**Predicting Allstate Customer Insurance Coverages**

Team #5: Joshua Bryer, Kindell Holst, William McGuire, Denesha Record

April 29th, 2025

**Table of Contents**

1. Executive Summary............................................................................................................3
2. Business Understanding......................................................................................................3
   1. Objective.................................................................................................................3
   2. Background.............................................................................................................3
   3. Success Criteria.......................................................................................................4
3. Data Understanding............................................................................................................4
   1. Data Sources...........................................................................................................4
   2. Initial Data Exploration...........................................................................................4
   3. Data Quality Issues.................................................................................................8
4. Data Preparation..................................................................................................................9
   1. Cleaning Steps........................................................................................................9
   2. Feature Engineering..............................................................................................10
   3. Final Dataset.........................................................................................................10
5. Modeling...........................................................................................................................11
   1. Model Selection....................................................................................................11
   2. Training Procedure................................................................................................12
   3. Hyperparameter Tuning........................................................................................12
   4. Model Evaluation Metris......................................................................................13
6. Evaluation.........................................................................................................................13
   1. Model Performance..............................................................................................13
   2. Interpretation.........................................................................................................14
   3. Error Analysis.......................................................................................................15
7. Deployment.......................................................................................................................15
   1. Prototype Integration............................................................................................15
   2. Automated Scoring Pipeline.................................................................................15
   3. Model Retraining Strategy....................................................................................15
   4. Performance Monitoring and Feeback Loops.......................................................16
   5. Documentation and Knowledge Transfer.............................................................16
   6. Project Retrospective and Continuous Improvement...........................................16
8. Conclusion and Recommendations...................................................................................16
   1. Summary...............................................................................................................16
   2. Recommendations.................................................................................................17
   3. Limitations............................................................................................................17
9. Bibliography.....................................................................................................................
10. Appendix...........................................................................................................................18
    1. Code Snippets.......................................................................................................18
    2. Glossary of Terms.................................................................................................24
    3. Data Dictionary.....................................................................................................25

**Executive Summary**

Insurance carriers are always looking for an edge to stand out in a hypercompetitive industry. An edge may be in the form of better pricing, catchier jingles, or better data solutions. In this project we will act as consultants helping the insurer Allstate to stand out from the crowd. We intend to develop an advanced data solution for Allstate to help them outperform their competitors. This solution comes in the form of a data model which will be used to identify what policies consumers are likely to purchase so the company can make better recommendations in the purchase process, saving the customer time and frustration. The data model will be trained on Allstate’s client database and depending on the model, it should allow them to inform clients why we propose certain policies based on their characteristics. This is valuable to many consumers as insurance can be confusing, and knowing which coverage makes sense based on your individual needs is hard to determine. Providing informed recommendations based on the client’s information will ease the process for consumers looking to find the right coverage at the right price. A client's feelings of comfortability in making a purchase are critical to gaining and maintaining business. A company that does that better than others will always stand out and through this method, Allstate will stand out in the insurance market.

**Business Understanding**

**Objective**

The business problem we will address is how to increase customer satisfaction, boost overall sales, and improve retention rate by using Allstate customer’s shopping history data. Our efforts will be centered on developing new bundle options for customers, streamlining the quoting experience, and providing information for personalized marketing. This all hinges on a model that can predict which coverage options suit a customer the best. This creates our technical goal, creating a model that can predict customer coverage preferences as accurately as possible.

**Background**

As key consultants who have been hired by Allstate, we have identified the primary stakeholder as Allstate itself. This includes Allstate’s board of directors, shareholders and especially the employees that hired us for this particular task. Their primary goals are to increase the profitability of the company, which leads back to the business problem stated above.

**Success Criteria**

The primary technical success criteria we will use to evaluate the model is an accuracy rate above 70% when predicting customer insurance coverage choices. We kept this threshold low because there are 7 different types of coverage, all with varying levels. Outside of the technical criteria, the stakeholders would want to see an increase in insurance plans sold of 2% within the first year of deployment. Allstate is already the one of the largest insurance providers in the United States, so a 2% difference has a significant impact.

**Data Understanding**

**Data Sources**

The data we are using is an insurance database exported into a .csv file. The data comes from the major insurer Allstate’s internal customer database. This data includes customer’s interactions with Allstate, the quotes Allstate provided as well as the purchases customers ultimately made. This dataset has been filtered to only include Allstate clients who went on to make a purchase of at least one of Allstate’s coverage options.

**Initial Data Exploration**

The dataset consists of 25 columns and 198857 rows. The dataset contains a mixture of primarily categorical variables with some continuous and discrete variables. The following tables contain the descriptive statistics for each variable according to their type as well as the distribution for discrete and continuous variables.

**Nominal Variables**

|  |  |  |
| --- | --- | --- |
| Column name | Format | Anomalies? |
| customer\_id | ######## | Repeats client ids for every shopping\_pt |

**Categorical Variables**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Column Name | Values | | | | | | | Anomalies? |
| day | 0 | 1 | 2 | 3 | 4 | 5 | 6 |  |
| 21% | 21% | 20% | 19% | 18% | 1% | <1% | Few clients on the weekends |
| state | FL | NY | PA | OH | MD | IN | WA | This dataset does not include Texas or California |
| 16% | 14% | 9% | 7% | 4% | 4% | 4% |
| CO | AL | CT | TN | KY | NV | MO |
| 4% | 4% | 3% | 3% | 2% | 2% | 2% |
| OR | UT | OK | MS | AR | WI | GA |
| 2% | 2% | 2% | 2% | 2% | 2% | 1% |
| NH | ME | NM | ID | RI | KS | Other |
| 1% | 1% | 1% | 1% | 1% | 1% | 4% |
| location | 10083 | 10213 | 10348 | 11195 | 11517 | Other |  |  |
| 16% | 15% | 15% | 13% | 13% | 28% |
| car\_value | e | f | d | g | h | c | Other | car\_value is not a number |
| 33% | 27% | 17% | 15% | 4% | 3% | 1% |
| risk\_factor | 3.0 | 4.0 | 1.0 | 2.0 |  | | | NaN |
| 28% | 26% | 23% | 23% |
| C\_previous | 3.0 | 1.0 | 2.0 | 4.0 |  | | | NaN |
| 42% | 27% | 17% | 15% |
| A | 1 | 0 | 2 |  | | | |  |
| 64% | 22% | 14% |
| B | 0 | 1 |  | | | | |  |
| 55% | 45% |
| C | 3 | 1 | 2 | 4 |  | | |  |
| 41% | 31% | 20% | 9% |
| D | 3 | 2 | 1 |  | | | |  |
| 61% | 23% | 16% |
| E | 0 | 1 |  | | | | |  |
| 55% | 45% |
| F | 2 | 0 | 1 | 3 |  | | |  |
| 38% | 33% | 24% | 5% |
| G | 2 | 3 | 1 | 4 |  | | |  |
| 40% | 29% | 21% | 10% |

**Binomial Variables**

|  |  |  |  |
| --- | --- | --- | --- |
| Column Name | 0 | 1 | Anomalies? |
| married\_couple | 79% | 21% |  |
| homeowner | 54% | 46% |  |
| record\_type | 85% | 15% |  |

**Continuous and Discrete Variables**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Column Name | Mean | Std. | Min | Max | Anomalies? |
| time (in minutes) | 789.65 | 161.91 | 1 | 1439 | Average time is past mid-day |
|  | | | | | |
| shopping\_pt | 4.22 | 2.39 | 1 | 13 |  |
|  | | | | | |
| group\_size | 1.23 | 0.46 | 1 | 4 |  |
|  | | | | | |
| car\_age | 8.14 | 5.76 | 0 | 85 |  |
|  | | | | | |
| age\_oldest | 44.99 | 17.40 | 18 | 75 |  |
|  | | | | | |
| age\_youngest | 42.58 | 17.46 | 16 | 75 |  |
|  | | | | | |
| duration\_previous | 6.00 | 4.69 | 0 | 15 | NaN |
|  | | | | | |
| cost | 635.79 | 45.99 | 260 | 922 |  |
|  | | | | | |

**Data Quality Issues**

For a majority of the data, it is fully populated and does not have issues impeding data modelling. There were though some issues that were discovered. Firstly, the dataset is composed of duplicate client IDs reported for each client shopping point. This makes it more complicated to model the data, but we are able to move forward by only looking at shopping points of purchase. Additionally, it was discovered that there were 18,000 records that were missing values in the “C\_previous” and “duration\_previous” columns. This creates an issue in our progress and substitution is the path we are going to take to overcome it. Lastly, the “car\_value” column did not reflect the numeric amount, but instead a letter classification. This means we will have to approach this field as a categorical variable instead of the more ideal continuous variable. These issues are troublesome, and we will go over how we dealt with them.

**Data Preparation**

**Cleaning Steps**

Most data collected needs to be cleaned and ours is no different. There are 18000 records missing in the C\_previous and duration\_previous columns. A cursory glance seems to show that these all have associated customer records so these missing values can be replaced with accurate values. These records are a small portion of the total records so it would seem like simply removing them wouldn’t have a significant impact on the analysis of over 600000 records however, removing some of the records from an individual customer while retaining others would give as an incomplete picture of the customer and could skew the results. In reviewing the missing fields, each customer we checked had a value for the final record. Therefore, we decided the best course of action was to fill in the blank entries with the field below using excel. Because the data set is ordered by customer id, and all the last entries are the purchase point, this maps the correct values to each customer. There may be a few cases where the incorrect value is filled in but there should not be enough to significantly affect the final analysis. The next more significant set of missing values was in the risk\_factor column. For almost all customers we checked, if the risk factor was missing from one customer record, it was missing from all records for that customer. The risk factor was missing from over 200,000 records, roughly one third of the records in the dataset. Eliminating these records would result in a significant loss of data that could greatly affect the accuracy of the analysis. There is no good way to calculate the risk factor from the data, so we decided to treat the blank entries as a separate rating level, unknown. This seems reasonable as an unknown risk factor would make an insurer more hesitant to insure a potential customer. We replaced the blank entries with a risk factor of 5 to enable easier analysis. After these replacements were done, there was no missing data in the dataset.

Next, we checked for outliers. For car age and previous duration, we used the quartiles and interquartile range to find upper and lower limits of 1.5 times the IQR above the third quartile and below the first quartile. We found about 6 thousand records for car age that were above the upper limit of 25.5. These records represent a group of vehicles considered classic cars. Classic cars have special considerations such as stated value policies and values that may go up over time and should be considered separately. We will need to consider removing these vehicles from the dataset and may perform analysis with and without them to see how they affect the model’s accuracy. We applied the same quartile strategy to the cost and found around 7000 outliers. However, we created a histogram of the data, and the cost is normally distributed with no visible outliers. We believe these “outliers” are reasonable to include in the analysis.

**Feature Engineering**

There were two columns we added to get some more insights from the data set. First, to make time of day analysis easier, we created an hour of the day column. This alters any time of day from 8:22, to 8. This allows for easier categorical analysis with this variable.

The second is creating a new column that is very similar to car value. This column is called car value number and simply puts the same values on a numerical scale instead of alphabetical. So, for example, a is equal to 1 and b is equal to 2. This, much like the first calculated variable, helps with analysis based on car value, because numeric values are simply easier to navigate.

After delving into models, it was found that one hot encoding could help input categorical variables into the model. This meant a drastic increase in columns, as the state attribute has 48 attributes and the location attribute has many, many more. Because this one hot encoding was used for specific moments in select models, our final dataset presented in the next section sub section does not include these additional attributes.

**Final Dataset**

The original dataset had a size of 665250 rows and 25 columns. After performing the data preparation, the new dataset has a size of 665250 rows and 27 columns. This aligns with what we performed, because all our cleaning didn’t delete any rows, and we added two new columns. It is important to note that the most important information is the final sell rows. These will be of the highest value, so noting their size is important. There are 97010 final sale records in the original and new dataset, meaning we have all the information we need to start making a model.

**Modeling**

**Model Selection**

To get the best model possible, we tried three different models: Random Forest, Logisitic Regression, and Artificial Neural Networks. These models were chosen due to their ability to process large data sets in a quick fashion compared to other models. Logisitic Regression could only be used on insurance policies B and E, because they have only two possible values. There are ways to expand Logisitic regression through one vs rest, but it did not seem appropriate for this dataset with other models' capabilities.

The first model attempted was random forest. We developed a model in r that allowed all insurance policies to be predicted sequentially. The true problem was that it never performed well. Running the model with every possible parameter and variable led to every insurance policy to be 71% accurate or lower. This was very dissatisfying and clearly would not fulfill our goals. To see if logistic regression would help for B and E, we attempted it next and received results that were very similar to random forest. This led us to our next model, artificial neural networks.

Unlike the previous models, the code for neural networks was performed in python. Unfortunately, many of the same results were happening with only the numerical variables and neural networks do not work with traditional categorical variables. The work around is using one hot encoding to make every category a column of its own that can only be true or false. Once this change was made, the performance of the model improved. This improvement leads us to choose artificial neural networks as our model for predicting insurance policy choices of customers.

There were some important findings that led to better and worse performance. We previously mentioned the use of one hot encoding for specific attributes. The most daunting attribute was using location as a predictor. Location is a categorical variable with a large number of categories, making it very process heavy to create dummy variables. Not just creating, but running those variables takes a long time for the model. For this reason, only a select number of policies were chosen to have the one hot encoded location columns. The policies were B, D, F, and G. During development there was a mistake that led to perfect results for some of the policies. This later was found to be the model predicting using knowledge of other policies. This error was found after the progress report, leading to a drop in performance since then.

**Training Procedure**

The cross-validation technique used is stratified k-fold. We chose to use k-fold cross validation because it helps build our confidence that this model is generalizable and that it will perform well on unseen data sets. The reason we chose a stratified version is because most of the insurance policies in the data set have a non-uniform distribution. Stratified samples help to ensure the model is receiving a similar distribution to the whole data set.

**Hyperparameter Tuning**

The hyperparameters in this model are the number of layers, number of neurons in each layer, the activation function for each layer, and the number of epochs. For this model, the number of epochs is reasonably low, at 3. This is because overfitting quickly became a problem once the number of epochs was higher, and the accuracy and loss typically stagnated between 2 and 3 epochs. Another benefit is the quicker run times. The activation functions come next. For the hidden layers, our activation function of choice was the relu function. Relu is a prime choice for hidden layers, along with sigmoid and tanh, with the main benefit of relu avoiding the vanishing gradient problem. Relu also is less computationally expensive because of its simplicity, but that doesn’t mean it is perfect. The main worry is called dying relu, where specific neurons are forever stick on the negative side of the function and never activate. Ensuring good normalization can help avoid this defect. For the output layer, we chose the softmax function. Softmax is the go-to choice for multiple category classification and should be exactly what we need due the insurance policies with more than two options. The final two hyperparameters are the number of layers and number of neurons. After a tremendous amount of experimentation, the model benefitted most with one hidden layer. The number of neurons for the output layer was made to always be the number of options for each insurance policy. The input and hidden layer were decided by more experimentation, that led to 64 in the input layer and 128 in the hidden layer.

**Model Evaluation Metrics**

The primary metric for this classification problem is accuracy. The approximate accuracy, f1 score, and loss performance of the of model is as follows:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | A | B | C | D | E | F | G |
| Accuracy | 71.9% | 66.1% | 73.4% | 74.2% | 67.9% | 64.2% | 67.2% |
| F1 score | 67.9% | 66.1% | 73.1% | 72.9% | 67.9% | 62.4% | 65.9% |
| Loss | 0.670 | 0.631 | 0.715 | 0.647 | 0.589 | 0.882 | 0.851 |

These results leave much to be desired. Not a single policy is predicted above 75% with policy D being the closest at 74.2%. Most of the f1 scores are close to the accuracies which gives us confidence that the errors are distributed quite evenly. Three out of the seven policies hit our goal of 70% accuracy or above. Further tweaking should be attempted to get the other four policies to that threshold.

**Evaluation**

**Model Performance**

|  |  |
| --- | --- |
| Target Variable - A | Confusion Matrix |
| We can see in the confusion matrix that the model does well in predicting values of ‘A’, but that it struggled to accurately predict when values were equal to two. This suggests that our model will not always capture when two is the best plan for a customer. |  |
|  | |
| Target Variable - B | Confusion Matrix |
| The model predicting values of ‘B ‘was fairly accurate, often would predict 0 or 1 when the other was the true value. |  |
|  | |
| Target Variable - C | Confusion Matrix |
| Column ‘C’ was one of our more accurate target variables. It did well in predicting each variable but missed about 30% of the time on each outcome. |  |
|  | |
| Target Variable - D | Confusion Matrix |
| The model did well in predicting values of ‘D’. The model did strongly to predict values of 2 but struggled with values of 0 and 1. This could be because of an imbalance in the dataset. |  |
|  | |
| Target Variable - E | Confusion Matrix |
| Column ‘E’ was a target variable our model did well with. It appears though that our model struggled with misclassifying zeros for ones, but worse with ones for zeros. |  |
|  | |
| Target Variable - F | Confusion Matrix |
| The model struggled the most with variable ‘F’. The model missed more values of three than it got correct and seemed to only do well in predicting values of zero. |  |
|  | |
| Target Variable - G | Confusion Matrix |
| The model did well and predicted the values of ‘G’ It did not appear to struggle with any one value. |  |

**Interpretation**

Our target accuracy value for our model is >70% for each target value. This would mean that when we make suggestions for what clients should purchase, it should meet their needs more than 7 out of ten times for each different recommended plan. From the model evaluation metrics, we can see that we are over that statistic on 3 variables of the target variables, but not on variables B, E, F, and G. The rest are over the threshold and meet our objective. We are not at our desired accuracy, but we believe our model still shows valuable predictive power.

**Error Analysis**

There were no discernible trends to our model’s shortcomings. Our model will indiscriminately struggle with various target variables. Additionally, there were no discernible limitations to our data and no unexpected findings from our model. The lack of errors and surprises in our data could be because of some of our choices we made in data preparation, such as only including purchase rows. Regardless of the reason, the lack of errors was a boon to our efforts.

**Deployment**

To transition our predictive model from development into Allstate’s production environment, we propose the following phased deployment plan:

**Prototype Integration**  
 We will package the trained neural network as a RESTful microservice that exposes a “predict coverage” endpoint. This service can be containerized and deployed within Allstate’s existing quoting system or customer-facing portal. When a customer enters their details, the front end will call the microservice in real time and display not only the recommended coverage levels, but also a brief rationale pulled from the model’s top feature importances to enhance transparency and trust.

**Automated Scoring Pipeline**  
 For returning customers or bulk renewals, we will implement a nightly batch job that reads new or updated customer records from the data warehouse, applies the model to generate updated coverage recommendations, and writes the results back to the CRM. This will ensure that Allstate’s agents always have the latest personalized recommendations at their fingertips. We’ll leverage the same microservice in a scheduled fashion via an orchestrator such as Apache Airflow.

**Model Retraining Strategy**  
 To maintain accuracy as customer behavior and market conditions evolve, we will establish a monthly retraining pipeline. At the end of each month, newly labeled policy purchase data will be appended to the training set; data preparation and one‑hot encoding scripts will run automatically; and the model will be retrained, validated, and versioned. We will promote the new model into production only after it surpasses the current model on a hold‑out validation set.

**Performance Monitoring and Feedback Loops**  
 We will build a real‑time monitoring dashboard that tracks key metrics prediction latency, input feature distributions, overall accuracy on sampled hold‑out data, and business KPIs such as conversion rate uplift. If we detect data drift or performance degradation (e.g., accuracy drops below 70%), automated alerts will notify the data science and operations teams. In parallel, customer feedback (via post‑quote surveys) will be incorporated as an additional labeled data source for future retraining.

**Documentation and Knowledge Transfer**

A comprehensive deployment playbook will be delivered, covering API specifications, data schemas, environment configurations, rollback procedures, and troubleshooting guides. We will conduct hands‑on workshops with Allstate’s IT and business teams to ensure they can operate, monitor, and extend the solution independently.

**Project Retrospective and Continuous Improvement**  
 After the initial go‑live, we will host a retrospective to capture lessons learned across data collection, modeling, and deployment. This will inform enhancements for subsequent iterations such as adding new data sources, exploring alternative architectures (e.g., serverless inference), or refining the user interface to better communicate recommendations.

**Conclusion & Recommendations**

**Conclusion**

Our group set out to help Allstate stand out in the crowd by creating a data solution in their policy selection process. After experimenting with multiple approaches and multiple models, we settled on an artificial neural network model trained on customer’s purchase information. Unfortunately, we did not reach our accuracy goal for each target variable. Despite our self-imposed shortcomings, we were successful in creating a model that has a high success rate in predicting one or more selections for a consumer’s insurance policy. This application will absolutely enable a streamlined process in the client’s purchasing journey and give Allstate an edge over their competitors.

**Recommendations**

We recommend that Allstate implement this model into their recommendation systems on their client’s purchasing journey. We also recommend that they install a series of regular updates to keep the model trained on the most relevant up to date data. We see room for improvement in the model and believe that we can improve our model’s accuracy through further experimentation and the inclusion of more relevant data that Allstate may be able to provide us with (i.e. Marketing data).

**Limitations**

We experienced some computational limits with our model building and see room for model revisions using more robust systems. This was particularly true for us in our attempt to model using the variable “Location”. This variable was informative to our model’s creation and informs the risk factors associated with our client’s. Unfortunately, due to the sheer number of values in this variable, our computational power was insufficient to model with the variable and meet our timeline. Another crucial limitation we experienced was a lack of understanding of the Allstate purchase process. We were not provided with the details of client purchase order. Had we known this information, it is plausible, we would have been able to make more accurate suggestions for clients based on their previous selections.

**Bibliography**

“Allstate Purchase Prediction Challenge.” *Kaggle*, 2014, [www.kaggle.com/competitions/allstate-purchase-prediction-challenge/data](https://www.kaggle.com/competitions/allstate-purchase-prediction-challenge/data).

**Appendix**

**Code Snippets**

Artificial Neural Network (Python)

# Get packages

import numpy as np

import pandas as pd

from sklearn.preprocessing import StandardScaler

import tensorflow as tf from tensorflow

import keras from sklearn.metrics

import f1\_score from sklearn.metrics

import confusion\_matrix from sklearn.model\_selection

import StratifiedKFold

def set\_up():

'''Sets up the data for the model'''

# Load in data  
data = pd.read\_csv("AllState\_clean\_data.csv")  
  
# Grab only final purchase  
data = data[data['record\_type'] == 1]  
  
# Get important columns  
data = data.iloc[:,[3,5,6,7,8,9,10,12,13,14,15,16,17,18,19,20,21,22,23,24,25]]  
  
# Make car\_value\_number an integer  
data['car\_value\_number'] = data['car\_value\_number'].astype('int')  
  
# Create dummy variables and remove location  
data = pd.get\_dummies(data, columns = ['homeowner','married\_couple','state'])  
loc = data['location']  
data.drop('location', axis = 1, inplace = True)

return data, loc

def run\_model(x,y,v):

'''Runs the artificial neural network model'''  
 acc\_temp = []  
 loss\_temp = []  
 f1\_temp = []  
  
# Cross validation using stratified k-fold  
skf = StratifiedKFold(n\_splits=3, shuffle=True, random\_state=30) for train\_index, test\_index in skf.split(x, y): X\_train, X\_test = x.iloc[train\_index], x.iloc[test\_index]  
 y\_train, y\_test = y.iloc[train\_index], y.iloc[test\_index]  
   
 # Standardize the data  
 sc = StandardScaler()  
 X\_train = sc.fit\_transform(X\_train)  
 X\_test = sc.transform(X\_test)  
   
 # Configure the model  
 model = keras.Sequential()  
 model.add(tf.keras.layers.Dense(64, activation = 'relu',  
 input\_shape = (X\_train.shape[1],)))  
 model.add(tf.keras.layers.Dense(128, activation = 'relu'))  
 model.add(tf.keras.layers.Dense(v, activation = 'softmax'))  
   
 # Compile the model  
 model.compile(optimizer = 'adam', loss = 'sparse\_categorical\_crossentropy', metrics = ['accuracy'])  
   
 # Model summary  
 model.summary()  
   
 # Actual Learning  
 model.fit(X\_train, y\_train, epochs = 3)  
   
 # Using test data to gauge accuracy, loss and f1 score  
 testLoss, testAccuracy = model.evaluate(X\_test, y\_test)  
 acc\_temp.append(testAccuracy)  
 loss\_temp.append(testLoss)  
 pred = model.predict(X\_test)  
 pred\_labeled = []  
 for i in range(len(pred)):  
 pred\_labeled.append(np.argmax(pred[i]))  
 print(confusion\_matrix(y\_test, pred\_labeled))  
 f1\_temp.append(f1\_score(y\_test, pred\_labeled, average = 'weighted'))  
   
return acc\_temp, loss\_temp, f1\_temp

def print\_metrics(accuracy, loss, f1):

'''Prints the three metrics'''

print(f"Accuracy: {accuracy}")  
print(f"Loss: {loss}")  
print(f"f1: {f1}")

def main():

data, loc = set\_up()  
  
# Set up insurance policies to select and empty lists  
policies = ['A','B','C','D','E','F','G']  
accuracy = []  
loss = []  
f1 = []  
  
# Setting up x with and without location  
x\_1 = data.drop(policies, axis = 1)  
x\_2 = x\_1  
x\_2['location'] = loc  
x\_2 = pd.get\_dummies(x\_1, columns = ['location'])  
  
# Loop over insurance policies  
for i in policies:  
  
 # Depending on insurance policy, setting x, y, and v  
 if i == 'A':  
 x = x\_1  
 y = data['A']  
 v = 3  
 if i == 'B':  
 x = x\_2  
 y = data['B']  
 v = 2  
 if i == 'C':  
 x = x\_1  
 y = data['C']  
 y = y - 1  
 v = 4  
 if i == 'D':  
 x = x\_2  
 y = data['D']  
 y = y - 1  
 v = 3  
 if i == 'E':  
 x = x\_1  
 y = data['E']  
 v = 2  
 if i == 'F':  
 x = x\_2  
 y = data['F']  
 v = 4  
 if i == 'G':  
 x = x\_2  
 y = data['G']  
 y = y - 1  
 v = 4  
   
 acc\_temp, loss\_temp, f1\_temp = run\_model(x,y,v)  
   
 # Add new results to lists  
 accuracy.append(sum(acc\_temp)/len(acc\_temp))  
 loss.append(sum(loss\_temp)/len(loss\_temp))  
 f1.append(sum(f1\_temp)/len(f1\_temp))  
   
print\_metrics(accuracy, loss, f1)

if name == "main":

main()

Random Forest (R)

# Load in data

data <- read.csv("AllState\_clean\_data.csv") set.seed(123)

# Grab only final purchase

data <- data[data$record\_type == 1,]

# Select important fields

data <- data[,c(4,6,7,9,10,11,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27)]

# Change attributes to correct types

data$car\_value\_number <- sapply(data$car\_value\_number, as.integer)

data$homeowner <- sapply(data$homeowner, as.factor)

data$married\_couple <- sapply(data$married\_couple, as.factor)

data$state <- sapply(data$state, as.factor)

data$C\_previous <- sapply(data$C\_previous, as.factor)

# Prep variables and libraries

x <- c('A','B','C','D','E','F','G')

accuracyforest <- c()

library(randomForest)

library(dplyr)

# Run model

for (i in x) {

if (i == 'A') {

data\_new <- data[,1:14] data\_new <- rename(data\_new, y = A)

}

if (i == 'B') {

data\_new <- data[,c(1:13,15)] data\_new <- rename(data\_new, y = B)

}

if (i == 'C') {

data\_new <- data[,c(1:13,16)] data\_new <- rename(data\_new, y = C)

}

if (i == 'D') {

data\_new <- data[,c(1:13,17)] data\_new <- rename(data\_new, y = D)

}

if (i == 'E') {

data\_new <- data[,c(1:13,18)] data\_new <- rename(data\_new, y = E)

}

if (i == 'F') {

data\_new <- data[,c(1:13,19)] data\_new <- rename(data\_new, y = 'F')

}

if (i == 'G') {

data\_new <- data[,c(1:13,20)] data\_new <- rename(data\_new, y = G)

}

data\_new$y <- sapply(data\_new$y, as.factor)

myForest <- randomForest(y ~ ., nodesize = 100, mtry = 3, ntree = 50, data = data\_new) confusionforest <- myForest$confusion

if (i == 'A' || i == 'D') {

accuracyforest <- c(accuracyforest, sum(confusionforest[c(1,5,9)]) / sum(confusionforest[c(1,2,3,4,5,6,7,8,9)]))

}

if (i == 'B' || i == 'E') {

accuracyforest <- c(accuracyforest, sum(confusionforest[c(1,4)]) / sum(confusionforest[1:4]))

}

if (i == 'C' || i == 'F' || i == 'G') {

accuracyforest <- c(accuracyforest, sum(confusionforest[c(1,6,11,16)]) / sum(confusionforest[c(1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16)]))

}

}

Logistic Regression (R)

#Load in data

data <- read.csv("AllState\_clean\_data.csv") set.seed(123) library(dplyr)

#Grab only final purchase

data <- data[data$record\_type == 1,]

#Select important fields

data <- data[,c(4,6,7,9,10,11,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27)] data$car\_value\_number <- sapply(data$car\_value\_number, as.integer)

data$homeowner <- sapply(data$homeowner, as.factor)

data$married\_couple <- sapply(data$married\_couple, as.factor)

data$state <- sapply(data$state, as.factor)

data$C\_previous <- sapply(data$C\_previous, as.factor)

data[1:13] <- data[1:13] %>% mutate\_if(is.numeric, scale)

x <- c('B','E')

accuracylr <- c()

for (i in x) {

if (i == 'B') {

data\_new <- data[,c(1:13,15)]

data\_new <- rename(data\_new, y = B)

}

if (i == 'E') {

data\_new <- data[,c(1:13,18)]

data\_new <- rename(data\_new, y = E)

}

data\_new$y <- factor(data\_new$y, levels = c("0","1"))

#Partitioning are data, training and testing sets

library(caret)

partition <- createDataPartition(data\_new$y, p = 0.7, list = FALSE)

train <- data\_new[partition,] test <- data\_new[-partition,]

#Logistic model

m1 <- glm(y ~ ., family = binomial(link = "cauchit"), data = train)

#Test

fitted.results <- predict(m1, test, type = 'response')

fitted.results <- ifelse(fitted.results > 0.5, '1','0')

accuracylr <- c(accuracylr, confusionMatrix(as.factor(fitted.results), as.factor(test$y))$overall[[1]])

}

**Glossary of Terms**

Allstate – An insurance company with a wide range of coverage options and one of the biggest insurers in the United States.

Policies – The coverage options that are the predicted variable in the model. Variables A, B, C, D, E, F, G in the data set.

Random Forest – Supervised machine learning technique that creates a random number of decision trees based off a set of rules. Then each tree gets a vote with the highest vote getter being the predicted class.

Artificial Neural Network – Supervised machine learning technique that mimics the process of a biological neural network. It uses layers of neurons that activate depending on a specific mathematical function. Each neuron has weight which needs to be optimized to yield the best result.

Epoch – One complete use of the training set in a machine learning model, primarily used terminology in artificial neural networks.

F1 Score – A way to measure predictive performance using true positives, true negatives, and false negatives. Traditionally used when classes are imbalanced in the data set.

API – Application Programming Interface. Allows for communication between different applications using a set of rules and protocols.

**Data Dictionary**

* customer\_ID: A unique identifier for each customer
* shopping\_pt: The point of the shopping experience, e.i. how many offers so far
* record\_type: Whether this is the purchase instance or not. 1 = purchase, 0 = non-purchase
* day: Day of the week
* time: Time of the day in hour and minutes
* time(hour): The hour at which the instance occurred
* state: The US state where the instance occured
* location: The unique location the instance took place at
* group\_size: The number of people under the policy (1 to 4)
* homeowner: Whether the customer is a homeowner. 1 = yes, 0 = no
* car\_age: The age of the car
* car\_value: How valuable the car is when originally purchased (a to g)
* car\_value\_number: Converted car\_value into integers from 1 to 7
* risk\_factor: An assessment of how risky a driver is (1 to 4)
* age\_oldest: The age of the oldest person in the group
* age\_youngest: The age of the youngest person in the group
* married\_couple: Whether the customer group has a married couple. 1 = yes, 0 = no
* C\_previous: What the customer currently or formerly had for policy C (0 to 4)
* duration\_previous: How many years the customer was covered by their previous insurer
* A, B, C, D, E, F, G: The different coverage options
* cost: The cost of the quoted coverage options