## ***Project Overview:***

As a starting point, my project is a collaborative recommender system for BMW cars. The user would provide their preference for a number of car features and the system would compare their feature vector to other users and then provide a recommendation based on the most similar users.

## ***Project Tasks:***

The main tasks involved for the project was dataset collection and curation, implementation of KNN and Naive Bayes algorithms, test data collection, analysis of the performance of the algorithms and the project report.

## ***Approach:***

I decided to do a collaborative system over a content based system because people with similar tastes/preferences will usually buy similar cars. Next, I chose the KNN algorithm because it has good performance even with a high number of classes and it provides a good baseline accuracy to beat. Lastly, I decided to use Naive Bayes as the 2nd algorithm to implement because it is more challenging and can provide better performance than KNN.

## ***Steps Implemented:***

1. Dataset collection and curation
2. Implementation of KNN and Naive Bayes algorithms using Spark
3. Test Data collection
4. Analysis of performance of the algorithms

## ***Context/Project Motivation***

My motivation for this project was to create a recommender system for car consumers that are similar to me. Most of the existing systems only work with basic car features like body style, engine size and price. Furthermore, car consumers that are similar to me will have requirements for less basic features, for example minimum 0-60 and minimum horsepower. However, these features are not present in any recommender system that I have found. I think it would be really cool/interesting if a recommender system were created for car consumers that are similar to me. It would be much easier to identify cars that meet my specific needs and would be very popular amongst consumers similar to myself.

## ***Results and Performance Evaluation***

The algorithms were able to easily classify inputs that had preferences similar to most users. However, both did struggle with rare preferences that were less common in the training data.
For example, one test sample had a maximum price of 118,000 and they purchased an 840I. However, the users maximum price is much higher than the actual price of the car(103,000). As a result, both algorithms classified this sample into a BMW M850I because the price was unusual and both cars have similar features.

KNN: I implemented functionality that tested the best value for K on the test data. The best accuracy was obtained when K was equal to 3. The K value is small because the dataset is small and completely balanced. A higher value for K results in the algorithm performing worse on test data with rare preferences because most of the top K samples will not be related to the rare sample.

The best accuracy achieved by KNN was 75% over the 25 test samples.

Naive Bayes: Naive bayes performed much better on the dataset compared to KNN, as it achieved an accuracy of 83.33%.
After analyzing the predictions of both algorithms, Naive Bayes most likely performed better because euclidean distance in KNN was unable to distinguish the difference between samples from different classes that are similar. The Gaussian probability density function used in Naive Bayes is more effective with similar samples from different classes. It incorporates the mean and standard deviation, which is more informative than euclidean distance.

The accuracy could have been much higher from both algorithms if I had used more features. The features were not informative enough to distinguish between 2 similar classes/BMW models.

## ***Dataset***

I created the dataset myself using my knowledge of BMW models and how consumer preferences differ based on the model they purchase. For example, someone who purchases a BMW M850i is going to have a higher preference for horsepower and speed, compared to someone who purchases a base model BMW 840i.

Next, the features used in the dataset are: Maximum Price, Minimum Horsepower, Minimum Engine Size, Body Style(Convertible, sedan), Body Size (Small, Medium, Large).

The training data contained 200 samples and the test data contained 25 samples.

## ***Aspects of Proposal Completed:***

1. Definitely Accomplish: Collected and annotated a dataset, implemented KNN algorithm, provided an analysis of the performance of the algorithms.

2. Likely to Accomplish: I initially wanted to enable input from users but that was not the focus of the project.

3. Ideally Like to Accomplish: Implemented the Naive Bayes Algorithm

## ***Additional Observations/Comments***

Challenges: The main challenge was time management, as I started the project late. I also have 3 classes and each had a project due within a week period. Not to mention, I am a TA for ITIS 4166 so I had a lot going on.

The Naive bayes algorithm was also challenging, due to my inexperience with python. I had to familiarize myself with python data structures like the dictionary and list, in order to implement the algorithm.

Dataset collection and curation was much more challenging than I anticipated. When I initially tested the algorithms with the test data the performance was much lower. The algorithms did not work well for unusual/rare preferences amongst the features. I had to expand the training set to incorporate samples that were difficult for the algorithms to classify. Lastly, some of the classes/BMW models had very similar features and it was difficult to differentiate the samples between the two classes. For example, the M850I and M550I both have the same horsepower, engine size and the 0-60 times only differ by .2 seconds. With this in mind, it was difficult for the algorithms to distinguish between these classes.