Assignment 1: A Mathematical essay on Linear Regression

Vishal Rishi MK
Department of Chemical Engineering
IIT Madras
ch18b013@smail.iitm.ac.in

***Abstract*—Linear regression is a statistical procedure for calculating the value of a dependent variable from an independent variable. Linear regression measures the association between two variables. It is a modeling technique where a dependent variable is predicted based on one or more independent variables. Linear regression analysis is the most widely used of all statistical techniques. Linear regression can not only be viewed as an estimation technique. It also helps us to understand the strength of relationships between variables and their statistical significance (causal analysis). In this article, we explore the relationships between socioeconomic status and cancer mortality (and incidence) rates in several states in the USA using regression analysis.**

Keywords—linear regression, significance, estimation

I. Introduction

The concept of linear regression was first proposed by Sir Francis Galton in 1894. Linear regression is a statistical test applied to a data set to define and quantify the relation between the considered variables. Univariate statistical tests such as Chi-square, Fisher's exact test, *t*-test, and analysis of variance (ANOVA) do not allow taking into account the effect of other covariates/confounders during analyses. However, linear regression allows the researcher to control the effect of confounders in the understanding of the relation between two variables. Linear regression assumes a linear relationship (in parameters) between the independent variables and the dependent variable. It also assumes a random, unpredictable, zero mean error term that is being added to the dependent variable. Using the least squares technique, we estimate the coefficients of the independent variables (including the intercept). During the process of estimation, we make several assumptions on the data and the residuals. Residuals are the difference between the observed values and the predicted values of the dependent variable. If some of the assumptions are broken, our estimates of the coefficients might not be reliable. Hence, it is important to assess the significance of our model and estimates through various testing methods before we use the model to derive insights and conclusions about the data. Otherwise, we would be arriving at spurious correlations between variables.

In this article, we explore relationships between cancer incidence (and mortality) and socioeconomic status in several states in the USA. We have cancer incidence and mortality data for 51 states in the USA along with socioeconomic data for several counties in those states. The features that describe socioeconomic status of a county are 1) Median income 2) Number of people below the poverty line (male and female) 3) Median income of native Americans, Black, White, Asian and Hispanic. Cancer mortality (or incidence) is best described by the Incidence and Mortality rates in a county. Incidence rate of a disease is the number of new cases per 100,000 people at risk. Mortality rate of a disease is the number of new deaths per 100,000 people at risk. Along with this, we also have the average annual incidence and average annual deaths due to cancer in each county.

Section 2 of this article gives a mathematical background to linear regression and the statistical assumptions behind the estimation of the parameters. Section 3 of the article describes the data processing pipeline and several observations, insights, visualizations. We also provide quantitative results after applying regression analysis. Finally, section 4 draws conclusions from our explorations and analysis.

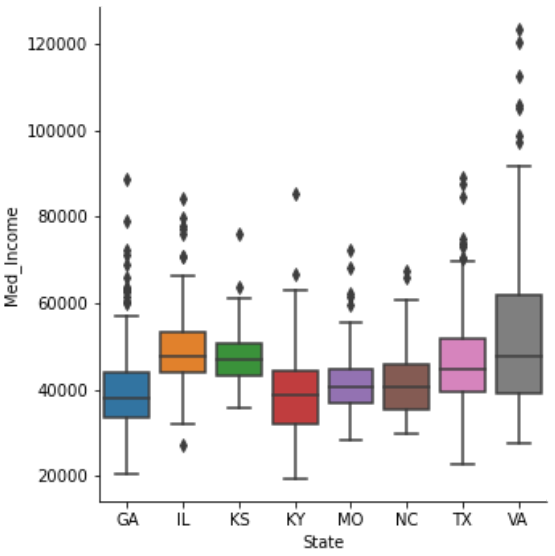

Figure 1: Boxplot distribution of Med_Income for the top eight states

II. Linear Regression

In this section, we develop the linear regression methodology for building a model of the relation between two or more variables of interest on the basis of available data. An interesting feature of this methodology is that it may be explained and developed simply as a least squares approximation procedure, without any probabilistic assumptions. Yet, the linear regression formulas may also be interpreted in the context of various probabilistic frameworks, which provide perspective and a mechanism for quantitative analysis. Suppose that our data consist of triples of the form (***xi*** , ***yi***, ***zi***) and that we wish to estimate the parameters ***θj*** of a model of the form

$$y \approx \theta_0 + \theta_1 x + \theta_2 z$$

We then seek to minimize the sum of the squared residuals

$$\sum_{i=1}^{n} (y_i - \theta_0 - \theta_1 x_i - \theta_2 z_i)^2 \quad \text{---------- (1)}$$

over ***θj, j = 0, 1, 2.***

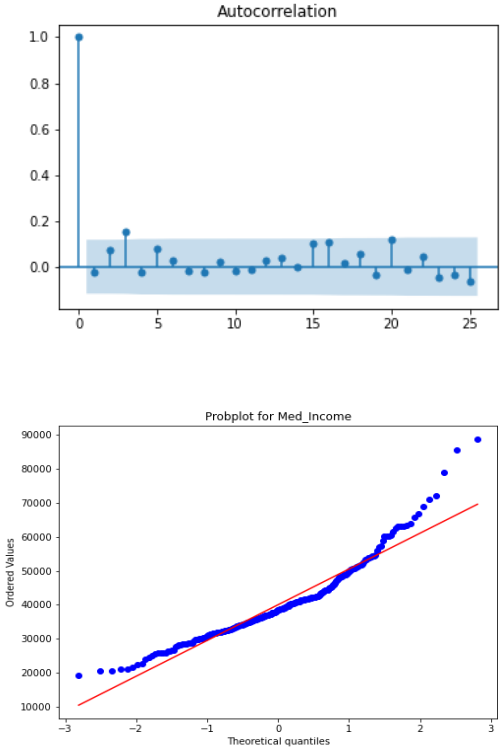

Figure 3: Autocorrelation plot (top) and the normal probability plot (bottom) of Med_Income for the states GA and KY

A. Approximate Bayesian LMS estimation (linear model)

Let the pairs ***(Xi, Yi, Zi)*** be random and i.i.d (identical and independently distributed random variables). Let us also make the additional assumption that the pairs satisfy a linear model of the form

$$Y_i = \theta_0 + \theta_1 X_i + \theta_2 Z_i + W_i$$

where the ***Wi*** is i.i.d. , zero-mean noise terms, independent of ***Xi, Zi***. From the least mean squares property of conditional expectations, we know that **E[*Yi*|*Xi,Zi*]** minimizes the mean squared estimation error. Under our assumptions,

$$E[Y_i | X_i, Z_i] = \theta_0 + \theta_1 X_i + \theta_2 Z_i$$

Thus, the true parameters minimize

$$E[(Y_i - \theta_0 - \theta_1 X_i - \theta_2 Z_i)^2]$$

By the weak law of large numbers, this expression is the limit as **n**→∞ of

$$\frac{1}{n} \sum_{i=1}^{n} (Y_i - \theta_0 - \theta_1 X_i - \theta_2 Z_i)^2 \quad \text{---------- (2)}$$

This indicates that we will obtain a good approximation of the minimizers of $E[(Y_i - \theta_0 - \theta_1 X_i - \theta_2 Z_i)^2]$ (the true parameters), by minimizing the above expression (2) (with ***Xi, Yi*** and ***Zi*** replaced by their observed values ***xi***, ***yi***, and ***zi*** respectively). But minimizing this expression is the same as minimizing the sum of the squared residuals given in (1).

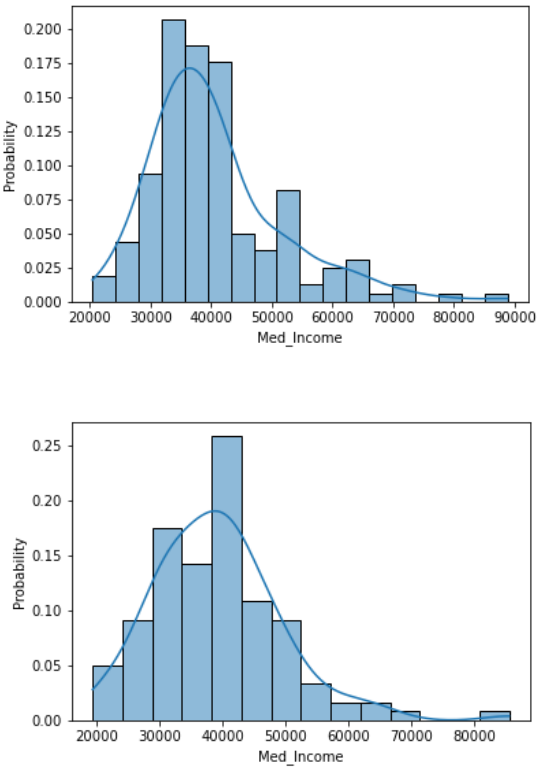

Figure 2: Histogram of Med_Income for the state GA (Top), and histogram of Med_Income for the state KY (Bottom)

III. APPLICATION TO THE PROBLEM

The data that we have contains data records for 3134 counties in the USA. The counties belong to one of the 51 states. We have 23 features (categorical and numerical) that describe the socioeconomic status and cancer incidence (and mortality) of the counties.

A. Data Preparation

For linear regression, one of the main assumptions is that the variables are independent and identically distributed. Hence, we sort all 51 states in descending order based on the number of counties in a particular state. The top eight states having the most number of counties (and hence, the most number of data points) are shown in Table 1 . Figure 1 shows the boxplot distribution of Med_Income (refer to Table 1) for the top eight states. From Figure 1, we can see that the states GA and KY have a similar distribution of Med_Income. Figure 2 shows the histogram of Med_Income for the states GA and KY. From this, we can say qualitatively that Med_Income is identically distributed in the states GA and KY. We also need to assure the independence of the observations. Though independence is a strong statement and is hard to prove, we can fairly assume that there is no autocorrelation between the observations. Figure 3 shows the autocorrelation plot and the normal probability plot of Med_Income for the states GA and KY. They show that the assumption of independence is justifiable. We group the data records for

the counties in the states GA and KY in a separate dataset and analyze this further.

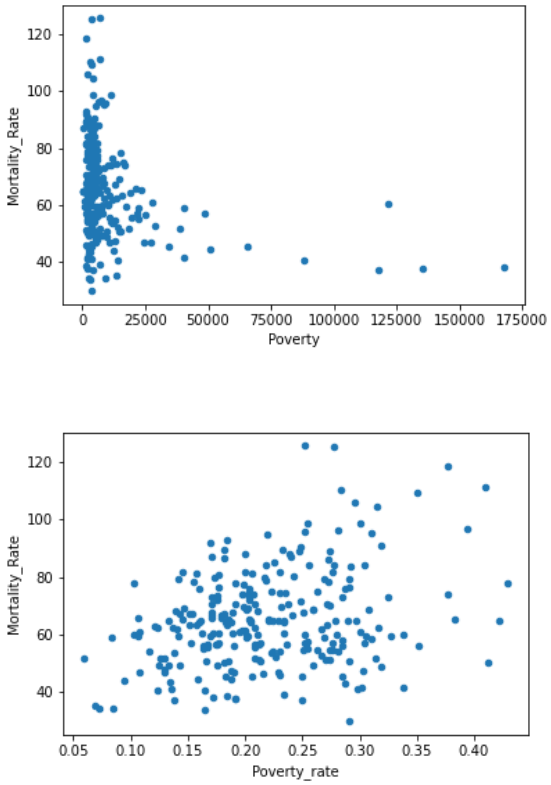

Figure 4: Effect of All_Poverty and Poverty_rate on Mortality rate

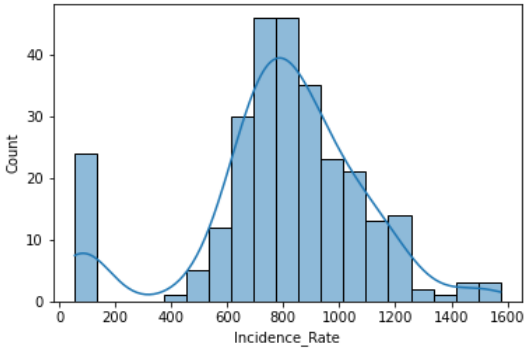

Figure 6: Distribution of Incidence_Rate

B. Feature selection and engineering

Along with the median income of the county, we also have the median income of several ethnicities (eg. Hispanic, White, Black, Asian, etc). But without the actual composition of the ethnic groups in the entire population, they would not create meaningful correlations with the incidence or mortality rates (which does not take ethnicity into account). The same argument goes with gender. Hence, we can discard variables that indicate ethnicity or gender. Figure 4 shows a scatter plot between All_Poverty and Mortality (and Incidence rates). There are no obvious patterns or trends visible. On the other hand, there is an increasing trend when we plot Mortality (and Incidence rates) against All_Poverty_rate. All_Poverty_rate is the ratio of the number of people below the poverty line in a county. This shows that rates are better features than absolute counts. Hence, we also introduce another variable All_Without_rate, which is the ratio of the number of people without health insurance. Our dataset contains 279 data points and we do encounter some missing values. The missing entity is replaced by the mean of the corresponding variable. Missing values are encountered only in Incidence_Rate and Mortality_Rate. The main features that we worked with are as follows 1) Poverty_rate, 2) All_Without_rate, 3) Med_Income, 4) Incidence_Rate, and 5) Mortality_Rate.

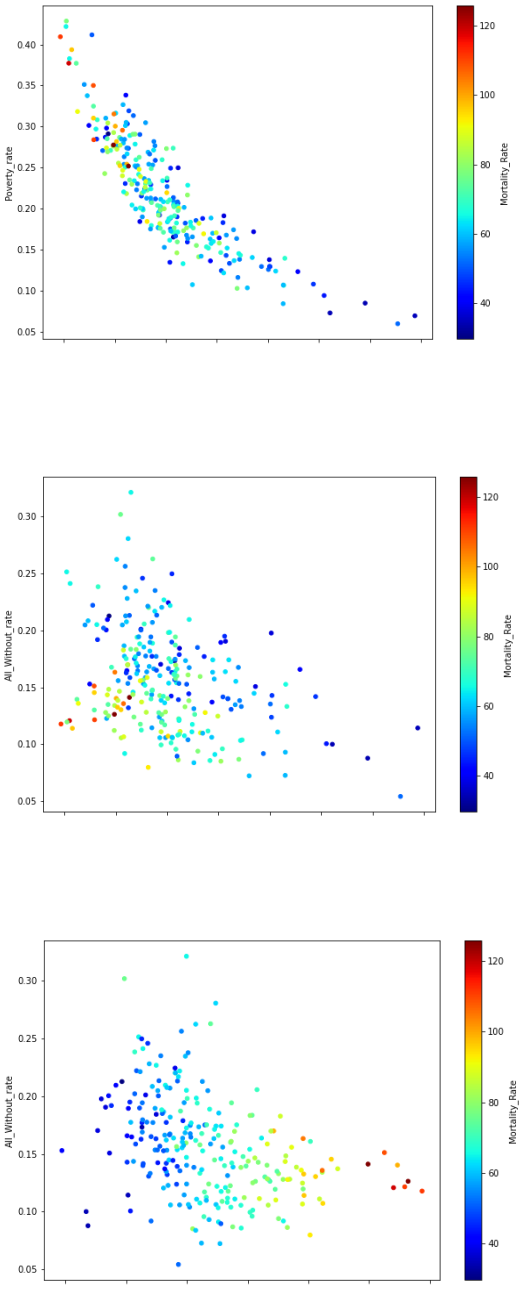

Figure 5: a) Poverty_rate vs Med_Income (top), b) All_Without_rate vs Med_Income (middle), c) All_Without_rate vs Incidence_Rate (bottom)

C. Exploratory Data Analysis

Figure 5 shows scatter plots between several combinations of the filtered variables. An important point to note is that the features are highly correlated. Poverty_rate and Med_Income have a very correlation of -0.87. Including both these variables in our regression can lead to unstable coefficients and unreliable results due to multicollinearity.

In Figure 5, the scatter plot between Med_Income and Poverty_rate shows qualitatively that Med_Income is a good indicator of Mortality and Incidence rates.

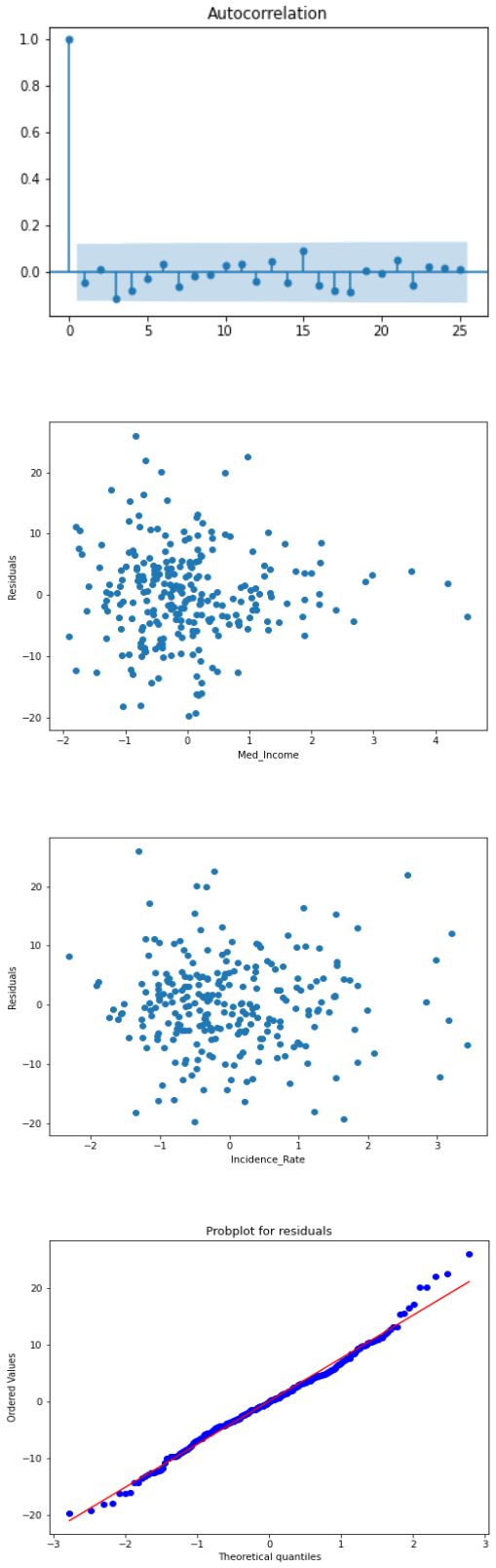

Figure 7: Autocorrelation between residuals (top), scatter plots between explanatory variables and residuals (second and third), and normal probability plot of residuals (bottom)

Counties with a higher value of Med_Income have higher mortality rates and those with a lower value of Med_Income have lower mortality rates. An interesting thing to note is that the correlation between All_Without_rate and Mortality_Rate is -0.25. This might seem counterintuitive. This might be due to the effect of unaccounted confounding variables. In Figure 5, in the scatter plot between Med_Income and All_Without_rate, for a fixed value of Med_Income, we still observe that the value of Mortality_Rate decreases as All_Without_rate increases. This interesting phenomenon might be due to specific local effects present only in these selected counties. Also in the scatter plot between Incidence_Rate and All_Without_rate, we cannot see any relationship between All_Without_rate and Mortality_Rate, for a fixed value of Incidence_Rate. This indicates that Incidence_Rate could well be one of those confounding variables. We should make sure to include them in our model. Otherwise, we would get biased coefficients leading to spurious correlations. Figure 6 shows the distribution of Incidence_Rate. We can see a considerable number of outliers in the left tail of the distribution. If these outliers are not removed, they might lead to residuals that are not normally distributed. Hence, these outliers are removed, reducing our data points to 255.

D. Regression analysis - Mortality rate

We tried several regression models with different feature combinations. Table 2 shows the results for all the regression models. The model that gave reliable results, in terms of accuracy and statistical significance is shown at the bottom of the table. This model used Med_Income, All_Without_rate, and Incidence_Rate as the features and Mortality_Rate as the target variable. It achieved an R-squared value of 0.798. This means that model was able to explain 79.8% of the variance in Mortality_Rate. The statistical significance of the measure is given by the F - score, which is, in our case, equal to 329.9 (a p-value ≈ 0). A higher F - score leads to rejecting the null hypothesis that the coefficients of the model are insignificant. The statistical significance of each of the coefficients is given in Table 2. We can see that the coefficient of All_Without_rate is insignificant (p-value = 0.198). This can be qualitatively verified by the scatter plot between Incidence_Rate and All_Without_rate in Figure 5. We cannot see any relationship between All_Without_rate and Mortality_Rate, for a fixed value of Incidence_Rate. This can be quantitatively verified by looking at the model that uses only Med_Income and Incidence_Rate to model Mortality_Rate (refer to Table 2). Though this model does not use the variable All_Without_rate, its R - squared value does not drop significantly from the best model. This shows that the variable All_Without_rate does not add significant explanatory power to the model. The coefficient of the variable Med_Income in our best model is -1.61. The p-value suggests that it is statistically significant (p-value = 0.007). This suggests that the median income of a county is inversely related to the cancer mortality rate (as seen in Figure 5). The variable Poverty_rate is not included in our final model because of multicollinearity. Table 2 shows such a model with all the variables included. Since Poverty_rate is highly correlated with Med_Income, we can see that the p - values of these variables have been distorted. This is because of the high standard error of the coefficients. To avoid such unreliable estimates, we did not include Poverty_rate in the final model. The reason for including Incidence_Rate in the final model can be easily understood by looking at the model that includes Med_Income and All_Without_rate as the features (refer to Table 2). The

coefficient of All_Without_rate in this model is -7.5610 (p-value = 0). This is clearly a biased estimate and indicates a spurious relationship. This is due to omitted variable bias. The omitted variable, in this case, is the Incidence_Rate. Though we need to model the relationship between socioeconomic status and cancer mortality (or incidence), we need to include Incidence_Rate in our model so that we do not model incorrect relationships between variables. Once we include it in the final model, we can see that there is no clear relationship between All_Without_rate and Mortality_Rate.

Figure 7 shows the autocorrelation plots of residuals, plots to check no endogeneity, and plots to check the normality of the residuals. The normal probability plot shows that the errors are normally distributed. From the autocorrelation plot of the residuals, we can see that the residuals are linearly independent of each other. Also, the scatter plot between the residuals and each of the explanatory variables shows a random scatter. It indicates that the explanatory variables are not able to further explain the residuals (no endogeneity). This also assures us that we are not leaving any confounding variables in our model.

E. Effect of outliers

The presence of outliers can severely affect the results in our model. The results in Table 2 are conducted with the outliers in Incidence_Rate removed from the data. Table 3 shows the results obtained when these outliers are not removed. Once again, we can see the biased estimate for the coefficient of All_Without_rate, even when the confounding variable Incidence_Rate is included. The coefficient of All_Without_rate is also statistically significant (p-value = 0). These are wrong results. This clearly depicts the importance of removing outliers from our data.

F. The Problem of Heteroscedasticity

The problem of heteroscedasticity arises when the population error does not have constant variance. This leads to inefficient estimates of the coefficients. When the population errors are homoscedastic, OLS estimators (Ordinary Least Squares) for the regression coefficients are found to be the Best Linear Unbiased Estimators (BLUE). To test for heteroskedasticity, we perform the White's test on the residuals obtained from the regression of cancer mortality rate on Med_Income, All_Without_Rate, and Incidence_Rate. The test yields a p-value of 0.0275 which results in the rejection of the null hypothesis (Population errors are homoscedastic) at 5% significance. Hence the estimates for the regression coefficients are inefficient. To overcome this problem of heteroscedasticity, we perform Weighted Least Squares (WLS). This method scales the residuals using weights that are inversely proportional to their variance. For our problem, we use the squared reciprocal of the residuals obtained from the regression of cancer mortality rate on Med_Income, All_Without_Rate, and Incidence_Rate as the weights. Table 4 shows the results obtained from the WLS regression. When we test the residuals obtained from the WLS regression for heteroscedasticity, we obtain a p-value of 0.0264 which results in the rejection of the null hypothesis (Population errors are homoscedastic) at 5% significance. This shows that we have not estimated the population error covariance matrix correctly. Efficient estimation of the error covariance matrix is essential to overcome heteroscedasticity.

IV. Conclusion

This article illustrates the use of linear regression on a practical dataset - cancer mortality. The mathematical background for linear regression is explained. We further employ linear regression to explore relationships between cancer mortality and socioeconomic data.

1) The results show that in the states GA and KY, there is strong evidence that the median income has a strong influence on cancer mortality.
2) Also, the ratio of the number of people without health insurance does not affect the mortality rate in the states GA and KY.
3) The effect of outliers on the results is explained by including the outliers in our analysis and computing the estimates. We observed that outliers in the data can lead to spurious correlations between the variables.
4) Whits's test has revealed the presence of heteroscedasticity in the population errors. This calls for the efficient estimation of the error covariance matrix.

There is room for improvement. We can extend our analysis to other states as well. We might be able to find different patterns in different states. Similar to modeling mortality rate, we can also try to model incidence rate. But mortality rate cannot be used as an explanatory variable because mortality does not cause incidence. The racial and gender composition of the population can be included to study the socio-economic effects on cancer mortality within an ethnic or gender group. We can also include several other factors like medical infrastructure, public awareness, food insecurity, lifestyle habits like smoking, obesity, etc. These factors might be able to explain cancer incidence which in turn helps us to understand cancer mortality.

States	No. of Counties
TX	254
GA	159
VA	132
KY	120
MO	115
KS	105
IL	102
NC	100

Table 1: The top eight states having the most number of counties (and hence, the most number of data points)

References

[1] D. Bertsekas, and J. Tsitsiklis, Introduction to Probability, 2nd ed., Massachusetts Institute of Technology, 2008, pp.475-483

[2] D. Montgomery, C. Jennings, and M. Kulahci, Introduction to Time Series Analysis and Forecasting, Wiley, 2008, pp.73-138

[3] J. O'Connor, T. Sedghi, M. Dhodapkar, M. Kane, and C. Gross, "Factors associated with cancer disparities among low-, medium-, and high-income US counties ", 2018

[4] J. Kim, "Multicollinearity and misleading statistical results", 2019

[5] G. Wilkes et al., "Cancer and poverty: breaking the cycle", 1994

	coef	p val	std err	F Val	R squared
cons	65.1	0	0.47	246.5	0.798
x0	-0.1	0.96	0.957	246.5	0.798
x1	-1.6	0.008	0.966	246.5	0.798
x2	0.81	0.198	0.63	246.5	0.798
x3	14.6	0	0.60	246.5	0.798

	coef	p val	std err	F Val	R squared
cons	65.1	0	0.87	59.32	0.32
x1	-9.1	0.0	0.936	59.32	0.32
x2	-7.5	0.0	0.936	59.32	0.32

	coef	p val	std err	F Val	R squared
cons	65.1	0	0.47	492.8	0.796
x1	-2.1	0.00	0.51	492.8	0.796
x3	14.2	0.0	0.51	492.8	0.796

	coef	p val	std err	F Val	R squared
cons	65.1	0	0.47	329.9	0.798
x1	-1.6	0.007	0.59	329.9	0.798
x2	0.79	0.198	0.61	329.9	0.798
x3	14.6	0	0.60	329.9	0.798

Table 2: Regression results for all linear models are shown above. The final model is shown in the last table
(x0 : Poverty_rate, x1 : Med_Income,
x2 : All_Without_rate, x3 : Incidence_Rate)

	coef	p val	std err	F Val	R squared
cons	64.9	0	0.72	83.9	0.478
x1	-6.6	0	0.83	83.9	0.478
x2	-4.9	0	0.84	83.9	0.478
x3	7.07	0	0.79	83.9	0.478

Table 3: Regression results for the linear model built with outliers in the data (x1: Med_Income,
x2 : All_Without_rate, x3 : Incidence_Rate)

	coef	p val	std err	F Val	R squared
cons	65.1	0	0.057	14970	0.994
x1	-1.7	0	0.079	14970	0.994
x2	0.68	0	0.084	14970	0.994
x3	14.6	0	0.082	14970	0.994

Table 4: Regression results for the linear model estimated using WLS (x1 : Med_Income, x2 : All_Without_rate,
x3 : Incidence_Rate)

Assignment 2 : A Mathematical essay on Logistic Regression

Vishal Rishi MK
Department of Chemical Engineering
IIT Madras
ch18b013@smail.iitm.ac.in

***Abstract*—In statistics, the logistic model (or logit model) is used to model the probability of a certain class or event existing such as pass/fail, win/lose, alive/dead or healthy/sick. This can be extended to model several classes of events such as determining whether an image contains a cat, dog, lion, etc. Each object being detected in the image would be assigned a probability between 0 and 1, with a sum of one. Logistic regression is a statistical model that in its basic form uses a logistic function to model a binary dependent variable. In regression analysis, logistic regression (or logit regression) is estimating the parameters of a logistic model (a form of binary regression). The sinking of the Titanic is one of the most infamous shipwrecks in history. On April 15, 1912, during her maiden voyage, the widely considered "unsinkable" RMS Titanic sank after colliding with an iceberg. Unfortunately, there weren't enough lifeboats for everyone onboard, resulting in the death of 1502 out of 2224 passengers and crew. While there was some element of luck involved in surviving, it seems some groups of people were more likely to survive than others. In this article, we use logistic regression to build a predictive model that helps us to determine the groups of people that were more likely to have survived the accident.**

Keywords—logistic regression, logits

I. Introduction

In many situations, the variables are qualitative. For example, eye color is qualitative, taking qualitative values blue, brown, or green. Often qualitative variables are referred to as categorical. Approaches for predicting qualitative responses is known as classification. Predicting a qualitative response for an observation can be referred to as classifying that observation, since it involves assigning the observation to a category, or class. On the other hand, often the methods used for classification first predict the probability of each of the categories of a qualitative variable, as the basis for making the classification. In this sense they also behave like regression methods. There are many possible classification techniques, or classifiers, that one might use to predict a qualitative response. In this article, we discuss the logistic regression approach for classification. Just as in the regression setting, in the classification setting we have a set of training observations ***(x1, y1),...,(xn, yn)*** that we can use to build a classifier. We want our classifier to perform well not only on the training data, but also on test observations that were not used to train the classifier.

Mathematically, a binary logistic model has a dependent variable with two possible values, where the two values are labeled "0" and "1". In the logistic model, the log-odds (the logarithm of the odds) for the value labeled "1" is a linear combination of one or more independent variables or predictors. The function that converts log-odds to probability is the logistic function. The defining characteristic of the logistic model is that increasing one of the independent variables multiplicatively scales the odds of the given outcome at a constant rate, with each independent variable having its own parameter. Outputs with more than two values are modeled by multinomial logistic regression. The logistic regression model itself simply models probability of output in terms of input and does not perform statistical classification.

We try to understand the factors that determined the survival status of passengers in the Titanic, when it sank on April 15, 1992. We employ logistic regression for our analysis. This is an interesting issue in itself as the probability of survival differs greatly between individuals. For example, according to our analysis, men traveling first class were much more likely to survive than men in second and third class, and nearly all women traveling in first class survived compared to women traveling in the other two classes. Yet, the Titanic disaster is also relevant in a more general context. It allows us to analyze behavior under extraordinary conditions, namely, in a life and death situation.

Section 2 of this article gives a mathematical background to logistic regression and the estimation of the parameters. Section 3 of the article describes the data processing pipeline and several observations, insights, visualizations. We also provide quantitative results after applying logistic regression analysis. Finally, section 4 draws conclusions from our explorations and analysis.

II. Logistic Regression

Consider a model with two predictors, ***x1*** and ***x2***, , and one binary (Bernoulli) response variable ***Y***, which we denote ***p = P(Y = 1 | x1, x2).*** We assume a linear relationship between the predictor variables and the log-odds (also called logit) of the event that ***Y = 1,*** given ***x1 and x2***. This linear relationship can be written in the following mathematical form

$$l = log(\frac{p}{1-p}) = \beta_0 + \beta_1 x_1 + \beta_2 x_2$$

where ***l*** is the log-odds and ***βi, i = 0, 1, 2*** are parameters of the model. By simple algebraic manipulation, the probability that ***Y = 1*** is

$$p = \sigma(\beta_0 + \beta_1 x_1 + \beta_2 x_2) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x_1 + \beta_2 x_2)}}$$

where ***σ(.)*** is the sigmoid function. The above formula shows that once ***βi*** are fixed, we can easily compute the log-odds that ***Y = 1,*** given ***x1 and x2*** for a given observation. The coefficients ***βi*** are unknown, and must be estimated based on the available training data. The more

general method of maximum likelihood is used, since it has better statistical properties. The basic intuition behind using maximum likelihood to fit a logistic regression model is as follows: we seek estimates for $\boldsymbol{\beta i}$ such that the likelihood of observing the data is maximized. Hence, the likelihood function becomes the objective function. Optimization algorithms like gradient descent are employed to find the optimal parameters. In some instances, the model may not reach convergence. Non-convergence of a model indicates that the coefficients are not meaningful because the iterative process was unable to find appropriate solutions. A failure to converge may occur for a number of reasons: having a large ratio of predictors to cases, multicollinearity, sparseness, or complete separation.

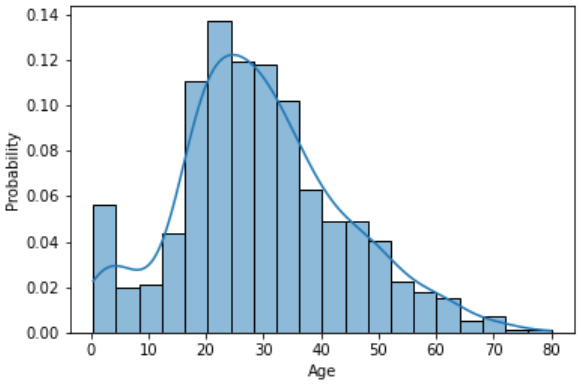

Figure 1: histogram of the feature Age. We can see some outliers in the left tail of the distribution

III. Application to the problem

The data that we have contains data records for 891 passengers who sailed on the Titanic. We have 11 features (categorical and numerical) that describe the survival status of the passengers. We have a target feature that indicates the survival status of each passenger. The 11 features are 1) Passenger Id, 2) Passenger class, 3) Name, 4) Sex, 5) Age, 6) Number of siblings and spouses, 7) Number of parents and children, 8) Ticket Id, 9) Fare, 10) Cabin Id, 11) Port of embarkation. The features Age, Cabin Id, and Port of Embarkation contain missing values. Specifically, feature Age has 177 missing values and feature Cabin Id has 687 missing values. These are too large to impute those data records with the mean, median, or mode of the particular feature. If we do so, we might miss some of the interesting patterns. Moreover, not all the features might be relevant in predicting the probability of survival. Clearly, the factors that could have played a significant role in deciding the survival status of a passenger at the time of sinking are Age, Sex, Passenger class (and hence Ticket fare), Cabin Id, number of siblings, spouses, parents, and children. The other features like Name, Passenger Id, and Ticket Id could not have played a role in deciding the survival status. Hence, we can conveniently discard them from our analysis.

A. Exploratory Data Analysis

Though we have data records for 891 passengers, we only have 681 unique Ticket IDs. This could be due to the presence of family tickets, which share a common ticket id for all the members of the family. We define a family ticket as one which is shared by at least two passengers. From the data, we can see that there are 131 family ticket ids. Also, the passengers sharing a family ticket id have the same port of embarkation. Hence, we can easily assume that there were at least 131 families on the ship. Figure 1 shows the histogram of the feature Age. We can see some outliers in the left tail of the distribution. We need to see the effect of these outliers on the logistic regression model. Also, Figure 2 shows the histogram of the feature Fare. This is highly skewed and is not normally distributed. We might need to do some transformations to remove the skewness. Fortunately, the Box-Cox transformation of this feature is shown in Figure 2. Though it shows multimodal nature, it has removed the skewness. Also, the features Fare and Passenger class are highly correlated. All the 3rd class ticket fares are less than 70. For passengers who have bought tickets with fares greater than 70, 100 out of the 105 passengers belong to the first class. Another interesting fact to note is that 15 passengers have traveled with zero fares (First class: 5, Second class: 6, Third class: 4). These could be erroneous values. Instead of working with the feature Fare, we can use the feature Passenger class, which neither has erroneous values nor skewness. Figure 3 shows the bar plots between Survival status and the features Sex and Passenger class. We can see that the features Sex and Passenger class individually played an important role in determining the survival status. But if we were to use both the features together in our logistic regression model, we need to check their combined significance. Figure 4 shows one such barplot. We can easily infer some points. Within a particular passenger class, females were given priority in an emergency situation. This results in a high survival rate. Compared to females, the survival rate of males is significantly lower. Also, for a given gender, the survival rate is the highest for passenger class 1. This proves the presence of economic bias at the time of the sinking of the ship. We also need to analyze the combined effect of age, sex, and passenger class on the survival status. Since age is a continuous variable, we discretize it into three bins - less than 15 years, greater than 48 years, and between 15 and 48 years. Figure 5 shows the bar plots showing the combined effect of sex and passenger class on survival rates for different age groups. We can see that the passengers less than 15 years of age have the highest survival rate and the passengers greater than 48 years of age have the least survival rate. Interestingly, though the overall survival rate for males is low, the survival rate of males less than 15 years of age (male children) belonging to passenger classes 1 and 2 strikes 100 percent. Also, none of the male passengers greater than 48 years of age belonging to passenger class 3 have survived. These facts show the strong effect of the feature Age on survival rate, combined with the features Sex and Passenger class. These features are a good starting point to use in our logistic regression model.

B. Data Preparation

We have used only three features in our initial model - Age (continuous variable), Sex, and Passenger class. Since the feature Age has 177 missing values, we have dropped those data records with missing age values. Hence, the size of our dataset is reduced to 714. We have now created two additional variants of this dataset. The first variant discretizes the feature Age into four groups - less than 15 years, between 15 and 30 years, between 30 and 48 years, greater than 48 years. The second variant removes the outliers in the feature Age (shown in Figure 1). Hence, the second variant has 674 data points. In all the variants of the

dataset, we have used ordinal encoding for the feature Passenger class and one-hot encoding for the feature Sex. There is no standardization applied to the feature Age, as it did not change the results drastically. In all the datasets, we have set aside 100 randomly selected data points as the test set. We have also made sure that the class representation in the train and test sets remain approximately the same.

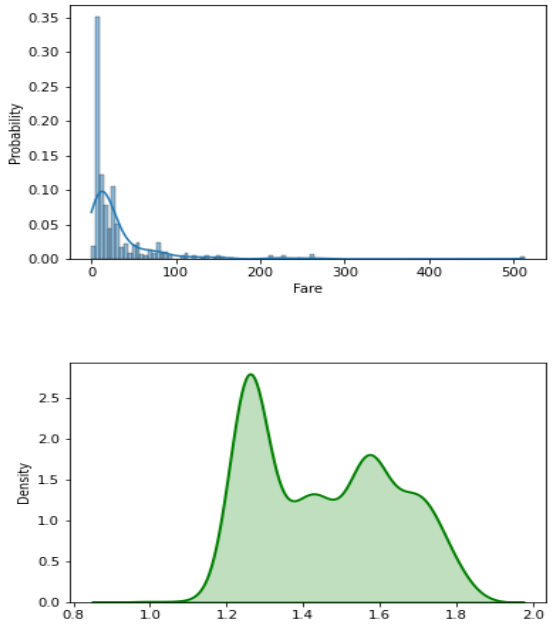

Figure 2: Top: Histogram of the feature Fare; Bottom: Box-Cox transformation of the feature Fare

C. *Logistic Regression Model*

We train logistic regression models for each of the datasets. The parameters of the model are tuned using 10-fold cross-validation. The best results are obtained when we use the L1 penalty (Lasso regression). The precision, recall, f1, and ROC AUC (Receiver Operator Characteristics Area Under Curve) scores on the test sets for each of the models is shown in Table 1. Figure 6 shows the ROC curves for each model. We can see that model 2 gives the best performance on the test set. This is due to the fact that we have removed the outliers in the feature Age. Model 1 is trained with these outliers and hence, its performance is worse than that of model 2. Model 3 is trained with the discretized version of the feature Age. Because of the discretization, the model loses access to the granularity of the feature and hence underperforms.

D. *Permutation importance*

Permutation feature importance is a model inspection technique that can be used for any fitted estimator when the data is tabular. This is especially useful for non-linear or opaque estimators. The permutation feature importance is defined to be the decrease in a model score (in our case, it is the F1 score) when a single feature value is randomly shuffled. This procedure breaks the relationship between the feature and the target, thus the drop in the model score is indicative of how much the model depends on the feature. This technique benefits from being model agnostic and can be calculated many times with different permutations of the feature. Permutation importance does not reflect the intrinsic predictive value of a feature by itself but how important this feature is for a particular model. We use permutation importance for model 2 since this is our best model. Figure 7 shows the relative importance of the features in both the training and test sets. Table 2 shows the decrease in F1 score in train and test sets when each of the features is permuted. We can easily see that the feature Sex plays the most important role in deciding the survival status of a passenger, followed by passenger class and age. Though the relative importance of the feature Sex remains the same in the train and test sets, the relative importance of the other features differs in the train and test sets. This indicates slight overfitting of the model. But still, the differences are small.

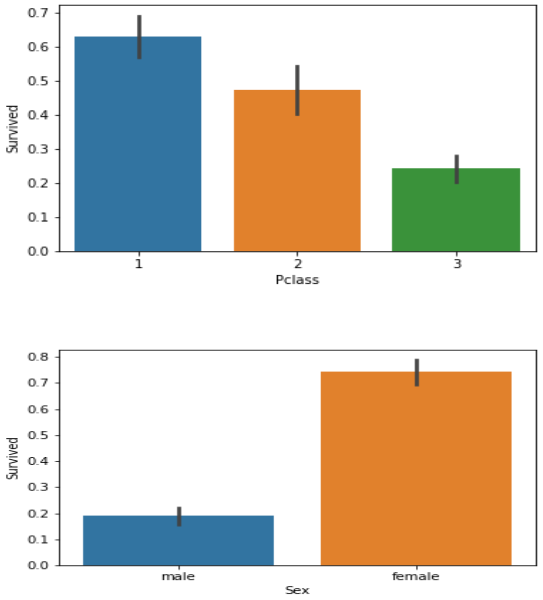

Figure 3: Top: Barplot between Survival rate and the feature Passenger class; Bottom: Barplot between Survival rate and the feature Sex

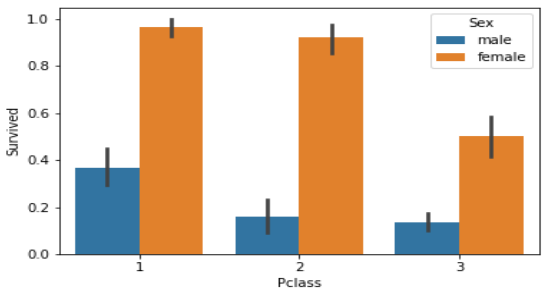

Figure 4: Barplot between Survival rate and the feature Passenger class conditioned on gender

E. *Confidence intervals for the predictions*

Our model outputs the probability of survival of a passenger given their sex, passenger class, and age. We need confidence intervals for our predictions. Hence, we turn to the method of bootstrapping. We resample the entire dataset with replacement several times. The number of times we resample is given by the bootstrap parameter. Every time we resample, we train a logistic regression model with the same hyperparameters used by model 2 and get the predicted probabilities on the test set. Hence, for each observation, we obtain a distribution for the predicted probability of survival. From this, we can construct confidence intervals. Figure 8 shows the distribution and the confidence intervals

for eight randomly selected observations from the test set for a bootstrap size of 2000. We can also use the bootstrapped probabilities as an ensemble technique. We obtain the sample mean of the bootstrapped probabilities for each observation and if this crosses a particular threshold, we classify the observation as "1" (survived). Otherwise, the observation is classified as "0" (not survived). This is called soft classification and the metrics we obtain are a function of the threshold we use. For all practical purposes, we impose a threshold of 0.5. Figure 9 shows the metrics - precision, recall, and F1 scores - as a function of threshold. We see that a threshold of 0.3 gives the best F1 score on the test set and it seems reasonable as well. In such an emergency, humans tend to put in huge effort to ensure survival. Hence, even if the situation indicates a low probability of survival, the passenger is able to survive the mishap.

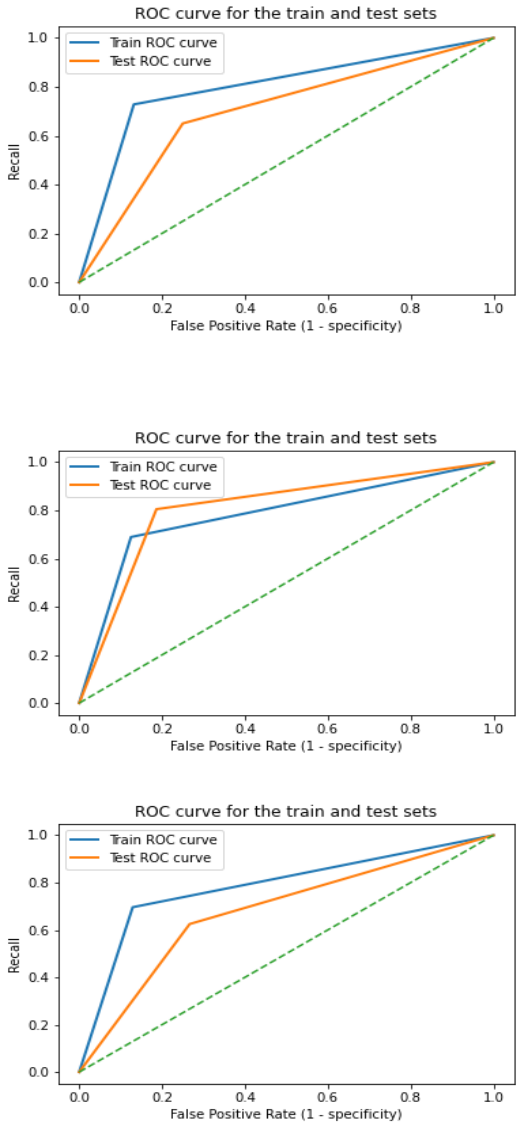

Figure 6: ROC curves for model 1 (top), model 2 (middle), model 3 (bottom)

F. Interpreting the coefficients

From the coefficients obtained from running a logistic regression model on all the features, we make the following quantitative interpretations:

1) The log-odds of survival decreases by 0.91 units as we move to an adjacent lower Passenger class (eg. from passenger class 2 to passenger class 3).
2) The log-odds of survival for females is greater than that for males by 2.72 units.
3) The log-odds of survival decreases by 0.255 units when we have parents or children aboard the Titanic.
4) The log-odds of survival decreases by 0.191 units when we have siblings aboard the Titanic
5) Average marginal decrease in survival rate (probability of survival) with respect to age is found to be -0.002. However, the marginal decrease in survival rate with respect to age is different for different persons.
6) Average marginal increase in survival rate (probability of survival) with respect to fare is found to be 0.0006. However, the marginal increase in survival rate with respect to fare is different for different persons.

IV. Conclusion

The estimates of the factors determining survival during the sinking of the Titanic produce a coherent story. Some noteworthy points are listed:

1) While people in their prime were more likely to be saved, it was women—rather than men—who had a better chance of being saved. Children also had a higher chance of surviving.
2) At the time of the disaster, the unwritten social norm of "saving women and children first" seems to have been enforced.
3) Passengers with high financial means, traveling in first class, we're better able to save themselves as passengers in second class (compared to third class).
4) The log-odds of survival decreases by 0.255 units when we have parents or children aboard the Titanic.
5) The log-odds of survival decreases by 0.191 units when we have siblings aboard the Titanic

The sinking of the Titanic represents a rare case of a well-documented and most dramatic life and death situation. However, even under these extreme situations, the behavior of human beings is not random or inexplicable but can be accounted for by statistical analysis. There is also scope for including other factors. There is documentation stating that crew members who had access to better informational and relational resources managed to survive more often than others aboard. This applies in particular to the deck crew who was partly in charge of the rescue operations. Including this factor in our model can improve its performance and generalization ability. Also, we need to take into account the position of the cabin in which the passengers were present at the time of sinking. Passengers who were present in cabins that were in the lower deck would have a very low chance of survival.

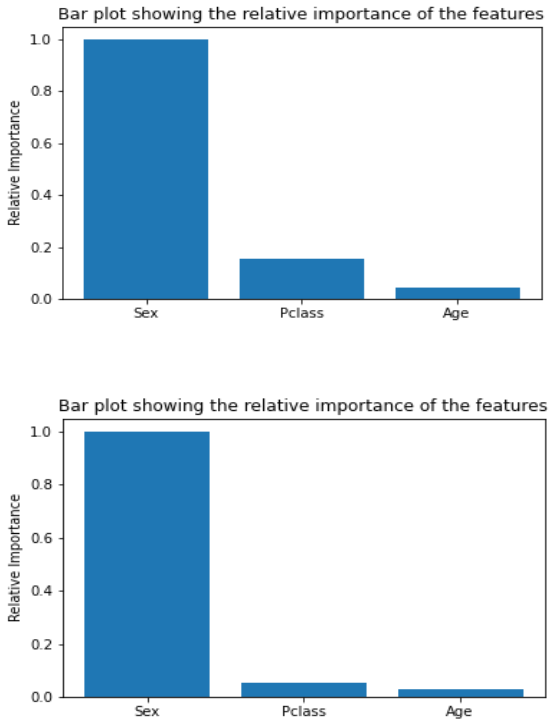

Figure 7: Relative importance of the features in both the training (top) and test (bottom) sets

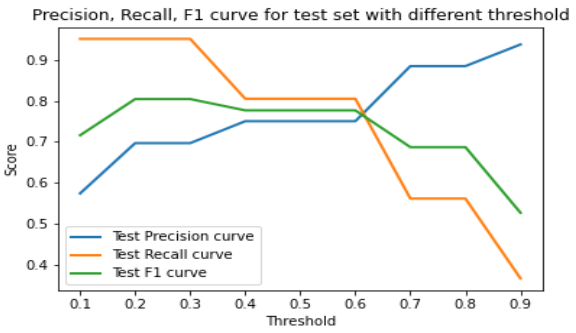

Figure 9: Precision, Recall, and F1 scores on the test set as a function of threshold. We can see that a threshold of 0.3 gives the best F1 score of 0.804 on the test set

References

[1] B. Frey, D. Savage, and B. Torgler, "Surviving the Titanic Disaster: Economic, Natural and Social Determinants", 2009

[2] "The Journal of Economic Perspectives", Vol. 25, No. 1 (Winter 2011), pp. 209-221

[3] W. Hall, "Social Class and the Survival on the S.S. Titanic"

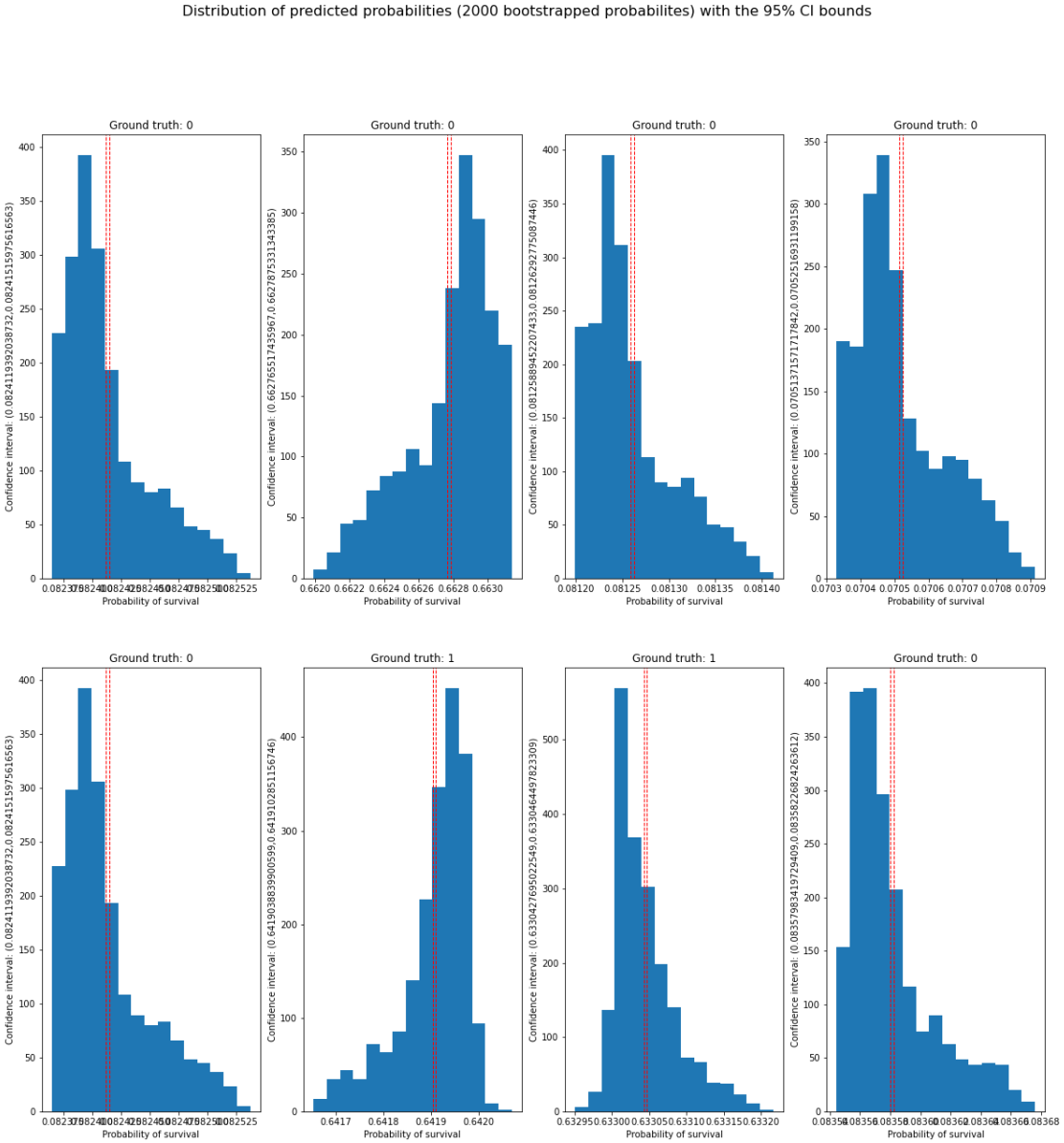

Figure 8: Distribution and the confidence intervals for the probability of survival for eight randomly selected observations from the test set for a bootstrap size of 2000. The ground truth is shown at the top of each plot. The red dotted lines indicate the confidence bounds for the probability of survival for that observation

Metrics	Model 1	Model 2	Model 3
Precision	0.63	0.75	0.61
Recall	0.65	0.81	0.63
F1 score	0.64	0.77	0.62
ROC AUC	0.70	0.81	0.68

Table 1: Model 1: Trained with outliers (L1 penalty; C = 1); Model 2: Trained without outliers (L1 penalty; C = 1); Model 3: Trained with Age feature being discretized (L1 penalty; C = 0.5)

Feature	Decrease in F1 score in Train set (standard deviation in brackets)	Decrease in F1 score in Test set (standard deviation in brackets)
Sex	0.349 (0.017)	0.298 (0.058)
Passenger class	0.054 (0.010)	0.016 (0.018)
Age	0.016 (0.008)	0.009 (0.032)

Table 2: Decrease in F1 score in train and test sets when each of the features is permuted. The feature Sex has the highest importance in predicting the probability of survival of a passenger

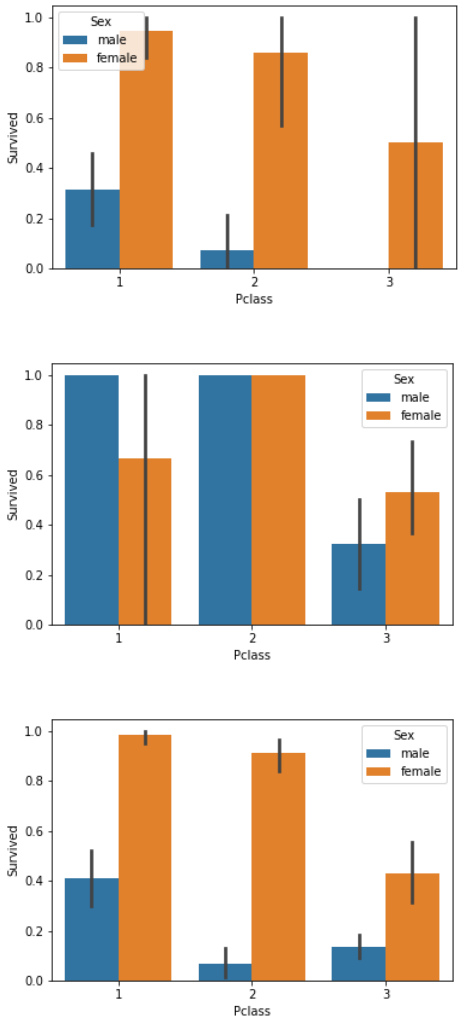

Figure 5: Barplots showing the combined effect of sex and passenger class on survival rate for different age groups. Top: greater than 48 years; Middle: less than 15 years; Bottom: between 15 and 48 years

Assignment 3: A Mathematical essay on Naive Bayes Classifier

Vishal Rishi MK
Department of Chemical Engineering
IIT Madras
ch18b013@smail.iitm.ac.in

***Abstract*—The Naive Bayes classifier is one of the simplest classifiers that can be implemented and interpreted very easily because of its simplifying formulations and assumptions. Though the assumptions rarely hold true in real-world scenarios, the Naive Bayes classifier produces decent classification performance (especially in text classification). In this article, we explain the Naive Bayes classifier and apply it to an income classification. We also explore some variants of the Naive Bayes classifier which would be helpful in dealing with imbalanced datasets.**

Keywords—classification, text, Naive Bayes, imbalanced

I. Introduction

In many situations, the variables we encounter in data analysis are qualitative. For example, gender is qualitative, taking qualitative values male, female. Often qualitative variables are referred to as categorical. Approaches for predicting qualitative responses is known as classification. Predicting a qualitative response for an observation can be referred to as classifying that observation, since it involves assigning the observation to a category, or class. Classification is a fundamental issue in machine learning and data mining. In classification, the goal of a learning algorithm is to construct a classifier given a set of training examples with class labels. Often the methods used for classification first predict the probability of each of the categories of a qualitative variable, as the basis for making the classification. In this sense they also behave like regression methods. On the other hand, there are classifiers that use a distance-based decision function to assign classes for instances. There are many possible classification techniques, or classifiers, that one might use to predict a qualitative response. In this article, we discuss the Naive Bayes approach for classification. We have a set of training observations that we can use to build a classifier. We want our classifier to perform well not only on the training data, but also on test observations that were not used to train the classifier.

Naive Bayes is one of the most efficient and effective inductive learning algorithms for machine learning and data mining. Its competitive performance in classification is surprising, because the conditional independence assumption on which it is based, is rarely true in real world applications. Naive Bayes is the simplest form of Bayesian network, in which all attributes are independent given the value of the class variable. This is called conditional independence. Intuitively, since the conditional independence assumption that it is based on is almost never held, its performance may be poor. It has been observed that, however, its classification accuracy does not depend on the dependencies; i.e Naive Bayes may still have high accuracy on the datasets in which strong dependencies exist among attributes. On the downside, since we make an oversimplifying assumption on the data, the probability estimates produced by the Naive Bayes classifier may not be reliable. Inspite of this limitation, Naive Bayes is termed as the "punching bag of classifiers" as it serves as a very good baseline for classification problems. It is also very easy and quick to implement. It is also very intuitive and easy to understand. This makes the Naive Bayes classifier highly interpretable.

We explain the performance of the Naive Bayes classifier on the problem of income classification. We also want to understand the effect of the oversimplifying assumption of conditional independence on the test precision and test recall scores. Since the model is interpretable, we should also be able to identify the important features that influence the classification performance.

Section 2 of this article gives a mathematical background to Naive Bayes classifier and the estimation of the parameters involved in the model. Section 3 of the article describes the data processing pipeline and several observations, insights, visualizations. We also provide quantitative results after applying Naive Bayes classifier to the train, validation, and test sets. Finally, section 4 draws conclusions from our explorations and analysis.

II. Naive Bayes Classification

Naive Bayes classifier is based on applying Bayes theorem with the "naive" assumption of conditional independence between every pair of features given the value of the class variable. Bayes' theorem states the following relationship, given class variable ***y*** and dependent feature vector ***x1*** through ***xn***,

$$P(y \mid x_1, ..., x_n) \;=\; \frac{P(x_1, ..., x_n \mid y)P(y)}{P(x_1, ..., x_n)}$$

Using the naive conditional independence assumption that

$$P(x_i \mid y, x_1, ..., x_{i-1}, x_{i+1}, ..., x_n) \;=\; P(x_i \mid y)$$

for all ***i,*** Bayes theorem simplifies to

$$P(y \mid x_1, ..., x_n) \;=\; \frac{P(y)\prod_{i=1}^{n} P(x_i \mid y))}{P(x_1, ..., x_n)} \qquad \text{---(i)}$$

Since $P(x_1, ..., x_n)$ is constant given the input, we can use the following classification rule:

$$\widehat{y} \;=\; argmax_y\, P(y)\prod_{i=1}^{n} P(x_i \mid y)) \qquad \text{---(1)}$$

and we can use Maximum A Posteriori (MAP) estimation to estimate $P(y)$ and $P(x_i \mid y)$; the former is then the

relative frequency of class ***y*** in the training set. The different naive Bayes classifiers differ mainly by the assumptions they make regarding the distribution of $P(x_i \mid y)$. Two other variants of the prediction rule given in (1) are

$$\hat{y} = argmax_y \frac{P(y)}{\prod_{i=1}^{n} P(x_i \mid y_c)} \quad \text{---(2)}$$

and

$$\hat{y} = argmax_y \frac{P(y)\prod_{i=1}^{n} P(x_i \mid y)}{\prod_{i=1}^{n} P(x_i \mid y_c)} \quad \text{---(3)}$$

For a given class ***y***, ***yc*** is called the complement of ***y*** (In binary classification, if ***y*** belongs to the positive class, ***yc*** becomes the negative class and vice versa). The likelihood probabilities are estimated using the method of Maximum Likelihood Estimation (MLE).

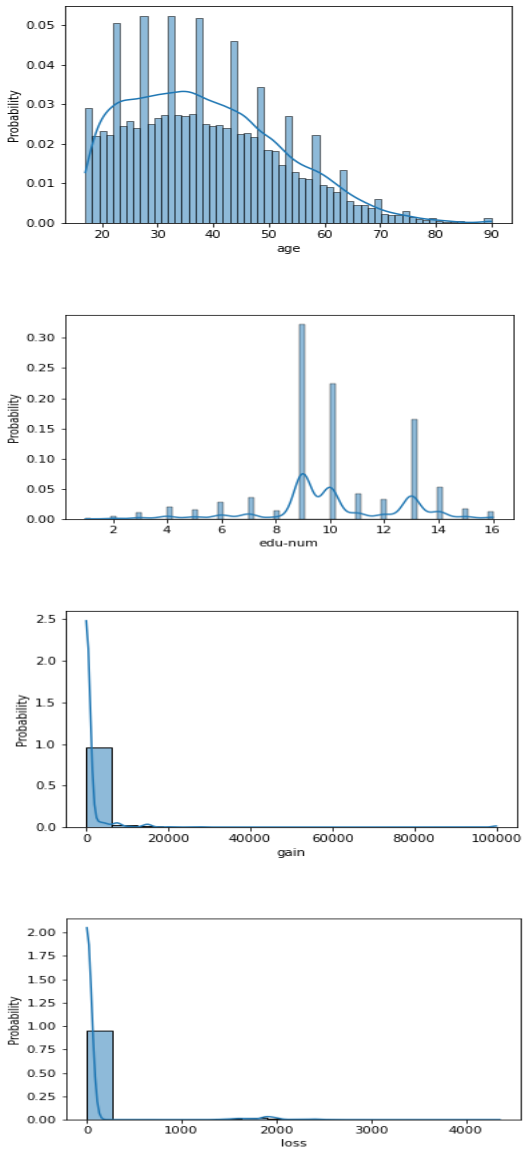

Figure 1: Histogram of the numeric features Age (top), Number of years of education (second), Gain (third), Loss (bottom)

III. Application to the Problem

The data that we have contains 32561 records and 13 features that describe the target variable. The target variable indicates whether the person has an income greater than fifty thousand units or not. Throughout our analysis, we consider "greater than 50K" as the positive class and "less than or equal to 50K" as the negative class. The thirteen features are 1) Age, 2) Working class, 3) Education, 4) Number of years of education, 5) Marital status, 6) Occupation, 7) Relationship status, 8) Race, 9) Sex, 10) Gain, 11) Loss, 12) Country and 13) Number of work hours per week.

A. Preliminary Feature Selection

The features Working class, Occupation, and Country had missing values. The total number of records with missing values amounted to 2399 (7% of the entire dataset). The records with missing values were removed since they did not cause too much loss of data. Table 1 shows the class split in the data. The positive class amounts to just 24.08% of the total records. This shows that we have an imbalance dataset that could potentially bias the model towards the negative class. Figure 1 shows the histogram of the numeric features Age, Number of years of education, Gain, Loss. We can see that the features Gain and Loss have most of their entries as zeroes (more than 75% of their entries). Hence, these features show very little variation and they do not qualify as good starting attributes to predict the income category of a candidate. The numeric features Age and Number of years of education are not continuous. They can be modeled as discrete variables. Hence, we do not need to fit probability distributions to estimate the likelihood probabilities. The Maximum Likelihood estimator of the likelihood for a discrete feature would result in the relative normalized frequency of the discrete feature value. Also, Figure 2 shows the scatter plot between Age and Number of years of education. We can see a pattern that indicates increasing income as age and the number of years of education increases. Also, if the number of years of education is less than 10, increasing age does not necessarily increase income. Figure 2 also shows the scatterplot between Age and working hours per week. We could not see any clear patterns that would be helpful in predicting the income category. We can see a considerable number of candidates earning more even after working for less time. This is prevalent in all age groups. These observations show that the features Age and Number of years of education are good features for our problem. For a baseline model, in addition to these numeric features, we will also work with the remaining eight categorical variables. Hence, with ten features, we create the train, validation, and test datasets. The class split is maintained the same in all the datasets for representativeness. Table 2 shows the number of records in each dataset. We use the held-out validation set to compare between different frameworks and to select the best model. Finally, we report the performance of the selected model on the test dataset.

B. Naive Bayes classifier and its variants

Equation (1) gives the vanilla Naive Bayes classifier and equations (2) and (3) are some variants. These variants are known to perform well when we have imbalanced datasets. We train three models - Model 1 (equation (1)), Model 2 (equation (2)), and Model 3 (equation (3)) on the train dataset. Their performance in the validation set is given in Table 3. While evaluating in the validation set, Laplace smoothing (smoothing parameter equal to 1.0) is used when a new feature value that did not occur in the training set is encountered. We also noted that the value of the Laplace

smoothing parameter did not have a significant effect on the performance of the models. An advantage of model 1 is that it can output the posterior probabilities along with the predicted classes for the validation instances. These probabilities can be further used for ROC (Receiver - Operator Characteristics) analysis. But models 2 and 3 do not give probabilities that have a physical significance. Hence, they are not amenable to ROC analysis. From Table 3, we can see that model 3 achieves a greater recall (81%) at the cost of a lower precision (56%) when compared to models 1 and 2. This means that model 3 produces a lot more false positives compared to models 1 and 2. By including the complementary likelihood probabilities in the decision function (given in equation (3)), we correct for the reduced representation of the positive class in the training dataset. By doing this, we also increase the false positives of the model. Though the accuracy is close to 80% for all the models, we cannot use that as a fair assessment tool because of class imbalance. Hence, we turn to F1-score which is close to 66% for all the models. The numbers indicate that there is certainly room for improvement.

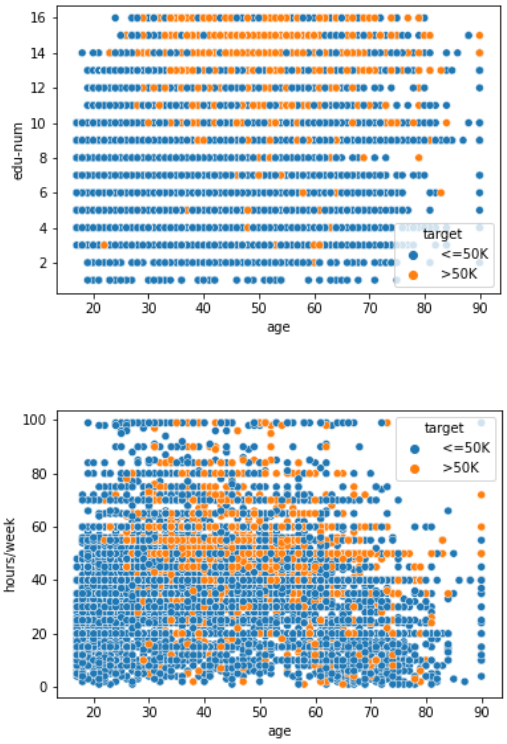

Figure 2: Scatter plot between Age and Number of years of education (top) and scatterplot between Age and working hours per week (bottom)

C. Sequential Forward Feature Selection

The performance of the classifier can be improved by careful selection of features. Naive Bayes classifier relies on the important assumption of conditional independence. Hence, if we have correlated features in our data (like Education, Number of years of education), the posterior probabilities as shown in (i) get affected badly because of the weakening of the conditional independence assumption. Hence it is important to select features that are uncorrelated (the weakest form of independence). Table 4 shows the mutual information between the numeric features and the target variable. The mutual information between the Number of work hours per week and the target variable is very low which can also be verified from Figure 2. But the mutual information between Gain and the target variable is higher than expected. This contradicts our observation from Figure 1. This calls for further analysis on the feature Gain. Since only 6.9% of the records have a non-zero gain, we convert the feature to a binary variable ("0" indicating no gain and "1" indicating non-zero gain). Since the models were producing a lot of false positives, we take a look at the proportion of the false positives that have a value "1" in the transformed feature Gain. Table 5 summarizes this split. We can see that a majority of false positives have zero gain. By reducing this part of the error, we can improve the precision of the model. Figure 3 shows the conditional probabilities of gain, given the target variable. From this figure, the conditional probability P(gain = 0 | target = 0) is greater than

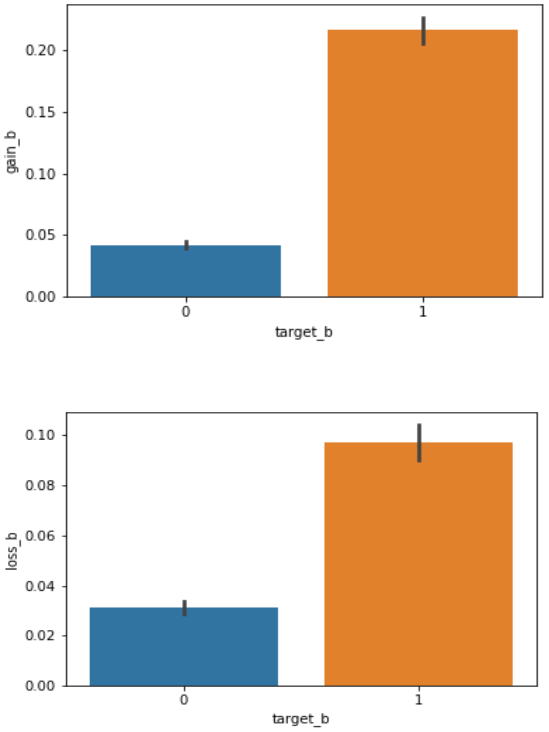

Figure 3: Conditional probabilities of gain, given the target variable (top), and conditional probabilities of loss, given the target variable (bottom). (gain_b = fraction of records with gain greater than zero, loss_b = fraction of records with loss greater than zero, target_b = binary target variable)

P(gain = 0 | target = 1). By including the binary variable Gain in our analysis, some of the false positives with zero gain would be pushed to the negative class thereby improving the precision. The same argument can be put forth for the variable Loss as well. Table 6 shows the performance of the three models after including the variables Gain and Loss (binary variables) on the validation set. Though the number of false positives has decreased, the increase in precision is not significant. This is due to the presence of correlated features that create an opposing effect.

To select the best set of features, we recursively add features to a list that increase the ROC AUC score of a baseline classifier. We remove a feature if it does not increase the ROC AUC score. This way, we progressively build a list that gives the best ROC AUC score on the

validation set. This is called the Sequential Forward Feature Selection strategy. After applying this, we find the following set of features - 1) Age, 2) Gain, 3) Loss, 4) Working class, 5) Number of years of education, 6) Relationship status, and 7) Sex. Table 7 shows the performance of the three models on the validation set when the best subset is used. Clearly, the precision has increased to 64% and model 1 produces a good ROC AUC score of 87%.

D. Abstaining Classifiers

Classifiers that refrain from classification in certain cases can significantly reduce the misclassification cost. These are called Abstaining classifiers. An Abstaining classifier that can refrain from classification in certain cases is analogous to a human expert, who in certain cases can say "I don't know". In many domains (e.g medical diagnosis) such experts are preferred to those who always make a decision and are sometimes wrong. For our problem, since we have a classifier that produces false positives and false negatives, it will be beneficial to have an abstaining classifier that refrains from classification in certain unsure scenarios. In this way, we can reduce the classification error to a certain extent. But at the same time, we need to keep track of the coverage as well. Coverage is defined as the fraction of all instances for which the abstaining classifier predicts a class. We need to build an abstaining classifier that has sufficiently large coverage and enhanced classification performance. Since we have model 1 and model 3 producing decent performance on the validation set (when trained with the best subset of features), we build our abstaining classifier as follows: If models 1 and 3 produce the same class label, the abstaining classifier also outputs the same class label. Otherwise, the abstaining classifier refrains from classification. Table 7 shows the classification metrics for the abstaining classifier on the validation set. We can see that the classifier has a precision of 64% and a recall of 74%. The increase in recall suggests that a lot of false negatives have refrained from classification. By combining models 1 and 3 into a single abstaining classifier, we are able to combine the goodness of both models (increase in precision and recall). As a result, the abstaining classifier achieves the best F1 score of 69% (at a coverage of 89%). Finally, Table 8 shows the classification performance of the abstaining classifier on the test set.

IV. Conclusion

In this article, we discuss the mathematical formulation behind the Naive Bayes classifier and its variants. We use this technique to solve the problem of income classification. Some of the key insights that we obtained are listed :

1. The features age, working-class, number of years of education, gain, loss, sex, and relationship status are the subset of features that best describe the income category of a person.
2. The variants of the Naive Bayes classifier shown in equations (2) and (3) tend to oppose the tendency of the classifier to output the majority class by introducing complementary likelihood probabilities. Because of this, they are able to achieve high recall (close to 81%) but at the cost of lower precision.
3. The abstaining classifier discussed in this article combines the best of all the variants of the Naive Bayes classifier while maintaining the coverage at 89%.
4. The presence of correlated features in our analysis affects the posterior probability estimates badly.

There is still room for improvement. We can employ model agnostic techniques to sort the features according to their importance. Careful hypothesis testing would reveal the presence of gender or racial bias. That would be an interesting study by itself.

References

[1] T. Pietraszek, "Optimizing Abstaining Classifiers using ROC Analysis", IBM Zurich Research Laboratory

[2] J. D. M. Rennie, L. Shih, J. Teevan, D. R. Karger, "Tackling the Poor Assumptions of Naive Bayes Text Classifiers", Artificial Intelligence Laboratory; Massachusetts Institute of Technology; Cambridge

[3] M. Zhang, J. M. Pena, V. Robles, "Feature selection for multi-label naive Bayes classification", 2009

[4] H. Zhang. "The Optimality of Naive Bayes"

	Positive class (income >50K)	Negative class (income <=50K)
Number of records	7841	24720

Table 1: Class split in the dataset

	Training set	Validation set	Testing set
Number of records	24430	2715	3017

Table 2: Size of the train, validation, and test datasets

	Model 1	Model 2	Model 3
Accuracy	0.80	0.81	0.79
Precision	0.59	0.60	0.55
Recall	0.75	0.73	0.82
F1 score	0.66	0.66	0.66
ROC AUC	0.87	NA	NA

Table 3: Classification metrics of all the models on the validation set. These models are trained with the preliminary set of ten features (NA - Not Applicable)

	Age	Work-class	Gain	Loss	Hours/week
Mutual Information	0.0628	0.0713	0.0863	0.0363	0.0450

Table 4: Mutual Information between numeric features and the target variable

	Model 1	Model 2	Model 3
Zero Gain	3075	2677	3810
Gain > 0	183	303	229

Table 5: The number of false-positive instances produced by the models with zero gain and non-zero gain. We can see that a majority of false positives have zero gain. By reducing this part of the error, we can improve the precision of the corresponding model

	Model 1	Model 2	Model 3
Accuracy	0.81	0.81	0.79
Precision	0.60	0.60	0.56
Recall	0.73	0.73	0.80
F1 score	0.66	0.66	0.66
ROC AUC	0.88	NA	NA

Table 6: Classification metrics of all the models on the validation set. These models are trained with the preliminary set of ten features in addition to the binary variables ‘gain’ and ‘loss’ (NA - Not Applicable)

	Model 1	Model 2	Model 3	Abstaining classifier
Accuracy	0.82	0.82	0.79	0.84
Precision	0.64	0.65	0.56	0.65
Recall	0.62	0.62	0.77	0.73
F1 score	0.63	0.63	0.65	0.69
ROC AUC	0.87	NA	NA	0.89

Table 7: Classification metrics of all the models (including the abstaining classifier) on the validation set. These models are trained with the best subset of features obtained using sequential forward feature selection. It is to be noted that the abstaining classifier achieves coverage of 89% (NA - Not Applicable)

	Abstaining classifier
Accuracy	0.85
Precision	0.67
Recall	0.75
F1 score	0.71
ROC AUC	0.91

Table 8: Classification metrics of the abstaining classifier on the test set. The model is trained with the best subset of features obtained using sequential forward feature selection. It is to be noted that the abstaining classifier achieves coverage of 89% on the test set as well

Assignment 4: A Mathematical essay on Decision Tree

Visha Rishi MK
Department of Chemical Engineering
IIT Madras
ch18b013@smail.iitm.ac.in

***Abstract*—Decision trees are non-parametric supervised learning methods that do not assume anything about the data distribution. Understanding the decision tree structure will help in gaining more insights about how the decision tree makes predictions, which is important for understanding the important features in the data. This is used as a preliminary step in exploring the data in several complex problems. In this article, we tackle the problem of classifying cars based on their safety with the help of a decision tree classifier.**

Keywords—Decision trees, supervised, non-parametric

I. Introduction

In many situations, the variables we encounter in data analysis are qualitative. For example, gender is qualitative, taking qualitative values male, female. Often qualitative variables are referred to as categorical. Approaches for predicting qualitative responses is known as classification. Predicting a qualitative response for an observation can be referred to as classifying that observation, since it involves assigning the observation to a category, or class. Classification is a fundamental issue in machine learning and data mining. In classification, the goal of a learning algorithm is to construct a classifier given a set of training examples with class labels. Often the methods used for classification first predict the probability of each of the categories of a qualitative variable, as the basis for making the classification. In this sense they also behave like regression methods. On the other hand, there are classifiers that use a distance-based decision function to assign classes for instances. There are many possible classification techniques, or classifiers, that one might use to predict a qualitative response. In this article, we discuss the Decision tree approach for classification. We have a set of training observations that we can use to build the classifier. We want our classifier to perform well not only on the training data, but also on test observations that were not used to train the classifier.

Decision trees are a non-parametric supervised learning method used for classification and regression. The goal is to create a model that predicts the value of a target variable by learning simple decision rules inferred from the data features. A tree can be seen as a piecewise constant approximation. Some of the advantages of working with a decision tree are 1) They are simple and easy to understand and the decisions made by the tree can be visualized. Hence, this is a white box model, 2) They require little data preparation. Other techniques often require data normalisation, dummy variables need to be created and blank values to be removed, 3) They perform well even if their assumptions are somewhat violated by the true model from which the data were generated. On the other hand, Decision-tree learners can create over-complex trees that do not generalise the data well. This is called overfitting. Mechanisms such as pruning, setting the minimum number of samples required at a leaf node or setting the maximum depth of the tree are necessary to avoid this problem. They can be unstable because small variations in the data might result in a completely different tree being generated. This problem is mitigated by using decision trees within an ensemble. Also, the problem of learning an optimal decision tree is known to be NP-complete under several aspects of optimality and even for simple concepts. Consequently, practical decision-tree learning algorithms are based on heuristic algorithms such as the greedy algorithm where locally optimal decisions are made at each node. Such algorithms cannot guarantee to return the globally optimal decision tree. This can be mitigated by training multiple trees in an ensemble learner, where the features and samples are randomly sampled with replacement. Decision-tree learners create biased trees if some classes dominate. It is therefore recommended to balance the dataset prior to fitting with the decision tree. It is to be noted that predictions of decision trees are neither smooth nor continuous, but piecewise constant approximations.

In this article, we want to classify cars based on their safety. Given that the data is obtained from a hierarchical decision model, we use decision trees to retrieve the decision rules that were used to classify the cars. These simple decision rules retrieved from the model are a proof to the explainability of the decision tree classifier.

Section 2 of this article gives a mathematical background to the decision tree classifier. Section 3 of the article describes the data processing pipeline and several observations, insights, and visualisations. We also provide quantitative results after applying a decision tree classifier to the train and test sets. Finally, section 4 draws conclusions from our explorations and analysis.

II. Decision Tree Classifier

Given training vectors $\boldsymbol{xi} \in R^n$**, i = 1 to n**, and a label vector $\boldsymbol{y} \in R^l$, a decision tree recursively partitions the feature space such that the samples with the same labels or similar target values are grouped together. a decision tree recursively partitions the feature space such that the samples with the same labels or similar target values are grouped together. Let the data at node ***m*** be represented by Q_m with N_m samples. For each candidate split $\boldsymbol{\theta} \in \boldsymbol{(j,\ t_m)}$, consisting of a feature ***j*** and threshold t_m, partition the data into ${Q_m}^{left}(\theta)$ and ${Q_m}^{right}(\theta)$ subsets

$${Q_m}^{left}(\theta) = \{(x,\ y) \mid x_j <= t_m\}$$

$${Q_m}^{right}(\theta) = Q_m \setminus {Q_m}^{left}(\theta)$$

The quality of a candidate split of node ***m*** is then computed using an impurity function or loss function ***H(),*** the choice of which depends on the task being solved (classification or regression)

$$G(Q_m, \theta) = \frac{N_m{}^{left}}{N_m} H(Q_m{}^{left}(\theta)) + \frac{N_m{}^{right}}{N_m} H(Q_m{}^{right}(\theta$$

We select the parameters that minimize the impurity,

$$\widehat{\theta} = arg\, min_\theta\, G(Q_m, \theta)$$

We recursively apply this procedure to the subsets $Q_m{}^{left}(\widehat{\theta})$ and $Q_m{}^{left}(\widehat{\theta})$ until the maximum allowable depth is reached or $N_m < min_{samples}$ or $N_m = 1$.

A. Classification Criterion

If a target is a classification outcome taking on values 0,1,…, K-1, for node ***m,*** let

$$p_{mk} = \frac{1}{N_m} \sum_{y \in Q_m} I(y = k)$$

be the proportion of class ***k*** observations in node ***m***. If ***m*** is a terminal node, the predicted probability for this region is set to p_{mk}. Common measures of impurity are the following

Gini impurity:

$$H(Q_m) = \sum_k p_{mk}(1 - p_{mk})$$

Entropy:

$$H(Q_m) = -\sum_k p_{mk}\, log(p_{mk})$$

Misclassification:

$$H(Q_m) = 1 - max(p_{mk})$$

III. Application to the Problem

The data that we have contains 1728 records and 6 features that describe the target variable. The target variable indicates car safety. The target variable consists of four classes - “unaccountable”, “accountable”, “good”, and “very good”. Hence, it qualifies as an ordinal qualitative variable. The six features are 1)Buying price, 2) Price of maintenance, 3) Number of doors, 4) Capacity of the car (in terms of the people to carry), 5) Size of the luggage boot, and 6) Estimated safety of the car.

A. Data exploration

The features are qualitative (ordinal) in nature. None of the features are numerical. Hence, we use ordinal encoding to transform the features for further analysis. Table 1 shows the ordinal encoding for all the features. Also, none of the features have missing values. Table 2 shows the number of data points (cars) in each class. This shows an imbalance in the number of data points between the classes. Most of the cars in the dataset are being classified as “unaccountable”. Class imbalance can lead to trees that are biased towards the majority class (ie. “unaccountable”). Hence we group the classes “accountable”, “good”, and “very good” into a single class - “accountable”. This turns our problem into a binary classification problem. Since the original target variable is ordinal, a multiclass classification strategy might be inappropriate. Hence, working with a binary target variable seems relevant for classification algorithms (binary classification). We should make sure that we group similar classes into a single class. In our case, it makes sense to combine “accountable”, “good”, and “very good” into a single class. In our analysis, we have used “accountable” as the positive class and “unaccountable” as the negative class. Once we frame a binary classification problem, we mitigate the problem of class imbalance in the data (29.9% of the data points belong to the positive class).

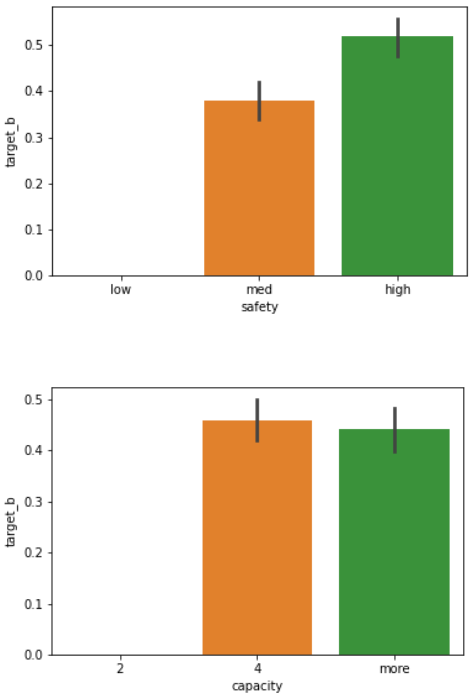

Figure 1: Barplots between the target variable and “Estimated safety of the car” (top) and the “Capacity of the car” (bottom). target_b represents the probability of belonging to the positive class

B. Preliminary feature selection

Table 3 shows the mutual information between each feature and the transformed target variable. We can see that the variables “Capacity of the car” and the “Estimated safety of the car” have the highest mutual information. We work only with these variables in our initial analysis. Figure 1 shows the bar plots between the target variable and “Capacity of the car” and the “Estimated safety of the car”. We can see that when the estimated safety is low or the capacity of the car is 2, all the cars belong to the negative class (all the cars are “unaccountable”). Also, when the capacity of the car increases, the probability of the car belonging to the positive class increases. A similar trend is also seen with the variable “Estimated safety of the car”. We also want to see the combined effect of these variables on the target variable. Figure 2 shows the barplot between the target variable and the car capacity conditioned on the estimated safety of the car. The patterns that we discussed before are visible in this plot as well. This shows that the

capacity and estimated safety of a car are good indicators of the target variable.

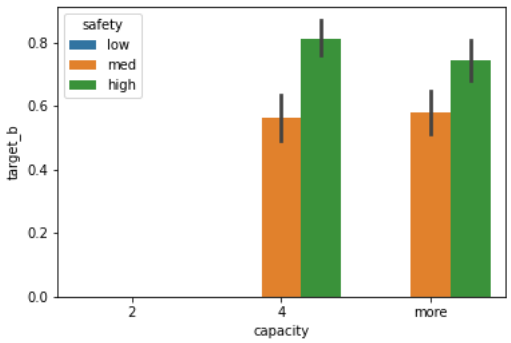

Figure 2: Barplot between the target variable and the car capacity conditioned on the estimated safety of the car. target_b represents the probability of belonging to the positive class

C. *Decision Tree Classifier*

The data is split into train and test datasets. The train data has 1555 data points and the test dataset has 173 data points. While performing the train-test split, we ensure that the class split remains the same in the train and test data. This is to ensure that the test data is representative of the train data. We train a decision tree classifier on the training dataset with 10-fold cross-validation (Tree 0). The cross-validation scores yielded a ROC AUC score of 0.92 (with a standard deviation of 0.026). Table 4 shows the evaluation metrics on both the train and test datasets. Figure 3 shows the ROC curve of the classifier. Though the classifier does well on the test recall and the test ROC AUC score, the test precision score is modest and can be improved. Figure 4 is a visualization of the decision tree. The decision tree is fairly simple since we have only used two attributes. The depth of the tree is four and we have six leaf nodes. Out of the six nodes, two leaf nodes have a Gini impurity close to 0.5. The samples in those nodes are predicted positive. Since the Gini impurity is high, the tree produces a lot of false positives by predicting a positive tag for all the samples in those leaf nodes. The false positives reduce the precision of the classifier. It is to be noted that all the samples in those leaf nodes have a capacity greater than 2 and fall under the "medium" estimated safety category.

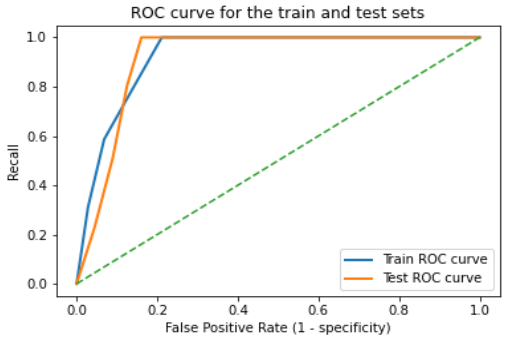

Figure 3: ROC curve of Tree 0 (Test ROC AUC: 0.916)

D. *Abstaining classifier*

Classifiers that refrain from classification in certain cases can significantly reduce the misclassification cost. These are called Abstaining classifiers. An Abstaining classifier that can refrain from classification in certain cases is analogous to a human expert, who in certain cases can say "I don't know". In many domains (e.g medical diagnosis) such experts are preferred to those who always make a decision and are sometimes wrong. For our problem, since we have a classifier that produces false positives, it will be beneficial to have an abstaining classifier that refrains from classification in certain unsure scenarios. In this way, we can reduce the classification error to a certain extent. But at the same time, we need to keep track of the coverage as well. Coverage is defined as the fraction of all instances for which the abstaining classifier predicts a class. We need to build an abstaining classifier that has sufficiently large coverage and enhanced classification performance. We follow a simple rule to increase the precision of the early model. If a particular data point has a capacity greater than 2 and if it falls under the "medium" estimated safety category, we refrain from classifying it. Otherwise, the decision tree we trained earlier (Tree 0) is used for prediction. Table 5 shows the evaluation metrics of the abstaining classifier on the train and test data. Figure 5 shows the ROC curve of the abstaining classifier. The abstaining classifier has achieved a ROC AUC of 95% with a coverage of 78% on the test set compared to the ROC AUC of 92% produced by Tree 0. But the test precision score of the abstaining classifier has not improved much, contrary to what we had expected. This is because the abstaining classifier has reduced the number of true positives (by refraining from classification) in addition to reducing the false positives. These opposing effects cancel each other out, thereby, not affecting the test precision score.

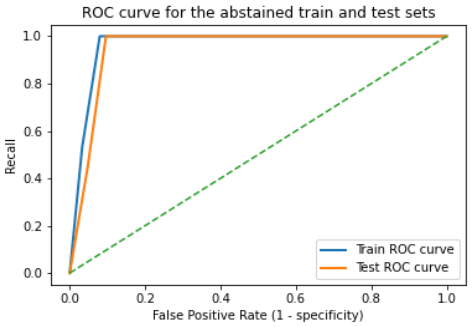

Figure 5: ROC curve of the Abstaining classifier (Test ROC AUC: 0.949)

E. *Improving the Precision - A Hybrid Approach*

As we saw in the previous section, the abstaining classifier does not significantly increase the precision. Hence we need to further analyze those data points that have a capacity greater than 2 and fall under the "medium" estimated safety category. We create a separate dataset with data points that fulfill the above criterion. We split this into train and test data. Once again, we make sure that the class split remains the same in both the train and test datasets (almost 50% of the data points belong to the positive class in both the datasets). The train data has 346 data points and the test data has 38 data points. Table 6 shows the mutual information between the features and the target variable. We can infer that the features "Buying price", "Price of maintenance", and "Size of the luggage boot" have a

significant effect on the target, and hence we include them in our analysis. Since the variable "Number of doors" is insignificant (low mutual information), we discard it from our analysis. It should also be noted that the capacity and estimated safety of the cars do not play an important role in this analysis. Since all the cars in the dataset fall under the "medium" estimated safety category with a capacity greater than 2. Figure 6 shows the barplot between the target variable and the buying price, conditioned on the price of maintenance.

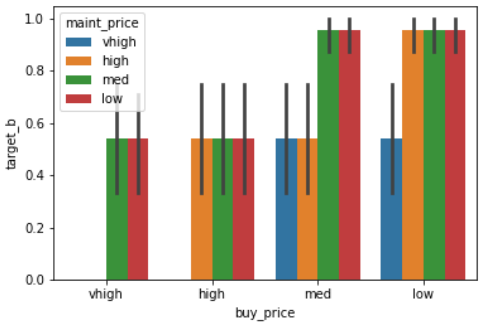

Figure 6: Barplot between the target variable and the buying price, conditioned on the price of maintenance. target_b represents the probability of belonging to the positive class

It seems to suggest that high costs do not guarantee safety. This observation is a bit surprising as we would expect costly cars to have better safety features. Particularly, when the buying price and the price of maintenance are either "low" or "medium", almost all the cars belong to the positive class ("accountable"). Given these observations, we train a decision tree classifier (Tree 1) on the train data with 10-fold cross-validation. Since trees are prone to overfitting, we set the maximum depth to 4. Figure 7 shows an example of an overfitted tree. Once the training is complete, the cross-validation scores yield a mean ROC AUC of 0.922 (with a standard deviation of 0.041). Figure 8 is a visual representation of Tree 1. Table 7 displays the evaluation metrics of Tree 1 on the train and test data. Also, Figure 9 is the ROC curve of Tree 1.

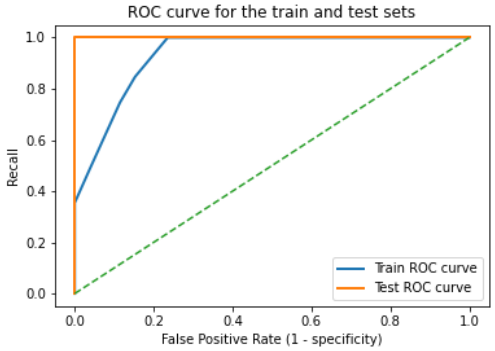

Figure 9: ROC curve of Tree 1 (Test ROC AUC: 1.0)

Since it shows promising results, we combine Tree 0 and Tree 1 to beat the performance of Tree 0 and the abstaining classifier. The logic is as follows: If a data point has a capacity greater than 2 and if it falls under the "medium" estimated safety category, Tree 1 is used to predict the tag. Otherwise, we use Tree 0 for prediction. Table 8 displays the evaluation metrics of this hybrid classifier on the train and test data. Also, Figure 10 shows the ROC curve and the Precision-Recall curve of the hybrid classifier.

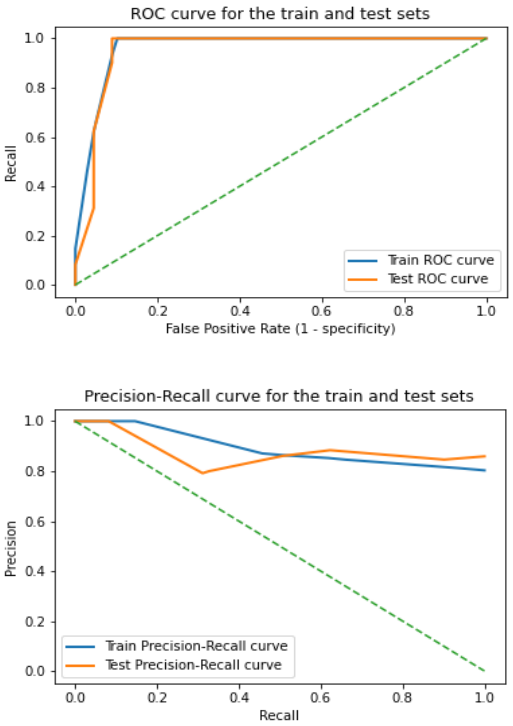

Figure 10: ROC curve and the Precision-Recall curve of the hybrid classifier (Test ROC AUC: 0.953, Test Precision: 0.859, Test Recall: 1.0)

The hybrid classifier achieves the best precision score of 86% on the test set while not affecting the ROC AUC and the recall scores. Hence, the hybrid classifier can be seen as a boosting algorithm where Tree 1 complements Tree 0 by reducing the misclassification errors made by Tree 0. Also, the hybrid approach reduces overfitting by gradually increasing the number of features considered by the model (Tree 0 considers only two features whereas Tree 1 considers three different features. The probability of overfitting would be very high if we train a decision tree starting with all the five features). The hybrid approach also improves the explainability, thus giving us better insights into the data.

F. Multiclass classification

After analyzing the results for binary classification with decision trees, we analyze the results for multiclass classification in this section. In our case, we have four classes - "unaccountable", "accountable", "good", and "very good". Hence, the target qualifies as an ordinal qualitative variable. We train a decision tree classifier (Tree 2) with buying price, price of maintenance, size of luggage boot, estimated safety, and the capacity of the car as features. Table 9 shows the evaluation metrics for Tree 2 in the test dataset. Figure 11 is a graphic visualization of Tree 2. We can see that the tree is much more complex than the previous classifiers. From Table 9, we can observe that the classifier is able to distinguish between the "unaccountable", and "accountable" classes. But it faces difficulty in differentiating the "good", and "very good" classes. This is mainly due to the underrepresentation of

these classes in the dataset. Hence, we conclude that a 4-class classification strategy seems redundant. As an improvement to the binary classification strategy, we propose a 3-class classification strategy in which the classes "good", and "very good" are combined. This would make the analysis more meaningful and efficient.

IV. Conclusion

In this article, we discussed the mathematical formulation behind Decision Tree classifiers and applied them to the problem of the classification of cars based on safety. Some of the key insights that we obtained are listed below:

1. Most of the cars that have high buying prices and high prices of maintenance belong to the negative class ("unaccountable"). This shows that costly cars are not worthy when it comes to safety.
2. All the cars that fall under the "low" estimated safety category belong to the negative class ("unaccountable").
3. All the cars that have a capacity of 2 belong to the negative class ("unaccountable").
4. More features might lead to trees that overfit the data. Hence, we adopted a hybrid approach which gradually increases the number of features considered by the model. This induces a boosting mechanism where the errors produced by a particular model are rectified by another model which considers a different subset of features. As a consequence, the hybrid classifier achieves the best precision score on the test data. The hybrid approach also improves the explainability, thus giving us better insights into the data.
5. In the multiclass classification approach, we cannot clearly separate the classes "good", and "very good" because of their underrepresentation. To overcome this shortcoming, we propose to work with three classes - "unaccountable", "accountable", and "good"

In this article, we have predominantly approached this problem as a binary classification task. An area of research would be to bring several ranking-based metrics for multiclass classification, apart from the traditional classification metrics, into the picture. We have to check whether decision tree classifiers provide any advantage over random forests in the multi-class classification task.

References

[1] H. Patel, P. Prajapati, "Study and Analysis of Decision Tree-Based Classification Algorithms", International Journal of Computer Sciences And Engineering, 6(10):74-78, 2018

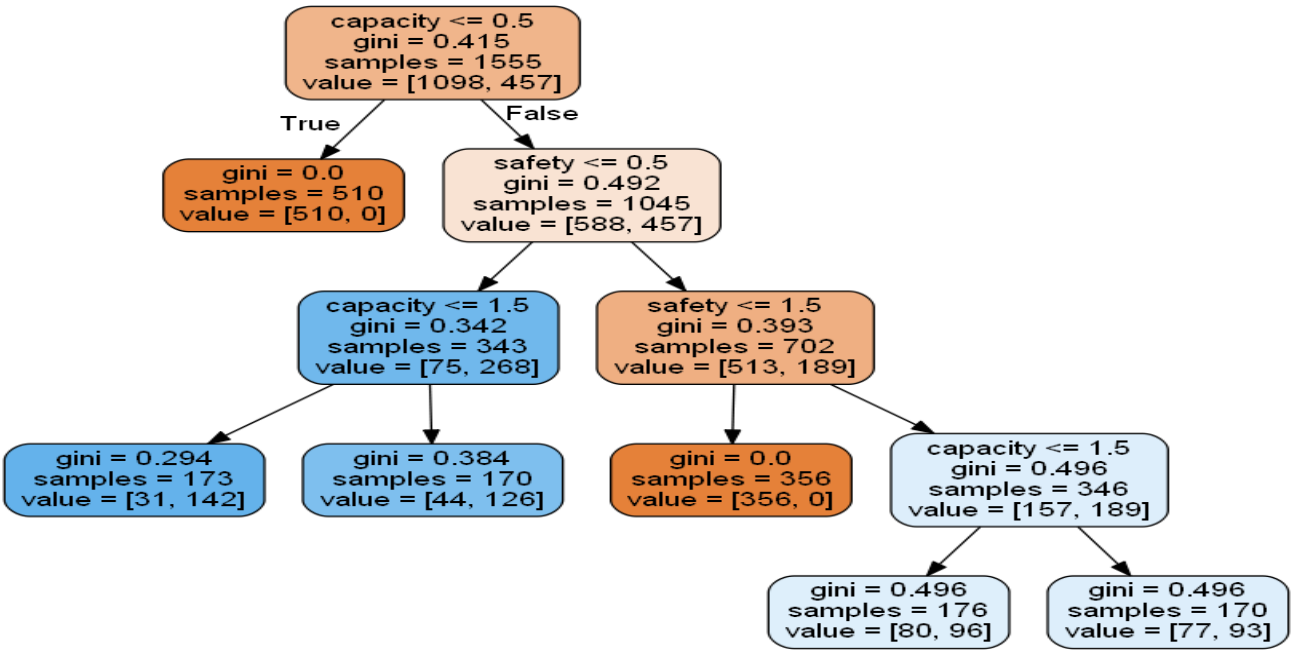

Figure 4: Visualization of Tree 0. Refer to Table 1 for ordinal encoding

Buying price		Price of maintenance		Number of doors		Capacity of car		Size of the luggage boot		Estimated safety	
Feature	Value	Feature	Value	Feature	Value	Feature	Value	Feature	Value	Feature	Value
high	0	high	0	2	0	2	0	big	0	high	0
low	1	low	1	3	1	4	1	medium	1	low	1
medium	2	medium	2	4	2	more	2	small	2	medium	2
vhigh	3	vhigh	3	5more	3	---	---	---	---	---	---

Table 1: Ordinal Encoding of all the features (vhigh = very high)

Class	Number of data points
Unaccountable	1210
Accountable	384
Good	69
Very good	65

Table 2: Number of data points in each class. We can clearly see the class imbalance in the data

	Buying price	Price of maintenance	Number of doors	Capacity	Size of luggage booty	Estimated Safety
Mutual Information	0.06	0.05	0.003	0.15	0.02	0.18

Table 3: Mutual information between each feature and the transformed binary target variable

	Train set	Test set
Accuracy	0.85	0.89
Precision	0.66	0.77
Recall	1.0	1.0
F1 score	0.79	0.87
ROC AUC	0.92	0.92

Table 4: Evaluation metrics of Tree 0 on both the train and test datasets

	Train set	Test set
Accuracy	0.94	0.93
Precision	0.78	0.75
Recall	1.0	1.0
F1 score	0.87	0.86
ROC AUC	0.96	0.95

Table 5: Evaluation metrics of Abstaining classifier on both the train and test datasets. The abstaining classifier achieves coverage of 78%

	Buying price	Price of maintenance	Number of doors	Capacity	Size of luggage booty	Estimated Safety
Mutual Information	0.17	0.14	0.02	0.001	0.13	0.00

Table 6: Mutual information between each feature and the transformed binary target variable. This time, we consider only those samples that are abstained from classification by the abstaining classifier

	Train set	Test set
Accuracy	0.89	1.0
Precision	0.83	1.0
Recall	1.0	1.0
F1 score	0.91	1.0
ROC AUC	0.93	1.0

Table 7: Evaluation metrics of Tree 1 on both the train and test datasets

	Train set	Test set
Accuracy	0.93	0.94
Precision	0.80	0.86
Recall	1.0	1.0
F1 score	0.89	0.92
ROC AUC	0.96	0.95

Table 8: Evaluation metrics of the hybrid classifier on both the train and test datasets

Class	Precision	Recall	F1 score	Support
Unaccountable	1.00	0.99	1.00	112
Accountable	0.87	0.91	0.89	44
Good	0.40	0.29	0.33	7
Very good	0.73	0.80	0.76	10

Table 9: Evaluation metrics of Tree 2 on the test dataset. Notice that the support for the class “good” is too low. Hence, the metrics for this class are not good

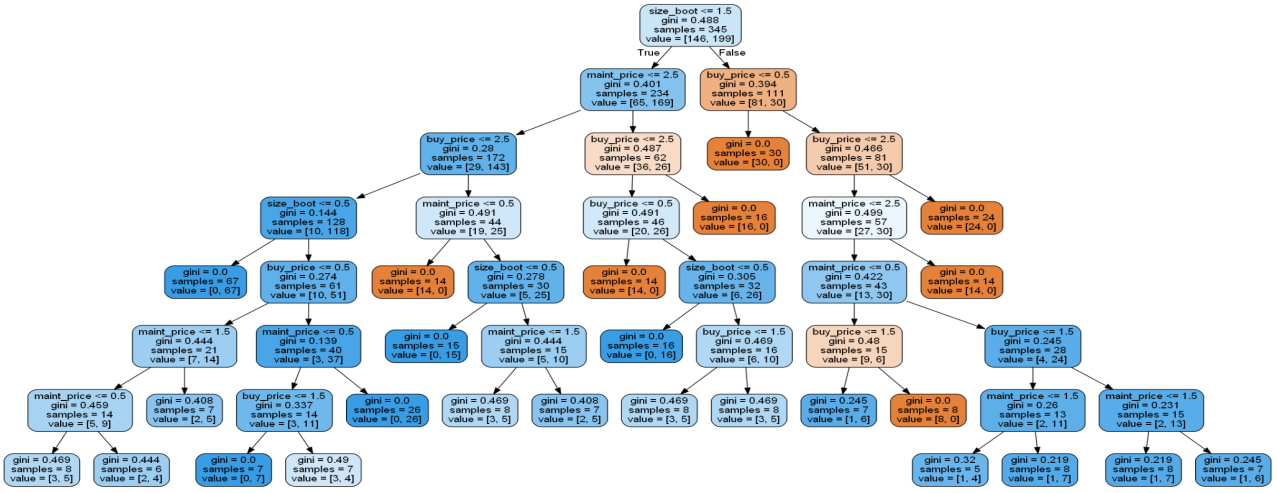

Figure 7: This is an example of an overfitted tree

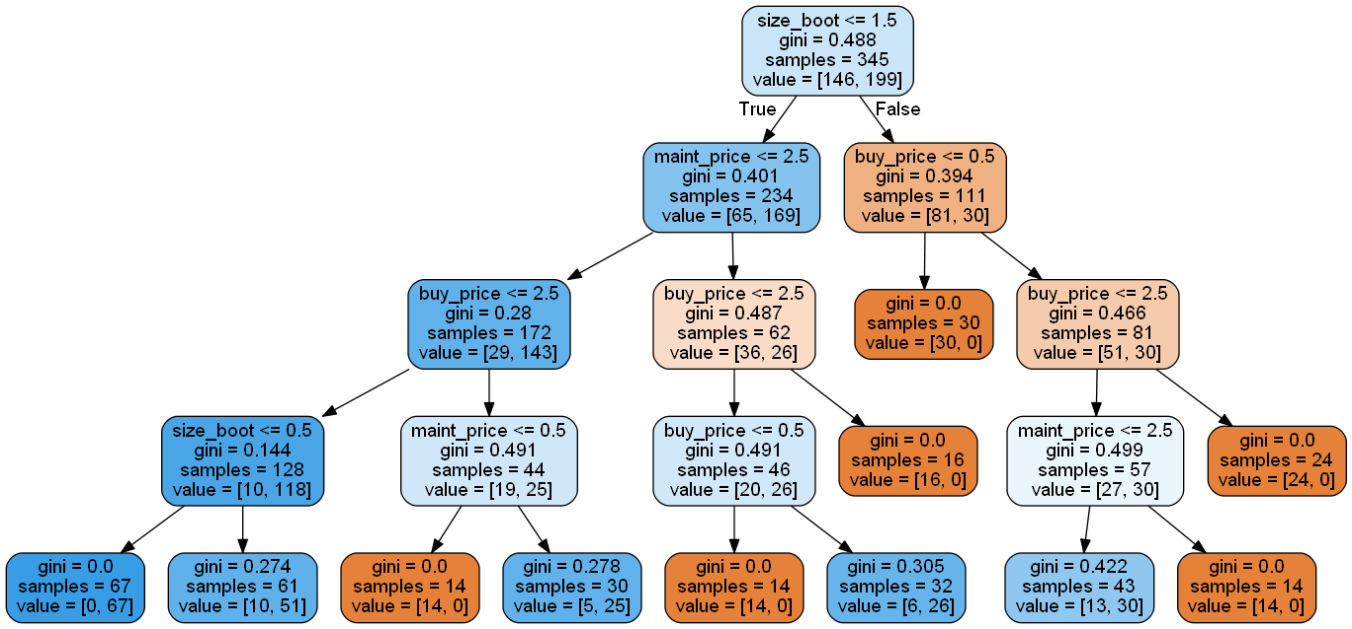

Figure 8: A visualization of Tree 1 (maximum depth is set to 4 prior to training to prevent overfitting)

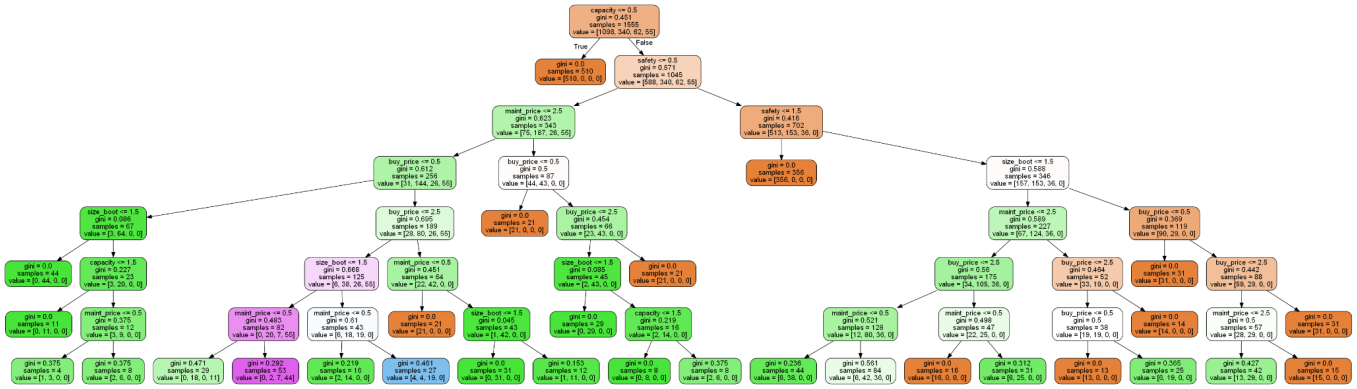

Figure 11: A visualization of Tree 2 (maximum depth is set to 7 prior to training to prevent overfitting)

Assignment 5: A Mathematical essay on Random Forest

Visha Rishi MK
Department of Chemical Engineering
IIT Madras
ch18b013@smail.iitm.ac.in

***Abstract*—Random forests or random decision forests are an ensemble learning method for classification, regression and other tasks that operate by constructing a multitude of decision trees at training time. Decision trees are non-parametric supervised learning methods that do not assume anything about the data distribution. Random forests are frequently used as "blackbox" models in businesses, as they generate reasonable predictions across a wide range of data while requiring little configuration. In this article, we tackle the problem of classifying cars based on their safety with the help of a random forest classifier.**

Keywords—Random Forest, Decision trees, supervised, non-parametric, classification, regression

I. Introduction

In many situations, the variables we encounter in data analysis are qualitative. For example, gender is qualitative, taking qualitative values male, female. Often qualitative variables are referred to as categorical. Approaches for predicting qualitative responses is known as classification. Predicting a qualitative response for an observation can be referred to as classifying that observation, since it involves assigning the observation to a category, or class. Classification is a fundamental issue in machine learning and data mining. In classification, the goal of a learning algorithm is to construct a classifier given a set of training examples with class labels. Often the methods used for classification first predict the probability of each of the categories of a qualitative variable, as the basis for making the classification. In this sense they also behave like regression methods. On the other hand, there are classifiers that use a distance-based decision function to assign classes for instances. There are many possible classification techniques, or classifiers, that one might use to predict a qualitative response. In this article, we discuss the Random forest approach for classification. We have a set of training observations that we can use to build the classifier. We want our classifier to perform well not only on the training data, but also on test observations that were not used to train the classifier.

In statistics and machine learning, ensemble methods use multiple learning algorithms to obtain better predictive performance than could be obtained from any of the constituent learning algorithms alone. Unlike a statistical ensemble in statistical mechanics, which is usually infinite, a machine learning ensemble consists of only a concrete finite set of alternative models, but typically allows for much more flexible structure to exist among those alternatives. Empirically, ensembles tend to yield better results when there is a significant diversity among the models. Many ensemble methods, therefore, seek to promote diversity among the models they combine. Although perhaps non-intuitive, more random algorithms can be used to produce a stronger ensemble than very deliberate algorithms. Random forests are nothing but an ensemble of decision trees. Decision trees are a non-parametric supervised learning method used for classification and regression. The goal is to create a model that predicts the value of a target variable by learning simple decision rules inferred from the data features. A tree can be seen as a piecewise constant approximation. Decision-tree learners can create over-complex trees that do not generalise the data well. They can be unstable because small variations in the data might result in a completely different tree being generated. This is called overfitting. Random decision forests correct for decision trees' habit of overfitting to their training set. Random forests generally outperform decision trees, but data characteristics can affect their performance. Also, the problem of learning an optimal decision tree is known to be NP-complete under several aspects of optimality and even for simple concepts. Consequently, practical decision-tree learning algorithms are based on heuristic algorithms such as the greedy algorithm where locally optimal decisions are made at each node. Such algorithms cannot guarantee to return the globally optimal decision tree. This can be mitigated by training multiple trees in an ensemble learner, where the features and samples are randomly sampled with replacement.

In this article, we want to classify cars based on their safety. Given that the data is obtained from a hierarchical decision model, we use random forests to outperform the simple decision tree classifier in terms of the variance and the bias.

Section 2 of this article gives a mathematical background to the random forest classifier. Section 3 of the article describes the data processing pipeline and several observations, insights, and visualisations. We also provide quantitative results after applying a random forest classifier to the train and test sets. Finally, section 4 draws conclusions from our explorations and analysis.

II. Random Forest Classifier

Decision tree classifier

Given training vectors $\boldsymbol{x_i} \in R^n$, $\boldsymbol{i = 1 \text{ to } n}$, and a label vector $\boldsymbol{y} \in R^l$, a decision tree recursively partitions the feature space such that the samples with the same labels or similar target values are grouped together. a decision tree

recursively partitions the feature space such that the samples with the same labels or similar target values are grouped together. Let the data at node ***m*** be represented by Q_m with N_m samples. For each candidate split $\theta \in (j, t_m)$, consisting of a feature ***j*** and threshold t_m, partition the data into ${Q_m}^{left}(\theta)$ and ${Q_m}^{right}(\theta)$ subsets

$${Q_m}^{left}(\theta) = \{(x, y) \mid x_j <= t_m\}$$

$${Q_m}^{right}(\theta) = Q_m \setminus {Q_m}^{left}(\theta)$$

The quality of a candidate split of node ***m*** is then computed using an impurity function or loss function ***H(),*** the choice of which depends on the task being solved (classification or regression)

$$G(Q_m, \theta) = \frac{{N_m}^{left}}{N_m} H({Q_m}^{left}(\theta)) + \frac{{N_m}^{right}}{N_m} H({Q_m}^{right}(\theta$$

We select the parameters that minimize the impurity,

$$\widehat{\theta} = arg\, min_\theta\, G(Q_m, \theta)$$

We recursively apply this procedure to the subsets ${Q_m}^{left}(\widehat{\theta})$ and ${Q_m}^{left}(\widehat{\theta})$ until the maximum allowable depth is reached or $N_m < min_{samples}$ or $N_m = 1$.

A. Classification Criterion

If a target is a classification outcome taking on values 0,1,…, K-1, for node ***m,*** let

$$p_{mk} = \frac{1}{N_m} \sum_{y \in Q_m} I(y = k)$$

be the proportion of class ***k*** observations in node ***m***. If ***m*** is a terminal node, the predicted probability for this region is set to p_{mk}. Common measures of impurity are the following

Gini impurity:

$$H(Q_m) = \sum_k p_{mk}(1 - p_{mk})$$

Entropy:

$$H(Q_m) = -\sum_k p_{mk}\, log(p_{mk})$$

Misclassification:

$$H(Q_m) = 1 - max(p_{mk})$$

B. Bootstrap aggregating

The training algorithm for random forests applies the general technique of bootstrap aggregating, or bagging, to tree learners. Given a training set $\boldsymbol{X = x_1, ..., x_n}$ with responses $\boldsymbol{Y = y_1, ..., y_n}$, bagging repeatedly (B times) selects a random sample with replacement of the training set and fits trees to these samples:

For b = 1, ..., B:

1. Sample, with replacement, n training examples from ***X, Y;*** call these $\boldsymbol{X_b, Y_b}$

2. Train a classification or regression tree $\boldsymbol{f_b}$ on $\boldsymbol{X_b, Y_b}$

After training, predictions for unseen samples ***x′*** can be made by averaging the predictions from all the individual regression trees on ***x′***:

$$\widehat{f} = \frac{1}{B} \sum_{b=1}^{B} f_b(x')$$

or by taking the majority vote in the case of classification trees. This bootstrapping procedure leads to better model performance because it decreases the variance of the model, without increasing the bias. This means that while the predictions of a single tree are highly sensitive to noise in its training set, the average of many trees is not, as long as the trees are not correlated. Simply training many trees on a single training set would give strongly correlated trees (or even the same tree many times, if the training algorithm is deterministic); bootstrap sampling is a way of de-correlating the trees by showing them different training sets. Additionally, an estimate of the uncertainty of the prediction can be made as to the standard deviation of the predictions from all the individual regression trees on ***x′***:

$$\sigma = \sqrt{\frac{\sum_{b=1}^{B} (f_b(x') - \widehat{f})^2}{B - 1}}$$

The number of samples or trees, B, is a free parameter. Typically, a few hundred to several thousand trees are used, depending on the size and nature of the training set. An optimal number of trees B can be found using cross-validation, or by observing the out-of-bag error (the mean prediction error on each training sample $\boldsymbol{x_i}$ using only the trees that did not have $\boldsymbol{x_i}$ in their bootstrap sample). The training and test error tends to level off after some number of trees have been fit.

III. Application to the Problem

The data that we have contains 1728 records and 6 features that describe the target variable. The target variable indicates car safety. The target variable consists of four classes - “unaccountable”, “accountable”, “good”, and “very good”. Hence, it qualifies as an ordinal qualitative variable. The six features are 1)Buying price, 2) Price of maintenance, 3) Number of doors, 4) Capacity of the car (in terms of the people to carry), 5) Size of the luggage boot, and 6) Estimated safety of the car.

A. Data exploration

The features are qualitative (ordinal) in nature. None of the features are numerical. Hence, we use ordinal encoding to transform the features for further analysis. Table 1 shows the ordinal encoding for all the features. Also, none of the

features have missing values. Table 2 shows the number of data points (cars) in each class. This shows an imbalance in the number of data points between the classes. Most of the cars in the dataset are being classified as "unaccountable". Class imbalance can lead to trees that are biased towards the majority class (ie. "unaccountable"). Hence we group the classes "accountable", "good", and "very good" into a single class - "accountable". This turns our problem into a binary classification problem. Since the original target variable is ordinal, a multiclass classification strategy might be inappropriate. Hence, working with a binary target variable seems relevant for classification algorithms (binary classification). We should make sure that we group similar classes into a single class. In our case, it makes sense to combine "accountable", "good", and "very good" into a single class. In our analysis, we have used "accountable" as the positive class and "unaccountable" as the negative class. Once we frame a binary classification problem, we mitigate the problem of class imbalance in the data (29.9% of the data points belong to the positive class).

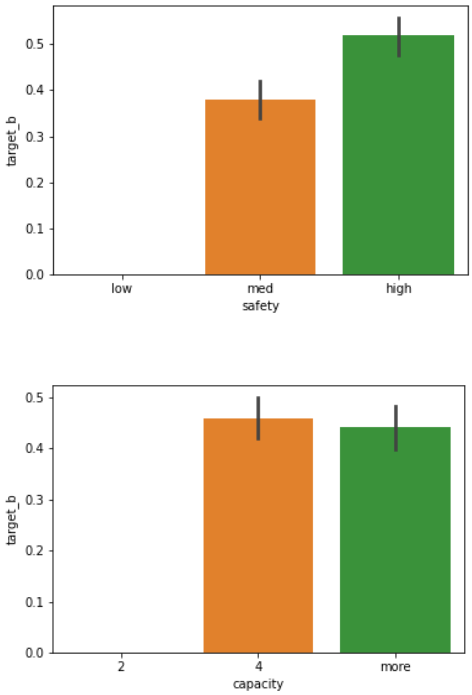

Figure 1: Barplots between the target variable and "Estimated safety of the car" (top) and the "Capacity of the car" (bottom). target_b represents the probability of belonging to the positive class

B. *Preliminary feature selection*

Table 3 shows the mutual information between each feature and the transformed target variable. We can see that the variables "Capacity of the car" and the "Estimated safety of the car" have the highest mutual information. We work with these variables in our analysis. Figure 1 shows the bar plots between the target variable and "Capacity of the car" and the "Estimated safety of the car". We can see that when the estimated safety is low or the capacity of the car is 2, all the cars belong to the negative class (all the cars are "unaccountable"). Also, when the capacity of the car increases, the probability of the car belonging to the positive class increases. A similar trend is also seen with the variable "Estimated safety of the car". We also want to see the combined effect of these variables on the target variable. Figure 2 shows the barplot between the target variable and the car capacity conditioned on the estimated safety of the car. The patterns that we discussed before are visible in this plot as well. This shows that the capacity and estimated safety of a car are good indicators of the target variable.

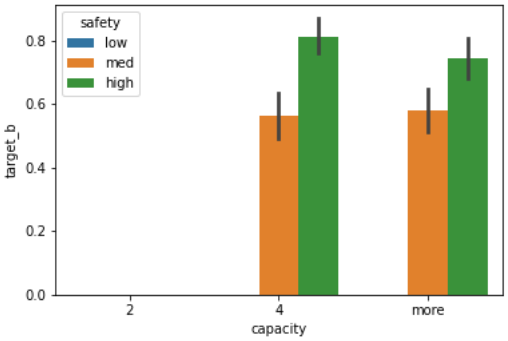

Figure 2: Barplot between the target variable and the car capacity conditioned on the estimated safety of the car. target_b represents the probability of belonging to the positive class

We can infer that the features "Buying price", and "Price of maintenance" also have a significant effect on the target, and hence we also include them in our analysis. Since the variables "Size of the luggage boot", and "Number of doors" are insignificant (low mutual information), we discard them from our analysis. Figure 6 shows the barplot between the target variable and the buying price, conditioned on the price of maintenance. It seems to suggest that high costs do not guarantee safety. This observation is a bit surprising as we would expect costly cars to have better safety features. Particularly, when the buying price and the price of maintenance are either "low" or "medium", almost all the cars belong to the positive class ("accountable").

C. *Random Forest Classifier*

The data is split into train and test datasets. The train data has 1555 data points and the test dataset has 173 data points. While performing the train-test split, we ensure that the class split remains the same in the train and test data. This is to ensure that the test data is representative of the train data. We have the decision tree classifier as our baseline model. We train a decision tree classifier on the training dataset with 10-fold cross-validation. The cross-validation scores yielded an average ROC AUC score of 0.97 (with a standard deviation of 0.009). Table 4 shows the evaluation metrics achieved by the baseline on both the train and test datasets. Figure 3 shows the ROC curve of the baseline classifier. We now train a random forest classifier on our training data. To start with, we need to determine the number of estimators or decision trees in the forest. Hence, we plot the out-of-bag evaluation score against the number of estimators. With bagging, some instances may be

sampled several times for any given predictor, while others may not be sampled at all. This means that only about 63% of the training instances are sampled on average for each estimator. The remaining 37% of the training instances that are not sampled are called out-of-bag instances. Since a predictor never sees them during training, out-of-bag samples can be evaluated without the need for a separate validation set. The ensemble is evaluated by averaging the out-of-bag evaluation scores for each estimator. They give us an idea about the generalization capabilities of the ensemble. Figure 4 shows the plot of out-of-bag scores against the number of estimators. We can see that the out-of-bag score is the highest when the number of estimators is set to 400. Hence we train a random forest classifier with 400 decision trees. Table 5 shows the evaluation metrics achieved by the random forest on both the train and test datasets. Figure 5 shows the ROC curve of the random forest classifier. We can see that the random forest achieves a better F1 score while retaining the ROC AUC score compared to the decision tree classifier.

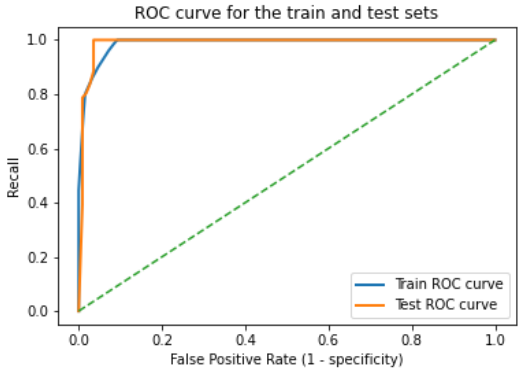

Figure 3: ROC curve of the Baseline classifier (Test ROC AUC: 0.988)

Since the dataset was obtained from a simple hierarchical decision model, we cannot clearly see the advantages of the random forest classifier. In this case, both models seem to have nearly the same generalization power. One of the advantages of using a random forest is that we can obtain the bootstrapped probabilities for a particular instance. From these bootstrapped probabilities, we could estimate the bias, variance, and confidence intervals for our predictions. In the case of decision trees, we need to employ resampling with or without replacement as a post-processing step to obtain the confidence bounds. Figure 7 shows the bootstrapped probabilities obtained from the random forest classifier along with the bias and variance of the estimates. One particular disadvantage of random forests is that they are non-interpretable, unlike decision trees. We look into this problem of interpretability in the upcoming sections.

D. Mean Decrease in Impurity

A great quality of random forests is that they make it easy to compute the relative importance of every feature. Feature importance can be measured by looking at how much the tree nodes that use that feature reduce impurity on average (across all trees in the forest). More precisely, it is a weighted average where each node's weight is equal to the number of training samples that are associated with it. Scikit-Learn computes this score automatically for each feature after training. The importance is scaled so that they sum up to one. Figure 8 shows the relative feature importance. It is not surprising that the estimated safety of the car occupies the top position followed by the capacity of the car, buying, and maintenance price of the car. This is the order we would get even if we rank features based on the mutual information with the target variable. Hence, we can be confident with our analysis that the random forest has correctly captured the patterns in the data without overfitting.

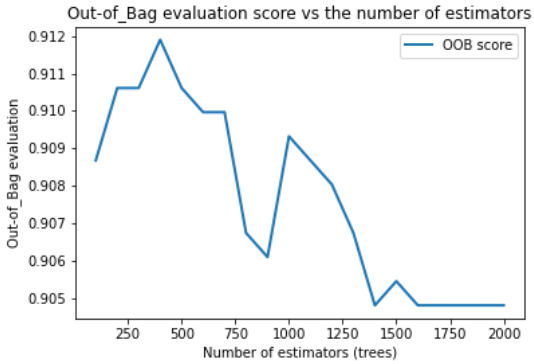

Figure 4: The plot of out-of-bag scores against the number of trees in the forest. Peak is attained when the number of trees is 400

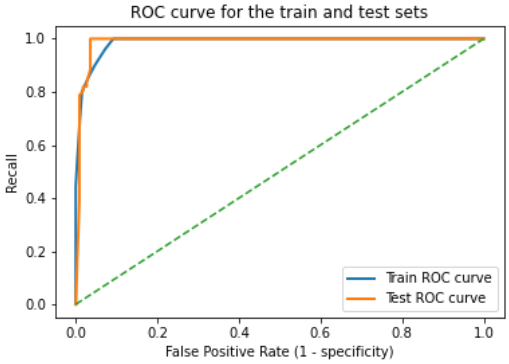

Figure 5: ROC curve of the random forest classifier (Test ROC AUC: 0.988)

E. Integration with Logistic Regression

Since random forests can be hard to interpret, we need to integrate them with a much simpler model whose parameters are easy to understand and intuitive. One of the models that come to our mind is the logistic regression model. For the purpose of integration, we transform the original data to a higher dimensional space with the help of the random forest classifier. We have a random forest with 400 trees that are trained on the training data. Then each leaf of each tree in the ensemble is assigned a fixed arbitrary feature index in a new feature space. Each instance of the training data, when passed through a tree, would end up on a leaf node with a particular index. Since we have 400 trees, we would then end up with 400 indices for the training

instance. With this, we transform the original data into a higher dimensional feature space. These are called random forest embeddings. We use one-hot encoding and train a logistic regression model on it. Clearly, the number of parameters in the model would be very high. This might call for regularization. We have employed lasso regression which performs feature selection. Table 6 shows the evaluation metrics achieved by the logistic regression model on both the train and test datasets. Figure 9 shows the ROC curve of the logistic regression classifier. In this case, all three models perform equally well and we cannot pick a single model as the best one. The baseline model is highly explainable, unlike the random forest. Taking explainability, simplicity, and performance into account, we pick the decision tree classifier as the best choice for analyzing this dataset. However, the logistic regression model has a lot of parameters (greater than 20,000) whose interpretability should be a topic for further research. It has the potential to deliver interesting insights into the data.

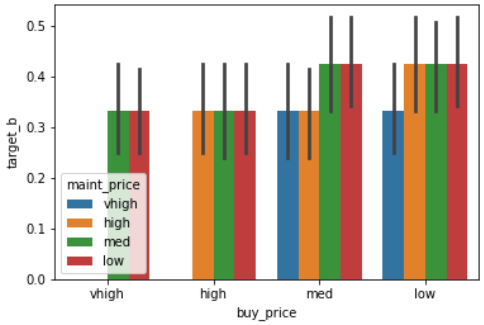

Figure 6: Barplot between the target variable and the buying price, conditioned on the price of maintenance

F. Multiclass classification

After analyzing the results for binary classification with random forests, we analyze the results for multiclass classification in this section. In our case, we have four classes - “unaccountable”, “accountable”, “good”, and “very good”. Hence, the target qualifies as an ordinal qualitative variable. We train a random forest classifier (model 1) with buying price, price of maintenance, size of luggage boot, estimated safety, and the capacity of the car as features. The number of decision trees in the forest is determined by plotting the out-of-bag evaluation score against the number of estimators. Since the curve was found to be increasing, we chose to train the forest with 2000 decision trees. From model 1, we obtain the random forest embeddings which are used to train a logistic regression classifier (model 2). Tables 7 and 8 show the evaluation metrics for both the models in the test dataset. We can observe that both the models are able to distinguish between the “unaccountable”, and “accountable” classes. But they face difficulty in differentiating the “good”, and “very good” classes. This is mainly due to the underrepresentation of these classes in the dataset. Hence, we conclude that a 4-class classification strategy seems redundant. As an improvement to the binary classification strategy, we propose a 3-class classification strategy in which the classes “good”, and “very good” are combined. This would make the analysis more meaningful and efficient.

IV. Conclusion

In this article, we discussed the mathematical formulation behind random forest classifiers and applied them to the problem of the classification of cars based on safety. Some of the key insights that we obtained are listed below:

1. Most of the cars that have high buying prices and high prices of maintenance belong to the negative class (“unaccountable”). This shows that costly cars are not worthy when it comes to safety.
2. All the cars that fall under the “low” estimated safety category belong to the negative class (“unaccountable”).
3. All the cars that have a capacity of 2 belong to the negative class (“unaccountable”).
4. In this particular dataset, the random forest classifier does not offer any clear advantage compared to the decision tree classifier. Being very simple and highly explainable, decision trees would be the best option to work with this data.
5. In the multiclass classification approach, we cannot clearly separate the classes “good”, and “very good” because of their underrepresentation. To overcome this shortcoming, we propose to work with three classes - “unaccountable”, “accountable”, and “good”

In this article, we have predominantly approached this problem as a binary classification task. An area of research would be to bring several ranking-based metrics for multiclass classification, apart from the traditional classification metrics, into the picture. We have to check whether random forest classifiers provide any advantage over decision trees in the multi-class classification task. Also, we have tried to interpret random forest classifiers using random forest embeddings. Interpreting the coefficients of the logistic regression model might give us interesting insights into the data.

References

[1] H. Patel, P. Prajapati, “Study and Analysis of Decision Tree-Based Classification Algorithms”, International Journal of Computer Sciences And Engineering, 6(10):74-78, 2018

[2] J. Ali, N. Ahmad, I. Maqsood, R. Khan, “Random Forests and Decision Trees”, International Journal of Computer Sciences And Engineering, 2012

Buying price		Price of maintenance		Number of doors		Capacity of car		Size of the luggage boot		Estimated safety	
Feature	Value	Feature	Value	Feature	Value	Feature	Value	Feature	Value	Feature	Value
high	0	high	0	2	0	2	0	big	0	high	0
low	1	low	1	3	1	4	1	medium	1	low	1
medium	2	medium	2	4	2	more	2	small	2	medium	2
vhigh	3	vhigh	3	5more	3	---	---	---	---	---	---

Table 1: Ordinal Encoding of all the features (vhigh = very high)

Class	Number of data points
Unaccountable	1210
Accountable	384
Good	69
Very good	65

Table 2: Number of data points in each class. We can clearly see the class imbalance in the data

	Buying price	Price of maintenance	Number of doors	Capacity	Size of luggage booty	Estimated Safety
Mutual Information	0.06	0.05	0.003	0.15	0.02	0.18

Table 3: Mutual information between each feature and the transformed binary target variable

	Train set	Test set
Accuracy	0.93	0.92
Precision	0.88	0.94
Recall	0.89	0.83
F1 score	0.89	0.88
ROC AUC	0.98	0.98

Table 4: Evaluation metrics of the baseline on both the train and test datasets

	Train set	Test set
Accuracy	0.93	0.93
Precision	0.85	0.93
Recall	0.95	0.88
F1 score	0.89	0.91
ROC AUC	0.98	0.98

Table 5: Evaluation metrics of the random forest classifier on both the train and test datasets

	Train set	Test set
Accuracy	0.93	0.92
Precision	0.88	0.94
Recall	0.91	0.83
F1 score	0.89	0.88
ROC AUC	0.98	0.98

Table 6: Evaluation metrics of the logistic regression classifier on both the train and test datasets

Class	Precision	Recall	F1 score	Support
Unaccountable	0.97	0.99	0.98	112
Accountable	0.92	0.80	0.85	44
Good	0.50	0.71	0.59	7
Very good	0.73	0.80	0.76	10

Table 7: Evaluation metrics of Model 1 on the test dataset. Notice that the support for the class “good” is too low. Hence, the metrics for this class are not good

Class	Precision	Recall	F1 score	Support
Unaccountable	0.97	0.99	0.98	112
Accountable	0.90	0.86	0.88	44
Good	0.38	0.43	0.40	7
Very good	0.78	0.70	0.74	10

Table 8: Evaluation metrics of Model 2 on the test dataset. Notice that the support for the class “good” is too low. Hence, the metrics for this class are not good

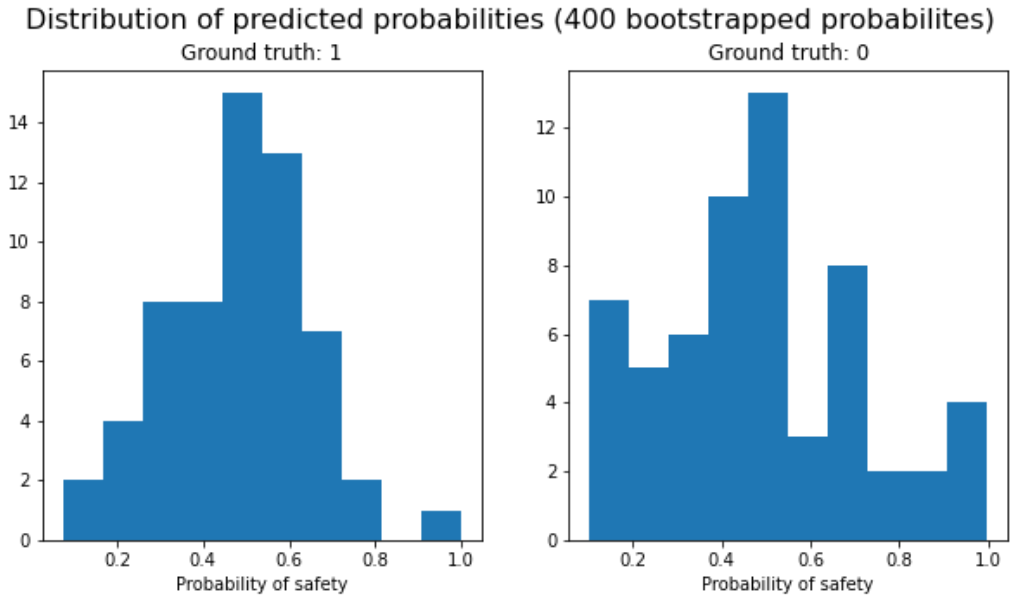

Figure 7: Bootstrapped probabilities computed using the random forest classifier for two samples from the test dataset. Bias and standard deviation in sample 1 (left): -0.51, 0.16 respectively. Bias and standard deviation in sample 2 (right): 0.48, 0.22 respectively

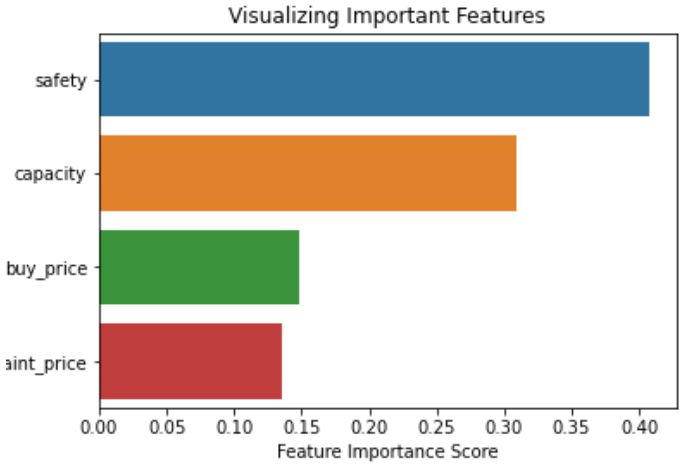

Figure 8: Relative feature importance computed using the Mean Decrease in Impurity (MDI)

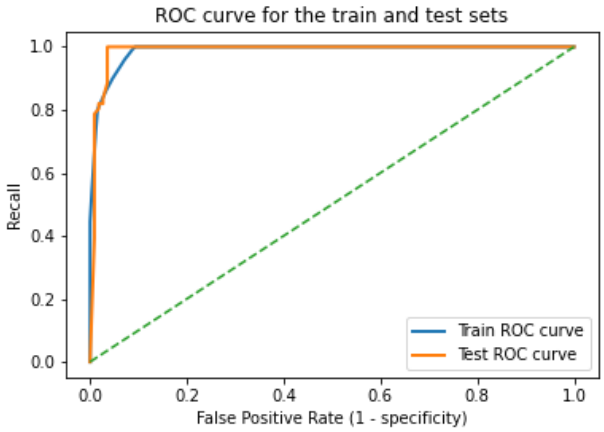

Figure 9: ROC curve of the logistic regression classifier
(Test ROC AUC: 0.989)