Justin Pickel
December 10, 2020
CS 315

CS 315 PROJECT REPORT

INTRODUCTION

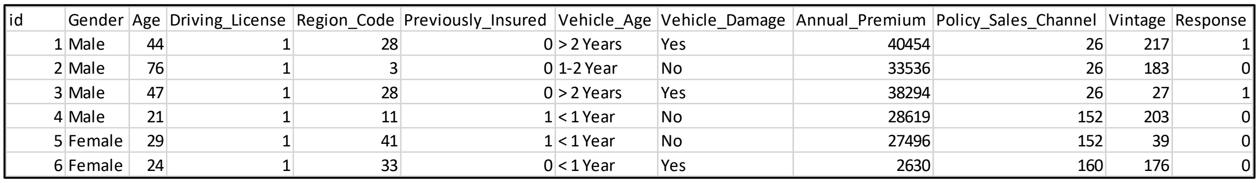
The purpose of this project is to build a machine learning model that can accurately predict if an individual is likely to purchase vehicle insurance based on demographic and vehicle features. One of my motivations for understanding how to build a machine learning classifier comes from the many uses this type of algorithm has in real-world applications. I have always been motivated to figure out which features produce the highest classification accuracy and whether the features change depending on the model you implement. Knowing how data is classified and then predicting the label for a given dataset has many uses, from healthcare to sports betting. Another motivation I have for understanding machine learning classifiers is to have the ability to automate classification. We all know the world has an awe-inspiring amount of data. Understanding which features should be used to classify data in an automated manner brings tremendous value to any organization. With the dataset used in this project, we can implement this very model to help narrow down the call list given to insurance salesman, saving the company time and money. Instead of cold calling every individual on the call list, we can leverage the machine learning classifier to help reduce the number of people listed on the calling list.

The challenge for me on this project is learning how to represent the data so the model can easily understand the data. Another challenge for me was creating a data pipeline that would allow me to observe the accuracy of each feature combination across different models. My approach to this project included creating all possible feature combinations and recording the feature combination that produced the highest accuracy for each model. In this project, I will be using different types of classifiers to determine if the feature combination that produces the highest accuracy changes between models. The models I decided to observe was a KNN, SVM, and a decision tree. I will be using the python library sklearn and python's collection library to produce the feature combinations.

I have discovered that the feature combinations are different across various models used in this project. I also discovered that the highest accuracy achieved was 87.2% using a support vector machine. To my surprise, within a given model, there are multiple feature combinations that achieve the highest accuracy.

DATA MINING TASK

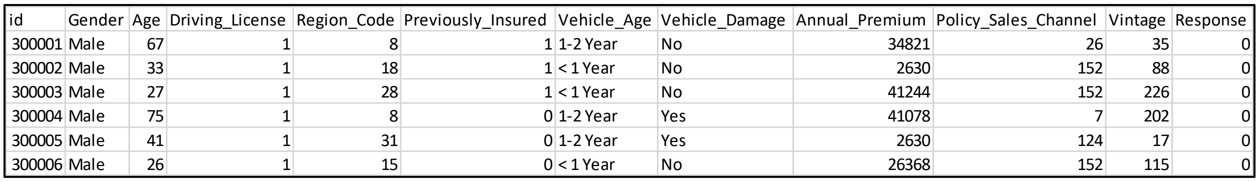
This task includes two different inputs and produces a feature combination as the output. The feature combination produced will be associated with the specific model I am observing. The first input data is known as the training data. This dataset contains eleven different features with a total of 1,000 rows. The features are id(int), Gender(str), Driving_License(Bool), Regional_code(int), Previously_Insured(Bool), Vehicle_Age(str), Vehicle_Damage(Bool), Annual_Premium(int), Policy_Sales_Channel(int), Vintage(int), Response(Bool). I used all ten features in the project, ignoring the id feature since this adds no value to the algorithm. The id is a personal identification associated with the row of data.



id	Gender	Age	Driving_License	Region_Code	Previously_Insured	Vehicle_Age	Vehicle_Damage	Annual_Premium	Policy_Sales_Channel	Vintage	Response
1	Male	44	1	28	0	> 2 Years	Yes	40454	26	217	1
2	Male	76	1	3	0	1-2 Year	No	33536	26	183	0
3	Male	47	1	28	0	> 2 Years	Yes	38294	26	27	1
4	Male	21	1	11	1	< 1 Year	No	28619	152	203	0
5	Female	29	1	41	1	< 1 Year	No	27496	152	39	0
6	Female	24	1	33	0	< 1 Year	Yes	2630	160	176	0

Figure 1: example of the training dataset.

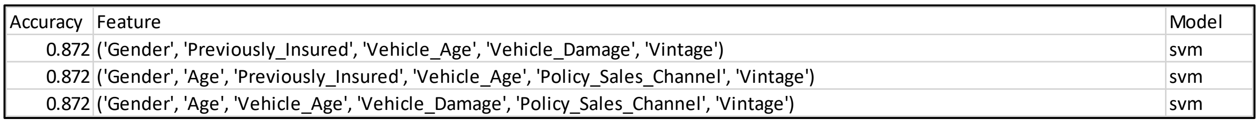
The second input is the test data which contains 1,000 rows and has the same information as the training dataset. The response column is the label assigned to the customer in which a 1 represents they have purchased vehicle insurance and a 0 indicates they have not.



id	Gender	Age	Driving_License	Region_Code	Previously_Insured	Vehicle_Age	Vehicle_Damage	Annual_Premium	Policy_Sales_Channel	Vintage	Response
300001	Male	67	1	8	1	1-2 Year	No	34821	26	35	0
300002	Male	33	1	18	1	< 1 Year	No	2630	152	88	0
300003	Male	27	1	28	1	< 1 Year	No	41244	152	226	0
300004	Male	75	1	8	0	1-2 Year	Yes	41078	7	202	0
300005	Male	41	1	31	0	1-2 Year	Yes	2630	124	17	0
300006	Male	26	1	15	0	< 1 Year	No	26368	152	115	0

Figure 2: example of the testing dataset.

The output of the model is a csv file named “output” that contains the accuracy of the model at testing time and the feature combination that was used to produce the accuracy. There will be three different outputs each corresponding to the observed model.



Accuracy	Feature	Model
0.872	('Gender', 'Previously_Insured', 'Vehicle_Age', 'Vehicle_Damage', 'Vintage')	svm
0.872	('Gender', 'Age', 'Previously_Insured', 'Vehicle_Age', 'Policy_Sales_Channel', 'Vintage')	svm
0.872	('Gender', 'Age', 'Vehicle_Age', 'Vehicle_Damage', 'Policy_Sales_Channel', 'Vintage')	svm

Figure 3: Output from model

The data mining questions I had asked during this project are:

- Does the feature combination that produces the highest accuracy change between models?
  - What combination of features produces the highest accuracy for each model?
  - What model produces the highest accuracy?

The key challenge I faced was understanding how to manipulate the data best so that the model can easily understand the data. This includes converting the gender, vechicle_damage and vechicle_age column to a numerical value. Another key challenge was going to be learning the sklearn library. I have not used this library to its full extent, and beings there is a lot of information within the docs, it would be challenging to understand everything that I can do.

TECHNICAL APPROACH

I will begin my project by reading in the training and testing datasets using pandas read_csv method. Once the data is read in, I will remove the "id" column and then convert all the features to numeric values for both datasets. After the datasets have been converted to numerical values, I will extract the features and create a list that contains all possible feature combinations. I will then loop through all feature combinations and record the feature combination and accuracy measure that produced each model's highest test accuracy. I will then compare the accuracies between the different models and whether the combination feature is different.

Some ways I am addressing the key challenges listed above is to convert all" Male"," Female" and "Yes", "No" values to a 1 or a 0, respectively. I have read the sklearn docs and have watched a couple of YouTube tutorials about how to work with the sklearn library. My last challenge was understanding how to convert the categorical label for the column vehicle_age to a numerical value. To accomplish this last challenge, I read some articles about how to perform label encoding in python. I have decided to give a numerical representation for the three different categories found in the column. The following encoding was performed for the vehicle_age column:

“< 1 Year”: 1
“1-2 Year”: 2
“> 2 Years : 3

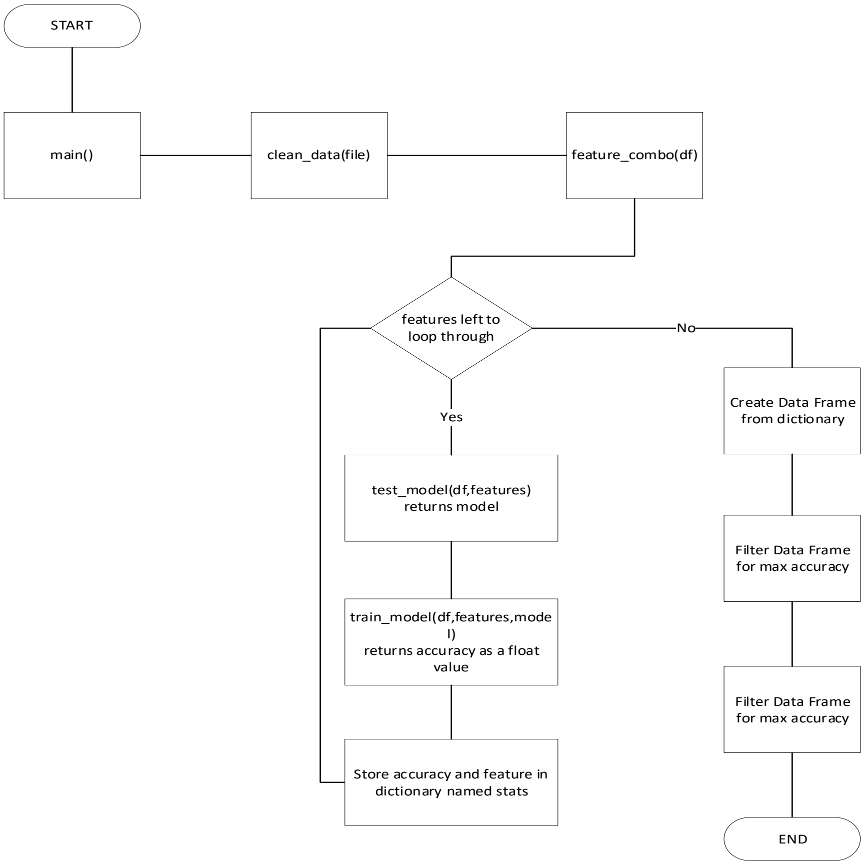

START
main()
clean_data(file)
feature_combo(df)
features left to loop through
No
Yes
test_model(df,features) returns model
train_model(df,features,model) returns accuracy as a float value
Store accuracy and feature in dictionary named stats
Create Data Frame from dictionary
Filter Data Frame for max accuracy
Filter Data Frame for max accuracy
END

Figure 4: Block diagram of program

EVALUATION METHODOLOGY

The data used in this project was sourced from Kaggle and some information about the data includes information about demographics (gender, age, region code type), Vehicles (Vehicle Age, Damage), Policy (Premium, sourcing channel). The data is open source and contains some preprocessed steps such as all NA's have already been removed and the train data is separate from the testing data (two files). The context description about the data from Kaggle states "Our client is an Insurance company that has provided Health Insurance to its customers now they need your help in building a model to predict whether the policyholders (customers) from past year will also be interested in Vehicle Insurance provided by the company.

An insurance policy is an arrangement by which a company undertakes to provide a guarantee of compensation for specified loss, damage, illness, or death in return for the payment of a specified premium. A premium is a sum of money that the customer needs to pay regularly to an insurance company for this guarantee." Some challenges I faced when using this data from Kaggle includes converting the gender, vehicle_damage and vehicle_age column from a categorical variable to a numerical value.

The metrics used to evaluate if the model is a success is the average mean accuracy score which is a method implemented by the sklearn library. The higher the accuracy score the higher the success of the model.

RESULTS AND DISCUSSION

The results are shown in figures 5 through 7; each table shows the highest accuracy achieved and the associated feature vector used to train the model. To my surprise, multiple combinations produce the highest accuracy for a given model. I am astonished that eight different feature combinations produce the same accuracy when using the support vector machine model. I am also surprised that each model used a different feature combination to produce the highest accuracy. No feature combination is shared between models, at least not for the most accurate ones.

For the support vector machine model, I used the radial basis function kernel and set the hyperparameter C, also known as the regularization parameter, to 1.0 and also did some reading about the gamma parameter offered in the sklearn library. Upon reading, it showed that a gamma of 0.1 and a C of 1.0 was the best-tuned model for the radial basis function.

When setting these parameters, the model achieved the highest accuracy from several tests I had conducted. Once the hyperparameters were set, I fed in all 1,012 feature combinations and recorded the test time's resulting accuracy for each feature combination. After observing the output shown in figure 5. You can see eight different combinations of feature vectors, each resulting with an accuracy of 87.1871%.

Looking at figure 6. We can see the results from the decision tree classifier. The parameters used int his project are all the default values from the sklearn library. Looking at the results, we have four feature combinations that resulted in an accuracy of 87.0870%, not much of a difference from the SVM model, yet the feature combinations are entirely different. Interestingly, I noticed that 'gender,' 'policy_insured,' and 'policy_sales_channel' appeared to be the most common feature used within the combinations.

Looking at figure 7, we can see the results from the K-nearest-neighbors model with the K parameter set to three. This model's highest accuracy was 86.9869%, again not too far off from the previous two models. However, once again, they share no common feature combination between any of them. With only two feature vectors producing the highest accuracy, you can observe from figure 7. that we do not have a common feature among the combinations. Interestingly, each model has a different number of combinations that produce the highest accuracy, no combination feature is shared across the three models, yet the accuracy between them is not that far off.

So, to answer the research questions; Does the feature combination that produces the highest accuracy change between models? yes! The feature combination that produces the highest accuracy changes between models and it even changes within the model. What combination of features produces the highest accuracy for each model? This can be observed by looking at figures 5 through 7. As we have stated before, the support vector model has eight different feature combinations. The decision tree model has four different feature combinations, and the k-nearest neighbors has 2 different feature combinations. Lastly, what model produces the highest accuracy? The support vector machine using the radial basis function kernel and setting the hyperparameter C to 1.0 as well as the gamma parameter to 0.1 produces the most accurate model with an incredible 87.1871%.

Support Vector Machine

Accuracy	Feature	Model
0.872	('Gender', 'Previously_Insured', 'Vehicle_Age', 'Vehicle_Damage', 'Vintage')	svm
0.872	('Gender', 'Age', 'Previously_Insured', 'Vehicle_Age', 'Policy_Sales_Channel', 'Vintage')	svm
0.872	('Gender', 'Age', 'Vehicle_Age', 'Vehicle_Damage', 'Policy_Sales_Channel', 'Vintage')	svm
0.872	('Gender', 'Driving_License', 'Previously_Insured', 'Vehicle_Age', 'Vehicle_Damage', 'Vintage')	svm
0.872	('Gender', 'Age', 'Driving_License', 'Previously_Insured', 'Vehicle_Age', 'Policy_Sales_Channel', 'Vintage')	svm
0.872	('Gender', 'Age', 'Driving_License', 'Vehicle_Age', 'Vehicle_Damage', 'Policy_Sales_Channel', 'Vintage')	svm
0.872	('Gender', 'Age', 'Previously_Insured', 'Vehicle_Age', 'Vehicle_Damage', 'Policy_Sales_Channel', 'Vintage')	svm
0.872	('Gender', 'Age', 'Driving_License', 'Previously_Insured', 'Vehicle_Age', 'Vehicle_Damage', 'Policy_Sales_Channel', 'Vintage')	svm

Figure 5: Results for Support Vector Machine with hyperparameter C = 1.0 and gamma = 0.1

Decision Tree

Accuracy	Feature	Model
0.871	('Gender', 'Previously_Insured', 'Policy_Sales_Channel')	clf
0.871	('Gender', 'Driving_License', 'Previously_Insured', 'Policy_Sales_Channel')	clf
0.871	('Gender', 'Previously_Insured', 'Vehicle_Damage', 'Policy_Sales_Channel')	clf
0.871	('Gender', 'Driving_License', 'Previously_Insured', 'Vehicle_Damage', 'Policy_Sales_Channel')	clf

Figure 6: Results for Decision Tree

K-Nearest Neighbor

Accuracy	Feature	Model
0.87	('Gender', 'Region_Code')	knn
0.87	('Previously_Insured', 'Vehicle_Damage')	knn

Figure 7: Results for K-Nearest Neighbor using K=3

LESSONS LEARNED

I learned a great deal from this project; most importantly, I am way more comfortable with the sklearn library as I now understand most of the lingo. The most interesting part for me during this project was seeing the different feature vectors used to produce the most accurate model. I am still mind-blown about the features that produced the highest accuracy changes when using different models. I also learn a lot about the radial basis function when using a support vector machine. I would stick with a smaller data set in hindsight because having three hundred thousand plus rows took a very long time for all the models to be trained and tested. Therefore, I decided to only train on 1000 rows and test on 1000 rows.

As far as improving the project, I am happy with the results, but I now have more questions to investigate that I cannot fit in this report. I am interested in seeing the relationship between the different parameters and how exactly all these knobs work with or against each other.

ACKNOWLEDGMENTS

I want to personally thank Dr. Jillepalli for his continuous and timely responses to my e-mails throughout the semester. I also want to thank Dr. Doppa for creating the CS 315 data mining class as it has expanded my understanding about machine learning skills.

Below are the website links I used to help accomplish this project:

https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html#sklearn.svm.SVC.score

https://towardsdatascience.com/how-to-find-the-optimal-value-of-k-in-knn-35d936e554eb

Justin Pickel
December 10, 2020
CS 315

https://www.kaggle.com/anmolkumar/health-insurance-cross-sell-prediction

https://scikit-learn.org/stable/auto_examples/svm/plot_rbf_parameters.html

https://pbpython.com/categorical-encoding.html

https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html

https://www.youtube.com/watch?v=0Lt9w-BxKFQ