DSE 6311 – Data Science Capstone

Team Gamma - 12.12.2024

**Team Lead:** Canfield, Ryan

canfieldr@merrimack.edu

**Reporter:** Crozier, Amber

croziera@merrimack.edu

**Spokesperson:** Sabri, Abdelmalek

sabria@merrimack.edu

\* Each author contributed equally to the design, coding & development, analysis, and writing of this project

EVALUATING AI’s ROLE IN CUSTOMER SATISFACTION AND RETENTION

Final Deliverables

Table of Contents

[**Abstract 2**](#_om0yyyk83sp)

[**Background & Question 2**](#_gjdgxs)

[Question: 3](#_30j0zll)

[Hypothesis and Prediction 3](#_3dy6vkm)

[Hypothesis: 3](#_1t3h5sf)

[Prediction: 3](#_4d34og8)

[**Data 3**](#_2s8eyo1)

[Data Acquisition: 3](#_d4kml1ln70p1)

[Target Variable: 3](#_26in1rg)

[Predictor Variables: 3](#_lnxbz9)

[Cleaning: 4](#_n8fkg4ikw9p5)

[Data Exploration: 4](#_fq2obe3xjgkc)

[**Models 6**](#_m31njub75k71)

[Pre-processing: 6](#_2qt1erxo48wl)

[Dimensionality: 7](#_9tyw0xnj8rz)

[Feature Engineering: 7](#_wbhao9xmtzg5)

[Algorithm Selection: 7](#_uskc121jr9rk)

[Final Model Results: 8](#_t6r0g93kh4zq)

[Conclusions 12](#_rxtn7j3y1qn5)

[**Discussion and Next Steps 13**](#_ig86pdrxqrst)

[Summary of Key Takeaways: 13](#_oevtynuhanj9)

[Recommendations: 13](#_37lgxar2zkt3)

[**Code Availability 13**](#_k3pl36efu8r5)

[**References 14**](#_2xcytpi)

[**Appendix 15**](#_m3adjyvj41xm)

# Abstract

Artificial Intelligence (AI) use is continually increasing within the business model specifically in addressing customer interactions. As this increase occurs, it’s important to identify when AI is beneficial to customer relationships, and when it’s harmful. This insight can guide companies in using AI effectively without risking customer loyalty or driving down revenue due to customer dissatisfaction. This project sought to predict customer satisfaction with AI-driven tools, focusing on identifying customer segments most likely to have positive experiences. We used neural networks, dimensionality reduction through PCA, and robust preprocessing methods in order to uncover actionable insights.

The model chosen was a neural network due to its capability to model non-linear relationships effectively. Its selection was further justified by its adaptability to categorical features. Results showed 85% accuracy and highlighted significant trade-offs, such as improved recall but reduced specificity. These findings validated our methodology and has the potential to create actionable targets for our stakeholders for AI deployment in retail which underscores the importance of balancing automation and human interaction.

# Background & Question

Customer satisfaction remains a critical yet complex element in the modern business landscape, particularly as companies increasingly rely on artificial intelligence to enhance customer service. According to Business Insider, while AI chatbots theoretically improve efficiency and reduce costs, they often fall short in delivering satisfactory customer experiences. Common issues such as misinterpreted requests and lack of human-like empathy highlight the nuanced challenges businesses face when implementing AI tools. Business Insider even goes so far as to say, “Customers are the guinea pigs in customers’ experimentation with AI.” Customers want to feel cared about, sympathized with and that they are important… not associated with a guinea pig.

In contrast, American Public University emphasizes the practical benefits of AI in customer service, including 24/7 availability, personalized interactions, and streamlined operations. This division between AI’s potential and its real-world shortcomings is what motivated our team to investigate the factors influencing customer satisfaction with AI tools. Understanding these predictive factors could guide companies in optimizing AI-driven customer interactions, ensuring a balance between operational efficiency and user satisfaction.

By addressing this research question, our project aims to bridge the gap between AI’s theoretical utility and practical challenges, contributing valuable insights to the field of customer experience management.

## Question:

Can we predict customer satisfaction with AI content to help businesses identify the type of customer base most likely to have a positive experience with AI tools they deploy?

## Hypothesis and Prediction

### Hypothesis:

While the implementation of AI in retail settings improves cost efficiency and reduces errors (Kasareni, 2021), excessive reliance on AI in customer service without human interaction will reduce customer satisfaction.

### Prediction:

Customer satisfaction with AI-driven customer service will be higher among specific demographic groups and for certain AI tools.

# Data

## Data Acquisition:

The “Customer Satisfaction Response to Artificial Intelligence Tools Usage During Online Shopping” dataset has much of the data that we will need to analyze customer satisfaction and use of AI. Our dataset includes customer demographic and behavioral variables from three different countries, focusing on interactions and satisfaction with AI tools. It includes 23 columns and 656 rows, covering variables such as country, age, gender, education, preferred online services, annual salary, payment methods, AI endorsement and privacy perceptions, and specific AI tools used.

### Target Variable:

Customer Satisfaction (AI\_Satisfaction) - Whether or not a customer was satisfied with their experience.

### Predictor Variables:

1. Demographic: Country, Generation (Age), Gender, Education, and Living\_Region
2. AI tools used: Chatbots, Virtual Assistants, Voice & Photo Search
3. AI\_Trust: (AI\_Privacy\_Trust + AI\_Endorsement) / 2 Created Column
4. AI\_Enhance\_Experience
5. AI\_Usage: AI tools (recode for high, moderate, or low)

This dataset aligns with the hypothesis that excessive reliance on AI in customer service may reduce satisfaction. The variables allow us to test demographic and AI tool-related predictors, providing insights into which groups are more likely to have positive experiences. The data collection process was not difficult. Additionally, the data source was funded by multiple universities which enhances the validity of our project. The only thing that stands out as a potential threat to our validity would be that our dataset is relatively small. Increasing the number of observations would increase our validity.

Original data can be found here: [Link to data](https://figshare.com/articles/dataset/Customer_Satisfaction_Response_to_Artificial_Intelligence_Tools_Usage_During_Online_Shopping/24633105?file=43284342)

## Cleaning:

To prepare the dataset for analysis, we cleaned and prepared the dataset for analysis by handling missing values, encoding our categorical variables, normalizing our data, inverting AI\_Privacy\_No\_Trust to become AI\_Privacy\_Trust and creating new variables such as AI\_Trust which will combine AI\_Privacy\_Trust and AI\_Endorsement and AI\_Usage which allows us to see exposure to different variations of AI tools (high, moderate, and low exposure). We also grouped AI use by country since some countries can overall have differing opinions about AI than others. We will also remove records of individuals who reported they do not shop online, as well as the single data point missing a gender report. This ensures our analysis remains focused on the online sales context where AI tools are used. This data only constitutes 3% of the overall data, so this cleaning step is unlikely to impact the overall results and also ensures a relevant sample to provide to online retailers.

## Data Exploration:

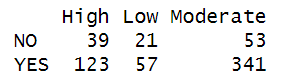
One of the most important things to consider when first taking a look at your data, far before building the models, is to complete a thorough data exploration. The data exploration for our group was very straightforward because we had a clear understanding of our target variable and how we wanted to approach the project. Below is the analysis we conducted of a couple tables and a few graphs along with an explanation of each.

The table *(Figure 1)* below is very straightforward showing a distribution of our target variable. Distribution for our target variable is an important piece of analysis to make sure that we have a balanced data set that we can work with… which we very clearly do!



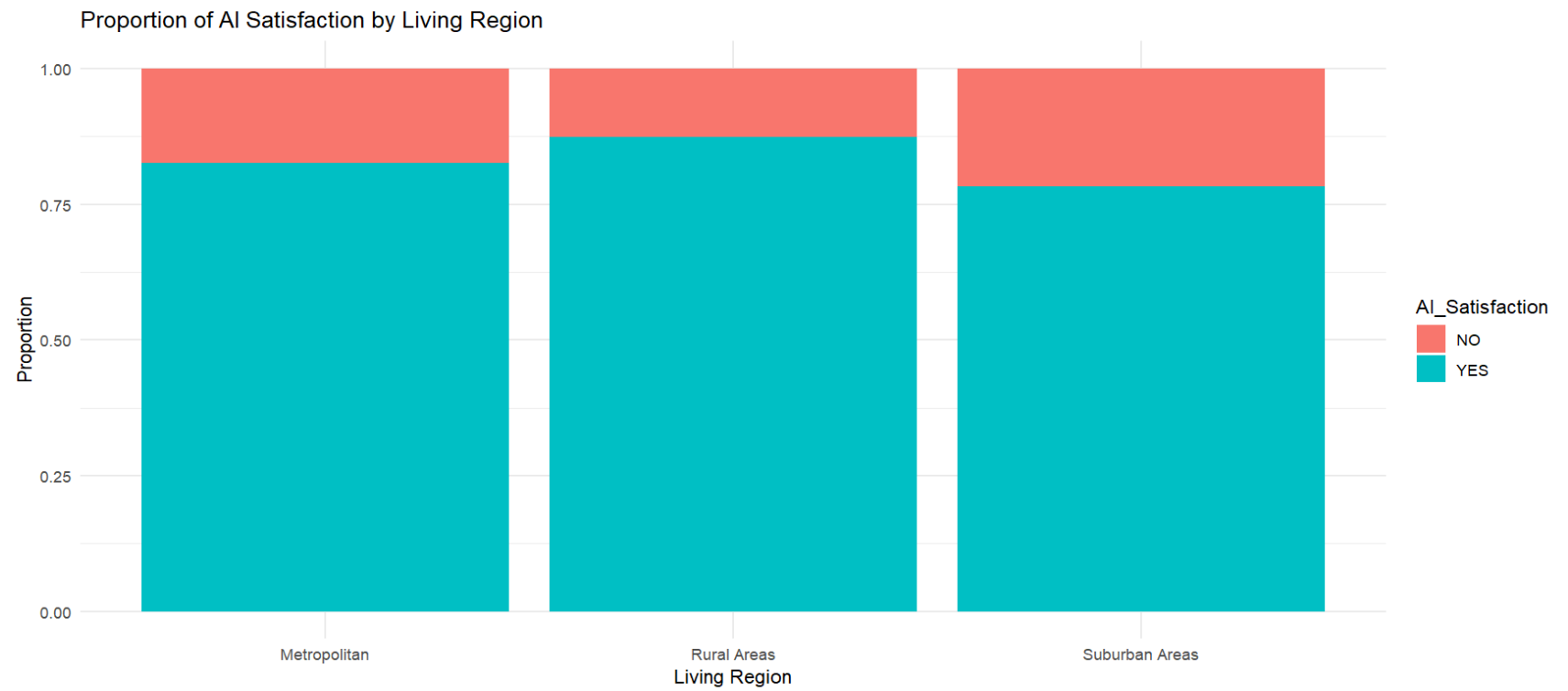
*Figure 1 - Table*

This next table (*Figure 2)* is looking at AI satisfaction and AI trust as a cross table. This is helpful because it provides the relationship between the two variables. When working with a predictor variable, we want to make sure that does not correlate highly with the target variable.

**

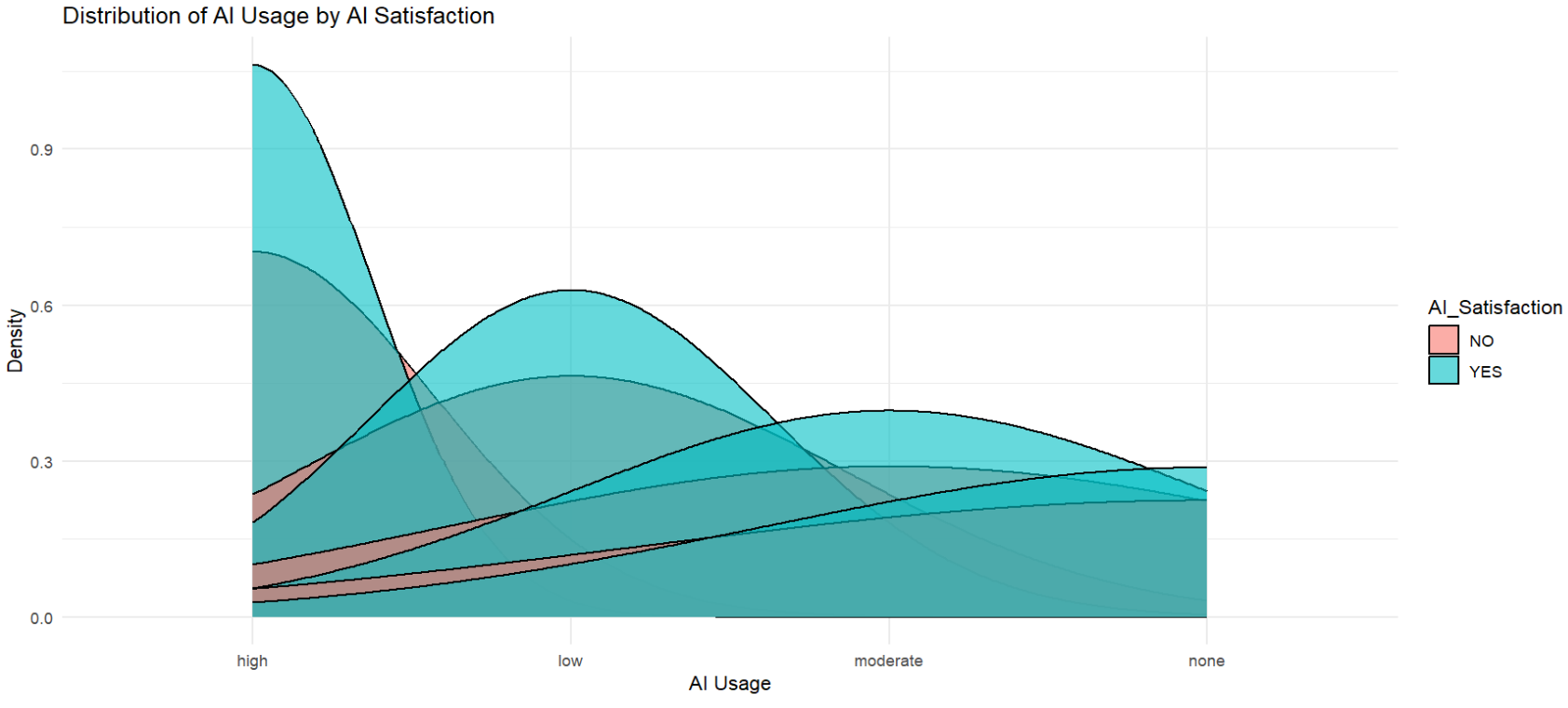
*Figure 2 - Table*

The first graphwas useful for us because it showed AI satisfaction across different regions. This helped our group determine whether or not we should split up those regions in any way. We ended up taking a deeper look at the division of countries (*Figure 3)*.

**

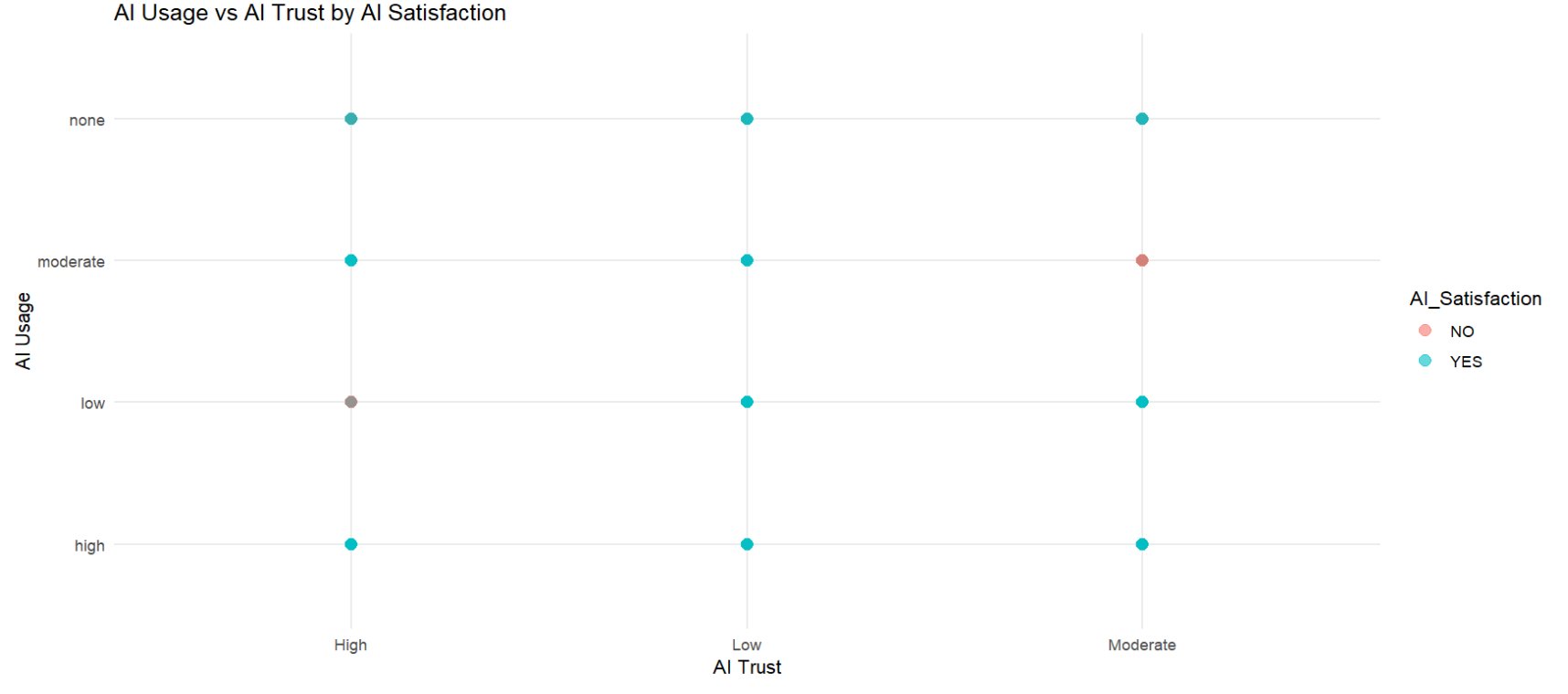
*Figure 3 - Bar Chart*

The next graph (*Figure 4)* shows the distribution of AI usage for each category of AI satisfaction. Here we see that those that use AI highly have the highest level of satisfaction. This density plot is interesting because across all levels of AI usage it looks like the proportion of those satisfied and dissatisfied is relatively similar.

**

*Figure 4 - Density Plot*

This last graph (*Figure 5)* shows the relationship between AI usage and AI trust with the target variable AI satisfaction thrown in there. This is useful to categorize users, for example, those with high trust and high usage are satisfied with AI.

**

*Figure 5 - Bar Chart*

# Models

## Pre-processing:

To make sure our dataset was ready to be used by our model we had to employ a lot of encoding techniques. The reason why we had to do this was that most of our data had a character data type instead of a numerical data type. The first encoding technique was to use binary encoding since most of our data was either yes or no, or one option or the other, we converted all of them into 1’s and 0’s. The next bit of encoding we had to do was for all the ordered data. Features like education, age, and salary had to be mapped differently since they have multiple values in their column. For example, in our data we converted a salary column that had low, medium, medium high and high to 1, 2, 3, and 4. If you had a high salary or a 4 value, you already covered the low and medium categories. The last bit of encoding we did was for our “Country” and “Living\_Region” features. From one of our lectures our professor taught us that this was a good technique to use when you want to represent categorical data in a way that captures the distribution of occurrences in that feature. It does this and still transforms it into a numerical format that is useful for machine learning models. After encoding the categorical variables, we split the dataset into training and testing sets using a 78:22 ratio, which was determined using the calcSplitRatio function (Geist, 2019) to ensure an optimal split based on the number of predictors. Lastly, to fit the data into our model we decided to standardize the data. Standardization is when you mathematically transform the features of your dataset to have a mean of 0 and a standard deviation of 1.

## Dimensionality:

Our team decided that we were not going to reduce any of the features in our dataset. This decision was based on the need to keep the number of features manageable (under 25 columns) due to having a relatively small dataset of 634 observations. Standard techniques like one-hot encoding would have resulted in an excessively large number of columns, which could lead to overfitting and increased model complexity.

## Feature Engineering:

Our team added a total of three columns into our data before we fed it into our model. These three variables came from us employing two methods: Linear Discriminant Analysis (LDA) and K-Nearest Neighbors (KNN). We chose LDA because we had a categorical target variable and pre-existing labels.This meant that it was an appropriate technique for pulling out any underlying patterns or hidden features from our data. After applying LDA, we obtained a single component since we only had two classes in our target variable. The output of the LDA gives continuous values ranging from -2 to 4. To Make things simple and to leave no value undefined we applied a threshold of 0, categorizing values below zero as 0 and values above zero as 1.

Next, we used KNN to identify the top two clusters by calculating the densities of each cluster. We selected these top two clusters and added their information as two new columns to the dataset. In total, we generated three new features: one from LDA (the LDA component) and two from KNN (representing their cluster information). These new features will hopefully capture underlying patterns that were not apparent from exploratory data analysis. We believe that including these features will provide additional insights when applied to our machine learning models.

## Algorithm Selection:

The choice of a neural network model was based on its ability to handle categorical data effectively, and its capacity to model complex interactions between features. Neural networks are ideal for learning non-linear relationships. Part of our assumptions were that the relationship between the features were not linear and more complex. We wanted to make these connections even if we could not see them from the surface. We also used Principal Component Analysis (PCA) to reduce dimensionality and to reduce feature space and address overfitting after the initial model was created.

For the initial model, we implemented a deep learning model using the python package Keras. Our Neural Network was initially created with three dense neuron layers, and a ReLU activation function. The final output layer had one neuron and a sigmoid activation function to predict the binary outcome of satisfied vs. not satisfied. When initializing and training the model, we had a RMSProp optimizer to reduce noise. We used a binary cross-entropy loss function because we had a binary categorical problem that we wanted the model to handle. Assumptions of the model included the expectation that the relationships between features and customer satisfaction are non-linear and that PCA would keep enough variance in the dataset to be able to see important patterns to predict. We also assumed that the features were independent, this assumption seemed reasonable based on the exploratory data analysis.

To address overfitting, we created an updated model and included dropout layers, PCA, and early stoppage. The dropout layer was to enhance regularization, PCA was to reduce features that did not contribute a lot of variance and early stoppage was to monitor the validation loss. The idea of early stoppage is to run consecutive epochs to see if there was any improvement. When improvement in the training did not occur, the training was halted and best model weights were restored.

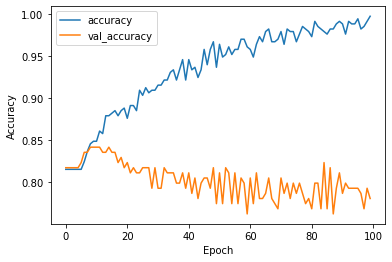
The new model had a batch size of 512 and used 100 epochs, although it never completed the full 100 due to the methods listed above. For hyperparameter tuning, we started by splitting the dataset into training and test sets. We optimized the split at 78% in the training set and held out 22% for the test set. We then optimized the neural network’s architecture and training settings by manually selecting the number of hidden layers and neurons in each layer (4 hidden layers with 256, 124, 64, and 12 neurons) based on previous experience and model experimentation. A dropout rate of 0.2 was chosen to reduce overfitting without significantly inhibiting the model’s ability to learn. The rest was kept the same because after testing multiple optimizers and loss functions the two that were first used were the best.

## Final Model Results:

Upon analyzing the initial neural network model and every complete pass the neural network did, it revealed many things to us. If we look at the first run or epoch the model completed, it showed us that the model has a promising accuracy of 82% with a loss of 0.6. When testing the model on the validation data, the model had a validation accuracy of 82% and a validation loss of 0.52. By the last or 100th completed pass, the neural network had an accuracy of 99.7% with a loss of 0.78. It also got a validation accuracy of 78% and a validation loss of 1.13. Below is a table *(Figure 6)* to summarize the information above which compares the first and final pass that the Neural Network made and a graph *(Figure 7)* that shows accuracy across all epochs.

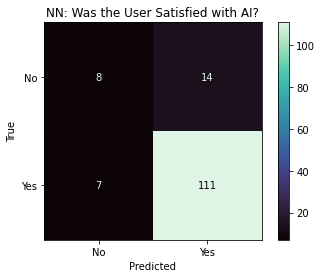
| Epoch | 1 | 100 |
| --- | --- | --- |
| Accuracy | 0.8152 | 0.9970 |
| Loss | 0.5933 | 0.0346 |
| Validation Accuracy | 0.8171 | 0.7805 |
| Validation Loss | 0.5213 | 1.1283 |

*Figure 6: Model 1 Table*



*Figure 7: Model 1 Accuracy Plot*

Loss is a metric that shows how well the model's predictions match the actual target values. The goal of a good model is to have a low loss value since the lower the loss value the more accurate the model will be. So even though our model is increasing its accuracy and decreasing loss over the 100 runs, it has also decreased its validation accuracy and validation loss every run which is a sign of overfitting. The reason why we think the model is overfitting is because it is making some general assumptions on the data. Even though the model seems to be overfitting, here are the evaluation metrics after the model predicted on the test data. A confusion matrix of the test results *(Figure 8)* showed that the model achieved an accuracy of 85%, recall of 94.07%, precision of 88.8%, specificity of 36.36%, and an F1 score of 91.36%. The accuracy indicates that 85% of the model’s predictions were correct, but the lower specificity suggests that the model has a higher rate of false positives, meaning it incorrectly classifies customers who are not satisfied as satisfied. On the other hand, the high recall shows that the model is effective at detecting most of the instances where a customer is satisfied. With all these base metrics we can still improve the model by dealing with some of its assumptions and fine tuning the hyper parameters.



|  |  |
| --- | --- |
| **Evaluation Metrics** |  |
| Accuracy | **85.00 %** |
| Recall | **94.07 %** |
| Precision | **88.80 %** |
| Specificity | **36.36 %** |
| F1 | **91.36 %** |

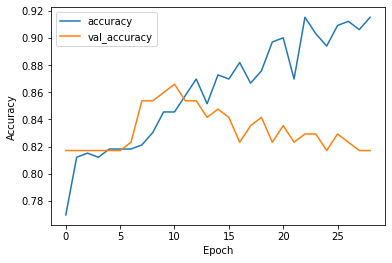
### 

*Figure 8: Model 1 Confusion Matrix and Evaluation Metrics*

We are using the same metrics to evaluate the overfitting and the model’s overall improvement. For the overfitting we looked at the new training accuracy and validation loss. From the first epoch of the new neural network model we got an accuracy of 0.7697 and a validation loss of 0.5254. On the last epoch of this new model which is number 29 we got an accuracy of 0.9152 and a validation loss of 0.4540. The table *(Figure 9)* below summarizes these new values as well as compares the previous values from the first model. There is also a plot *(Figure 10)* showing the new model's accuracy across all of the training.

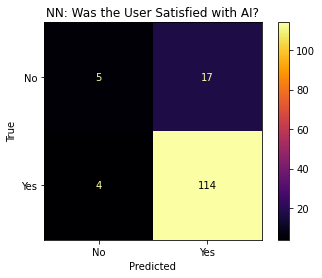
|  | New Model | New Model | Old Model | Old Model |
| --- | --- | --- | --- | --- |
| Epoch | 1 | 29 | 1 | 100 |
| Accuracy | 0.7697 | 0.9152 | 0.8152 | 0.9970 |
| Loss | 0.6017 | 0.2416 | 0.5933 | 0.0346 |
| Validation Accuracy | 0.8171 | 0.8171 | 0.8171 | 0.7805 |
| Validation Loss | 0.5254 | 0.4540 | 0.5213 | 1.1283 |

*Figure 9: Model Comparison*



*Figure 10: Model 1 Accuracy Plot*

As you can see from the table and graph above the model is overfitting less then the initial model. Even though the new model has a slightly lower accuracy than the initial model it has a greatly reduced validation loss value. For the model’s improvement we can look at the confusion matrix and how it did on the test data *(Figure 11)*.



|  |  |  |
| --- | --- | --- |
| **Evaluation Metrics** | Initial Model | Improved Model |
| Accuracy | **85.00 %** | **85.00 %** |
| Recall | **94.07 %** | **96.61 %** |
| Precision | **88.80 %** | **87.02 %** |
| Specificity | **36.36 %** | **22.73 %** |
| F1 | **91.36 %** | **91.57%** |

*Figure 11: Model 2 Confusion Matrix and Evaluation Metrics*

Even though the initial model performed well, there are noticeable improvements in the evaluation metrics after all the adjustments were made. The accuracy of the model remained the same at 85% for both the initial and improved models. This can suggest that the overall correct classification rate didn’t change. However, the recall improved from 94.07% to 96.61%, showing that the improved model is now better at detecting positive instances, meaning it’s more effective at identifying satisfied customers. The precision slightly decreased from 88.8% to 87.02%, indicating a small increase in false positives, where the model mistakenly labels some unsatisfied customers as satisfied. The specificity decreased from 36.36% to 22.73%, meaning the model is now less effective at correctly identifying negative instances, or unsatisfied customers. Lastly, the F1 score improved slightly from 91.36% to 91.57%, reflecting a slight increase in the balance between precision and recall. Based on these results, the improved model performs better in terms of recall and F1 score, but there is a trade-off with specificity. This suggests that while the improved model is better at detecting positive instances, it has become slightly more prone to misclassifying negatives as positives.

# Conclusions

Our model is designed to predict customer satisfaction with AI-driven tools by identifying demographic and usage patterns which correlate with positive AI Satisfaction. Using a neural network model, we addressed our hypothesis that excessive AI exposure could lower satisfaction but that certain demographics and tools might elicit better outcomes. The model achieved an accuracy of 85%, with strong recall (96.61%), which effectively identified customers who were satisfied with AI but also showed reduced specificity (22.73%).

These results indicate that our model can predict AI satisfaction by leveraging demographic and usage data, aligning with existing studies highlighting customer trust and engagement as critical factors in AI adoption (Ng, et. al, 2020). For instance, previous research on AI-enhanced retail platforms also emphasized the importance of understanding user trust levels and interaction styles to optimize outcomes (Bach, et. al, 2022). Our model provides a good predictive framework for businesses to analyze individual customers to be able to create targeted AI implementation strategies.

Our project provides value to our stakeholders, both customers and businesses by gaining a better understanding of how demographic and usage patterns influence satisfaction with AI-driven tools. For customers, it can ensure that AI tools are targeted to their preferences and needs which can enhance their experiences and their overall trust in technology. Companies can use our model to create a more strategic deployment of AI tools and create a better balance between AI and human interaction which can still keep costs low without sacrificing customer loyalty.

# Discussion and Next Steps

## Summary of Key Takeaways:

Our final model demonstrated strong predictive ability, achieving an accuracy of 85% on the test data, with particularly high recall (94.07%) and an F1 score of 91.36% *(Figure 8)*. This indicated the model’s effectiveness at identifying customers who are satisfied could be captured by neural networks, which aligned with our hypothesis. However, the specificity was lower (36.36%), which indicates that the model struggled with differentiating satisfied customers from unsatisfied ones and lead to a higher false positive rate.

When we compared the initial and updated model we found that incorporating regularization techniques like dropout layers, early stoppage, and dimensionality reduction through PCA reduced overfitting. The initial model achieved a higher training accuracy of 99.7% but it showed signs of overfitting, with validation accuracy dropping to 78% and validation loss increased to 1.13 after 100 epochs *(Figure 6)*. The updated model balanced this and maintained validation accuracy (81.71%) while reducing validation loss to 0.4540 by the final epoch *(Figure 9)*.

## Recommendations:

Future directions include extending this work to improve model specificity, potentially by experimenting with ensemble methods like Random Forest or Gradient Boosting. These methods could enhance prediction accuracy for dissatisfied customers while maintaining the neural network's recall. We also recommend integrating additional features, such as customer sentiment from reviews or real-time feedback, to further enhance the predictive model.

Caveats to this analysis include the relatively small dataset size (634 observations) and its potential limitations in generalizability. We also had overfitting concerns which we mitigated through dropout layers and early stopping but the overfitting should be monitored as we scale the model up. Addressing these limitations through larger datasets and enhancing our model to accommodate new data will be essential for broader applicability. We also recommend businesses use our findings when implementing and aligning AI deployments with customer expectations and focus on the tools and demographics most likely to create positive AI experiences.

# Code Availability

<https://github.com/Canfieldr/DSE6311-Graduate-DS-Capstone>

# 

# References

Bergen, Mark, and Lynn Doan. “Tech Giants Are Set to Spend $200 Billion This Year Chasing AI.” *Bloomberg.com*, Bloomberg, Nov. 2024, www.bloomberg.com/news/articles/2024-11-01/tech-giants-are-set-to-spend-200-billion-this-year-chasing-ai. Accessed 3 Nov. 2024.

Bach, T. A., Khan, A., Hallock, H., Beltrão, G., & Sousa, S. (2022). A Systematic Literature Review of User Trust in AI-Enabled Systems: An HCI Perspective. International Journal of Human–Computer Interaction, 40(5), 1251–1266. <https://doi.org/10.1080/10447318.2022.2138826>

Bueler, T. Leigh (2024) AI in Customer Service: Revolutionizing Digital Retail https://www.apu.apus.edu/area-of-study/business-and-management/resources/ai-in-customer-service/

Chollet, F., Kalinowski, T., and Allaire, J. J. (2022). Deep Learning with R. (Second Edition). Manning.

Geist, K.S. (2019) *calcSplitRatio-3*, *GitHub*. Merrimack College. Available at: https://github.com/ksgeist (Accessed: 14 November 2024).

James G., Witten D., Hastie T., & Tibshirani R. (2015). An Introduction to Statistical Learning with Applications in R (2nd ed.). Springer.

Kannan, Rathimala; Ramakrishnan, Kannan; Ersoy, Ayse Begum; Contu, Davide (2023). Customer Satisfaction Response to Artificial Intelligence Tools Usage During Online Shopping. figshare. Dataset. https://doi.org/10.6084/m9.figshare.24633105.v1

Kasaraneni, Ramana Kumar (2021). AI-Enhanced Supply Chain Collaboration Platforms for Retail: Improving Coordination and Reducing Costs. *Journal of Bioinformatics and Artificial Intelligence*, 1(1), pp. 410–450. Available at: https://biotechjournal.org/index.php/jbai/article/view/98 (Accessed: 3 November 2024).

Ng, S. C., Sweeney, J. C., & Plewa, C. (2020). Customer Engagement: A Systematic Review and Future Research Priorities. Australasian Marketing Journal, 28(4), 235-252. <https://doi.org/10.1016/j.ausmj.2020.05.004>

Stewart, Emily (2024) Customer service hits a new lowAI chatbots are answering the phones. Customers are not happy about it. https://www.businessinsider.com/ai-chatbots-customer-service-call-center-annoying-problems-2024-11

What is the difference between one-hot and dummy encoding? (n.d.) *Data Science Stack Exchange*. Available at: https://datascience.stackexchange.com/ questions/98172/what-is-the-difference-between-one-hot-and-dummy-encoding.

# Appendix

Data Dictionary:

|  |  |  |
| --- | --- | --- |
| Variable | Data Type | Description |
| AI\_Satisfaction | String | Overall satisfaction level |
| Country | String | Country of the respondent |
| Online\_Consumer | String | Whether the consumer shops online (YES/NO) |
| Age | String | Age group of the respondent (Gen X, Gen Z, etc.) |
| Annual\_Salary | String | Salary range of the respondent (Low, Medium, High) |
| Gender | String | Gender of the respondent (Male/Female) |
| Education | String | Highest education level achieved by the respondent (Masters' Degree, University Graduate, etc.) |
| Payment\_Method\_Credit/Debit | String | Whether the respondent uses credit or debit cards for payments (YES/NO) |
| Living\_Region | String | Type of living area (Metropolitan, Rural Areas) |
| Online\_Service\_Preference | String | Preference for using online services (YES/NO) |
| AI\_Enhance\_Experience | String | Whether the respondent feels AI enhances their shopping experience (YES/NO) |
| AI\_Trust | String | (AI\_Privacy\_No\_Trust + 1 + AI\_Endorsement+1)/2 (Will create High (2), Medium (1), Low Trust(0)) |
| AI\_Usage | String | AI\_Tools variables/3 (Will create High (3), Medium (2), Low Usage(1)) |
| AI\_Tools\_Used\_Chatbots | String | Use of chatbots for assistance (YES/NO) |
| AI\_Tools\_Used\_Virtual\_Assistant | String | Use of virtual assistants (YES/NO) |
| AI\_Tools\_Used\_Voice&Photo\_Search | String | Use of voice and photo search tools (YES/NO) |
| Paymeny\_Method\_COD | String | Use of Cash on Delivery as a payment method (YES/NO) |
| Payment\_Method\_Ewallet | String | Use of e-wallets as a payment method (YES/NO) |
| Product\_Category\_Appliances | String | Interest or purchase in appliances category (YES/NO) |
| Product\_Category\_Electronics | String | Interest or purchase in electronics category (YES/NO) |
| Product\_Category\_Groceries | String | Interest or purchase in groceries category (YES/NO) |
| Product\_Category\_Personal\_Care | String | Interest or purchase in personal care category (YES/NO) |
| Product\_Category\_Clothing | String | Interest or purchase in clothing category (YES/NO) |

***Appendix***

*# Import necessary libraries*

**import** pandas **as** pd

**import** numpy **as** np

**import** matplotlib.pyplot **as** plt

**import** seaborn **as** sns

**import** random

**from** sklearn.model\_selection **import** train\_test\_split

**from** sklearn.preprocessing **import** StandardScaler

**from** sklearn.decomposition **import** PCA

**from** sklearn.metrics **import** confusion\_matrix, roc\_curve, auc, ConfusionMatrixDisplay

**from** sklearn.metrics **import** accuracy\_score, recall\_score, precision\_score

**import** tensorflow **as** tf

**from** tensorflow.keras.models **import** Sequential

**from** tensorflow.keras.layers **import** Dense, Dropout

**from** tensorflow.keras.callbacks **import** EarlyStopping

**import** warnings

warnings**.**filterwarnings("ignore")

In [47]:

*# Load in the full dataset*

df **=** pd**.**read\_csv("FE\_final\_data.csv")

df**.**head()

In [48]:

*# Split the dataset into training and testing sets (78% - 22% split) based off of our calculations*

X **=** df**.**drop(columns **=** ['AI\_Satisfaction'])

y **=** df['AI\_Satisfaction']

X\_train, X\_test, y\_train, y\_test **=** train\_test\_split(X, y, test\_size **=** 0.22, random\_state **=** 3)

*# Scaling the features*

scaler **=** StandardScaler()

X\_train\_scaled **=** scaler**.**fit\_transform(X\_train)

X\_test\_scaled **=** scaler**.**transform(X\_test)

In [104]:

*# This is for producing the same results*

random**.**seed(3)

*# Defining the model*

model **=** Sequential()

model**.**add(Dense(256, activation **=** 'relu', input\_dim **=** X\_train\_scaled**.**shape[1]))

model**.**add(Dropout(0.2))

model**.**add(Dense(124, activation **=** 'relu'))

model**.**add(Dropout(0.2))

model**.**add(Dense(64, activation **=** 'relu'))

model**.**add(Dropout(0.2))

model**.**add(Dense(12, activation **=** 'relu'))

model**.**add(Dense(1, activation **=** 'sigmoid'))

*# Compiling the model*

model**.**compile(optimizer **=** 'RMSprop', loss **=** 'binary\_crossentropy', metrics **=** ['accuracy'])

*# Fitting the model*

history **=** model**.**fit(X\_train\_scaled, y\_train, epochs **=** 100, batch\_size **=** 512, validation\_split **=** 0.33)

In [105]:

*# Plotting the training history*

plt**.**plot(history**.**history['accuracy'], label **=** 'accuracy')

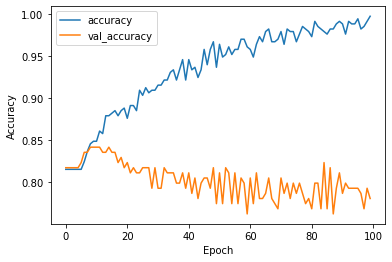
plt**.**plot(history**.**history['val\_accuracy'], label **=** 'val\_accuracy')

plt**.**xlabel('Epoch')

plt**.**ylabel('Accuracy')

plt**.**legend()

plt**.**show()



In [106]:

*# Predicting on the test set*

predictions **=** model**.**predict(X\_test\_scaled)

predicted\_class **=** (predictions **>=** 0.5)**.**astype(int)

*# Displaying the Confusion Matrix*

cf\_matrix **=** confusion\_matrix(y\_test, predicted\_class)

cmd **=** ConfusionMatrixDisplay(cf\_matrix, display\_labels **=** ['No', 'Yes'])

cmd**.**plot(cmap **=** 'mako')

cmd**.**ax\_**.**set(xlabel **=** 'Predicted', ylabel **=** 'True', title **=** 'NN: Was the User Satisfied with AI?')

*# Printing results*

print("When predicting on the test dataset, the accuracy of NN was:", np**.**round((accuracy\_score(y\_test, predicted\_class) **\*** 100), 2), "%")

print("When predicting on the test dataset, the recall of NN was:", np**.**round((recall\_score(y\_test, predicted\_class) **\*** 100), 2), "%")

print("When predicting on the test dataset, the precision of NN was:", np**.**round((precision\_score(y\_test, predicted\_class) **\*** 100), 2), "%")

print("When predicting on the test dataset, the F1 of NN was:", np.round((f1\_score(y\_test, predicted\_class) \* 100), 2), "%")

print("When predicting on the test dataset, the specificity of NN was:", np.round((recall\_score(y\_test, predicted\_class, pos\_label = 0) \* 100), 2), "%")

**5/5** ━━━━━━━━━━━━━━━━━━━━ **0s** 1ms/step

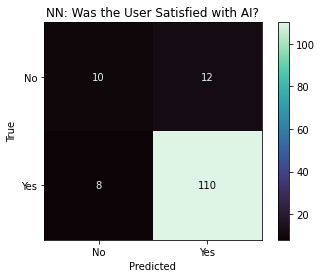
When predicting on the test dataset, the accuracy of NN was: 85.0 %

When predicting on the test dataset, the recall of NN was: 94.07 %

When predicting on the test dataset, the precision of NN was: 88.8 %

When predicting on the test dataset, the F1 of NN was: 91.36 %

When predicting on the test dataset, the specificity of NN was: 36.36 %



In [107]:

*# This is to try and stop the NN from overfitting*

*# Performing PCA on the training features.*

pca **=** PCA(n\_components **=** 0.95)

X\_train\_pca **=** pca**.**fit\_transform(X\_train\_scaled)

X\_test\_pca **=** pca**.**transform(X\_test\_scaled)

*# Creating a new model for PCA-reduced data and adding a early dropout to see if the model's accuracy will increase.*

model2 **=** Sequential()

model2**.**add(Dense(256, activation **=** 'relu', input\_dim **=** X\_train\_pca**.**shape[1]))

model2**.**add(Dropout(0.2))

model2**.**add(Dense(124, activation **=** 'relu'))

model2**.**add(Dropout(0.2))

model2**.**add(Dense(64, activation **=** 'relu'))

model2**.**add(Dropout(0.2))

model2**.**add(Dense(12, activation **=** 'relu'))

model2**.**add(Dropout(0.2))

model2**.**add(Dense(1, activation **=** 'sigmoid'))

In [108]:

*# Compile the second model*

model2**.**compile(optimizer **=** 'rmsprop', loss **=** 'binary\_crossentropy', metrics **=** ['accuracy'])

*# Define early stopping*

early\_stopping **=** EarlyStopping(monitor **=** 'val\_loss', patience **=** 10, restore\_best\_weights **=** **True**)

*# Fit the second model with early stopping*

history\_pca **=** model2**.**fit(X\_train\_pca, y\_train, epochs **=** 100, batch\_size **=** 512,

validation\_split **=** 0.33, callbacks **=** [early\_stopping])

In [109]:

*# Plot training history for PCA model*

plt**.**plot(history\_pca**.**history['accuracy'], label**=**'accuracy')

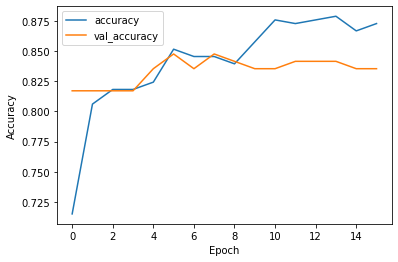
plt**.**plot(history\_pca**.**history['val\_accuracy'], label**=**'val\_accuracy')

plt**.**xlabel('Epoch')

plt**.**ylabel('Accuracy')

plt**.**legend()

plt**.**show()



In [110]:

*# Predict on the test set using PCA-reduced data*

predictions\_pca **=** model2**.**predict(X\_test\_pca)

predicted\_class\_pca **=** (predictions\_pca **>=** 0.5)**.**astype(int)

*# Displaying the Confusion Matrix*

cf\_matrix **=** confusion\_matrix(y\_test, predicted\_class\_pca)

cmd **=** ConfusionMatrixDisplay(cf\_matrix, display\_labels **=** ['No', 'Yes'])

cmd**.**plot(cmap **=** 'inferno')

cmd**.**ax\_**.**set(xlabel **=** 'Predicted', ylabel **=** 'True', title **=** 'NN: Was the User Satisfied with AI?')

*# Printing results*

print("When predicting on the test dataset, the accuracy of NN was:", np**.**round((accuracy\_score(y\_test, predicted\_class\_pca) **\*** 100), 2), "%")

print("When predicting on the test dataset, the recall of NN was:", np**.**round((recall\_score(y\_test, predicted\_class\_pca) **\*** 100), 2), "%")

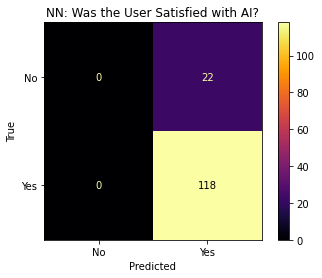
print("When predicting on the test dataset, the precision of NN was:", np**.**round((precision\_score(y\_test, predicted\_class\_pca) **\*** 100), 2), "%")

**5/5** ━━━━━━━━━━━━━━━━━━━━ **0s** 8ms/step

When predicting on the test dataset, the accuracy of NN was: 84.29 %

When predicting on the test dataset, the recall of NN was: 100.0 %

When predicting on the test dataset, the precision of NN was: 84.29 %



In [111]:

*# ROC Curve and AUC*

fpr, tpr, \_ **=** roc\_curve(y\_test, predictions)

roc\_auc **=** auc(fpr, tpr)

fpr\_pca, tpr\_pca, \_ **=** roc\_curve(y\_test, predictions\_pca)

roc\_auc\_pca **=** auc(fpr\_pca, tpr\_pca)

*# Plot ROC curves*

plt**.**figure()

plt**.**plot(fpr, tpr, color **=** 'blue', lw **=** 2, label **=** f'Model 1 (AUC = {roc\_auc:.2f})')

plt**.**plot(fpr\_pca, tpr\_pca, color**=**'red', lw **=** 2, label **=** f'Model 2 (AUC = {roc\_auc\_pca:.2f})')

plt**.**plot([0, 1], [0, 1], color **=** 'gray', lw **=** 2, linestyle **=** '--')

plt**.**xlim([0.0, 1.0])

plt**.**ylim([0.0, 1.05])

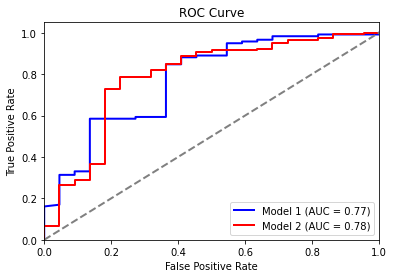
plt**.**xlabel('False Positive Rate')

plt**.**ylabel('True Positive Rate')

plt**.**title('ROC Curve')

plt**.**legend(loc **=** 'lower right')

plt**.**show()



In [ ]:

library(ggplot2)  
  
 data <- read.csv("C:/Users/amsab/Downloads/CapstoneCleaned.csv")  
  
 # 1. Frequency distribution of AI\_Satisfaction  
 table(data$AI\_Satisfaction)

##  
 ## NO YES  
 ## 113 521

# 2. Cross table of AI\_Satisfaction with AI\_Trust  
 table(data$AI\_Satisfaction, data$AI\_Trust)

##   
 ## High Low Moderate  
 ## NO 39 21 53  
 ## YES 123 57 341

# 3. Bar chart: Proportion of AI\_Satisfaction across different Living\_Region  
 ggplot(data, aes(x = Living\_Region, fill = AI\_Satisfaction)) +  
 geom\_bar(position = "fill") +  
 labs(title = "Proportion of AI Satisfaction by Living Region",  
 x = "Living Region",  
 y = "Proportion") +  
 theme\_minimal()

# 4. Density plot: Distribution of AI Usage grouped by AI Satisfaction

ggplot(data, aes(x = AI\_Usage, fill = AI\_Satisfaction)) +

geom\_density(alpha = 0.6) +

labs(title = "Distribution of AI Usage by AI Satisfaction",

x = "AI Usage",

y = "Density") +

theme\_minimal()

# 5. Scatter plot: AI\_Usage vs AI\_Trust, colored by AI\_Satisfaction  
 ggplot(data, aes(x = AI\_Trust, y = AI\_Usage, color = AI\_Satisfaction)) +  
 geom\_point(alpha = 0.6, size = 3) +  
 labs(title = "AI Usage vs AI Trust by AI Satisfaction",  
 x = "AI Trust",  
 y = "AI Usage") +  
 theme\_minimal()