Used Cars Listings Data Mining KSDAMIN1KU

Miguel Angel Crozzoli
Artistic Research at RMC
micr@itu.dk

Anders C. Ketelsen
Technical University of Denmark
aket@itu.dk

Emmanuel Adonis Turay
ITU Copenhagen
emmt@itu.dk

I. The Data

A. What knowledge are you trying to extract?

Anyone who has been in the used car market and who is somewhat interested in mathematics and data, has probably asked themselves questions like: "What is the best time to buy or sell your car?", "Which color best preserve value?" and just in general "What price should I put my car at?" These are the questions we will use data mining to try to answer in this report.

To do so we will make use of a dataset "US used cars dataset" retrieved from Kaggle. Furthermore, as a curiosity we will see if it's possible to detect whether a state is predominantly republican, or democrat purely based on the cars sold in the state and also, see which cars are typical for states of either party. Lastly, we will also do a clustering analysis, to see if any clusters of the cars can be found and how they are defined.

B. Why did you choose this data set?

The data set was chosen as one of our authors has always wanted to analyze the data behind bilbasen.dk (the biggest Danish online car marketplace) in order to better be able to make decisions about which car to buy. So, when realizing that an equivalent data set of a very high quality was publicly available through Kaggle, it was an obvious choice.

There was, actually, a few candidate data sets containing used cars, just on Kaggle alone. This data set was chosen as it had the highest number of features (66) for each car, and by far the largest number of observations at 3 million.

C. How did you extract, store and manage the data?

The data set was extracted by downloading a compressed folder from Kaggle containing the data as a .csv file. The size of the uncompressed file with its 3 million rows and 66 columns is 9.7 GB, thereby making handling it somewhat challenging. It was possible to load the entire .csv file to a pandas dataframe as the size did not exceed the RAM of most modern computers, processing and querying on the dataframe was very slow due to the size, however.

This was managed in different ways depending on the task at hand. When visualizing the data in a non-aggregated way such as scatter plots, a random sample of a large enough size to remain representative could be used, other times only specific slices of the data would be visualized, or it would be aggregated making sampling unnecessary. Sampling was also used for developing the models in order to make them run successfully, to then finally run them with the entire data set.

D. General Descriptive Analysis

The dataset contains 3 million observations of listed cars from the website "Cargurus" and was scraped around september 2020. It has 67 features. The data is heavily dominated by newer cars as can be seen in fig. 1.

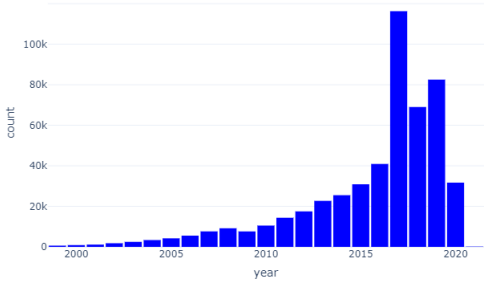

Fig. 1: Histogram showing distribution of the *year* feature, i.e. when the cars were produced.

II. Price Prediction (Anders Ketelsen)

A. Motivation

Having a model that can predict the prices of used cars has many uses. For example, it can be used on an online marketplace to suggest a price to users putting their car up for sale. It can also be used to guide buyers looking at a car for sale, telling them whether the price is above or below the predicted price level.

B. Visual Analysis

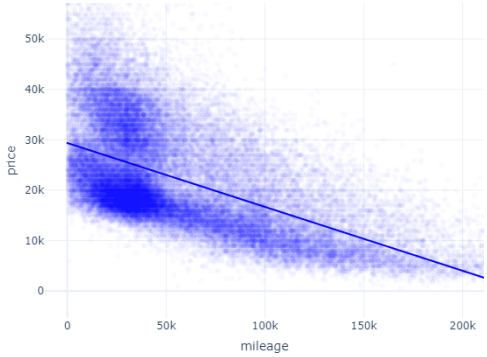

Fig. 2: Scatter plot of price and mileage with low opacity per point to show density of distribution, shown for all cars.

With its 66 variables we will focus this section on the features that have the biggest impact on the *price*. It is clear that a car's price is heavily dependent on what model it is, how far it has driven and how old it is. We'll start by examining the relationship between the *price* and *mileage* features.

As can be seen in the scatter plot shown in fig. 2 there is a clear negative correlation between the *mileage* and the *price*. Also shown by the linear regression line that achieves an $R^2 = 0.29$, meaning that 29% of the variance of *price* can be explained by the *mileage* feature. There is still a lot of variance left to be explained. One obvious explanatory variable is the model of the car given by the feature *model_name*. The car brand could also be used but the model is more specific and will presumably explain more variance. We will therefore create the same scatter plot as in fig. 2 but color it by *model_name*, to see if this feature can seperate the data, such that seperate linear regressions for each car model, better fits the data.

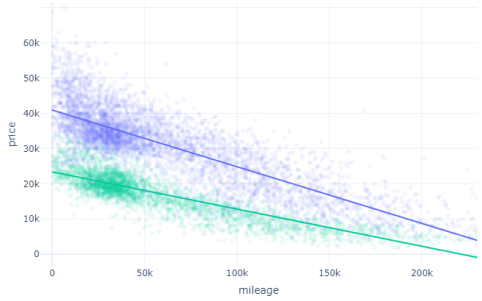

Fig. 3: Scatter plot of price and mileage colored by model name, here shown for Chevrolet Silverado 1500 (blue) and Toyota Camry (green).

As expected a lot of the variance observed around the regression line in fig. 2 can indeed be explained by the *model_name* as shown in fig. 3 where the linear regessions achieves R^2-values of 0.66 and 0.71 for the Silverado 1500 and the Camry, respectively. It's possible to seperate the models one level deeper with the variable *trim_name* that can be seen as a submodel of the models. Doing so for the Silverado 1500 model achieves the following result:

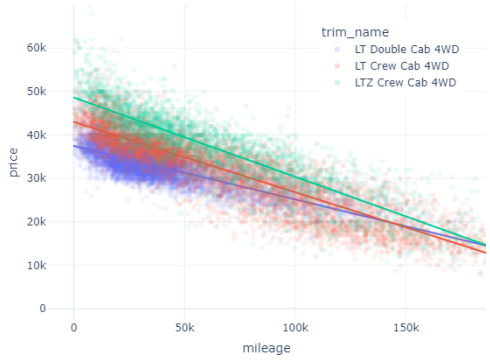

Fig. 4: Trim levels of Silverado 1500: Scatter plot of price and mileage colored by choice of trim, here shown for the three most popular trims of the Chevrolet Silverado 1500 model.

As can be seen from fig. 4 the trim levels succesfully seperate the model even further and thereby increase accuracy of the linear regressions further with the R^2-values from 0.66 when doing the linear regression on all of the trim levels of Silverado 1500, to a weighted (by count) average $R^2 = 0.70$ for the individual trim levels. The trim level is a good predictor as it holds information about a lot of variables. Whether it's 2WD or 4WD, the standard included options and in many cases also the engine variant. Information that can for the most part also be extracted from other variables, but are all summarized in one variable in the *trim_name*.

Another interesting observation is that in general, both when looking at the models and trim levels it can be seen that more expensive cars have a more negative correlation coefficient with mileage, meaning that they depreciate faster. It can also be seen that the relationship between mileage and price is not completely well described by the linear models as there seems to be a steeper price drop in the first miles than the last. And lastly it's evident almost no cars with more than 200k miles are sold, indicating the general lifespan of cars.

The visual analysis showed us, that *mileage* is a strong predictor of price, especially when creating seperate models for the different car variants.

C. Predictive Analysis

First a baseline performance will be established with a mean predictor on all cars:

	R^2	MAE
Mean Predictor	0.00	8352

TABLE I: Baseline accuracy from mean predictor.

The mean predictor achieves an expected $R^2 = 0.00$ as per defintion of the R^2-score. And a MAE of 8350 USD, which is of course a very poor performance.

1) Linear Regression: To build on the visual analysis we will begin the predictive analysis by seeing what can be accomplished by a simple univariate linear regression model. The ordinary least squares linear regression model works by minimizing the sum of squares of the residuals of the prediction which can be done analytically. The linear regression is done using *price* as the response variable and *mileage* as the predictor variable. We fit the model first to the entire dataset, then to a subset only containing the most popular model, the "Ford F-150" and then to the most popular trim level of the F-150 and lastly to specific engine version of that trim level. The performance is evaluated using 10-fold cross validation and the only data preparation is mean imputation on both *price* and *mileage* variables.

	R^2	MAE	MP MAE
All cars	0.29	7165	8352
F-150	0.56	5902	9700
F-150 (XLT SuperCrew 4WD)	0.77	2849	6392
F150 (XLT SuperCrew 4WD (375hp version))	0.36	2137	2740

TABLE II: Performance of univariate linear regression using *mileage* to predict *price*. "MP MAE" is the mean absolute error of mean prediction on the subset of the data.

From table II it can be seen by the decreasing MAE that, the predictions of the linear regression models improves the more specific the subset of cars becomes. The reason for the low $R2$-score of the most specific model is likely because there is much less variance left, so even though the model performs better the proportion of variance explained by the mileage is smaller.

One option is to create a linear regression model for each of the car models. But it's likely better to use a model that's non-linear and able to use learnings from each variable across the models, by creating a single model that predicts for all car models, using the car models as input features instead.

2) CatBoost: In this case we will use the CatBoost algorithm, since there are a lot of categorical variables that have many different possible values, that are not feasible to one-hot-encode as would be necessary with other gradient boosting algorithms such as XGBoost [1].

CatBoost is an ensemble method of the boosting type [4]. Ensemble meaning a method that adds the predictions of many (often weak) learners to form a strong learner, and boosting meaning that the learners are trained sequantially on the residuals of the predictions of the previous learners. Also, the observations are weighed in such a way that the observations that are difficult to predict for the previous learners get's a higher weight for the subsequent learners. The weak learners in this case are regression trees.

Since more variables will be used, it's necessary to first do some data preperation. This includes, removing columns with more than 20% nan-values, imputing missing values with the mean for continous variables and mode for categorical variables. Removing " -in" suffix from many spatial variables and then converting these to numerical data types. Normalization is not necessary when using CatBoost [2].

First, the model is trained on the variables used in the previous linear regression. That is: *mileage*, *model_name*, *trim_name* and *horsepower*. We will refer to this model as "CatBoost Minimal". Doing so achieves an MAE of 1798 for all cars, which is a significant improvement over the linear regression. This should be compared to the results achieved by the linear regression when trained on the smallest subset of data. Here the linear regression achieved an MAE = 2137 for the 375hp F-150 and when testing only on this subset CatBoost achieves MAE = 2025. So, a significant improvement, although it can be seen that catboost generally performs even better on other car models with the overall MAE of 1798.

When we include all variables CatBoost further improves to MAE=1118 when testing on the entire set of cars and 1391 for the 375hp F-150. We will refer to this model as "CatBoost allVars".

3) Including Major Options Variable in CatBoost: One variable that could be of great importance to the prediction is *major_options*. The variable contains a list of included options for the car. When loaded in the dataframe it is not stored as a list object, but a string literal, so until now CatBoost has simply treated it as a categorical feature, where each unique combination of options represents a category, which might not be optimal. The string literal is therefore converted to a list object, and then binarized using Sklearn's MultiLabelBinarizer:

major_options list	AC	ABS	Keyless
[AC, Keyless]	1	0	1
[]	0	0	0
[AC]	1	0	0
[ABS, Keyless]	0	1	1

TABLE III: Binarization of multilabel variable *major_options*

Adding the binarized major options, only resulted in a small improvement however, with an MAE=1106. This model will be referred to as "CatBoost allVars w/ major options".

4) Hyper Parameter Tuning: Grid search, using 3-fold CV on the train set, was performed by varying the number of iterations, the learning rate, the tree depth and the regularization of the leaf nodes. It was done on "CatBoost allVars", but not on "CatBoost allVars w/ major options" as this didn't add significant accuracy and had a severe impact on training speed, making it infeasable for grid search.

5) Interpreting the model: In this section we will use the SHAP library for feature interpretation and overall model explainability. In short, the Shapley values tells us how much the specific value of a variable attributed to the predictions deviation from the mean prediction for an observation. For more information about SHAP and shap values see [3].

First we will take a look at the the overall feature importance.

It's evident from fig. 5 that the same features we assumed to be of the greatest importance when dividing the dataset for the linear regression, are in fact the most important features. The only difference being that year is included, which is presumably very correlated with mileage, but also holds information in itself, as a car can for example be cheap even though it has a low mileage if it is very old. It can also be seen that the model makes use of the binarized options from the *major_options* variable, such as "Leather Seats".

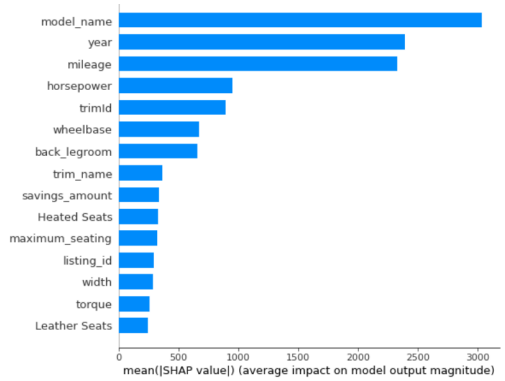

Fig. 5: Average SHAP value for the 15 most important features.

Using SHAP's dependency plot, it's possible to take a closer look at the effect of just one or two variables:

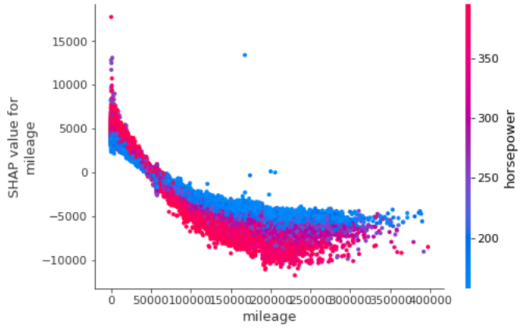

Fig. 6: SHAP values for the feature *mileage* colored by *horsepower* each dot is one observation's SHAP value addition from *mileage*.

In fig. 6 the very clear and expected negative impact of *milage* on the price prediction can be seen. The SHAP values are around 0 at the mean mileage of 58k miles. Higher horsepower cars are more positively impacted by a low mileage, than lower horsepower cars and vice versa for high mileage. This is probably because they are generally more expensive, and more expensive cars loose value faster when they add on mileage, at least in absolute terms.

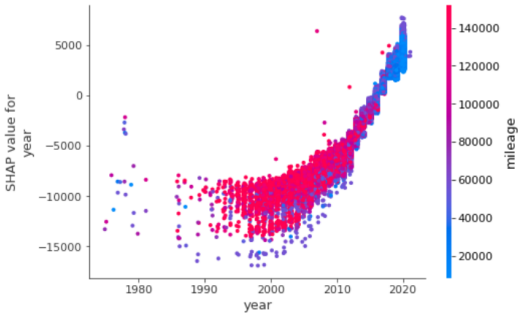

Fig. 7: SHAP values for the feature *mileage* colored by *horsepower* each dot is one observation's SHAP value addition from *mileage*.

Fig. 7 shows why both year and mileage have high feature importance despite their correlation, and also confirms the hypothesis put forth earlier, by showing the interdependency between the two features. The hypothesis was that old cars with low mileage, would be negatively impacted by the age. Which is also what can be observed here. Old cars have a more negative SHAP value if they have a lower mileage, as the other variables give them a higher prediction, so the age needs to drag down the prediction more.

6) Using description to predict residuals: We were able to use the *major_options* feature by binarizing it. But the dataset also has a text feature called *description* that holds the text of the listing. CatBoost is able to use text features, but only the classifier and not the regressor that we have so far been using. We will therefore make a classification model that use only the text feature *description* to predict the binarized residuals of the previous model's predictions. We do this by using the old regression model to predict prices for the training dataset and calculate the residuals of these predictions, which are then binarized according to whether they were over or under the actual price. This way we can use the classifier to predict whether the observation will have an over- or underprediction by the other model, based only on the description. Doing so results in an accuracy of 0.62, which is above the baseline 0.50 accuracy we would expect (the classes are balanced).

That means we should be able to slightly improve our model by adding the classifiers predicted label and using that to adjust our regression models score with a part of the mean over- and underpredictions of the regressor. Adding this correction factor from the classifier to the prediction of our regressor improves our MAE from 1105 to 1067, which is actually a significant improvement. It was tested on the original test set of the regression model, and the classifier was only trained on the train set of the regression model. The performance of the different CatBoost models is summarized in table IV.

Model	MAE
CatBoost Minimal	1798
CatBoost allVars	1121
CatBoost allVars w/ major options	1105
CatBoost allVars w/ major options + residuals classifier correction	1067

TABLE IV: MAE scores for the different CatBoost models.

III. Political Prediction from Car Market

To buy a car in many ways represents a standpoint in one's beliefs; electric or gasoline, family size, big or compact, luxurious or cost-efficient, comfort or performance? These parameters represent an ethical perspective; some of the same perspectives we think about when we vote in democracy. So as a group we wondered: **based on the used cars listed for a specific state, is it possible to predict which political party is going to win the elections?**

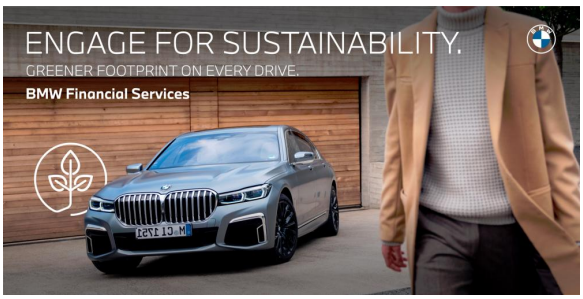

Fig. 8: BMW brand placement in regards to sustainability

For this part of our project, we used a historical data set from Keggle with the results for the US elections per state from 1976 to 2020, plus the used cars data set. To merge both data sets, first, we ran a zip code finder on the cars' data frame and added the state name into a new column for each input. We also used the cars' year model to create a relation to a specific election year. In this sense, we decided to do it one year before and two years after the election year. F.ex. for a car whose model is 2015, 2016, 2017, or 2018, we decided to classify it as 2016 election year. For the elections data set, we filtered the winner per state per year, and then we merged both data sets by finding inputs with the same state and election year. We added a new column to the cars' data frame with the winning political party for each entry.

transmission	transmission_display	trimId	trim_name	wheel_system	wheel_system_display	wheelbase	width	year	state	elec_year	city_fuel_economy	party
A	Automatic	t81782	SEL	FWD	Front-Wheel Drive	112.2	83.5	2019	TX	2020	23.0	1
A	6-Speed Automatic	t83968	SE AWD	AWD	All-Wheel Drive	105.1	72.8	2019	IN	2020	22.0	1
CVT	Continuously Variable Transmission	t45170	SV AWD	AWD	All-Wheel Drive	105.9	70.9	2013	VA	2012	22.0	2

Fig. 9: Improved data frame with 'election year' and 'winning party' classification (1=Republicans, 2=Democrats)

In addition, it was important to understand, which car features we could consider as part of political ethical concerns. We focused on the brand, body type, engine, seating capacity, fuel tank volume, city fuel economy, horsepower, and mileage. We made plots trying to find some relations between the data and political implications, among them: is the seating capacity related to conservative family values? Are pick-up trucks chosen more by Republicans and sedans by Democrats? Are cars with more mileage related to consumption habits? Who chooses powerful engines and who decides on economic fuel-consuming cars?

Plotting the data gave us some shallow indications that SUVs or pick-up trucks, Chevrolet and Ford, and more powerful engines are preferred by Republicans. Conversely, Toyota and Honda show a slight tendency to be preferred by Democrats, but no strong data relations were found, which suggested that our predictions were going to be close to a 50% chance.

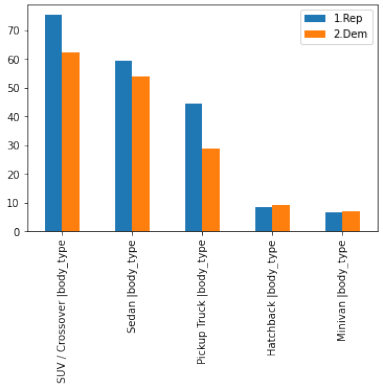

Fig. 10: Number of cars divided into body types for Republicans and Democrats

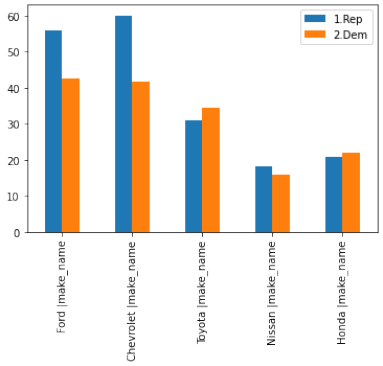

Fig. 11: Number of cars divided into brands/manufacturers for Republicans and Democrats

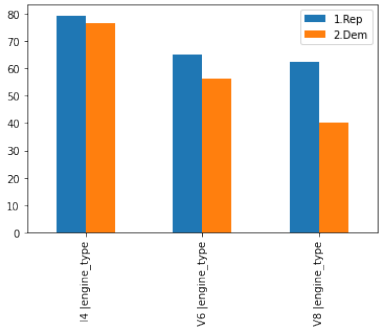

Fig. 12: Number of cars divided into engine types for Republicans and Democrats

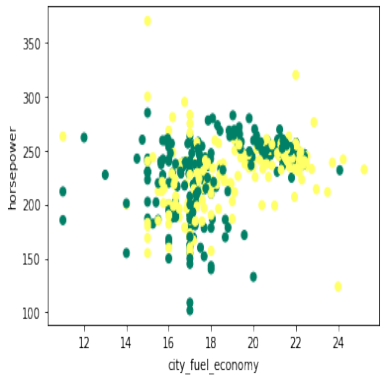

Fig. 13: No clear clustering between republicans or democrats regarding. On this scatter plott: city fuel economy and horsepower features

To train and test our models, we needed to create a new data frame structure, with likelihoods for the categorical data, and means for our numerical inputs, so we decided to calculate them per state and election year.

state	elec_year	party	SUV / Crossover |body_type	Sedan |body_type	Pickup Truck |body_type	Hatchback |body_type	Minivan |body_type	I4 |engine_type	V6 |engine_type	V8 |engine_type	Ford |make_name	Chevrolet |make_name
OR	2012	2	0.25	0.00	0.75	0.00	0.0	0.33	0.67	0.00	1.00	0.0
OR	2016	2	0.36	0.07	0.36	0.21	0.0	0.50	0.25	0.25	0.62	0.0

Fig. 14: data processed for training and testing models with likelihoods and means per election year and per state

We chose two methods to make our predictions: Naive Bayes and Perceptron. Naive Bayes considers all features as independent from each other and, predicts a classification by calculating the probabilities among the given features, into one class or the other. On the other hand, Perceptron takes a row of data as input and predicts a class label, which is achieved by calculating the weighted sum of the inputs and a bias.

We used both algorithms from Scikit Learn, and we got the following results:

Naive Bayes. Accuracy Score: 58.11%

Perceptron. Accuracy Score: 54.70%

For these results, we did not include the election year or state features. Giving this information to our model would improve our accuracy, but it would not mean that we can effectively predict who is going to win the elections based on a used cars database. What it actually means is that states do not change political parties so often -a state that is historically Republican will hardly choose a Democrat candidate.

state	party	%
AL	1	100.000000
AR	1	83.333333
	2	16.666667
AZ	1	77.777778
	2	22.222222
CA	2	72.727273
	1	27.272727
CO	2	62.500000
	1	37.500000

Fig. 15: percentage of winning parties per state (1=Republicans 2=Democrats)

To corroborate this assumption, we trained and tested our Naive Bayes model with data filtered by state (f.ex. only inputs for California), getting an **overall accuracy of 88.16%.**

Even though many political ethical perspectives can be linked to specific car features, the reason behind buying a specific car may be related to other factors, such as market-specific supply, purchasing power, loan options, geographical needs, terrain, types and cost of fuels available, among others.

We can conclude that **our political prediction model based on our used cars database did not help predict the winning party.**

IV. Clustering (Emmanuel)

On my perspective, this project has the aim of identifying the relationship between the manufacturing of cars in particular years and the elections held in those years. Mainly, this project will answer three questions: 1) which party wins the elections during the manufacturing of a particular car? 2) In percentage, which party is represented the most in a state to which a car belongs? 3) How shall we improve predictions in both cases defined above?

A. Preprocessing

Preprocessing is the way raw data is cleaned and structured. Raw data that contains some noise or missing values causes issues when mining, analyzing and modeling data so data must be to eliminate such noise. The process of preprocessing includes removing irrelevant attributes, removing missing values, calculating correlations for each dataset, and describing each dataset, etc.

B. Exploratory Data Analysis

Data Manipulation, Data Cleaning, Data Visualization, Dimensionality Reduction, Normalization, and Data Splitting among others were applied using Python's Pandas and Scikit-Learn libraries. The process of preprocessing the data is described below.

After loading the datasets, the first step on the political prediction code was to get the zip codes of each state using SearchEngine class of the uzipcode package and create a new column named 'state' containing those zip codes.

I dropped irrelevant columns from datasets using the drop function of Pandas library because irrelevant columns negatively impact the machine learning models. Then, I realized that there are columns in the dataset, which contain missing values, and machine-learning models do not accept missing or invalid values. Hence, I replaced invalid values of object columns with their mode and numeric columns with their mean.

C. Methods

I decided to try different techniques on the political prediction code, to get a deeper undestanding and corroborate if through other methods the accuracy can be improved. The methods that I choose are:

Naïve Bayes is the collection of classification algorithms that are derived from Bayes' theorem. It comes with different variations of the algorithm including GaussianNB (applied on continuous features), and CategoricalNB (applied on categorical data). The reason I chose CategoricalNB is that we have mostly categorical features and CategoricalNB provided better results than GaussianNB when compared.

Decision Tree is the type of supervised machine learning algorithm in which the model is trained on decision rules and predicts values according to those rules. It is applied in both regression and classification problems. Sometime applying Decision Trees does not provide promising results in cases where we have large datasets or features are a bit complicated for a single tree. There comes Random Forest Classifier which is also the supervised machine learning algorithm, categorized under ensemble machine learning algorithms, because it ensembles more than 1, even hundreds or thousands of trees; concept similar to a real forest. Random Forest is also applied in both regression and classification problems.

Also referred to as extreme gradient boosting, **Xgboost** is another ensemble learning method that is applied in regression and classification problems. The library is specially designed with a focus on high performance and computational speed.

Results list	F1-Score	Accuracy
[Naive Bayes]	0.56	0.56
[Decision Tree]	0.59	0.59
[Random Forest]	0.59	0.59
[Xgboost]	0.60	0.60

TABLE V: Political Prediction Results

Among the four techniques, **Xgboost provided the best results in terms of accuracy and f1-score.**

D. Principal Component Analysis (PCA)

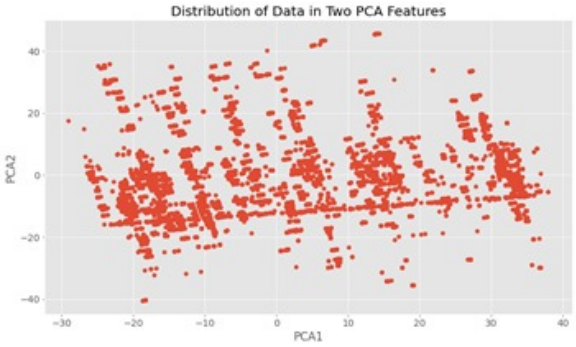

Fig. 16: Principal Component Analysis With 2 Components Showing Distribution of Data

E. K-Means Clustering

The reason why I used the clustering technique is to see which features are distributed across the number of clusters in a meaningful manner. Hence, it helped me to separate relevant features of data that I put back in classification models to get better prediction results.

F. Principal Component Analysis (PCA)

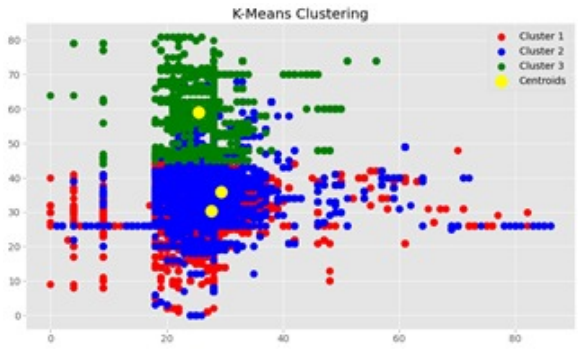

Fig. 17: Graph showing the data distributed across 3 clusters

Then, I colored the two components of PCA in which I reduced down overall data, using the values of clusters generated by K Means as shown in the figure below.

G. Principal Component Analysis (PCA)

Fig. 18: PCA components colored with the values of KMeans

H. CONCLUSION

I can personally conclude that thorugh my classification models, I can successfully predict which party wins the election for the year in which the car is manufactured. Finally, Principal Component Analysis (PCA) and K-Means Clustering techniques were applied to find insights and patterns from the data, hence relevant features were identified that were put in the model again which contributed in better results in predictions of both cases.

V. DIVISSION OF LABOR

Person	Part	Link
Anders	Price Prediction andIntroduction (Pages 1-4)	link
Miguel	Political Prediction (pages 5-6)	link
Emmanuel	Clustering (7-8)	link

TABLE VI: Division of labor (click link to see notebooks).

REFERENCES

[1] Catboost vs. light gbm vs. xgboost. ://towardsdatascience.com/catboost-vs-light-gbm-vs-xgboost-5f93620723db: :text=Unlike

[2] Feature normalization when using catboost. https://datascience.stackexchange.com/questions/16225/would-you-recommend-feature-normalization-when-using-boosting-trees.

[3] Shap values explained exactly how you wished someone explained to you. https://towardsdatascience.com/shap-explained-the-way-i-wish-someone-explained-it-to-me-ab81cc69ef30.

[4] Vasily Ershov Anna Veronika Dorogush and Andrey Gulin. Catboost: gradient boosting with categorical features support.