ISYE 7406 Project Presentation

ONLINE SHOPPING PURCHASE INTENT

(Group 5) Casey Stanfield, Jia Hui Mo, Mariah Carter, Simon Yee, Xu Anne Zhang

Project Overview

- ➔ Our project is designed to predict a binary dependent variable **revenue** indicating whether or not there was a sale based on various **purchase intent factors**, such as the number of pages visited per category, visitor type, information about the region, browser type, whether the visit was during a weekend, if the visit is close to a holiday, the month of the visit, and so forth
- ➔ We also used **web page categories** as a predictive factor, classifying them as informative, administrative, or product related
- ➔ The focus of this project is to **infer the relationship** between revenue and these predictor variables. We used a number of different binary classification methods to predict revenue and compare these different methods with one another

Research Questions

- We set out to address various questions about our dataset over the course of our project, but with a **differentiated approach** compared to previous analysis in academic literature
  - *There are several past articles and academic papers in relation to this dataset that showcase the use of advanced Neural Networks (NN), ensemble methods such as Random Forest (RF), and dimensionality reduction techniques such as Principal Component Analysis (PCA)*
- We focused our analysis on applying **different modeling and data mining techniques** to attempt to provide an alternative to Random Forests.
  - *We wanted to find an approach that achieves a similar accuracy level and more interpretability than NN or PCA*
- Our **key research questions** going into this project were:
  - *Are the uses of certain models like Random Forest across multiple research papers reasonable with our data set?*
  - *Do their results make sense?*
  - *How do other models respond?*
  - *How do the variables interact quantitatively and qualitatively?*

Exploratory Data Analysis

Exploratory Data Analysis - Data Overview

- We used the "**Online Shoppers Purchasing Intention**" dataset from the UCI Machine Learning Repository for our project [link to dataset here]. This dataset includes 17 independent or prediction variables covering a range of behavioral markers of online shoppers and an indicator variable (binary) for revenue earned. The dataset has a total of 12,330 rows. The revenue indicator variable is largely left leaning, towards the FALSE, or no revenue generated side. The vast majority of the predictor variables are not normally distributed but instead have a left skewed (towards 0) distribution - this includes variables such as duration on the website, how many related products there are, information distribution, and page values
  - *Page Value is a Google Analytics feature which measures the "value" of how likely a visitor is to make a purchase on a website - unsurprisingly, this is the most correlated variable with revenue earned*
  - *Any value for this Page Value feature other than 0 is an outlier as an indicator for revenue*
- The timing aspect of this dataset was interesting - while we would expect the month of November to be a strong month for revenue with holiday shopping, we found that **May was the most common month for purchases**, which was somewhat surprising.

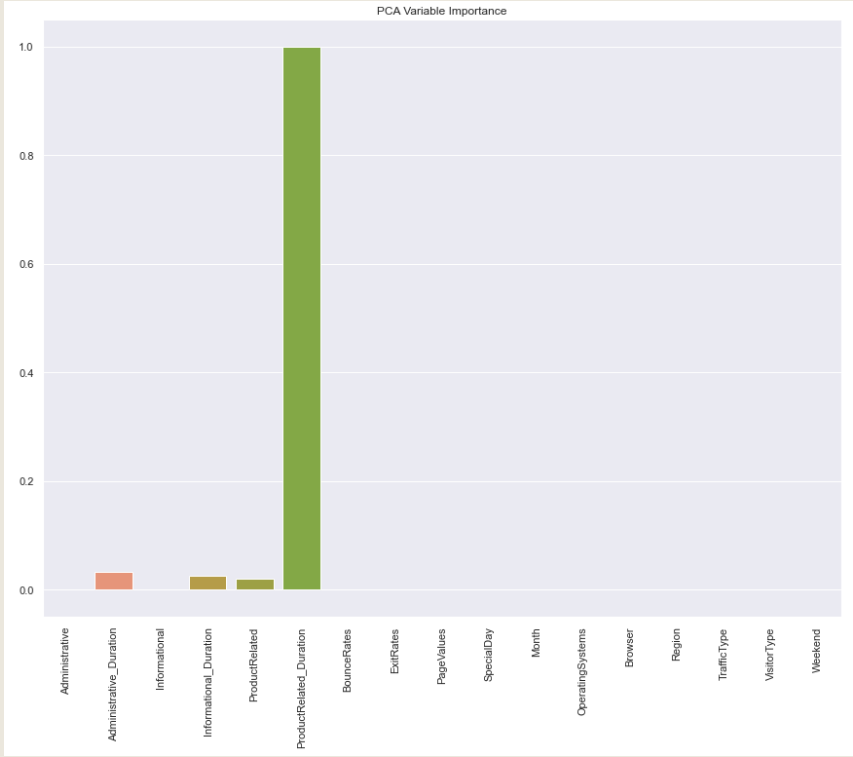
	Administrative	Administrative_Duration	Informational	Informational_Duration	ProductRelated	ProductRelated_Duration	BounceRates	ExitRates	PageValues	SpecialDay	Month	OperatingSystems	Browser	Region	TrafficType	Weekend	Revenue
Administrative																	
Administrative_Duration	0.602																
Informational	0.377	0.303															
Informational_Duration	0.256	0.238	0.619														
ProductRelated	0.431	0.289	0.374	0.280													
ProductRelated_Duration	0.374	0.355	0.388	0.347	0.861												
BounceRates	-0.224	-0.144	-0.116	-0.074	-0.205	-0.185											
ExitRates	-0.316	-0.206	-0.164	-0.105	-0.293	-0.252	0.913										
PageValues	0.099	0.068	0.049	0.031	0.056	0.053	-0.119	-0.174									
SpecialDay	-0.095	-0.073	-0.048	-0.031	-0.024	-0.036	0.073	0.102	-0.064								
Month	0.097	0.058	0.063	0.044	0.156	0.138	-0.067	-0.095	0.067	-0.257							
OperatingSystems	-0.006	-0.007	-0.010	-0.010	0.004	0.003	0.024	0.015	0.019	0.013	0.038						
Browser	-0.025	-0.015	-0.038	-0.019	-0.013	-0.007	-0.016	-0.004	0.046	0.003	0.020	0.223					
Region	-0.005	-0.006	-0.029	-0.027	-0.038	-0.033	-0.006	-0.009	0.011	-0.016	0.024	0.077	0.097				
TrafficType	-0.034	-0.014	-0.034	-0.025	-0.043	-0.036	0.078	0.079	0.013	0.052	0.055	0.189	0.112	0.048			
Weekend	0.026	0.015	0.036	0.024	0.016	0.007	-0.047	-0.063	0.012	-0.017	0.017	0.000	-0.040	-0.001	-0.002		
Revenue	0.139	0.094	0.095	0.070	0.159	0.152	-0.151	-0.207	0.493	-0.082	0.127	-0.015	0.024	-0.012	-0.005	0.029	

Exploratory Data Analysis - Data Exploration

We found that revenue was the most correlated with a high value for the variable **PageValues**

Exploratory Data Analysis – Data Variance

- ➔ We used a PCA analysis to explore the variance explained by each prediction variable as part of our data exploration
- ➔ We found that the first principal component was overwhelmingly responsible for most of the variance, nearly 100%. The explanatory variable ProductRelatedDuration was by far the most important and influential variable in the first principal component
- ➔ Interestingly, an unsupervised method in a low dimension setting like PCA showed us that most of the variance could be explained by ProductRelatedDuration but it could not help differentiate between revenue earned or not, which is where boosting would



Model Methods, Results, & Findings

Partial Least Squares

A PCA method was built prior to the PLS method, and the results indicated to us that the first component is able to explain nearly all of the variance, but is unable to demonstrate any meaningful relationships between those who made purchases and those who did not

We chose PLS because we wanted to understand and infer the relationship between the independent and dependent variable under a reduced dimension

PLS was used during the exploratory data analysis step as a means to develop an explainable relationship between variables

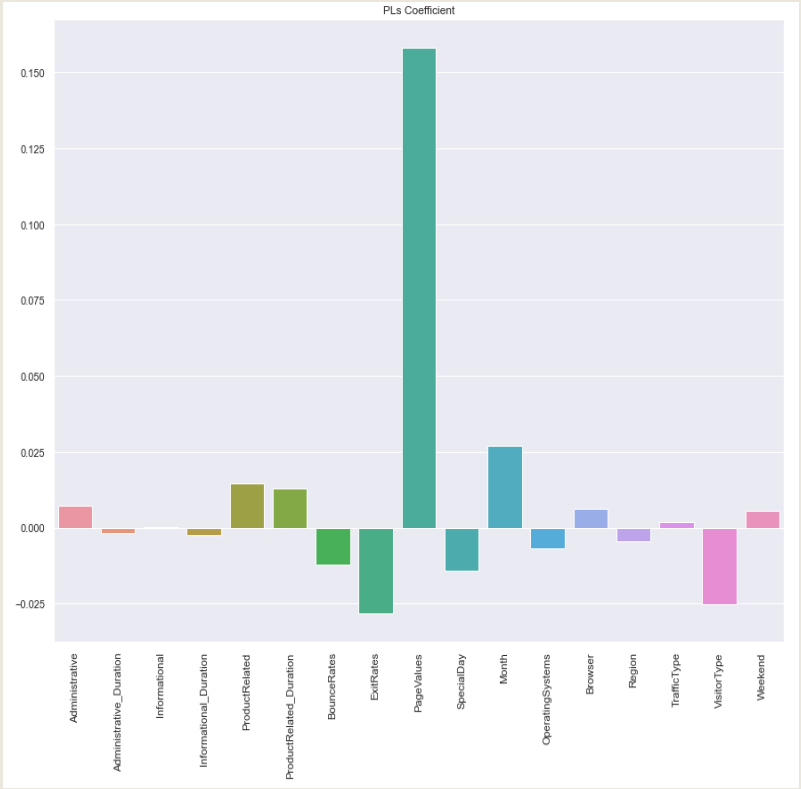
PLS

Results

- PageValues was found to be the most significant coefficient by far
- Month, VisitorType, and ExitRate share similar coefficient numbers, with the latter two being negative

Findings

- PageValues represents the average value for a web page that a user visited before completing an e-commerce transaction, and from a qualitative perspective we do not find that surprising
- There are many reasons for a large visit: some of these may include comparing prices, comparing similar items, and reading reviews before completing a purchase
- The ExitRate variable suggests that, unsurprisingly, those who leave your website are unlikely to make a purchase
- The VisitorType is surprising because it suggests that returning customers are less likely to make a purchase. But we hypothesize this could also be a returning customer leaving a review



LASSO

- One of our primary goals is to understand why complex methodologies like Random Forest and Boosting are commonly used for this data set. LASSO was used during the data analysis as a method to understand variable importance
- We specifically chose LASSO for this because we wanted to understand what happens when we minimize/penalize other coefficients and try to choose one strong predictor
- We ran a Monte Carlo cross validation with an 80/20 training and testing split 1000 times with a changing alpha parameter value. We then took the average R-squared and coefficients of each run

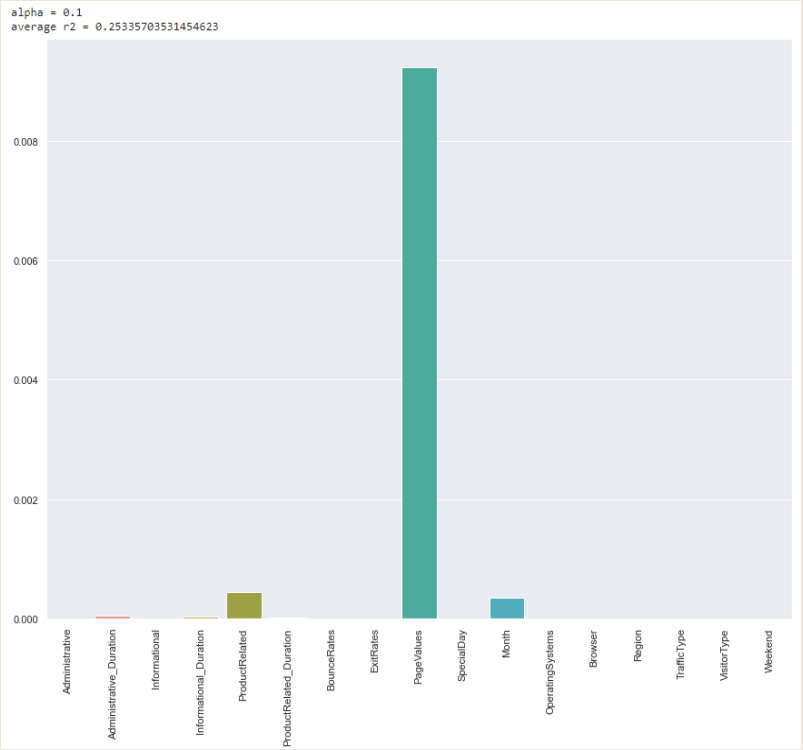
LASSO

Results

- While marginally, models with higher alphas tended to do worse. A larger alpha assigned a larger penalty term. We tested alpha values from 0.1 to 1 in increments of 0.1

Findings

- More variables helped classify the revenue variable beyond just the explanatory power of the PageValues variable
- All average R-squared are quite bad, averaging between .24 and .25 across all alphas, indicating that the relationships between the variables are NOT linear
- The lowest alpha (.1) performing the best, and a poor R-squared tells us that ensemble methods with “layers” tend to perform better because at a simple level, there is a lot of overlap between those who did and did not make purchases, and that there are very specific thresholds or variables that help make that differentiation



Discriminant Analysis

LDA is very similar to PCA as they both look for linear combinations of the features which best explain the data. The main difference is that LDA is a supervised dimension reduction technique that also achieves classification of data simultaneously

The aim of LDA is to find features that maximize the separation between two groups, meaning that the class group mean is placed as far as possible to ensure high confidence during prediction

We ran a Monte Carlo cross validation with an 80/20 training and testing split 100 times and compared the results for LDA and QDA

Discriminant Analysis

Results

- We found these variables have significant difference in group mean: Administrative Duration, Informational Duration, Product Related, ProductRelated_Duration, and PageValues
- When we fitted a model with all these variable above, model accuracy improved
- Compared with QDA, a linear classifier is more suitable for our dataset since it has higher model accuracy
- From our model output, we can see the separation between these two groups are quite close with lots of overlapping

Findings

Customers who made purchase have a page value of 27.3364 on average. For customers who made purchases, the amount of time spent on an administrative page (eg. login, entering shipping/billing info), specific product page and product category page (looking for similar products) are significantly higher than customers who did not make purchases

- Customers who did not make purchases have higher exit rate than customer who made purchases
- Customers who did not make purchases have higher bounce rate than customer who made purchases (bounce rate is the % of visitors who enter the website through a page and exit without any additional actions)
- Surprisingly, those variables that have significant difference in group means do not have large coefficient values

Logistic Regression

- ➔ Logistics regression is a popular algorithm to solve classification problems. It learns a linear relationship from the given dataset and introduces a non-linearity in the form of the Sigmond function. Logistic regression is easy to implement, interpret and very efficient to train
- ➔ Our dataset was a very imbalanced dataset, where about 85% of cases were FALSE and only 15% were TRUE for Revenue
- ➔ We were able to overcome the imbalanced data by using the R package ROSE. We have approximately an equal number of observations in the training dataset to build the model

Logistic Regression

Results

- By performed 10-fold cross validation for the full model and reduced model, the AUC values suggested that our model has been improved by stepwise variable selection in both directions by measuring AIC. In addition, testing for a subset of coefficients supports that Administrative_Duration, Informational_Duration, Region are not significant in making predictions
- The result of the Goodness of Fit (GOF) test suggested that the reduced model is not an adequate fit for the dataset. GOF does not guarantee good predictions and it does not mean that our independent variables are not good predictors of our dependent variable
- PageValue, exitRate, productRelated_Duration, month, and traffic type remain the most important variables in classifying Revenue, after performing variable selection

Findings

- The decision boundary helps to differentiate probability into positive class and negative class. Our final model has the highest accuracy when misclassification threshold reaches 0.75 and then it starts to decrease
- Even though we have a more accurate model, it resulted in more false positive cases, which could result in loss of revenue
- People are more likely to make purchases in the months of May, March, December based on the reduced model

Naive Bayes

We chose to include this model with 2 algorithms, Gaussian and Bernoulli, to test the effectiveness on the dataset against more standard classifications such as Logistic Regression and alternative ensemble methods such as XGBoost and AdaBoost

Naive Bayes models have strong assumptions including independence between each predictor and that all predictors contribute equally to the outcome

Despite these conditions not being met, some surveys of supervised classification models have shown that the model may still have strong predictive power on smaller datasets

To attempt to overcome some of the issues with the unmet assumptions, 3 subsets of data were tested. The first subest removed all of the "duration" predictors which were correlated directly to the page view e.g. "ProductRelated" and "ProductRelated_Duration". The second subset involved removing some interdependent variables including the dummy variables created for the "Visitor Types." Finally, all other interdependent variables were removed including all page view related variables and the "ExitRates" variable which is correlated to the "PageValues" predictor.

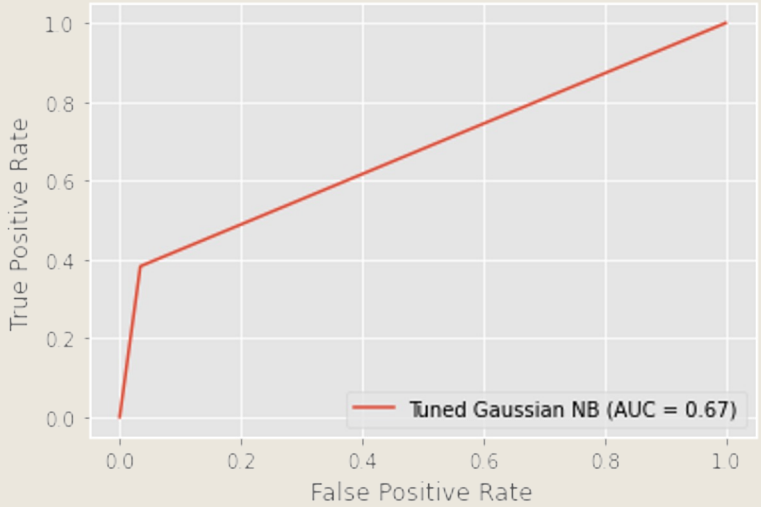
Naive Bayes

Results

- On the larger datasets the Gaussian Naive Bayes model performed slightly better than the Bernoulli model. Both models had increased accuracy up to roughly 87% with the smaller models, however the Bernoulli model misclassified the minority class to achieve that rate while Gaussian did not

Findings

- Naive Bayes, on this dataset, does perform roughly on par with more common models such as Logistic Regression without any tuning or data subseting
- Removing variables which were found to not have much predictive power in our EDA increased the performance of all Naive Bayes models tested with all permutations of parameter tuning



XGBoost

XGBoost is an open-source library that provides a regularizing gradient boost framework. This has been a very popular model implementation for variety of problems due to its flexibility and speed

The model achieves its speed and performance by using Newton Boosting and a unique approach to common boosting and tree based tuning such as shrinking leaf nodes, penalizating trees, and automatic feature selection

We included this model as an alternative to the AdaBoost model to attempt to achieve a better model with less training time

XGBoost

Results

- XGboost achieved an accuracy rate of roughly 88% with default parameters on the full dataset - this matched some of the other model performances after tuning them
- After tuning the number of estimators, cross trees sampling, and maximum tree depth parameters, an accuracy of 90% was achieved. This was done without the same degree of misclassification of the minority class as the Naive Bayes Bernoulli model

Findings

- XGBoost does require significantly less training time than the AdaBoosting model, roughly one third to one half of the time
- This model performs better on the full dataset with all columns, including those that are correlated with one another, than with any of the other predictor subsets
- XGBoost had a 5% worse accuracy in cross-validation and test sets when using oversampling to address class imbalance in the dependent variable Revenue

AdaBoost

- Adaptive Boosting (AdaBoost) is a common boosting algorithm that uses several poor performing, "weak learners", together to arrive at a final model with better performance
- AdaBoost can been done with a variety of base models as the classifier from Logistic Regression to SVC to decision trees though decision trees are considered the "base" version of most AdaBoost implementations
- We tested Logistic Regression, SVC, and decision trees as the base models for AdaBoost but only tuned the decision trees as all others were extremely poor performers

AdaBoost

Results

- AdaBoosting with a decision tree as the base model achieved an accuracy close to that of the XGBoost model at roughly 87%
- AdaBoost took 3-4 times as long to train as the XGBoost model and 5 times longer than the Naive Bayes model

Findings

- For this dataset AdaBoost did not perform very well, failing to achieve the same accuracy as the Logistic Regression once it had been tuned
- AdaBoost often suffers in performance from exploring extreme predictors in observations and using those as important features of the dataset
- The untuned model on the full dataset had an accuracy of 16% from attempting to assign all observations to the minority (Revenue TRUE)
- The accuracy of the model was only changed by 1% when using oversampling to address the class imbalance of the dependent variable Revenue

Concluding Thoughts

Our dataset seemed like it would lend itself to oversampling because of the strong class imbalance, however our models performed significantly worse when they were oversampled, which was somewhat surprising

We ran over half a dozen different models on the dataset to predict Revenue (a binary variable for revenue earned or not)

XGBoost had the best performance of the models we tested, nearly 90% when it was tuned

- Interestingly, that accuracy was achieved without the level of misclassification of minority classes seen in other models like Naive Bayes and Bernoulli
- The XGBoost model also took significantly less time to train compared to other complicated models such as AdaBoost