# Midterm Project

**Tyler Chang**

STA 160

Professor Fushing Hsieh

30 April 2020

Tyler Chang
914163826

# Midterm Project

STA 160
April 30, 2020

## Part 1: Automobile Data

### *Introduction*

The car insurance industry is one of the largest growing markets in the United States. In 2019, Policy Genius reported that the average car insurance premium was a stunning $1,099. Reports have also shown that rates have skyrocketed by 29.6% nationally since 2011. As a result, many people across the country wonder why insurance companies are so successful.

The main problem that the insurer faces is figuring out who they should insure and at what price. The consumer purchases a contract with the insurance company that specifies a (typically) monthly price premium, $\$p$, and deductible, $d$, the proportion of losses that the insurer will not cover. If the buyer gets into an accident, the insurer will cover $(1-d)\%$ of the damages costs. However, if the buyer does not get into an accident then the insurance company does not have to pay anything. Naturally, the goal of any company is to maximize their profit margins. Thus, the insurer has a strong incentive to insure customers that will not suffer any losses. However, there is no certainty in the market that can predict whether a customer will suffer a loss. Whether it be a natural disaster, surrounding bad drivers, or luck, insurance agencies cannot control such events from happening.

As a result, the insurance company must rely on the information they have- the data. Using the *Automobile* data set from the UC Irvine Machine Learning Repository, the goal of this project is to highlight information about a car that can be useful to insurance companies when thinking about who they should do business with. Moreover, we want to develop a model that can efficiently classify the risk of a car.

### *Target Variable: Symbolling*

The *Automobile* data set from the UC Irvine Machine Learning repository contains several pieces of information on 205 different cars, ranging from make, horsepower, price, etc.. One feature that stands out in particular is called "symbolling." Actuarians use the term "symbolling" to represent the riskiness of an asset. In this data set, symbolling is defined from

[-3,3], where lower values represent less risky cars.

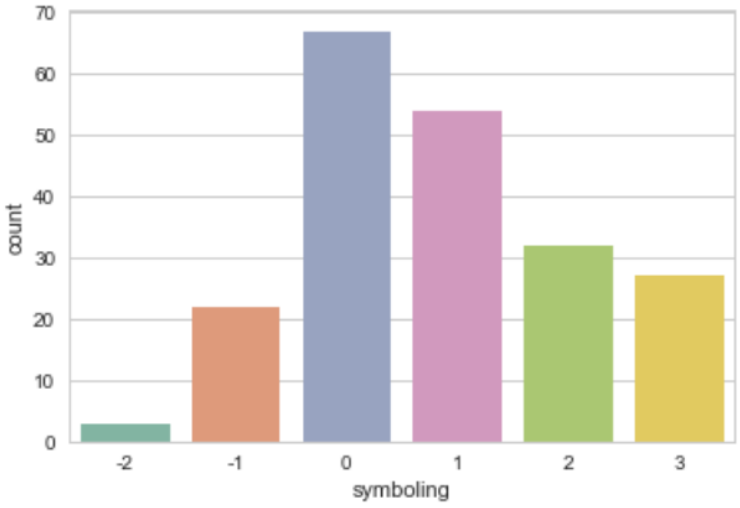


*Figure 1: Distribution of Risk*

The distribution of each car's risk is plotted above (*Figure 1*). It appears that most of the cars sampled have some risk associated with it, while a far less proportion of cars are considered to be safe. Furthermore, it is worth noting that none of the cars sampled had a symbolling level of $-3$. That is, none of the cars were at the highest tier of safety. Instead of using each car's original symbol, we will simply consider negative values as "not risky," values of zero as "neither," and positive values as "risky."

***Important Categorical Features***

Now knowing that each car has a label of risk attached to it, naturally we want to determine if there are certain features of the data that could affect a car's risk. In particular, we will focus on the categorical features of the data set that could have a potential effect. Out of the 10 categorical features, we will highlight two of the more intuitive features of the model-make and body style.

*Make*

The make of a car refers to the company that sold it. This would be particularly useful for

insurers so they have a better idea of what companies they should consider avoiding.

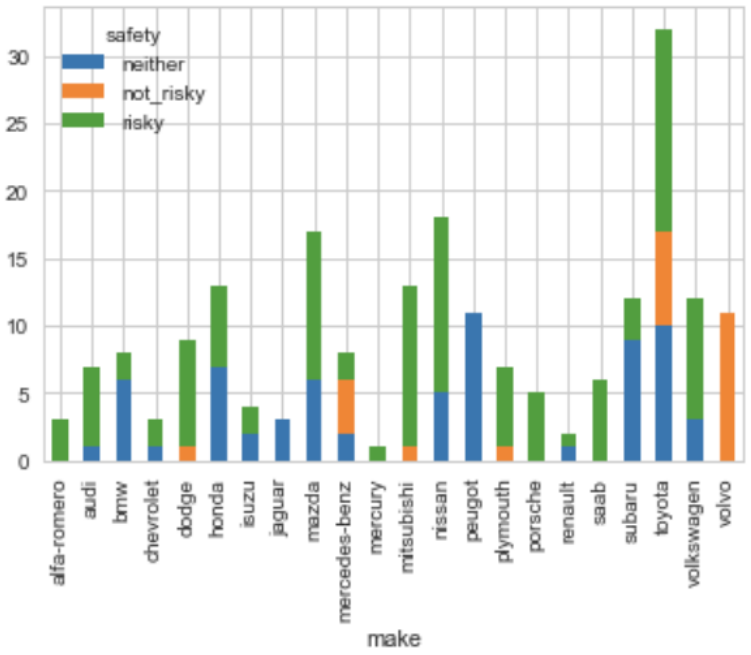


*Figure 2: Proportion of Risk Across Make*

Visually, it is obvious that there are some companies, such as Volvo, who produce particularly non-risky vehicles. While other companies, like Saab Automobile, have a significantly higher proportion of risky cars. Therefore, it may be in the insurer's best interest to pay special attention to the car's make when considering their contracts.

*Body Style*

There are five different body styles included in this data set: convertible, hard top, hatchback, sedan, and wagon. This could be useful for insurers because upon visual inspection of a car, they could possibly associate certain car styles with more or less risk.

Plotted in *Figure 3* shows the distribution of risk based on the a car's body style. It appears that more "sporty" looking cars (convertibles, hardtops, hatchbacks) are especially more risky than more economy, family-friendly sized cars (sedans, wagons). Similar to a car's make, insurance companies may also want to put special importance on the car's body style

when making their decision.

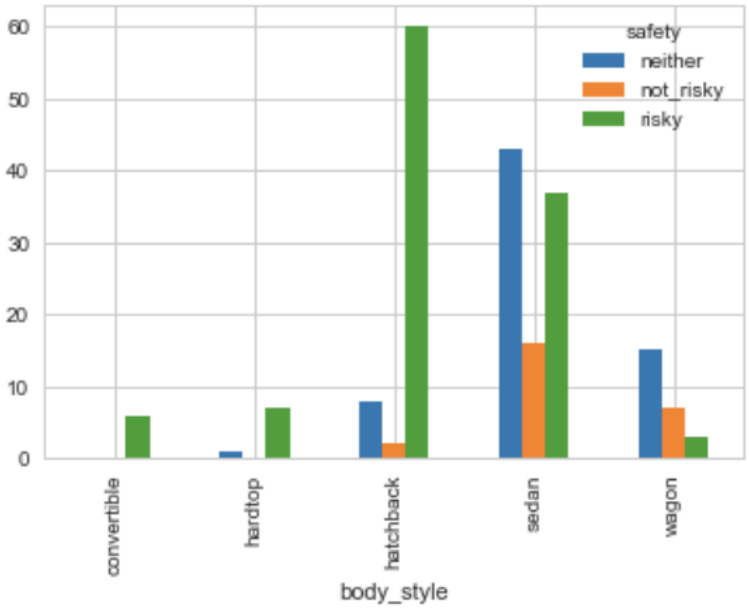


*Figure 3: Proportion of Risk Across Body Style*

***A More Rigorous Approach to Finding Categorical Importance***

Most of the categorical features in the data set have this similar pattern of having an effect on a car's safety. However, it is not enough for us to make this determination simply through visuals alone. Rather, we will use a more rigorous procedure to determine if a categorical feature is important.

In particular, we shall utilize the Chi-Square Test for Independence to test the following hypotheses for each of the categorical features:

$H_0$: *There is no association between the feature and its risk level*

$H_1$: *There is a association between the feature and its risk level*

After performing the test for each categorical variable, the results are as follows:

| Feature | Test Statistic | p-value |
|---|---|---|
| Number of Doors | 88.374070 | 6.453787e-20 |
| Make | 182.975338 | 1.636926e-19 |
| Body Style | 64.875648 | 5.107186e-11 |
| Drive Wheels | 21.644956 | 2.358120e-04 |
| Engine Type | 33.101207 | 9.334302e-04 |
| Fuel Type | 11.316578 | 3.488481e-03 |
| Fuel System | 31.048465 | 5.457554e-03 |
| Aspiration | 6.256719 | 4.378958e-02 |
| Engine Location | 2.478752 | 2.895648e-01 |
| Number of Cylinders | 11.115924 | 5.190110e-01 |

All but two categorical features were statistically significant at $\alpha = 0.05$, namely the car's engine location and number of cylinders. Furthermore, one can use each feature's corresponding p-value as a means of determining a feature's importance. Thus, the insurance company could take into special consideration of visible, qualitative features such as make and number of doors, and not pay much attention to features like engine location.

### *Classification*

It may also be particularly useful for an insurance company to have a machine learning model that can accurately classify a car's risk level. In this section, we will utilize several supervised learning techniques to use a car's given qualities to predict its level of risk.

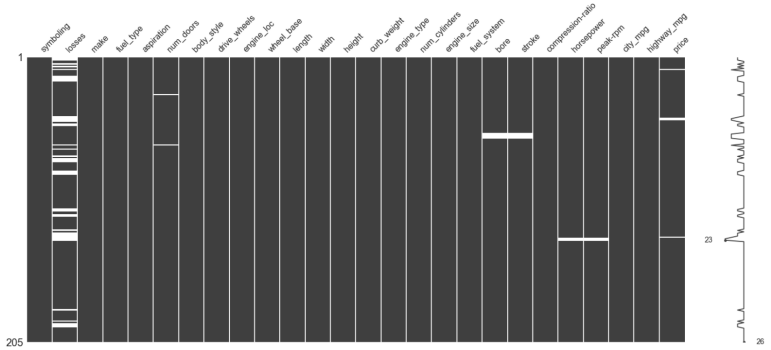


*Figure 4: Missing Data Matrix*

Prior to fitting the models, we first had several data pre-processing steps that had to be addressed first. The data set had quite a few missing values (*Figure 4*), especially for the feature called losses. A car's losses refers to its expected dollar loss in the event of an accident. We would expect there to be a relationship between a cars losses and its risk, so it was an important part of the pre-processing step to not simply get rid of the information altogether. Instead, we utilized the *sklearn* library to impute (KNN) the missing values for this feature and other missing values across the data matrix.

After imputing the data, the next pre-processing step was to encode all of the variables to properly run each model. The resulting encoding lead to over 100 total features for the models to take into account. Using over 100 different variates for classification lead to adequate but not optimal results in terms of 10-fold cross validation. Thus, we ranked each of the variates by means of Chi-Squared to determine how many "K-best" predictors were needed to maximize the performance of each model.

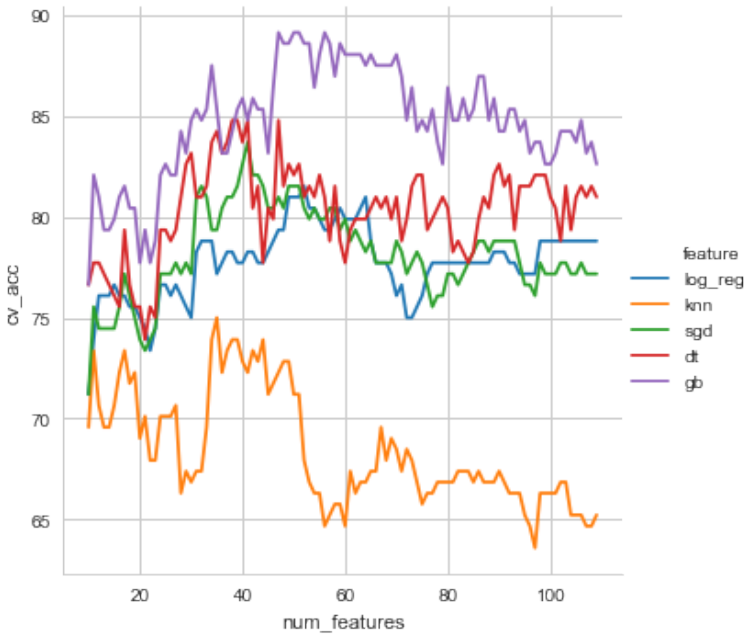


*Figure 5: Model Performance*

From this, it appears that the Gradient Boosting Classifier performed the best out of each of the classifiers, with a 10-fold cross validation score of over 90% (*Figure 5*). Interestingly, although it is clear that some models outperform others, each model follows the same general trend in accuracy results. That is, with a smaller amount of features, the model performs accuracy decreases until it meets a slightly higher number of features, where its accuracy peaks. Once each model reached its peak accuracy, their respective scores gradually decrease as the number of features increase. It is also worth reemphasizing that each model performs at its best when *not* using all of the encoded features. Rather, each of the models performed its best when using less than half the amount of predictors.

### *Conclusion*

It is crucial for car insurance companies to use any available information to aide them in determining how to price customers on an individual basis. With this data analysis, it is clear that there are some qualitative features than can be associated with a car's risk. However, it is equally important that the insurer carefully tests the significance of such associations

to ensure its validity. For example, it could be assumed that the number of cylinder's a car engine has can make a car more risky (since a higher number of cylinders can potentially make a car faster). However, we showed that in fact this assumption was not statistically valid, and concluded that this particular feature may not be a reliable indicator of risk.

# Part 2: Seeds Data

***Introduction***

A large part of machine learning and data science is the area of dimensionality reduction. In many real world settings, the number of features can be extremely large. These massive data matrices can take up a lot of computing power as well as long run times. The goal of dimensionality reduction is to reduce the number of features while maintaining a majority of the information captured by the full data set.

Quite similar to the famous *Iris* data set, the UC Irvine Machine Learning Repository provides data on wheat kernels. The data set provides several numerical features on over 200 different seeds. Each seed is labelled by its corresponding species: Karma, Rosa, or Canadian. Furthermore, each seed is described by several features that, at first glance, could possibly be related to one another. The goal of this section will be to utilize several dimensionality reduction techniques and compare classification performances among the full and reduced data matrices.

***Principal Component Analysis***

The *Seeds* data set has 7 quantitative features that describe 210 different seeds. More specifically, the seven features are: area, perimiter, compactness, length, width, asymmetry, and groove length. In this section, we will utilize a dimension reduction technique, principal component analysis (PCA), to determine if the seeds can be described by a fewer amount of features called principal components.

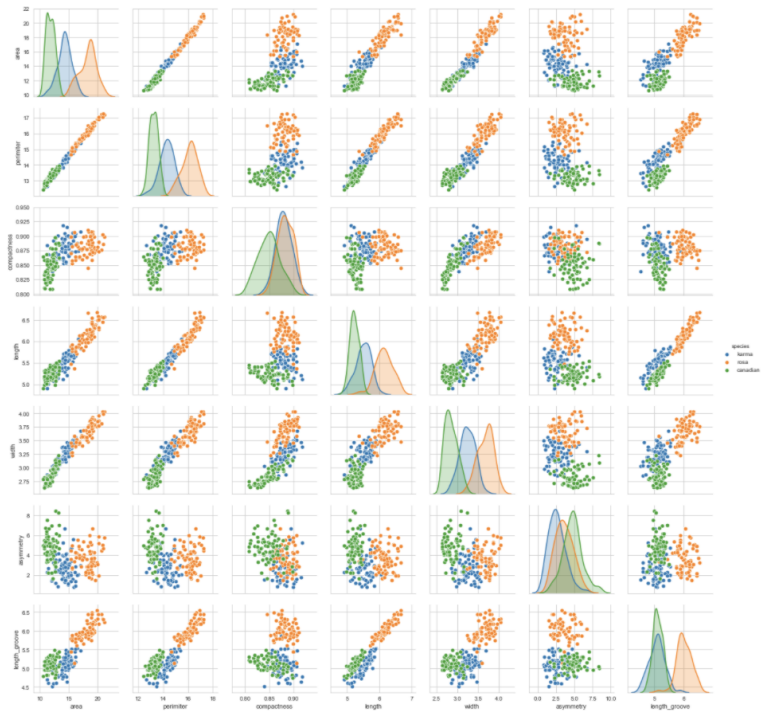

*Figure 6: Scatter Plot Across Features*

*Figure 6* shows a scatter plot of each combination of the variates. Notice that the species of each kernel is generally clustered together across every combination of variates. Also notice that for some of the variates there appears to be clear associations, but for some a relationship is not as clear. The goal of PCA is to capture the story told by all of the variates in a smaller number of principal components.

The exact values corresponding to each principal component are not particularly important, but rather the amount of information that is maintained after reducing the feature size. To quantify the amount of information maintained, we observe the porportion of variance each principal component contributes to the overall variance. Since $Var(Y_k) = \lambda_k$, the proportion of total variance the first $k$ principal components is defined as:

$$\frac{\lambda_1 + ... + \lambda_k}{\lambda_1 + ... + \lambda_p}$$

| | PC 1 | PC 2 | PC 3 | PC 4 |
|---|---|---|---|---|
| Proportion of Variance | 0.71874303 | 0.17108184 | 0.09685763 | 0.00976635 |
| Cumulative Proportion | 0.71874303 | 0.88982487 | 0.9866825 | 0.99644885 |

Notice that the cumulative proportion of variance for the first two principal components is almost 90%. That is, only about 10% of information was lost after the reduction. Furthermore, we were able to capture almost all (98.7%) of the variance with just three principal components, essentially cutting the number of variates in half.

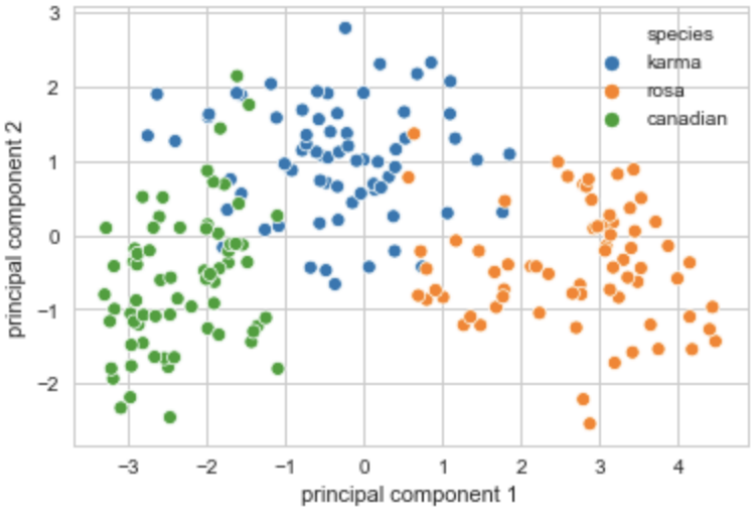


*Figure 7: First 2 Principal Components*

Visually, we can see that using either 2 principal components or 3 principal components, the same general pattern of the species clustering together was maintained. This is consistent with the results obtained from the proportion of variance maintained using those principal components. Thus, we can conclude that a vast majority of the information captured by the original feature set can be captured by a smaller number of principal components.

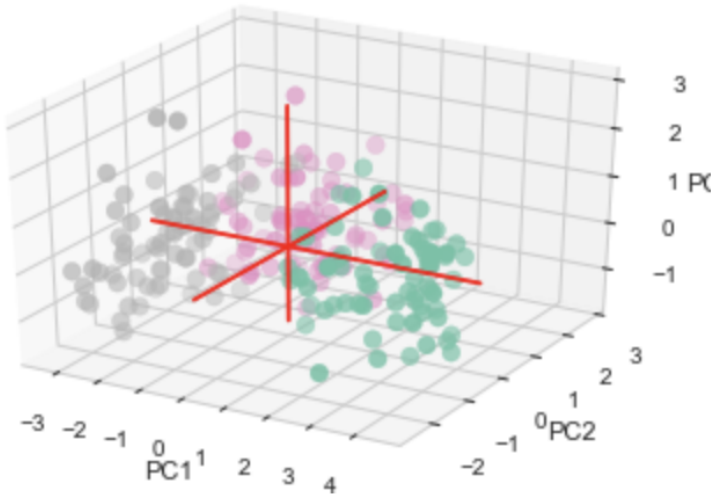


*Figure 8: First 3 Principal Components*

***Relationship Between Features***

Another approach we can take in reducing the dimensions of this data matrix is to simply remove some highly correlated features. In general, correlated features do not improve model performance by much. We will observe if this holds true when comparing model performance with reduced feature matrices. Furthermore, removing corrlated features also can prevent problems that arise from multicolinearity and overfitting.

Plotted in *Figure 9* is a heat map that shows visually how any two given features are correlated. It appears that many of the features are highly, positively correlated with one another. Since the most of the features of this data set essentially describe the size of the kernel, it is not surprising that each are highly correlated. This observation can be further seen through the asymmetry feature. Asymmetry is the only feature in the data set that is not a direct or indirect means of calculating the size of a kernel. Thus, the asymmetry feature does not behave similarly in correlation to the other features.

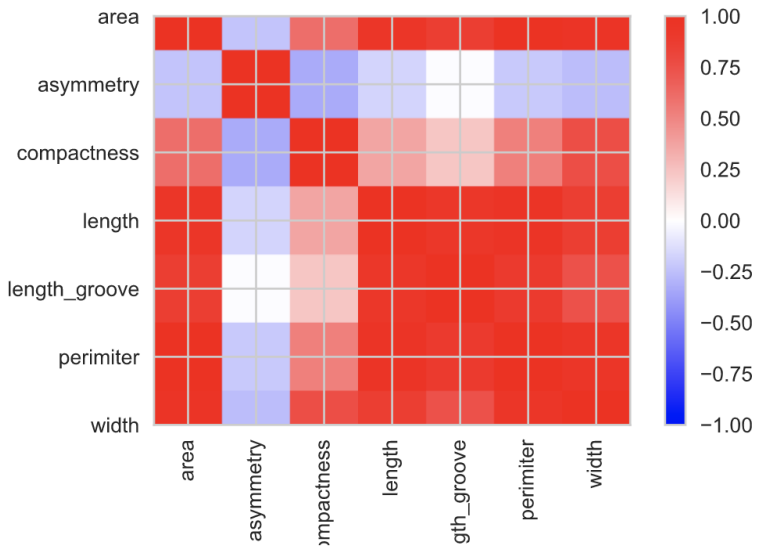


*Figure 9: Correlation Matrix*

***Classification***

Knowing that many of the features are highly correlated, one can reduce the size of the feature matrix by removing the highly correlated variates. In this section, we will perform classification using this reduced feature matrix, the first three principal components, and the original feature matrix.

| Model | All | Subset | PC 3 |
|---|---|---|---|
| Logistic Regression | 0.9107 | 0.8869 | 0.8988 |
| KNN | 0.8690 | 0.8869 | 0.8929 |
| Stochastic Gradient Descent | 0.9286 | 0.8690 | 0.8929 |
| Decision Tree | 0.8929 | 0.8750 | 0.8750 |
| Gradient Boost | 0.9167 | 0.8869 | 0.8750 |

The table above shows the differences in 10-fold cross validation scores for the given models. The first column of values shows the scores using all seven features. The second column under 'Subset' are the scores only using the features: area, asymmetry, and compactness. We effectively removed the other four highly correlated features (length, width, perimeter,

and groove length), since they all could be used when determining the area of any given kernel. Finally, the third column under 'PC 3' are the CV scores for the first three principal components which we determined captured approximately 98% of the variance.

Although using all of the variates had the highest CV scores (besides KNN) across the models, using every feature did not perform significantly better. In fact, the other two situations performed notably well considering each case only had 3 total variates to classify over 200 different seeds. Also, both reduced feature matrices were actually able to outperform the original feature set when using a KNN classifier.

***Conclusion***

From this analysis we have seen that, although it may be a good idea to use all pieces of information available, it may not be completely necessary. Based on the results, it was shown that we were able to essentially cut the number of predictors in half while obtaining only slightly worse performance. Fortunately, the size of the data matrix was quite small, so the benefit of using less computing power and shorter run time are not as obvious. However, the ability to reduce the size of the feature matrix is crucial when working with bigger, messier data sets, where the size can be quite large and the classes do not neatly cluster together.