Final Project Proposal

For the final project, we will be analyzing a dataset of airline customer satisfaction surveys, with the goal of being able to predict airline customer satisfaction. We selected this dataset based on our mutual interest in travel, and now that pandemic restrictions are slowly being lifted, we both have upcoming travel plans. Additionally, Meg has experience working as a marketing analyst and has a particular interest in customer satisfaction surveys. We agree that the features in the dataset are good indicators of customer satisfaction on a flight, as they are important factors to us when we fly. These features include convenience of departure/arrival time, ease of online booking, boarding, seat comfort, inflight entertainment, food and drink, leg room, cleanliness, delay time, and more. We are also interested in the features related to the demographics of the survey respondent, such as age, gender, customer loyalty, and type of travel. The target that we will train the network to predict is the result of the airline customer satisfaction survey: whether the respondent indicated they were satisfied with the airline or neutral/dissatisfied with the experience. The dataset has 129,487 rows and 24 features. The original dataset was posted by John D. on Kaggle three years ago.

We have chosen to use Databricks as our database and software of choice for this project, given our previous experience using the database. Databricks offers scalability and reliability in a cloud-based platform that combines the best attributes of data warehouses and data lakes. The free, open-source platform allows for flexibility and collaboration, which is important as we are still in the virtual environment. Databricks software is designed to be large enough to train a machine learning network without sacrificing speed, which is why we chose this database for our project. We will also be using PyCharm to format the file, check for errors, and make final adjustments prior to file submission.

For the purposes of this project, we will be using a Multilayer Perceptron (MLP) network to predict the overall customer satisfaction with the airline. This supervised learning network will be able to separate the survey responses into two categories, satisfied or neutral/dissatisfied, based on the features from the survey responses. The optimization algorithms we will be using to classify the responses include Stochastic Descent Gradient Backpropagation (SDG), Adam, and Limited-memory Broyden–Fletcher–Goldfarb–Shanno (LBFGS). We will compare the results of these models to the classical Random Forest classification model.

To create the neural network, we will be coding in Python and implementing the numpy, pandas, seaborn, matplotlib, scipy, and sklearn libraries. Over the course of the project, we will be referencing past research on neural networks, backpropagation optimization algorithms, and research on the Random Forest model for analyzing Likert-type scale survey responses. We will evaluate the success of the network using a Confusion Matrix, a Classification Report, F-1 Score, as well as several Classification metrics, such as accuracy, precision, and recall ensuring the network matches the targets at an acceptable rate.

Schedule

June 12th: First meeting to discuss project topic, ideas, and project plan
June 13th: Find a dataset and conduct research on topic
June 14th: Conduct research on which neural network and algorithm to use
June 15th through June 20th: Code network using Python and Databricks
June 21st and June 22nd: Finalize final project report and presentation
June 23rd: Give final project presentation and submit the results on Blackboard

Final Project Report

Customer satisfaction is paramount for any company, but particularly vital for the airline industry, as the competition between airlines for customer attention grows stiffer each year. However, gauging what really most matters in terms of customer satisfaction and what is superfluous can be challenging. Is the inflight experience more important than the online booking experience to the customer? Do customer demographics determine overall satisfaction? How can we predict customer satisfaction? This report explores these issues and more by analyzing the results of customer satisfaction surveys.

To measure customer satisfaction, we used a dataset of 129,487 survey responses that includes 24 features accessed from Kaggle and posted by John D. in 2018. Our target variable is the overall satisfaction with the airline from the survey respondent, either satisfied or neutral/dissatisfied. We included some demographic information on the survey respondents, such as age, gender, customer loyalty, and type of travel. The survey also gathered some information about the actual flight that could impact customer satisfaction, such as flight distance, flight delay times, and seating class (such as Business Class, Economy Class, etc.). The rest of the features measure customer satisfaction on a Likert-type scale of 1 to 5 with several different customer experiences before, during, and after the flight, 1 indicating not satisfied and 5 indicating very satisfied, with 0 meaning not applicable. A Likert scale gathers individual attitudes based on their level of agreement with a statement (Likert, 1932). However, a Likert-type scale allows an individual to indicate their level of satisfaction with a product or service (Lavrakas, 2008). These features include satisfaction with the inflight entertainment, booking experience, flight time, gate location, food and beverage service, seat comfort and leg room, baggage handling, cleanliness, and more.

Given overall satisfaction as well as the satisfaction within some of the features are measured on a Likert-type scale, we conducted some research on which classical models best classify the results of surveys using Likert-type scale questions. According to Anna Endresen and Laura A. Janda, in their analysis of Likert-type scale survey responses, Random Forest classification models work the best compared to parametric tests (Endresen and Janda, 2016). Random Forest is preferred by the authors given it is the “the most appropriate, informative, and user-friendly” (Endresen and Janda, 2016, p. 217). Going beyond this past research, we seek to find whether a Multilayer Perceptron Network is more accurate in predicting Likert-type scale survey responses than a classical model, such as the Random Forest classification model recommended by Endresen and Janda.

Prior to running the models, we did some exploratory data analysis and parametric tests of the data. We examined each of the features individually to determine if there is a relationship between the feature and the overall customer satisfaction with the airline. Starting with the categorical variables of gender, customer type, and travel class, we created a subroutine to determine the correlation with overall satisfaction in order to gauge how helpful they will be in the model. We discovered that, for the gender feature, women tend to be slightly more satisfied with the airline than men. Unsurprisingly, loyal customers were more likely to be satisfied with the airline than disloyal customers, given disloyal customers would potentially switch to another airline if they were dissatisfied with their experience. Likewise, passengers who flew in Business Class and Economy Plus tended to be more satisfied with the airline than Economy Class passengers, as they likely paid to have an upgraded experience. Interestingly, customers traveling for business tended to be more satisfied with the airline than customers who traveled for personal reasons. Figure 1 below is a screenshot demonstrating the exact correlations.

```
   Gender  satisfaction_v2
0  Female         0.651401
1    Male         0.440283
       Customer Type  satisfaction_v2
0     Loyal Customer         0.616358
1  disloyal Customer         0.239858
    Type of Travel  satisfaction_v2
0  Business travel         0.583677
1  Personal Travel         0.466385
      Class  satisfaction_v2
0  Business         0.709421
1       Eco         0.393998
2  Eco Plus         0.427186
```

Figure 1. Correlations between categorical features and overall satisfaction.

Meanwhile, our subroutine to complete exploratory data analysis of the continuous features revealed dramatically different results. Figure 2 below shows the exact correlations for these features. Neither departure delay in minutes nor arrival delay in minutes seemed to have a strong correlation with overall satisfaction, which is surprising as one would assume a significant flight delay would cause strong dissatisfaction with the airline. Figure 3 shows the skewed distribution of departure delay feature measured in minutes. Additionally, flight distance did not seem to have a significant correlation with satisfaction. Interestingly, age only had a marginal positive correlation with satisfaction, which led us to investigate further. We examined box plots of the distribution of the age feature and found there was no significant difference between ages that were satisfied and those that were neutral/dissatisfied with the airline. Figure 4 below visually demonstrates this relationship. We also sought to find if there was a relationship between age, satisfaction, and our other categorical variables, but as you can see in Figures 5 through 8, there does not appear a strong relationship.

```
                          Departure Delay in Minutes  satisfaction_v2
Departure Delay in Minutes                    1.00000         -0.07396
satisfaction_v2                              -0.07396          1.00000
                        Arrival Delay in Minutes  satisfaction_v2
Arrival Delay in Minutes                  1.000000        -0.080691
satisfaction_v2                          -0.080691         1.000000
                Flight Distance  satisfaction_v2
Flight Distance        1.000000        -0.039133
satisfaction_v2       -0.039133         1.000000
                     Age  satisfaction_v2
Age             1.000000         0.117913
satisfaction_v2 0.117913         1.000000
```

Figure 2. Correlations between continuous features and overall satisfaction.

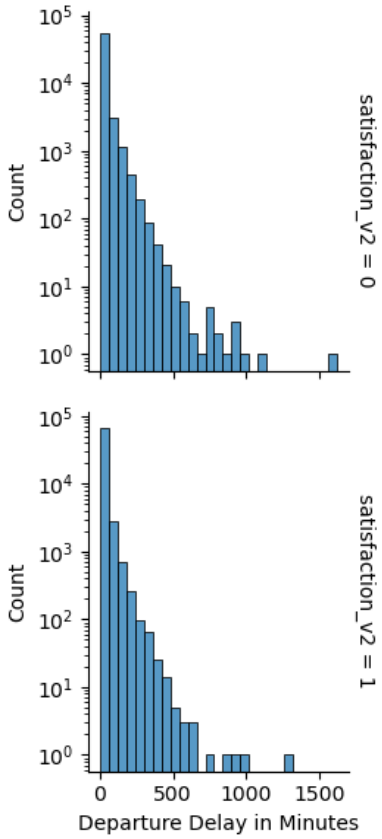

Figure 3. Skewed distributions of Departure Delay in Minutes by Level of Satisfaction - 1 indicating Satisfied, 0 indicating Neutral/Dissatisfied

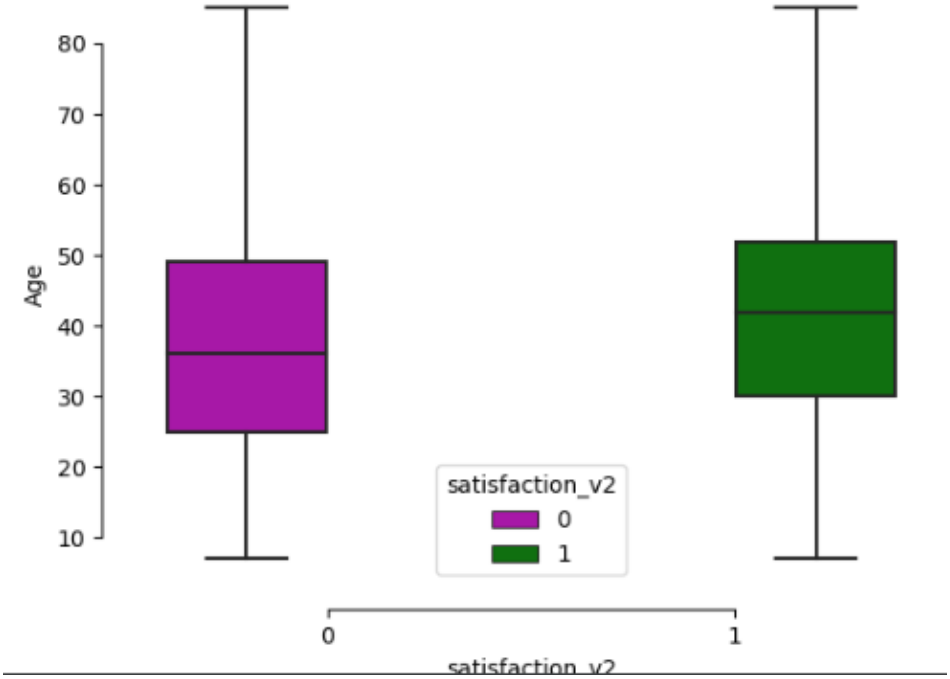

Figure 4. Distribution of Age by Level of Satisfaction - 1 indicating Satisfied, 0 indicating Neutral/Dissatisfied

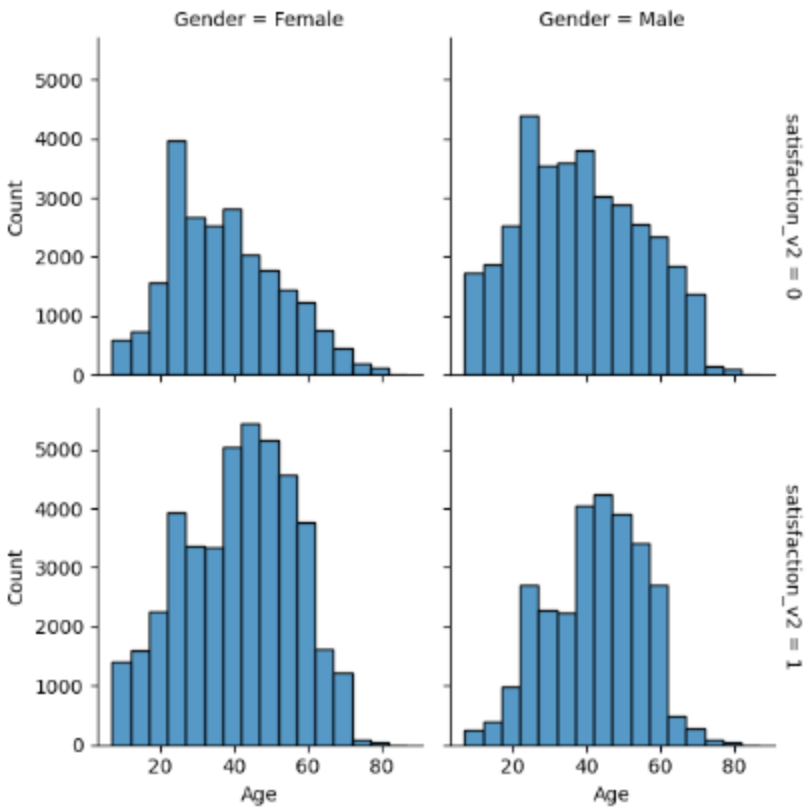

Figure 5. Distribution of Age by Gender and Level of Satisfaction - 1 indicating Satisfied, 0 indicating Neutral/Dissatisfied

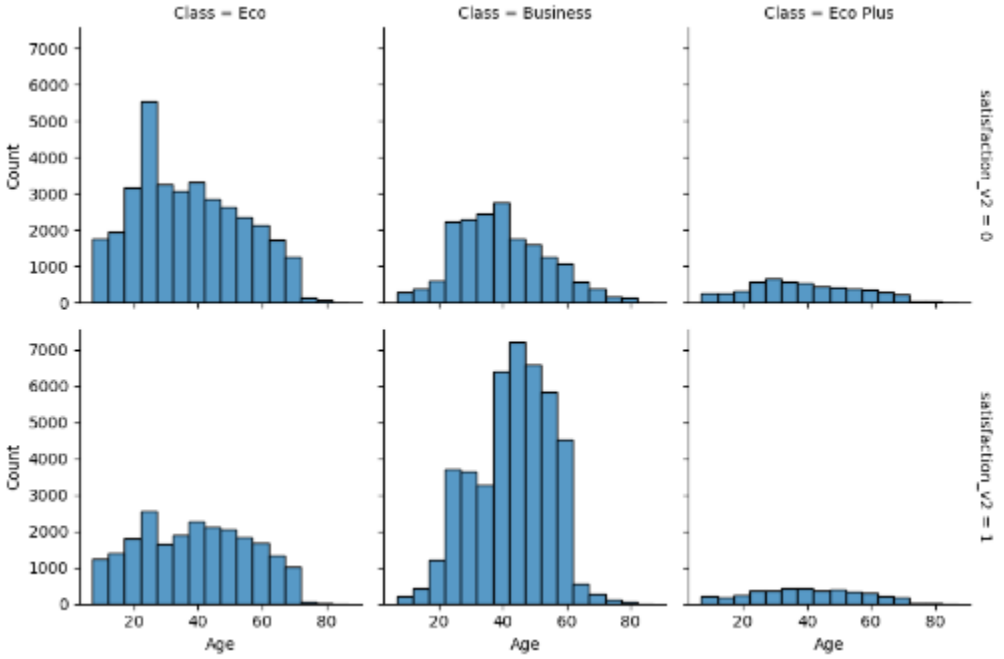

Figure 6. Distribution of Age by Passenger Class and Level of Satisfaction - 1 indicating Satisfied, 0 indicating Neutral/Dissatisfied

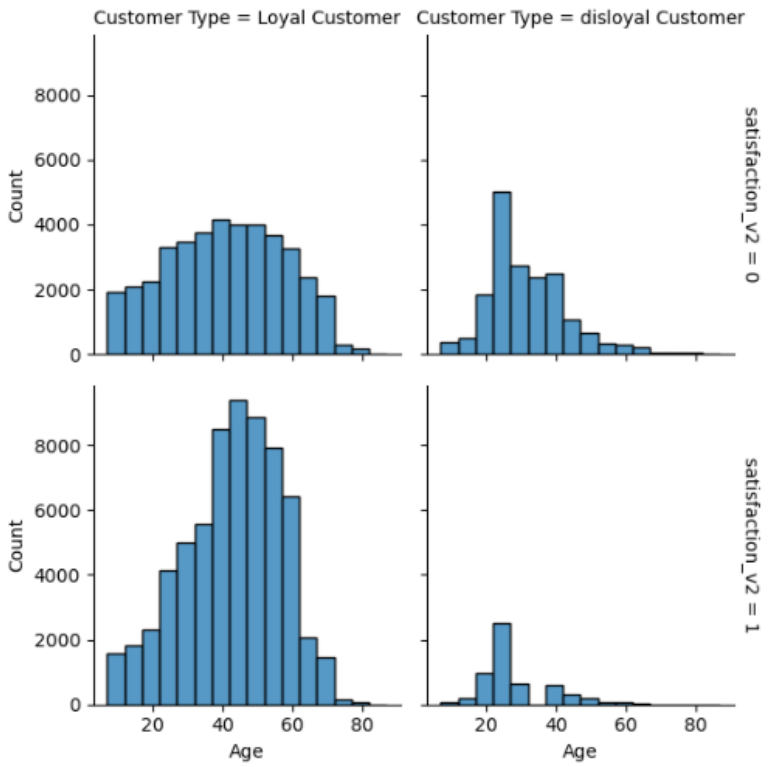

Figure 7. Distribution of Age by Customer Type and Level of Satisfaction - 1 indicating Satisfied, 0 indicating Neutral/Dissatisfied

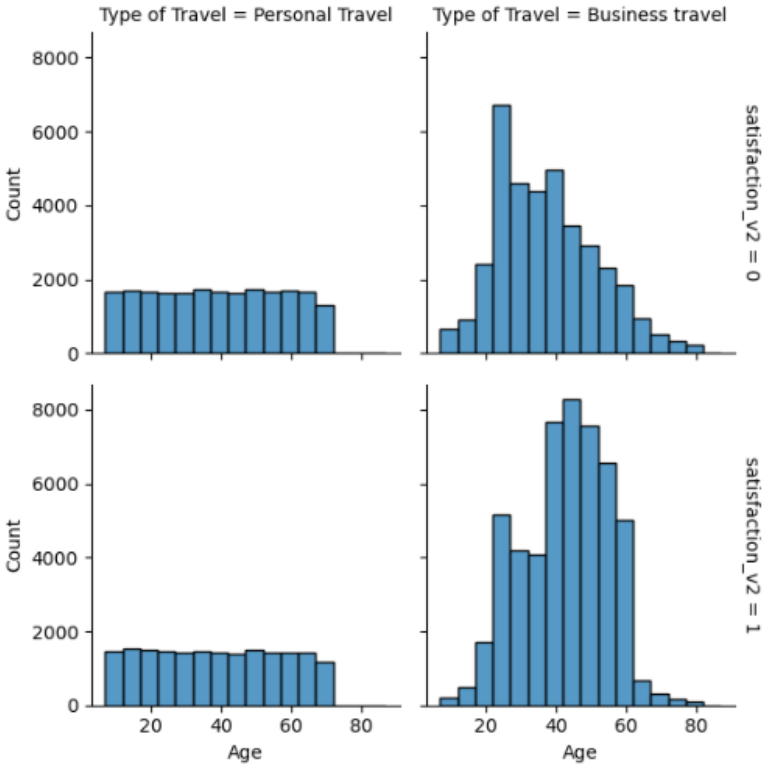

Figure 8. Distribution of Age by Type of Travel and Level of Satisfaction - 1 indicating Satisfied, 0 indicating Neutral/Dissatisfied

Finally, we created another subroutine to calculate the correlations of the remaining features, all of which measure satisfaction on a Likert-type scale. We calculated the correlation of overall customer satisfaction with the airline to each of the levels of satisfaction from 0 to 5 for all the Likert-type scale features. Focusing on the correlation of satisfaction with the Likert-type scores of 4 and 5, we found that inflight entertainment and seat comfort were the two most highly correlated features - see Figures 9 through 15 below for the exact correlations. The rest of the Likert-type scale features only had moderately high correlations with the mark of 5 and overall satisfaction. Of these Likert-type scale features, we can see that features that describe satisfaction with the inflight experience, such as seat comfort, inflight entertainment, food and drink, and on-board service, to name a few, are more highly correlated with overall satisfaction than features such as gate location and departure/arrival time convenience. If airlines are looking to improve overall customer satisfaction, they should therefore seek to invest in improving the inflight experience.

	Seat comfort	satisfaction_v2
0	0	0.997908
1	1	0.450915
2	2	0.357794
3	3	0.356131
4	4	0.651845
5	5	0.992064

	Inflight entertainment	satisfaction_v2
0	0	0.660714
1	1	0.210656
2	2	0.170363
3	3	0.199188
4	4	0.719870
5	5	0.952064

Figure 9. Exact Correlations of the Highest Correlated Likert-type Scale Features: Seat Comfort and Inflight Entertainment

	Gate location	satisfaction_v2
0	0	1.000000
1	1	0.610926
2	2	0.580705
3	3	0.463065
4	4	0.497850
5	5	0.655479

	Food and drink	satisfaction_v2
0	0	0.779635
1	1	0.508473
2	2	0.432713
3	3	0.428612
4	4	0.590254
5	5	0.780084

	Inflight wifi service	satisfaction_v2
0	0	0.446154
1	1	0.268507
2	2	0.502356
3	3	0.509557
4	4	0.638368
5	5	0.669114

	Online support	satisfaction_v2
0	0	0.000000
1	1	0.295464
2	2	0.296755
3	3	0.282690
4	4	0.680481
5	5	0.773152

Figures 10 and 11. Exact Correlations of Moderately Correlated Likert-type Scale Features: Gate Location, Food and Drink, Inflight WiFi, and Online Support

	Ease of Online booking	satisfaction_v2
0	0	0.000000
1	1	0.192730
2	2	0.286368
3	3	0.357053
4	4	0.717813
5	5	0.765499

	On-board service	satisfaction_v2
0	0	0.000000
1	1	0.265825
2	2	0.340831
3	3	0.410290
4	4	0.647394
5	5	0.765692

	Leg room service	satisfaction_v2
0	0	0.692308
1	1	0.283384
2	2	0.376378
3	3	0.371746
4	4	0.673345
5	5	0.708523

	Departure/Arrival time convenient	satisfaction_v2
0	0	0.542444
1	1	0.586154
2	2	0.540268
3	3	0.539680
4	4	0.524607
5	5	0.556450

Figures 12 and 13. Exact Correlations of Moderately Correlated Likert-type Scale Features: Ease of Online Booking, On-board Service, Leg room, and Departure/Arrival Time

	Baggage handling	satisfaction_v2
0	1	0.422574
1	2	0.396250
2	3	0.315119
3	4	0.588480
4	5	0.735817

	Checkin service	satisfaction_v2
0	0	0.000000
1	1	0.316277
2	2	0.332643
3	3	0.567485
4	4	0.577010
5	5	0.735763

	Cleanliness	satisfaction_v2
0	0	0.000000
1	1	0.403047
2	2	0.404760
3	3	0.317899
4	4	0.586767
5	5	0.731698

	Online boarding	satisfaction_v2
0	0	0.000000
1	1	0.264925
2	2	0.281309
3	3	0.549557
4	4	0.652527
5	5	0.731715

Figures 14 and 15. Exact Correlations of Moderately Correlated Likert-type Scale Features: Baggage Handling, Checkin Service, Cleanliness, and Online Boarding

With this knowledge in mind, we continued our data pre-processing prior to creating the model. First, we converted the categorical features from strings into numeric values and used LabelEncoder from the sklearn package to apply preprocessing transformations. After splitting our dataset into train and test sets, with 80% saved in the training set and 20% in the testing set, we used StandardScaler, also from the sklearn package, to further transform the data. Next, we conducted a Principal Component Analysis (PCA) within a subroutine to choose the best number of components to use in our model. Based on the results, we chose to use 12 components in our model, as it explains 84% of the variance without the risk of overfitting. See figure 16 below for the graph demonstrating the output of our Principal Component Analysis.

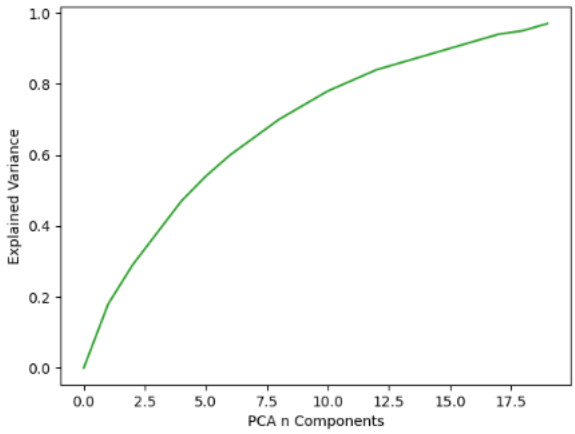

Figure 16. Principal Component Analysis Output; 12 features explain 84% of variance

Given the nature of our output data is binary, the customer is either satisfied or neutral/dissatisfied, we will create a supervised, binary classification Multilayer Perceptron Network. In order to train and test our network, we will be using several different backpropagation learning methods. Backpropagation was defined by Rumelhart et. al in 1986 as a method that “allows the information from the cost to then flow backward through the network in order to compute the gradient” (Goodfellow et. al, 2017). We chose to use three different backpropagation learning methods for our five layer networks: Stochastic Descent Gradient Backpropagation (SDG), Adam optimization, and Limited-memory Broyden–Fletcher–Goldfarb–Shanno (LBFGS). As a baseline for comparison, we will use hidden layer sizes of 10 - 10 - 10 for all three models initially. Additionally, we are going to make several different changes to the model to test if we can achieve better performance, such as changing the momentum, alpha, number of neurons in each layer, as well as changing the network architecture by increasing the number of hidden layers in the neural network to four. We will judge the performance of these neural networks based on the precision, recall, F1-score, and accuracy of the model, as stated in the Classification Report and Confusion Matrix.

Given our dataset is large with high-dimensional parameter spaces, we will begin with the Adam optimization algorithm. This method of backpropagation only requires the calculation of the first-order gradients and has very little storage requirements (Kingma and Ba, 2015). Adam combines the mastery of the AdaGrad algorithm (Duchi et al., 2011) in dealing with sparse gradients as well as RMSProp (Tieleman & Hinton, 2012) in handling online and non-stationary settings (Kingma and Ba, 2015). This algorithm has many advantages, including “that the magnitudes of parameter updates are invariant to rescaling of the gradient, its stepsizes are approximately bounded by the stepsize hyperparameter, it does not require a stationary objective, it works with sparse gradients, and it naturally performs a form of step size annealing” (Kingma and Ba, 2015). The pseudo-code below demonstrates how the Adam algorithm works in mathematical terms.

Require: α: Stepsize
Require: $\beta_1, \beta_2 \in [0, 1)$: Exponential decay rates for the moment estimates
Require: $f(\theta)$: Stochastic objective function with parameters θ
Require: θ_0: Initial parameter vector
$m_0 \leftarrow 0$ (Initialize 1st moment vector)
$v_0 \leftarrow 0$ (Initialize 2nd moment vector)
$t \leftarrow 0$ (Initialize timestep)
while θ_t not converged **do**
$t \leftarrow t + 1$
$g_t \leftarrow \nabla_\theta f_t(\theta_{t-1})$ (Get gradients w.r.t. stochastic objective at timestep t)
$m_t \leftarrow \beta_1 \cdot m_{t-1} + (1 - \beta_1) \cdot g_t$ (Update biased first moment estimate)
$v_t \leftarrow \beta_2 \cdot v_{t-1} + (1 - \beta_2) \cdot g_t^2$ (Update biased second raw moment estimate)
$\widehat{m}_t \leftarrow m_t / (1 - \beta_1^t)$ (Compute bias-corrected first moment estimate)
$\widehat{v}_t \leftarrow v_t / (1 - \beta_2^t)$ (Compute bias-corrected second raw moment estimate)
$\theta_t \leftarrow \theta_{t-1} - \alpha \cdot \widehat{m}_t / (\sqrt{\widehat{v}_t} + \epsilon)$ (Update parameters)
end while
return θ_t (Resulting parameters)

Figure 17. Adam optimization algorithm pseudo-code from Kingma and Ba, 2015, p. 2.

The unique update rule that is utilized in the Adam algorithm is based on the stepsize. As previously mentioned, the stepsize is invariant to the scale of the gradients (Kingman and Ba, 2015, p. 3):

$$(c \cdot \widehat{m}_t)/(\sqrt{c^2 \cdot \widehat{v}_t}) = \widehat{m}_t / \sqrt{\widehat{v}_t}$$

The Adam algorithm also uses an initialization bias correction term, “written as a function of the gradients at all previous timesteps” (Kingman and Ba, 2015, p. 3).

$$v_t = (1 - \beta_2) \sum_{i=1}^{t} \beta_2^{t-i} \cdot g_i^2$$

Utilizing the above equation, we could correct for the discrepancy between the expected value of the exponential moving average at timestep *t* and the true second moment by taking the expectations of both sides of the equation (Kingman and Ba, 2015, p. 3).

$$\begin{aligned}
\mathbb{E}[v_t] &= \mathbb{E}\left[(1 - \beta_2) \sum_{i=1}^{t} \beta_2^{t-i} \cdot g_i^2\right] \\
&= \mathbb{E}[g_t^2] \cdot (1 - \beta_2) \sum_{i=1}^{t} \beta_2^{t-i} + \zeta \\
&= \mathbb{E}[g_t^2] \cdot (1 - \beta_2^t) + \zeta
\end{aligned}$$

Finally, the Adam optimization algorithm employs the convergence algorithm proposed by Zinkevich in 2003 by utilizing regret. Regret is defined as (Kingman and Ba, 2015, p. 4):

$$R(T) = \sum_{t=1}^{T} [f_t(\theta_t) - f_t(\theta^*)]$$

The theorem that authors Kingman and Ba propose states (2015, p. 4):

Theorem 4.1. *Assume that the function f_t has bounded gradients, $\|\nabla f_t(\theta)\|_2 \leq G$, $\|\nabla f_t(\theta)\|_\infty \leq G_\infty$ for all $\theta \in R^d$ and distance between any θ_t generated by Adam is bounded, $\|\theta_n - \theta_m\|_2 \leq D$, $\|\theta_m - \theta_n\|_\infty \leq D_\infty$ for any $m, n \in \{1, ..., T\}$, and β_1, $\beta_2 \in [0, 1)$ satisfy $\frac{\beta_1^2}{\sqrt{\beta_2}} < 1$. Let $\alpha_t = \frac{\alpha}{\sqrt{t}}$ and $\beta_{1,t} = \beta_1 \lambda^{t-1}, \lambda \in (0, 1)$. Adam achieves the following guarantee, for all $T \geq 1$.*

$$R(T) \leq \frac{D^2}{2\alpha(1 - \beta_1)} \sum_{i=1}^{d} \sqrt{T\widehat{v}_{T,i}} + \frac{\alpha(1 + \beta_1) G_\infty}{(1 - \beta_1)\sqrt{1 - \beta_2}(1 - \gamma)^2} \sum_{i=1}^{d} \|g_{1:T,i}\|_2 + \sum_{i=1}^{d} \frac{D_\infty^2 G_\infty \sqrt{1 - \beta_2}}{2\alpha(1 - \beta_1)(1 - \lambda)^2}$$

Adam optimization algorithm converges when the average of regret is taken (Kingman and Ba, 2015, p. 4):

Corollary 4.2. *Assume that the function f_t has bounded gradients, $\|\nabla f_t(\theta)\|_2 \leq G$, $\|\nabla f_t(\theta)\|_\infty \leq G_\infty$ for all $\theta \in R^d$ and distance between any θ_t generated by Adam is bounded, $\|\theta_n - \theta_m\|_2 \leq D$, $\|\theta_m - \theta_n\|_\infty \leq D_\infty$ for any $m, n \in \{1, ..., T\}$. Adam achieves the following guarantee, for all $T \geq 1$.*

$$\frac{R(T)}{T} = O(\frac{1}{\sqrt{T}})$$

This result can be obtained by using Theorem 4.1 and $\sum_{i=1}^{d} \|g_{1:T,i}\|_2 \leq dG_\infty\sqrt{T}$. Thus, $\lim_{T\to\infty} \frac{R(T)}{T} = 0$.

The full mathematical proof demonstrating how and why this backpropagation learning method works can be found in the Appendix.

Applying these mathematical concepts to our airline customer satisfaction survey results, we find that the Adam optimization algorithm performs well as the “solver” in our Multilayer Perceptron Network, with hidden layer sizes of 10 - 10 - 10. The Classification Report and Confusion Matrix below shows the precision, recall, F1-score, and accuracy of the model.

	precision	recall	f1-score	support
0	0.91	0.90	0.90	11751
1	0.92	0.92	0.92	14147
accuracy			0.91	25898
macro avg	0.91	0.91	0.91	25898
weighted avg	0.91	0.91	0.91	25898

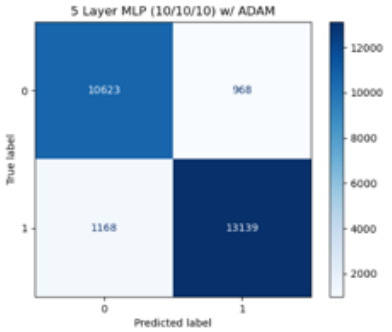

Figure 18. Classification Report and Confusion Matrix output of the Multilayer Perceptron Network with the Adam optimization algorithm.

Next in our experimental design, we will implement the quasi-Newton method of the Limited-memory Broyden–Fletcher–Goldfarb–Shanno (L-BFGS) optimization algorithm. This batch method of backpropagation uses “approximated second order gradient information which provides a faster convergence towards the minimum” (Najafabadi et. al, 2017, p. 2). The advantage of the L-BFGS algorithm is that it only uses a limited amount of memory, which is well suited for our dataset, as we have so many features. As stated by Xiao et. al, this “at each iteration, the [L-BFGS] method requires no more function or derivative evaluations, and hardly more storage or arithmetic operations” (2008, p. 1002). The pseudo-code to implement this algorithm is as follows:

Algorithm 1 L-BFGS

procedure L-BFGS
  Choose starting point x_0, and integer $m > 0$
  $k \leftarrow 0$
  while true **do**
    Calculate $\Delta f(x_k)$ at the current point x_k
    Calculate p_k using Algorithm 2
    Calculate α_k where it satisfies Wolfe conditions
    $x_{k+1} \leftarrow x_k + \alpha_k p_k$
    if $k > m$ **then**
      Discard the vector pair $\{S_{k-m}, y_{k-m}\}$ from storage
    end if
    Compute and Save $s_k = x_{k+1} - x_k$ and $y_k = \Delta f_{k+1} - \Delta f_k$
    $k \leftarrow k + 1$
  end while
end procedure

Figure 19. L-BFGS pseudo-code (Najafabadi et. al, 2017, p. 9).

The L-BFGS algorithm is a modification of the BFGS algorithm by obtaining “Hessian approximations that can be stored in just a few vectors of the length *n*” (Najafabadi et. al, 2017, p. 8). The Hessian matrix is therefore approximated using the following formula:

$$\begin{aligned} H_{k+1} &= V_k^{\mathrm{T}} H_k V_k + \rho_k s_k s_k^{\mathrm{T}} \\ &= V_k^{\mathrm{T}} [V_{k-1}^{\mathrm{T}} H_{k-1} V_{k-1} + \rho_{k-1} s_{k-1} s_{k-1}^{\mathrm{T}}] V_k + \rho_k s_k s_k^{\mathrm{T}} \\ &= \cdots \\ &= [V_k^{\mathrm{T}} \cdots V_{k-\widehat{m}+1}] H_{k-\widehat{m}+1} [V_{k-\widehat{m}+1} \cdots V_k] \\ &\quad + \rho_{k-\widehat{m}+1} [V_{k-1}^{\mathrm{T}} \cdots V_{k-\widehat{m}+2}^{\mathrm{T}}] s_{k-\widehat{m}+1} s_{k-\widehat{m}+1}^{\mathrm{T}} [V_{k-\widehat{m}+2} \cdots V_{k-1}] + \cdots + \rho_k s_k s_k^{\mathrm{T}}. \end{aligned}$$

Figure 20. Updated Hessian Matrix formula for L-BFGS (Xiao et. al, 2008, p. 1004).

The theorem utilized by Liu and Nocedal when they first created the L-BFGS optimization algorithm states (1989, p. 22):

Theorem 6.1 *Let x_0 be a starting point for which f satisfies Assumptions 6.1, and assume that the matrices $B_k^{(0)}$ are chosen so that $\{\|B_k^{(0)}\|\}$ and $\{\|B_k^{(0)^{-1}}\|\}$ are bounded. Then for any positive definite B_0, Algorithm 6.1 generates a sequence $\{x_k\}$ which converges to x_*. Moreover there is a constant $0 \leq r < 1$ such that*

$$f_k - f_* \leq r^k [f_0 - f_*], \tag{6.5}$$

which implies that $\{x_k\}$ converges R-linearly.

The full mathematical proof on how the L-BFGS algorithm works can be found in the appendix.

Applying this optimization algorithm to our airline customer satisfaction survey results, we find that implementing the L-BFGS “solver” in our Multilayer Perceptron Network with hidden layers sizes of 10 - 10 - 10 provides slightly better results in the Classification Report. In the figure below, you can see that the precision, recall, F1-score, and accuracy all have marginally improved results compared to the Adam optimization algorithm.

	precision	recall	f1-score	support
0	0.91	0.91	0.91	11751
1	0.93	0.92	0.92	14147
accuracy			0.92	25898
macro avg	0.92	0.92	0.92	25898
weighted avg	0.92	0.92	0.92	25898

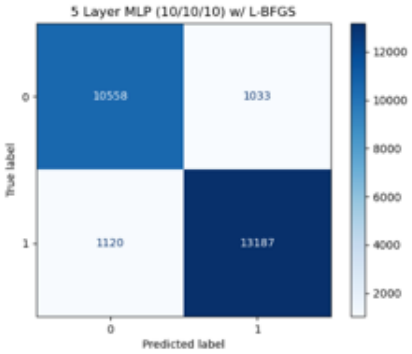

Figure 21. Classification Report and Confusion Matrix output of the Multilayer Perceptron Network with the L-BFGS optimization algorithm.

Last but not least, we will compare the results of these two optimization algorithms with the classic Stochastic Gradient Descent (SGD) backpropagation algorithm. The stochastic gradient descent trains the network incrementally, after each input is entered into the network (Hagan et. al, 2016, p. 11-17). SGD has been studied for decades, as it is an efficient, effective, and highly scalable optimization method. The SGD algorithm can be written mathematically as (Shamir and Zhang, 2013, p. 71):

$$\mathbf{w}_{t+1} = \Pi_{\mathcal{W}}(\mathbf{w}_t - \eta_t \hat{\mathbf{g}}_t)$$

The full mathematical proof of how Stochastic Gradient Descent algorithm works can be found in the appendix. Applying SGD as the “solver” to our Multilayer Perceptron Network with hidden layer sizes of 10 - 10 - 10, we see that the precision, recall, F1-score, and accuracy are all marginally worse than the L-BFGS optimization algorithm and about the same as the network using the Adam optimization algorithm.

	precision	recall	f1-score	support
0	0.90	0.91	0.90	11692
1	0.92	0.92	0.92	14206
accuracy			0.91	25898
macro avg	0.91	0.91	0.91	25898
weighted avg	0.91	0.91	0.91	25898

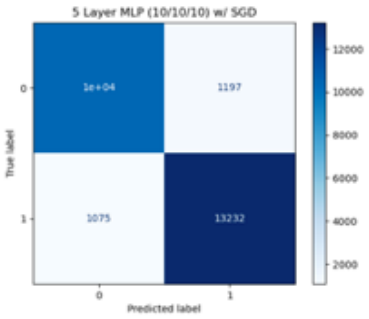

Figure 22. Classification Report and Confusion Matrix output of the Multilayer Perceptron Network with the SGD optimization algorithm.

Now that we have discovered that L-BFGS is the best optimization algorithm for our dataset, we sought out different ways to improve the performance of the network. First, we tried changing the network architecture from three hidden layers to four hidden layers. This seemed to have little to no effect on improving the model and actually lowered the precision score. See the Classification Report and Confusion Matrix in figure 23 for detailed output information.

	precision	recall	f1-score	support
0	0.90	0.91	0.91	11751
1	0.93	0.92	0.92	14147
accuracy			0.92	25898
macro avg	0.92	0.92	0.92	25898
weighted avg	0.92	0.92	0.92	25898

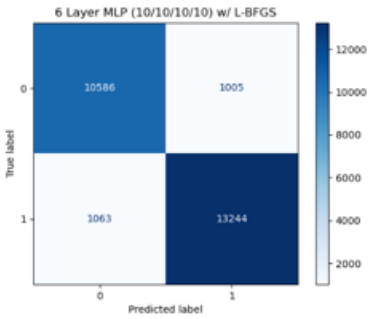

Figure 23. Classification Report and Confusion Matrix output for the Multilayer Perceptron Network with Four Hidden Layers and the L-BFGS optimization algorithm.

Given the four hidden layer network did not have any effect on the performance of the model, we returned to the network with three hidden layers, but this time increased the number of neurons in the second hidden layer to 15, as recommended by Goodfellow, Bengio, and Courville (2016). As is apparent from the Classification Report in Figure 24 below, there network performance increased marginally:

	precision	recall	f1-score	support
0	0.91	0.92	0.91	11751
1	0.93	0.92	0.93	14147
accuracy			0.92	25898
macro avg	0.92	0.92	0.92	25898
weighted avg	0.92	0.92	0.92	25898

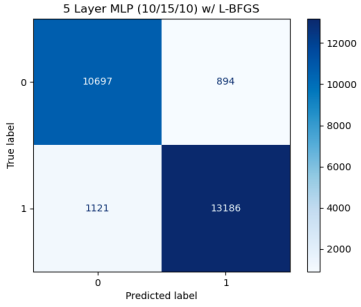

Figure 24. Classification Report and Confusion Matrix output for the Multilayer Perceptron Network with the L-BFGS optimization algorithm with hidden layer sizes of 10 - 15 - 10.

Noting that the increase in the number of neurons improved the output of the model, we then chose to increase the number of neurons in each hidden layer from 10 to 15. Below in figure 25, we achieve the highest precision and F1 scores of all the models in our analysis, gaining about 0.01 points across all metrics:

	precision	recall	f1-score	support
0	0.91	0.92	0.92	11591
1	0.94	0.92	0.93	14307
accuracy			0.92	25898
macro avg	0.92	0.92	0.92	25898
weighted avg	0.92	0.92	0.92	25898

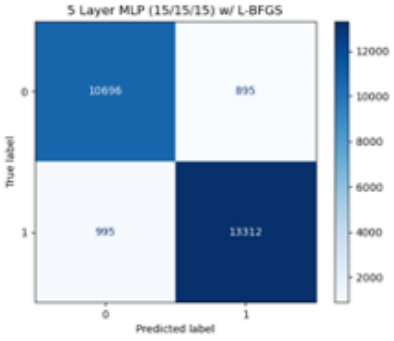

Figure 25. Classification Report and Confusion Matrix output for the Multilayer Perceptron Network with the L-BFGS optimization algorithm with hidden layer sizes of 15 - 15 - 15.

We next sought to test whether changing the activation function would have an impact on the performance of the Multilayer Perceptron Network with the L-BFGS optimization algorithm, keeping the hidden layer sizes of 15 - 15 - 15. Ultimately, as you can see in the Classification Report and Confusion Matrix below in Figure 26, this model with the hidden layer sizes of 15 - 15 - 15 that uses the L-BFGS optimization algorithm as the “solver” and the logistic activation function was the best performing model in our analysis.

	precision	recall	f1-score	support
0	0.91	0.93	0.92	11751
1	0.94	0.93	0.93	14147
accuracy			0.93	25898
macro avg	0.93	0.93	0.93	25898
weighted avg	0.93	0.93	0.93	25898

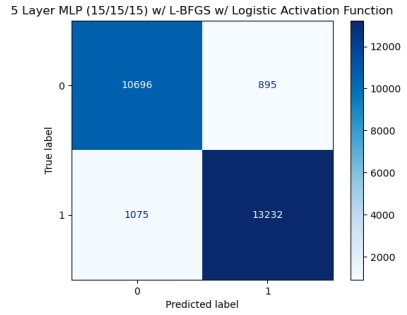

Figure 26. Classification Report and Confusion Matrix output for the Multilayer Perceptron Network with the L-BFGS optimization algorithm with hidden layer sizes of 15 - 15 - 15 and Logistic activation function in the hidden layers.

For a final test, we wanted to return to the SGD optimization algorithm and adjust the momentum and alpha to see if this improved the performance of the model. Changing the “solver” of the network back to SGD, we updated the number of neurons in each hidden layer to 15 - 15 - 15, the momentum to a value of 0.95, and the alpha to 0.01. While the performance of the updated SGD network is marginally better than the initial SGD network we created, it still did not outperform the updated L-BFGS network above. Figure 27 below shows the full Classification Report and Confusion Matrix output of the updated SGD network.

	precision	recall	f1-score	support
0	0.90	0.92	0.91	11751
1	0.93	0.92	0.93	14147
accuracy			0.92	25898
macro avg	0.92	0.92	0.92	25898
weighted avg	0.92	0.92	0.92	25898

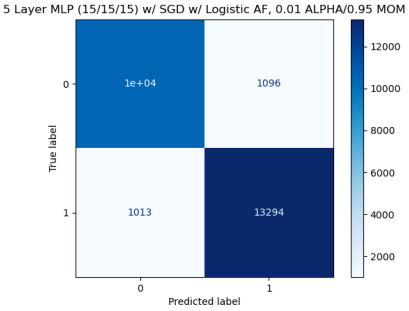

Figure 27. Classification Report and Confusion Matrix output for the Multilayer Perceptron Network with the SGD optimization algorithm, hidden layer sizes of 15 - 15 - 15, momentum of 0.95, and alpha of 0.01.

For comparison purposes, the listed modifications and performance of each model are shown in the appendix.

Returning to Endresen and Janda’s assertion that Random Forest classification models are better than parametric models for understanding Likert-type scale survey responses (2016), we used the airline customer satisfaction survey results through a classic Random Forest model for comparison. The accuracy score we achieved in the Random Forest model of these features was 0.92, quite close to the L-BFGS Multilayer Perceptron Network with hidden layer sizes of 15

- 15 - 15 and a logistic activation function. While Endresen and Janda ran multiple statistical tests with their Likert-type scale dataset, they did not create a neural network to analyze these types of questions. Our contribution to this area of study, therefore, is that a Multilayer Perceptron Network with three hidden layers of 15 neurons in each layer that uses a L-BFGS optimization backpropagation algorithm and logistic activation function provides more accurate results than the Random Forest model. However, given the complexity of this neural network, we agree with Endresen and Janda's argument that the Random Forest model is more user-friendly for a lay person to understand, and at times may be more informative and appropriate for the data (2016).

Over the course of the project, we learned so much about machine learning, backpropagation optimization algorithms, and the mathematics that makes these networks possible. In addition to the SGD algorithm we learned about in class, we implemented two additional learning algorithms in our project, Adam and L-BFGS. As shown in the appendix, we not only applied them to our dataset, but we also learned the mathematical proofs behind the algorithms. We realized the importance of fine-tuning the network through trial and error, changing both the network architecture as well as the number of neurons in each hidden layer. Comparing our best network to the classical Random Forest model, we found that our model marginally outperformed the classical Random Forest model in terms of accuracy.

We also discovered helpful insights about customer satisfaction with airlines. We found that satisfaction with the inflight experience leads to greater overall satisfaction with the airline, even more than satisfaction with service before and after the flight. In particular, customer satisfaction is most highly correlated with seat comfort and inflight entertainment. Meanwhile, features such as satisfaction with the gate location and the convenience of the departure/arrival time were only moderately correlated with overall satisfaction. Therefore, we recommend that airlines invest in improving the customer experience inflight if they hope to improve customer satisfaction overall, rather than investing in improvements before or after the flight.

While we assert that the classical Random Forest model is the best model to use to communicate the findings of surveys with Likert-type scale questions, further study of this dataset and these neural networks is needed. Increasing the sample size could have dramatically different results. Additionally, examining the customer satisfaction of each airline individually could provide alternate results, as some airlines do not offer inflight entertainment services. We would also like to try additional classical models for comparison with our neural networks, such as Classification Trees or Naive Bayesian models. Furthermore, additional tuning of the model could discover improvements in our findings, as we could not possibly go through every possible parameter update for the purpose of this project. This would allow us to see if we are truly at the ceiling of the dataset.

References

D, J. (2018, June 10). *Passenger Satisfaction*. Kaggle. Retrieved on 13 June 2021 from https://www.kaggle.com/johnddddd/customer-satisfaction.

Das, S. (2021, June 11). *Airline Passenger Satisfaction - Prediction > 95%*. Kaggle. Retrieved on 13 June 2021 from https://www.kaggle.com/codesagnik/airline-passenger-satisfaction-prediction-95.

Endresen, A., & Janda, L. A. (2016). Five statistical models for Likert-type experimental data on acceptability judgments. *Journal of Research Design and Statistics in Linguistics and Communication Science*, 3(2). Retrieved on 20 June 2021 from https://doi.org/10.1558/jrds.v2i1.2269

Goodfellow, I., Bengio, Y., & Courville, A. (2017). 6.5 Back-Propagation and Other Differentiation Algorithms. In *Deep Learning* (pp. 200–220). MIT Press.

Hagan, M. T., Demuth, H. B., Beale, M. H., & De Jesús, O. (2016). *Neural network design*. s. n. https://hagan.okstate.edu/nnd.html.

Jais, I. K. M., Ismail, A. R., & Nisa, S. Q. (2019). Adam Optimization Algorithm for Wide and Deep Neural Network . *Knowledge Engineering and Data Science (KEDS)* , 2(1), 41–46. https://core.ac.uk/download/pdf/287322851.pdf

Kingma, D. P., & Ba, J. (2017, January 30). *Adam: A Method for Stochastic Optimization.* arXiv.org. https://arxiv.org/abs/1412.6980.

Lavrakas, P. J. (2008). *Encyclopedia of Survey Research Methods*. Thousand Oaks, CA: SAGE Publications. https://doi.org/10.4135/9781412963947

Likert, R. (1932). A Technique for the Measurement of Attitudes. Doctoral dissertation. Columbia University. Series Archives of Psychology 22: 5–55. *NY: The Science Press.* Retrieved on 20 June 2021 from http://www.voteview.com/pdf/Likert_1932.pdf

Liu, D. C., & Nocedal, J. (1989). On the limited memory BFGS method for large scale optimization. *Mathematical Programming: Series A and B*, 45(1-3), 503–528. https://doi.org/10.5555/3112655.3112866

Moritz, P., Nishihara, R., & Jordan, M. I. (2016, April 13). *A Linearly-Convergent Stochastic L-BFGS Algorithm*. arXiv.org. https://arxiv.org//abs/1508.02087.

Najafabadi, M. M., Khoshgoftaar, T. M., Villanustre, F., & Holt, J. (2017). Large-scale distributed L-BFGS. *Journal of Big Data,* 4(22). https://doi.org/10.1186/s40537-017-0084-5

Rafati, J., & Marcia, R. F. (2018). Improving L-BFGS Initialization for Trust-Region Methods in Deep Learning. *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*. https://doi.org/10.1109/icmla.2018.00081

Rumelhart, D. E., Group, P. D. P. R., & McClelland, J. L. (1986). *Parallel Distributed Processing, Explorations in the Microstructure of Cognition: Foundations (Vol. 1)*. MIT Press.

Saad, D. (1998). *On-line learning in neural networks*. Cambridge.

Shamir, O. & Zhang,T. (2013). Stochastic Gradient Descent for Non-smooth Optimization: Convergence Results and Optimal Averaging Schemes. *Journal of Machine Learning Research: W & CP*: 28, p. 71-79. http://proceedings.mlr.press/v28/shamir13.pdf.

Xiao, Y., Wei, Z., Wang, Z. (2008). A limited memory BFGS-type method for large-scale unconstrained optimization. *Computers & Mathematics with Applications*, 56(4), 1001-1009. https://doi.org/10.1016/j.camwa.2008.01.028.

Appendix

Comparison Table of all Multilayer Perceptron Networks and their Performance:

Hidden Layers	Hidden Layer Transfer Function	Learning Method	Alpha	Momentum	Precision	Recall	F1 Score	Accuracy
10/10/10	relu	ADAM	0.0001	0.9	0.90 0.93	0.92 0.92	0.91 0.92	0.92
10/10/10	relu	L-BFGS	0.0001	0.9	0.90 0.93	0.91 0.92	0.91 0.92	0.92
10/10/10	relu	SGD	0.0001	0.9	0.90 0.93	0.91 0.92	0.90 0.92	0.91
10/10/10/10	relu	L-BFGS	0.0001	0.9	0.90 0.93	0.91 0.92	0.91 0.92	0.92
10/15/10	relu	L-BFGS	0.0001	0.9	0.91 0.93	0.92 0.92	0.91 0.93	0.92
15/15/15	relu	L-BFGS	0.0001	0.9	0.91 0.94	0.93 0.92	0.92 0.93	0.92
20/20/20	relu	L-BFGS	0.0001	0.9	0.91 0.94	0.92 0.92	0.92 0.93	0.92
15/15/15	logistic	L-BFGS	0.0001	0.9	0.91 0.94	0.93 0.93	0.92 0.93	0.93
15/15/15	relu	SGD	0.01	0.95	0.91 0.93	0.91 0.93	0.91 0.93	0.92

Mathematical Proof of Convergence for the Adam optimization algorithm (Kingman and Ba, 2015, p. 12-15).

Definition 10.1. *A function $f : R^d \to R$ is convex if for all $x, y \in R^d$, for all $\lambda \in [0, 1]$,*

$$\lambda f(x) + (1 - \lambda) f(y) \geq f(\lambda x + (1 - \lambda) y)$$

Also, notice that a convex function can be lower bounded by a hyperplane at its tangent.

Lemma 10.2. *If a function $f : R^d \to R$ is convex, then for all $x, y \in R^d$,*

$$f(y) \geq f(x) + \nabla f(x)^T (y - x)$$

The above lemma can be used to upper bound the regret and our proof for the main theorem is constructed by substituting the hyperplane with the Adam update rules.

The following two lemmas are used to support our main theorem. We also use some definitions simplify our notation, where $g_t \triangleq \nabla f_t(\theta_t)$ and $g_{t,i}$ as the i^{th} element. We define $g_{1:t,i} \in \mathbb{R}^t$ as a vector that contains the i^{th} dimension of the gradients over all iterations till t, $g_{1:t,i} = [g_{1,i}, g_{2,i}, \cdots, g_{t,i}]$

Lemma 10.3. *Let $g_t = \nabla f_t(\theta_t)$ and $g_{1:t}$ be defined as above and bounded, $\|g_t\|_2 \leq G$, $\|g_t\|_\infty \leq G_\infty$. Then,*

$$\sum_{t=1}^{T} \sqrt{\frac{g_{t,i}^2}{t}} \leq 2 G_\infty \|g_{1:T,i}\|_2$$

Proof. We will prove the inequality using induction over T.

The base case for $T = 1$, we have $\sqrt{g_{1,i}^2} \leq 2 G_\infty \|g_{1,i}\|_2$.

For the inductive step,

$$\begin{aligned}
\sum_{t=1}^{T} \sqrt{\frac{g_{t,i}^2}{t}} &= \sum_{t=1}^{T-1} \sqrt{\frac{g_{t,i}^2}{t}} + \sqrt{\frac{g_{T,i}^2}{T}} \\
&\leq 2 G_\infty \|g_{1:T-1,i}\|_2 + \sqrt{\frac{g_{T,i}^2}{T}} \\
&= 2 G_\infty \sqrt{\|g_{1:T,i}\|_2^2 - g_T^2} + \sqrt{\frac{g_{T,i}^2}{T}}
\end{aligned}$$

From, $\|g_{1:T,i}\|_2^2 - g_{T,i}^2 + \frac{g_{T,i}^4}{4\|g_{1:T,i}\|_2^2} \geq \|g_{1:T,i}\|_2^2 - g_{T,i}^2$, we can take square root of both side and have,

$$\begin{aligned}
\sqrt{\|g_{1:T,i}\|_2^2 - g_{T,i}^2} &\leq \|g_{1:T,i}\|_2 - \frac{g_{T,i}^2}{2\|g_{1:T,i}\|_2} \\
&\leq \|g_{1:T,i}\|_2 - \frac{g_{T,i}^2}{2\sqrt{T G_\infty^2}}
\end{aligned}$$

Rearrange the inequality and substitute the $\sqrt{\|g_{1:T,i}\|_2^2 - g_{T,i}^2}$ term,

$$G_\infty \sqrt{\|g_{1:T,i}\|_2^2 - g_T^2} + \sqrt{\frac{g_{T,i}^2}{T}} \leq 2 G_\infty \|g_{1:T,i}\|_2$$

Lemma 10.4. *Let* $\gamma \triangleq \frac{\beta_1^2}{\sqrt{\beta_2}}$. *For* $\beta_1, \beta_2 \in [0, 1)$ *that satisfy* $\frac{\beta_1^2}{\sqrt{\beta_2}} < 1$ *and bounded* g_t, $\|g_t\|_2 \leq G$, $\|g_t\|_\infty \leq G_\infty$, *the following inequality holds*

$$\sum_{t=1}^{T} \frac{\widehat{m}_{t,i}^2}{\sqrt{t\widehat{v}_{t,i}}} \leq \frac{2}{1-\gamma} \frac{1}{\sqrt{1-\beta_2}} \|g_{1:T,i}\|_2$$

Proof. Under the assumption, $\frac{\sqrt{1-\beta_2^t}}{(1-\beta_1^t)^2} \leq \frac{1}{(1-\beta_1)^2}$. We can expand the last term in the summation using the update rules in Algorithm 1,

$$\begin{aligned}
\sum_{t=1}^{T} \frac{\widehat{m}_{t,i}^2}{\sqrt{t\widehat{v}_{t,i}}} &= \sum_{t=1}^{T-1} \frac{\widehat{m}_{t,i}^2}{\sqrt{t\widehat{v}_{t,i}}} + \frac{\sqrt{1-\beta_2^T}}{(1-\beta_1^T)^2} \frac{(\sum_{k=1}^{T}(1-\beta_1)\beta_1^{T-k} g_{k,i})^2}{\sqrt{T\sum_{j=1}^{T}(1-\beta_2)\beta_2^{T-j} g_{j,i}^2}} \\
&\leq \sum_{t=1}^{T-1} \frac{\widehat{m}_{t,i}^2}{\sqrt{t\widehat{v}_{t,i}}} + \frac{\sqrt{1-\beta_2^T}}{(1-\beta_1^T)^2} \sum_{k=1}^{T} \frac{T((1-\beta_1)\beta_1^{T-k} g_{k,i})^2}{\sqrt{T\sum_{j=1}^{T}(1-\beta_2)\beta_2^{T-j} g_{j,i}^2}} \\
&\leq \sum_{t=1}^{T-1} \frac{\widehat{m}_{t,i}^2}{\sqrt{t\widehat{v}_{t,i}}} + \frac{\sqrt{1-\beta_2^T}}{(1-\beta_1^T)^2} \sum_{k=1}^{T} \frac{T((1-\beta_1)\beta_1^{T-k} g_{k,i})^2}{\sqrt{T(1-\beta_2)\beta_2^{T-k} g_{k,i}^2}} \\
&\leq \sum_{t=1}^{T-1} \frac{\widehat{m}_{t,i}^2}{\sqrt{t\widehat{v}_{t,i}}} + \frac{\sqrt{1-\beta_2^T}}{(1-\beta_1^T)^2} \frac{(1-\beta_1)^2}{\sqrt{T(1-\beta_2)}} \sum_{k=1}^{T} T\left(\frac{\beta_1^2}{\sqrt{\beta_2}}\right)^{T-k} \|g_{k,i}\|_2 \\
&\leq \sum_{t=1}^{T-1} \frac{\widehat{m}_{t,i}^2}{\sqrt{t\widehat{v}_{t,i}}} + \frac{T}{\sqrt{T(1-\beta_2)}} \sum_{k=1}^{T} \gamma^{T-k} \|g_{k,i}\|_2
\end{aligned}$$

Similarly, we can upper bound the rest of the terms in the summation.

$$\begin{aligned}
\sum_{t=1}^{T} \frac{\widehat{m}_{t,i}^2}{\sqrt{t\widehat{v}_{t,i}}} &\leq \sum_{t=1}^{T} \frac{\|g_{t,i}\|_2}{\sqrt{t(1-\beta_2)}} \sum_{j=0}^{T-t} t\gamma^j \\
&\leq \sum_{t=1}^{T} \frac{\|g_{t,i}\|_2}{\sqrt{t(1-\beta_2)}} \sum_{j=0}^{T} t\gamma^j
\end{aligned}$$

For $\gamma < 1$, using the upper bound on the arithmetic-geometric series, $\sum_t t\gamma^t < \frac{1}{(1-\gamma)^2}$:

$$\sum_{t=1}^{T} \frac{\|g_{t,i}\|_2}{\sqrt{t(1-\beta_2)}} \sum_{j=0}^{T} t\gamma^j \leq \frac{1}{(1-\gamma)^2\sqrt{1-\beta_2}} \sum_{t=1}^{T} \frac{\|g_{t,i}\|_2}{\sqrt{t}}$$

Apply Lemma 10.3,

$$\sum_{t=1}^{T} \frac{\widehat{m}_{t,i}^2}{\sqrt{t\widehat{v}_{t,i}}} \leq \frac{2G_\infty}{(1-\gamma)^2\sqrt{1-\beta_2}} \|g_{1:T,i}\|_2$$

□

To simplify the notation, we define $\gamma \triangleq \frac{\beta_1^2}{\sqrt{\beta_2}}$. Intuitively, our following theorem holds when the learning rate α_t is decaying at a rate of $t^{-\frac{1}{2}}$ and first moment running average coefficient $\beta_{1,t}$ decay exponentially with λ, that is typically close to 1, e.g. $1 - 10^{-8}$.

Theorem 10.5. *Assume that the function* f_t *has bounded gradients,* $\|\nabla f_t(\theta)\|_2 \leq G$, $\|\nabla f_t(\theta)\|_\infty \leq G_\infty$ *for all* $\theta \in R^d$ *and distance between any* θ_t *generated by Adam is bounded,* $\|\theta_n - \theta_m\|_2 \leq D$,

$\|\theta_m - \theta_n\|_\infty \leq D_\infty$ *for any* $m, n \in \{1, ..., T\}$, *and* β_1, $\beta_2 \in [0, 1)$ *satisfy* $\frac{\beta_1}{\sqrt{\beta_2}} < 1$. *Let* $\alpha_t = \frac{\alpha}{\sqrt{t}}$ *and* $\beta_{1,t} = \beta_1 \lambda^{t-1}, \lambda \in (0, 1)$. *Adam achieves the following guarantee, for all* $T \geq 1$.

$$R(T) \leq \frac{D^2}{2\alpha(1-\beta_1)} \sum_{i=1}^{d} \sqrt{T\widehat{v}_{T,i}} + \frac{\alpha(\beta_1+1)G_\infty}{(1-\beta_1)\sqrt{1-\beta_2}(1-\gamma)^2} \sum_{i=1}^{d} \|g_{1:T,i}\|_2 + \sum_{i=1}^{d} \frac{D_\infty^2 G_\infty \sqrt{1-\beta_2}}{2\alpha(1-\beta_1)(1-\lambda)^2}$$

Proof. Using Lemma 10.2, we have,

$$f_t(\theta_t) - f_t(\theta^*) \leq g_t^T(\theta_t - \theta^*) = \sum_{i=1}^{d} g_{t,i}(\theta_{t,i} - \theta_{,i}^*)$$

From the update rules presented in algorithm 1,

$$\begin{aligned}\theta_{t+1} &= \theta_t - \alpha_t \widehat{m}_t / \sqrt{\widehat{v}_t} \\ &= \theta_t - \frac{\alpha_t}{1-\beta_1^t}\left(\frac{\beta_{1,t}}{\sqrt{\widehat{v}_t}} m_{t-1} + \frac{(1-\beta_{1,t})}{\sqrt{\widehat{v}_t}} g_t\right)\end{aligned}$$

We focus on the i^{th} dimension of the parameter vector $\theta_t \in R^d$. Subtract the scalar $\theta_{,i}^*$ and square both sides of the above update rule, we have,

$$(\theta_{t+1,i} - \theta_{,i}^*)^2 = (\theta_{t,i} - \theta_{,i}^*)^2 - \frac{2\alpha_t}{1-\beta_1^t}\left(\frac{\beta_{1,t}}{\sqrt{\widehat{v}_{t,i}}} m_{t-1,i} + (1 - \frac{\beta_{1,t})}{\sqrt{\widehat{v}_{t,i}}} g_{t,i}\right)(\theta_{t,i} - \theta_{,i}^*) + \alpha_t^2 \left(\frac{\widehat{m}_{t,i}}{\sqrt{\widehat{v}_{t,i}}}\right)^2$$

We can rearrange the above equation and use Young's inequality, $ab \leq a^2/2 + b^2/2$. Also, it can be shown that $\sqrt{\widehat{v}_{t,i}} = \sqrt{\sum_{j=1}^{t}(1-\beta_2)\beta_2^{t-j} g_{j,i}^2} / \sqrt{1-\beta_2^t} \leq \|g_{1:t,i}\|_2$ and $\beta_{1,t} \leq \beta_1$. Then

$$\begin{aligned} g_{t,i}(\theta_{t,i} - \theta_{,i}^*) =& \frac{(1-\beta_1^t)\sqrt{\widehat{v}_{t,i}}}{2\alpha_t(1-\beta_{1,t})}\left((\theta_{t,i} - \theta_{,t}^*)^2 - (\theta_{t+1,i} - \theta_{,i}^*)^2\right) \\ &+ \frac{\beta_{1,t}}{(1-\beta_{1,t})} \frac{\widehat{v}_{t-1,i}^{\frac{1}{4}}}{\sqrt{\alpha_{t-1}}}(\theta_{,i}^* - \theta_{t,i})\sqrt{\alpha_{t-1}} \frac{m_{t-1,i}}{\widehat{v}_{t-1,i}^{\frac{1}{4}}} + \frac{\alpha_t(1-\beta_1^t)\sqrt{\widehat{v}_{t,i}}}{2(1-\beta_{1,t})}\left(\frac{\widehat{m}_{t,i}}{\sqrt{\widehat{v}_{t,i}}}\right)^2 \\ \leq& \frac{1}{2\alpha_t(1-\beta_1)}\left((\theta_{t,i} - \theta_{,t}^*)^2 - (\theta_{t+1,i} - \theta_{,i}^*)^2\right)\sqrt{\widehat{v}_{t,i}} + \frac{\beta_{1,t}}{2\alpha_{t-1}(1-\beta_{1,t})}(\theta_{,i}^* - \theta_{t,i})^2\sqrt{\widehat{v}_{t-1,i}} \\ &+ \frac{\beta_1 \alpha_{t-1}}{2(1-\beta_1)} \frac{m_{t-1,i}^2}{\sqrt{\widehat{v}_{t-1,i}}} + \frac{\alpha_t}{2(1-\beta_1)} \frac{\widehat{m}_{t,i}^2}{\sqrt{\widehat{v}_{t,i}}} \end{aligned}$$

We apply Lemma 10.4 to the above inequality and derive the regret bound by summing across all the dimensions for $i \in 1, ..., d$ in the upper bound of $f_t(\theta_t) - f_t(\theta^*)$ and the sequence of convex functions for $t \in 1, ..., T$:

$$\begin{aligned} R(T) \leq& \sum_{i=1}^{d} \frac{1}{2\alpha_1(1-\beta_1)}(\theta_{1,i} - \theta_{,i}^*)^2\sqrt{\widehat{v}_{1,i}} + \sum_{i=1}^{d}\sum_{t=2}^{T} \frac{1}{2(1-\beta_1)}(\theta_{t,i} - \theta_{,i}^*)^2\left(\frac{\sqrt{\widehat{v}_{t,i}}}{\alpha_t} - \frac{\sqrt{\widehat{v}_{t-1,i}}}{\alpha_{t-1}}\right) \\ &+ \frac{\beta_1 \alpha G_\infty}{(1-\beta_1)\sqrt{1-\beta_2}(1-\gamma)^2} \sum_{i=1}^{d} \|g_{1:T,i}\|_2 + \frac{\alpha G_\infty}{(1-\beta_1)\sqrt{1-\beta_2}(1-\gamma)^2} \sum_{i=1}^{d} \|g_{1:T,i}\|_2 \\ &+ \sum_{i=1}^{d}\sum_{t=1}^{T} \frac{\beta_{1,t}}{2\alpha_t(1-\beta_{1,t})}(\theta_{,i}^* - \theta_{t,i})^2\sqrt{\widehat{v}_{t,i}} \end{aligned}$$

From the assumption, $\|\theta_t - \theta^*\|_2 \leq D$, $\|\theta_m - \theta_n\|_\infty \leq D_\infty$, we have:

$$\begin{aligned} R(T) \leq & \frac{D^2}{2\alpha(1-\beta_1)} \sum_{i=1}^{d} \sqrt{T\widehat{v}_{T,i}} + \frac{\alpha(1+\beta_1)G_\infty}{(1-\beta_1)\sqrt{1-\beta_2}(1-\gamma)^2} \sum_{i=1}^{d} \|g_{1:T,i}\|_2 + \frac{D_\infty^2}{2\alpha} \sum_{i=1}^{d} \sum_{t=1}^{t} \frac{\beta_{1,t}}{(1-\beta_{1,t})} \sqrt{t\widehat{v}_{t,i}} \\ \leq & \frac{D^2}{2\alpha(1-\beta_1)} \sum_{i=1}^{d} \sqrt{T\widehat{v}_{T,i}} + \frac{\alpha(1+\beta_1)G_\infty}{(1-\beta_1)\sqrt{1-\beta_2}(1-\gamma)^2} \sum_{i=1}^{d} \|g_{1:T,i}\|_2 \\ & + \frac{D_\infty^2 G_\infty \sqrt{1-\beta_2}}{2\alpha} \sum_{i=1}^{d} \sum_{t=1}^{t} \frac{\beta_{1,t}}{(1-\beta_{1,t})} \sqrt{t} \end{aligned}$$

We can use arithmetic geometric series upper bound for the last term:

$$\begin{aligned} \sum_{t=1}^{t} \frac{\beta_{1,t}}{(1-\beta_{1,t})} \sqrt{t} & \leq \sum_{t=1}^{t} \frac{1}{(1-\beta_1)} \lambda^{t-1} \sqrt{t} \\ & \leq \sum_{t=1}^{t} \frac{1}{(1-\beta_1)} \lambda^{t-1} t \\ & \leq \frac{1}{(1-\beta_1)(1-\lambda)^2} \end{aligned}$$

Therefore, we have the following regret bound:

$$R(T) \leq \frac{D^2}{2\alpha(1-\beta_1)} \sum_{i=1}^{d} \sqrt{T\widehat{v}_{T,i}} + \frac{\alpha(1+\beta_1)G_\infty}{(1-\beta_1)\sqrt{1-\beta_2}(1-\gamma)^2} \sum_{i=1}^{d} \|g_{1:T,i}\|_2 + \sum_{i=1}^{d} \frac{D_\infty^2 G_\infty \sqrt{1-\beta_2}}{2\alpha\beta_1(1-\lambda)^2}$$

Mathematical proof of the Limited-memory Broyden–Fletcher–Goldfarb–Shanno (L-BFGS) optimization algorithm (Liu and Nocedal, 1989).

Assumptions 6.1

(1) The objective function f is twice continuously differentiable.

(2) The level set $D = \{x \in \mathbf{R}^n : f(x) \leq f(x_0)\}$ is convex.

(3) There exist positive constants M_1 and M_2 such that

$$M_1 \|z\|^2 \leq z^T G(x) z \leq M_2 \|z\|^2 \tag{6.4}$$

for all $z \in \mathbf{R}^n$ and all $x \in D$. Note that this implies that f has a unique minimizer x_* in D.

Theorem 6.1 *Let x_0 be a starting point for which f satisfies Assumptions 6.1, and assume that the matrices $B_k^{(0)}$ are chosen so that $\{\|B_k^{(0)}\|\}$ and $\{\|B_k^{(0)^{-1}}\|\}$ are bounded. Then for any positive definite B_0, Algorithm 6.1 generates a sequence $\{x_k\}$ which converges to x_*. Moreover there is a constant $0 \leq r < 1$ such that*

$$f_k - f_* \leq r^k [f_0 - f_*], \tag{6.5}$$

which implies that $\{x_k\}$ converges R-linearly.

Proof: If we define

$$\overline{G}_k = \int_0^1 G(x_k + \tau s_k) d\tau, \tag{6.6}$$

then

$$y_k = \overline{G}_k s_k. \tag{6.7}$$

Thus (6.4) and (6.7) give

$$M_1 \|s_k\|^2 \leq y_k^T s_k \leq M_2 \|s_k\|^2, \tag{6.8}$$

and

$$\frac{\|y_k\|^2}{y_k^T s_k} = \frac{s_k^T \overline{G}_k^2 s_k}{s_k^T \overline{G}_k s_k} \leq M_2. \tag{6.9}$$

Let $\mathrm{tr}(B)$ denote the trace of B. Then from (6.3), (6.9) and the boundedness of $\{\|B_k^{(0)}\|\}$

$$\begin{aligned} \mathrm{tr}(B_{k+1}) &\leq \mathrm{tr}(B_k^{(0)}) + \sum_{l=0}^{\bar{m}-1} \frac{\|y_{j_l}\|^2}{{y_{j_l}}^T s_{j_l}} \\ &\leq \mathrm{tr}(B_k^{(0)}) + \bar{m} M_2 \\ &\leq M_3, \end{aligned} \tag{6.10}$$

for some positive constant M_3. There is also a simple expression for the determinant (see Pearson (1969) or Powell (1976))

$$\begin{aligned} \det(B_{k+1}) &= \det(B_k^{(0)}) \prod_{l=0}^{\bar{m}-1} \frac{{y_{j_l}}^T s_{j_l}}{{s_{j_l}}^T B_k^{(l)} s_{j_l}} \\ &= \det(B_k^{(0)}) \prod_{l=0}^{\bar{m}-1} \frac{{y_{j_l}}^T s_{j_l}}{{s_{j_l}}^T s_{j_l}} \frac{{s_{j_l}}^T s_{j_l}}{{s_{j_l}}^T B_k^{(l)} s_{j_l}}. \end{aligned} \tag{6.11}$$

Since by (6.10) the largest eigenvalue of $B_k^{(l)}$ is also less than M_3, we have, using (6.8) and the boundedness of $\{\|B_k^{(0)^{-1}}\|\}$,

$$\begin{aligned} \det(B_{k+1}) &\geq \det(B_k^{(0)}) \left(\frac{M_1}{M_3}\right)^{\bar{m}} \\ &\geq M_4, \end{aligned} \tag{6.12}$$

for some positive constant M_4. Therefore from (6.10) and (6.12) we conclude that there

is a constant $\delta > 0$ such that

$$\cos\theta_k \equiv \frac{s_k^T B_k s_k}{\|s_k\| \|B_k s_k\|} \geq \delta. \tag{6.13}$$

One can show that the line search conditions (2.4)-(2.5) and Assumptions 6.1 imply that there is a constant $c > 0$ such that

$$f(x_{k+1}) - f(x_*) \leq (1 - c\cos^2\theta_k)(f(x_k) - f(x_*)),$$

see for example Powell (1976). Using (6.13) we obtain (6.5).

From (6.4)

$$\frac{1}{2} M_1 \|x_k - x_*\|^2 \leq f_k - f_*,$$

which together with (6.5) implies $\|x_k - x_*\| \leq r^{k/2}[2(f_0 - f_*)/M_1]^{1/2}$, so that the sequence $\{x_k\}$ is R-linearly convergent also.

Mathematical proof of Stochastic Gradient Descent optimization algorithm (Shamir and Zhang, 2013, p. 73-75).

Theorem 1. *Suppose F is λ-strongly convex, and that $\mathbb{E}[\|\hat{\mathbf{g}}_t\|^2] \leq G^2$ for all t. Consider SGD with step sizes*

$$\eta_t = 1/\lambda t.$$ *Then for any $T > 1$, it holds that*

$$\mathbb{E}[F(\mathbf{w}_T) - F(\mathbf{w}^*)] \leq \frac{17G^2(1 + \log(T))}{\lambda T}.$$

Proof. The beginning of the proof is standard. By convexity of $\mathcal{W}$, we have the following for any $\mathbf{w} \in \mathcal{W}$:

$$\begin{aligned}
\mathbb{E}\left[\|\mathbf{w}_{t+1} - \mathbf{w}\|^2\right] &= \mathbb{E}[\|\Pi_{\mathcal{W}}(\mathbf{w}_t - \eta_t\hat{\mathbf{g}}_t) - \mathbf{w}\|^2] \\
&\leq \mathbb{E}\left[\|\mathbf{w}_t - \eta_t\hat{\mathbf{g}}_t - \mathbf{w}\|^2\right] \\
&\leq \mathbb{E}\left[\|\mathbf{w}_t - \mathbf{w}\|^2\right] - 2\eta_t\mathbb{E}[\langle \mathbf{g}_t, \mathbf{w}_t - \mathbf{w}\rangle] + \eta_t^2 G^2.
\end{aligned}$$

Let k be an arbitrary element in $\{1, \ldots, \lfloor T/2 \rfloor\}$. Extracting the inner product, summing over all $t = T-k, \ldots, T$, and rearranging, we get

$$\begin{aligned}
\sum_{t=T-k}^{T} \mathbb{E}[\langle \mathbf{g}_t, \mathbf{w}_t - \mathbf{w}\rangle] &\leq \frac{1}{2\eta_{T-k}}\mathbb{E}[\|\mathbf{w}_{T-k} - \mathbf{w}\|^2] \\
&+ \sum_{t=T-k+1}^{T} \frac{\mathbb{E}[\|\mathbf{w}_t - \mathbf{w}\|^2]}{2}\left(\frac{1}{\eta_t} - \frac{1}{\eta_{t-1}}\right) + \frac{G^2}{2}\sum_{t=T-k}^{T}\eta_t.
\end{aligned} \tag{1}$$

By convexity of F, we can lower bound $\langle \mathbf{g}_t, \mathbf{w}_t - \mathbf{w} \rangle$ by $F(\mathbf{w}_t) - F(\mathbf{w})$. Plugging this in and substituting $\eta_t = 1/\lambda t$, we get

$$\mathbb{E}\left[\sum_{t=T-k}^{T} (F(\mathbf{w}_t) - F(\mathbf{w}))\right] \leq \frac{\lambda(T-k)}{2}\mathbb{E}[\|\mathbf{w}_{T-k} - \mathbf{w}\|^2]$$
$$+ \frac{\lambda}{2}\sum_{t=T-k+1}^{T} \mathbb{E}[\|\mathbf{w}_t - \mathbf{w}\|^2] + \frac{G^2}{2\lambda}\sum_{t=T-k}^{T}\frac{1}{t}. \qquad (2)$$

Now comes the crucial trick: instead of picking $\mathbf{w} = \mathbf{w}^*$, as done in standard analysis ((Hazan et al., 2007; Rakhlin et al., 2011)), we instead pick $\mathbf{w} = \mathbf{w}_{T-k}$. We also use the fact that $\mathbb{E}\left[\|\mathbf{w}_t - \mathbf{w}^*\|^2\right] \leq \frac{4G^2}{\lambda^2 t}$ ((Rakhlin et al., 2011), Lemma 1), which implies that for any $t \geq T-k$,

$$\begin{aligned}
&\mathbb{E}[\|\mathbf{w}_t - \mathbf{w}_{T-k}\|^2] \\
\leq &2\mathbb{E}\left[\|\mathbf{w}_t - \mathbf{w}^*\|^2 + \|\mathbf{w}_{T-k} - \mathbf{w}^*\|^2\right] \\
\leq &\frac{8G^2}{\lambda^2}\left(\frac{1}{t} + \frac{1}{T-k}\right) \leq \frac{16G^2}{\lambda^2(T-k)} \leq \frac{32G^2}{\lambda^2 T}.
\end{aligned}$$

Plugging this back into Eq. (2), we get

$$\mathbb{E}\left[\sum_{t=T-k}^{T} (F(\mathbf{w}_t) - F(\mathbf{w}_{T-k}))\right] \leq \frac{16G^2 k}{\lambda T} + \frac{G^2}{2\lambda}\sum_{t=T-k}^{T}\frac{1}{t}.$$

Let $S_k = \frac{1}{k+1}\sum_{t=T-k}^{T}\mathbb{E}[F(\mathbf{w}_t)]$ be the expected average value of the last $k+1$ iterates. The bound above implies that

$$-\mathbb{E}[F(\mathbf{w}_{T-k})] \leq -\mathbb{E}[S_k] + \frac{G^2}{2\lambda}\left(\frac{32}{T} + \sum_{t=T-k}^{T}\frac{1}{(k+1)t}\right).$$

By the definition of S_k and the inequality above, we have

$$\begin{aligned}
&k\mathbb{E}[S_{k-1}] = (k+1)\mathbb{E}[S_k] - \mathbb{E}[F(\mathbf{w}_{T-k})] \\
&\leq (k+1)\mathbb{E}[S_k] - \mathbb{E}[S_k] + \frac{G^2}{2\lambda}\left(\frac{32}{T} + \sum_{t=T-k}^{T}\frac{1}{(k+1)t}\right)
\end{aligned}$$

and dividing by k, implies

$$\mathbb{E}[S_{k-1}] \leq \mathbb{E}[S_k] + \frac{G^2}{2\lambda}\left(\frac{32}{kT} + \sum_{t=T-k}^{T}\frac{1}{k(k+1)t}\right). \qquad (3)$$

Using this inequality repeatedly and summing from $k = 1$ to $k = \lfloor T/2 \rfloor$, we have

$$\mathbb{E}[F(\mathbf{w}_T)] = \mathbb{E}[S_0] \leq \mathbb{E}[S_{\lfloor T/2 \rfloor}] + \frac{16G^2}{\lambda T} \sum_{k=1}^{\lfloor T/2 \rfloor} \frac{1}{k}$$
$$+ \frac{G^2}{2\lambda} \sum_{k=1}^{\lfloor T/2 \rfloor} \sum_{t=T-k}^{T} \frac{1}{k(k+1)t}. \tag{4}$$

It now just remains to bound these terms. $\mathbb{E}[S_{T/2}]$ is the expected average value of the last $\lfloor T/2 \rfloor$ iterates, which was already analyzed in ((Rakhlin et al., 2011), Theorem 5), yielding a bound of

$$\mathbb{E}[S_{\lfloor T/2 \rfloor}] \leq F(\mathbf{w}^*) + \frac{10G^2}{\lambda T}$$

for $T > 1$. Moreover, we have $\sum_{k=1}^{\lfloor T/2 \rfloor}(1/k) \leq 1 + \log(T/2)$. Finally, we have

$$\sum_{k=1}^{\lfloor T/2 \rfloor} \sum_{t=T-k}^{T} \frac{1}{k(k+1)t} \leq \sum_{k=1}^{\lfloor T/2 \rfloor} \frac{1}{k(T-k)}$$
$$= \frac{1}{T} \sum_{k=1}^{\lfloor T/2 \rfloor} \left(\frac{1}{k} + \frac{1}{T-k}\right) \leq (1 + \log(T))/T.$$

The result follows by substituting the above bounds into Eq. (4) and simplifying for readability. □

Theorem 2. *Suppose that F is convex, and that for some constants D, G, it holds that $\mathbb{E}[\|\hat{\mathbf{g}}_t\|] \leq G^2$ for all t, and $\sup_{\mathbf{w},\mathbf{w}' \in \mathcal{W}} \|\mathbf{w} - \mathbf{w}'\| \leq D$. Consider SGD with step sizes $\eta_t = c/\sqrt{t}$ where $c > 0$ is a constant. Then for any $T > 1$, it holds that*

$$\mathbb{E}[F(\mathbf{w}_T) - F(\mathbf{w}^*)] \leq \left(\frac{D^2}{c} + cG^2\right) \frac{2 + \log(T)}{\sqrt{T}}.$$

Proof. The proof begins the same as in Thm. 1 (this time letting k be an element in $\{1, \ldots, T-1\}$), up to Eq. (1). Instead of substituting $\eta_t = c/\lambda t$, we substitute $\eta_t = c/\sqrt{t}$, to get the, $\mathbb{E}[\|\mathbf{w}_t - \mathbf{w}\|^2]$ by D^2, pick $\mathbf{w} = \mathbf{w}_{T-k}$ and slightly simplify to get

$$\mathbb{E}\left[\langle \mathbf{g}_t, \mathbf{w}_t - \mathbf{w}_{T-k} \rangle\right]$$
$$\leq \frac{D^2}{2c}\left(\sqrt{T} - \sqrt{T-k}\right) + \frac{G^2}{2} \sum_{t=T-k}^{T} \frac{c}{\sqrt{t}}.$$

By convexity, we can lower bound $\langle \mathbf{g}_t, \mathbf{w}_t - \mathbf{w}_{T-k} \rangle$ by $F(\mathbf{w}_t) - F(\mathbf{w}_{T-k})$. Also, it is easy to verify (e.g. by integration) that $\sum_{t=T-k}^{T} \frac{1}{\sqrt{t}} \leq 2(\sqrt{T} - \sqrt{T-k-1})$, hence

$$\begin{aligned}
&\mathbb{E}\left[\sum_{t=T-k}^{T} (F(\mathbf{w}_t) - F(\mathbf{w}_{T-k}))\right] \\
&\leq \left(\frac{D^2}{2c} + cG^2\right)\left(\sqrt{T} - \sqrt{T-k-1}\right) \\
&= \left(\frac{D^2}{2c} + cG^2\right)\frac{k+1}{\sqrt{T} + \sqrt{T-k-1}} \\
&\leq \left(\frac{D^2}{2c} + cG^2\right)\frac{k+1}{\sqrt{T}}. \qquad (5)
\end{aligned}$$

As in the proof of Thm. 1, let $S_k = \frac{1}{k+1}\sum_{t=T-k}^{T} \mathbb{E}[F(\mathbf{w}_t)]$ be the expected average value of the last $K+1$ iterates. The bound above implies that

$$-\mathbb{E}[F(\mathbf{w}_{T-k})] \leq -\mathbb{E}[S_k] + \frac{D^2/2c + cG^2}{\sqrt{T}}.$$

By the definition of S_k and the inequality above, we have

$$\begin{aligned}
k\mathbb{E}[S_{k-1}] &= (k+1)\mathbb{E}[S_k] - \mathbb{E}[F(\mathbf{w}_{T-k})] \\
&\leq (k+1)\mathbb{E}[S_k] - \mathbb{E}[S_k] + \frac{D^2/2c + cG^2}{\sqrt{T}},
\end{aligned}$$

and dividing by k, implies

$$\mathbb{E}[S_{k-1}] \leq \mathbb{E}[S_k] + \frac{D^2/2c + cG^2}{k\sqrt{T}}.$$

Using this inequality repeatedly and by summing over $k = 1, \ldots, T-1$, we have

$$\mathbb{E}[F(\mathbf{w}_T)] = \mathbb{E}[S_0] \leq \mathbb{E}[S_{T-1}] + \frac{D^2/2c + cG^2}{\sqrt{T}} \sum_{k=1}^{T-1} \frac{1}{k}. \qquad (6)$$

It now just remains to bound the terms on the right hand side. Using Eq. (1) with $k = T-1$ and $\mathbf{w} = \mathbf{w}^*$,

and upper bounding the norms by D, it is easy to calculate that

$$\mathbb{E}[S_{T-1}] - F(\mathbf{w}^*) = \frac{1}{T}\mathbb{E}\left[\sum_{t=1}^{T}\mathbb{E}[F(\mathbf{w}_t) - F(\mathbf{w}^*)\right]$$
$$\leq \left(\frac{D^2}{c} + cG^2\right)\frac{1}{\sqrt{T}}.$$

Also, we have $\sum_{k=1}^{T-1} 1/k \leq (1 + \log(T))$. Plugging these upper bounds into Eq. (6) and simplifying for readability, we get the required bound. □

Python Libraries Utilized for this Analysis include:

- Pandas
- Numpy
- Seaborn
- Matplotlib.pyplot
- Sklearn

Python code is available for download using GitHub at the following link:

https://github.com/elliottfitzgerald/ML1---Group-Project/tree/main/Code