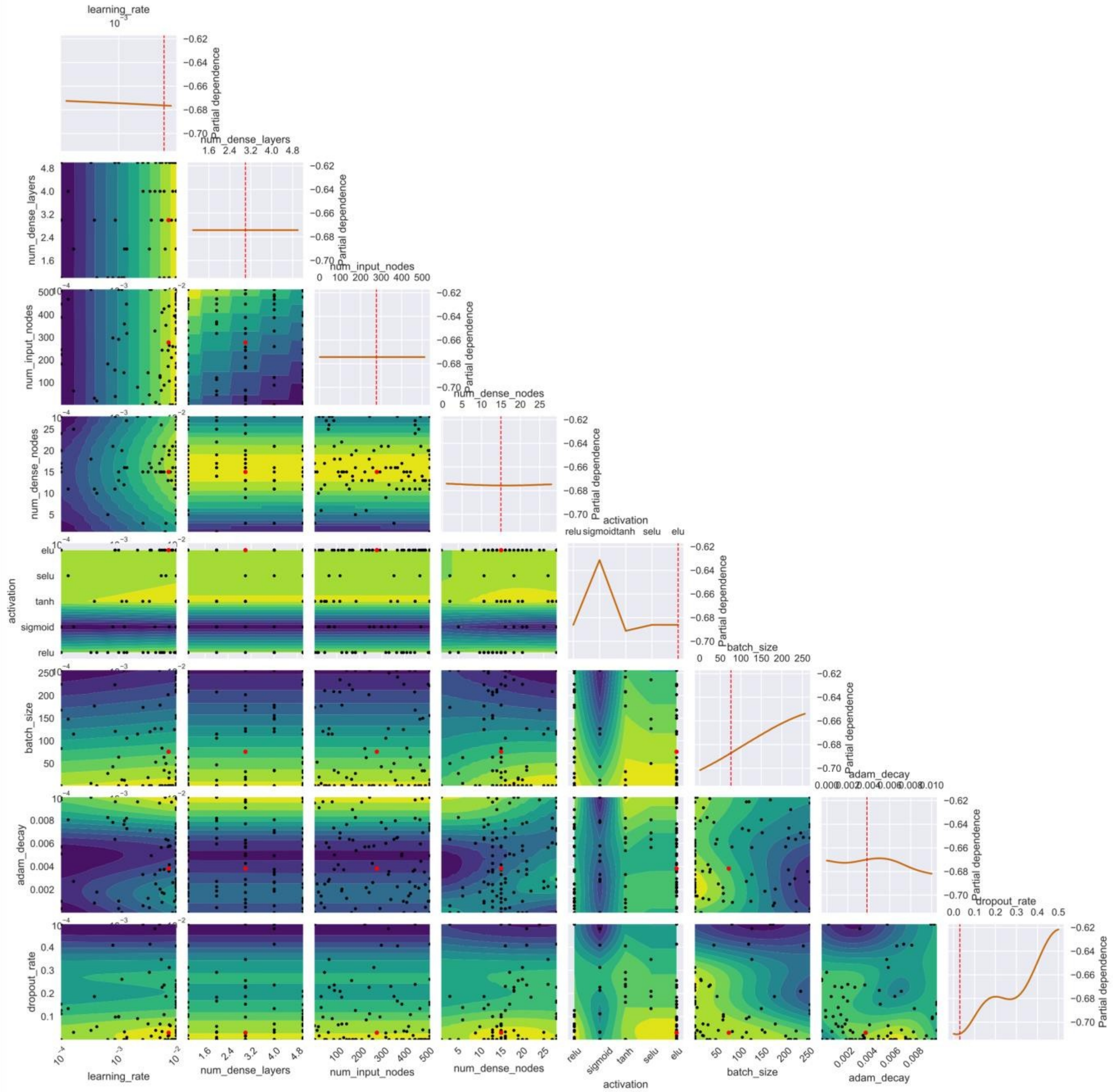
3. Bayesian Optimization Search Results (n_call = 100).

Index	Learning rate	Hidden layers	Input layer nodes	Hidden layer nodes	Activation function	Batch size	Adam learning rate decay	Dropout rate	Accuray	Time (seconds)
99	0.010000	5	512	28	elu	1	0.007296	0.000000	78.235292	43.58
60	0.010000	4	481	13	elu	150	0.006926	0.042091	78.235292	3.37
29	0.003240	5	143	15	relu	19	0.001240	0.068106	78.235292	4.59
75	0.010000	1	445	13	relu	208	0.003714	0.000000	78.235292	1.65
10	0.007348	3	277	15	elu	76	0.003821	0.029607	78.235292	2.9
81	0.007222	3	171	13	elu	1	0.006281	0.000000	78.235292	87.81
24	0.010000	5	397	15	elu	225	0.000001	0.037377	78.235292	3.98
92	0.002129	5	79	28	elu	115	0.009600	0.090900	78.235292	4.03
35	0.004392	5	12	15	relu	53	0.002489	0.040072	78.235292	4.07
80	0.010000	4	447	11	elu	256	0.000001	0.000000	77.647060	2.1
47	0.001236	1	143	12	relu	256	0.007640	0.000000	77.647060	2.81
72	0.010000	1	53	27	tanh	1	0.006051	0.000000	77.647060	33.28
16	0.010000	3	512	20	elu	1	0.007697	0.000000	77.647060	36.2
39	0.003678	3	512	16	elu	1	0.004938	0.015060	77.058822	29.02
43	0.000860	5	155	11	elu	224	0.006355	0.000000	77.058822	5.46
25	0.008683	4	259	16	elu	1	0.002608	0.000000	77.058822	26.72
20	0.010000	3	1	3	tanh	256	0.000001	0.000000	77.058822	4.15
32	0.010000	3	339	16	tanh	32	0.001143	0.230385	77.058822	2.87
88	0.010000	3	21	28	elu	20	0.009087	0.016378	77.058822	2.91
89	0.010000	2	121	22	elu	1	0.001881	0.183693	77.058822	9.09
97	0.010000	1	447	26	tanh	1	0.002272	0.239294	77.058822	9.17
57	0.010000	1	512	11	relu	256	0.007423	0.000000	76.470590	1.45
54	0.007209	3	242	13	elu	202	0.005712	0.015449	76.470590	3.3
67	0.010000	1	365	11	relu	256	0.006472	0.000000	76.470590	1.34
74	0.010000	1	512	28	tanh	1	0.010000	0.161682	76.470590	9.92
84	0.008268	3	210	13	elu	1	0.009574	0.000000	76.470590	26.64
91	0.010000	1	30	6	elu	6	0.001440	0.138962	76.470590	2.45
70	0.010000	4	387	13	relu	1	0.004932	0.000000	76.470590	25.43
0	0.001000	1	512	13	relu	64	0.001000	0.100000	76.470590	2.77
53	0.003415	2	45	17	tanh	1	0.000433	0.291390	75.882351	8.19
52	0.007554	1	264	20	sigmoid	1	0.005786	0.312361	75.882351	20.05
98	0.005665	3	243	18	elu	1	0.001748	0.226518	75.882351	14.75
62	0.001367	2	447	27	tanh	1	0.008179	0.000000	75.882351	97.42

94	0.010000	2	368	28	elu	1	0.002247	0.166889	75.882351	16.14
4	0.006148	4	88	21	elu	35	0.003747	0.236619	75.882351	3.87
36	0.006134	1	426	15	relu	64	0.001663	0.108286	75.882351	2.23
21	0.010000	3	317	9	elu	1	0.006005	0.028686	75.882351	19.6
38	0.001136	1	512	20	tanh	14	0.000001	0.261287	75.882351	3.59
86	0.010000	2	470	28	elu	47	0.005954	0.000000	75.882351	3.31
77	0.010000	3	257	13	elu	1	0.004268	0.000000	75.882351	25.9
73	0.010000	1	86	27	tanh	209	0.007330	0.000000	75.882351	2.84
69	0.010000	5	512	12	elu	256	0.010000	0.151501	75.294119	4.03
65	0.010000	4	488	13	elu	34	0.003518	0.000000	75.294119	2.37
68	0.010000	3	492	13	elu	1	0.002433	0.000000	75.294119	10.28
78	0.010000	1	46	12	relu	211	0.002471	0.000000	75.294119	1.45
55	0.010000	1	185	20	tanh	1	0.005378	0.000000	75.294119	14.02
87	0.010000	3	66	28	sigmoid	1	0.002881	0.226700	75.294119	11.29
31	0.010000	2	113	14	tanh	208	0.006449	0.000000	75.294119	1.52
82	0.010000	4	422	11	elu	256	0.002659	0.000000	74.705881	2.32
63	0.010000	1	463	25	elu	1	0.002131	0.000000	74.705881	16.32
95	0.010000	1	89	18	sigmoid	1	0.005002	0.213116	74.705881	20.81
17	0.010000	1	512	16	elu	222	0.002948	0.000000	74.705881	1.71
58	0.010000	3	165	19	elu	15	0.000001	0.255806	74.705881	3.54
64	0.010000	1	472	21	sigmoid	1	0.004455	0.000000	74.705881	64.27
50	0.001385	2	317	16	relu	106	0.007282	0.010000	74.705881	2.62
42	0.007567	5	106	15	elu	30	0.004155	0.064097	74.705881	4.75
37	0.000331	5	512	18	relu	6	0.010000	0.039929	74.705881	42.4
46	0.005223	5	130	15	elu	1	0.003708	0.008415	74.705881	7.79
33	0.000999	5	291	15	elu	1	0.000357	0.098302	74.117649	12.12
40	0.010000	2	418	14	elu	1	0.001444	0.057080	74.117649	6.28
5	0.000133	4	468	11	selu	148	0.008048	0.134381	74.117649	6.46
61	0.010000	4	512	13	elu	156	0.001860	0.031572	74.117649	3.11
76	0.010000	1	169	14	relu	213	0.008927	0.000000	74.117649	1.31
30	0.010000	5	478	14	elu	34	0.009874	0.000000	74.117649	2.61
56	0.010000	1	512	11	tanh	256	0.000001	0.000000	73.529410	1.55
66	0.007451	1	438	19	elu	18	0.003591	0.018033	73.529410	2.13
23	0.000648	1	182	5	relu	28	0.008244	0.000000	72.941178	5.6
48	0.005229	4	122	15	relu	179	0.002036	0.067612	72.941178	4.08
71	0.010000	2	512	19	tanh	1	0.010000	0.000000	72.941178	8.88
59	0.007869	5	1	15	relu	1	0.005768	0.060109	71.764708	10.55
51	0.000417	1	18	10	relu	1	0.001784	0.055935	71.764708	66.65
49	0.005651	5	410	15	relu	84	0.000891	0.034974	71.176469	3.39
19	0.000100	3	388	17	elu	1	0.005089	0.153600	70.588237	135.79
45	0.010000	1	369	16	tanh	241	0.006528	0.405735	69.411767	2.07
9	0.001259	2	359	16	sigmoid	13	0.007962	0.033381	69.411767	14.31
3	0.000379	3	30	11	tanh	177	0.000833	0.186489	69.411767	5.62
96	0.010000	5	230	6	elu	63	0.002167	0.085303	68.823528	3.75
1	0.000165	2	62	25	sigmoid	115	0.002711	0.031304	68.823528	5.34
18	0.000100	5	505	20	relu	225	0.000984	0.000000	68.235296	6.29
28	0.009277	5	395	11	relu	125	0.000984	0.000000	68.235296	2.67
6	0.000793	1	297	21	sigmoid	43	0.007821	0.408936	67.647058	5.82
13	0.005126	1	1	17	elu	256	0.009495	0.500000	67.058825	3.36

12	0.000100	3	447	24	elu	168	0.009769	0.500000	67.058825	5.95
8	0.005477	3	65	26	selu	127	0.005644	0.347275	65.882355	4.01
93	0.000875	3	38	28	elu	121	0.009610	0.097560	65.294117	2.72
15	0.000100	1	224	15	relu	83	0.005696	0.500000	63.529414	4.83
27	0.003499	4	512	28	elu	120	0.007124	0.413416	63.529414	2.81
34	0.010000	5	114	18	selu	235	0.006906	0.209167	62.941176	4.93
7	0.006611	5	353	3	selu	50	0.006352	0.187104	62.941176	5.72
79	0.010000	1	331	12	relu	250	0.003573	0.000000	62.352943	1.16
14	0.002651	4	1	3	relu	174	0.000599	0.500000	61.176473	5.93
83	0.010000	1	1	5	sigmoid	256	0.000001	0.000000	61.176473	1.33
11	0.006419	2	512	1	elu	1	0.010000	0.000000	61.176473	32.71
44	0.002623	5	1	15	tanh	1	0.003303	0.000000	60.000002	77.24
41	0.000100	5	182	16	relu	256	0.000340	0.126159	58.823532	7.46
85	0.010000	3	493	13	elu	1	0.000469	0.000000	54.705882	18.75
26	0.008612	5	1	21	elu	1	0.000001	0.000000	53.529412	8.44
22	0.010000	3	512	28	relu	1	0.000001	0.000000	45.294118	7.42
2	0.004257	4	328	21	sigmoid	127	0.003310	0.480414	38.823530	3.5
90	0.000100	5	245	28	sigmoid	256	0.010000	0.111936	38.823530	8.56

4. Bayesian Optimization Search (pairwise dependence plot of the objective function).



5. Probability and Class for Customers in Unseen Holdout Dataset.

StillCustomer_x	predicted_prob	Customer ID	predicted_class	cf_matrix
0	9%	[redacted]	0	TN
	22%	[redacted]	0	TN
	29%	[redacted]	0	TN
	29%	[redacted]	0	TN
	39%	[redacted]	0	TN
	43%	[redacted]	0	TN
	53%	[redacted]	1	FP
	63%	[redacted]	1	FP
	76%	[redacted]	1	FP
	76%	[redacted]	1	FP
	81%	[redacted]	1	FP
	90%	[redacted]	1	FP
	95%	[redacted]	1	FP
1	48%	[redacted]	0	FN
	53%	[redacted]	1	TP
	61%	[redacted]	1	TP
	74%	[redacted]	1	TP
	74%	[redacted]	1	TP
	76%	[redacted]	1	TP
	76%	[redacted]	1	TP
	86%	[redacted]	1	TP
	87%	[redacted]	1	TP
	89%	[redacted]	1	TP
	89%	[redacted]	1	TP
	90%	[redacted]	1	TP
	91%	[redacted]	1	TP
	91%	[redacted]	1	TP
	91%	[redacted]	1	TP
	92%	[redacted]	1	TP
	93%	[redacted]	1	TP
	95%	[redacted]	1	TP
	95%	[redacted]	1	TP
	95%	[redacted]	1	TP
	95%	[redacted]	1	TP
	95%	[redacted]	1	TP
	96%	[redacted]	1	TP