**Car Purchase Decision**

Saaki Vishnumolakala, [svishnumolakala@bellarmine.edu](mailto:student@bellarmine.edu)

**ABSTRACT**

Up to 150 word summary of your project.

1. **INTRODUCTION**

I found a data set on Kaggle. It is a dataset of customers’ car purchases based on their salaries. I used 4 ML regression models. Logistic regression, SVC, random forest, and decision tree through the voting classifier.

1. **BACKGROUND**
   1. *Data Set Description*

This dataset contains details of 1000 customers who intend to buy a car, considering their annual salaries. It is a purchase decision data set, indicating whether or not a client bought a car. It can be found on Kaggle as mentioned above. It has a usability rating of 10.0 and the dataset was made by for study purposes by a data scientist.

* 1. *Machine Learning Model*

[Logistic regression](https://www.statisticssolutions.com/free-resources/directory-of-statistical-analyses/logistic-regression/) is the appropriate regression analysis to conduct when the dependent variable is binary.  Logistic regression is a predictive analysis.  Logistic regression is used to describe data and to explain the relationship between one dependent binary variable and one or more independent variables.

Support Vector Regression is a supervised learning algorithm that is used to predict discrete values. Support Vector Regression uses the same principle as the SVMs. The basic idea behind SVR is to find the best fit line. In SVR, the best fit line is the hyperplane that has the maximum number of points.

Decision trees are a type of model used for both classification and regression. Trees answer sequential questions which send us down a certain route of the tree given the answer. The model behaves with “if this than that” conditions ultimately yielding a specific result. When a tree is deep, it is more prone to overfitting. That is why I have my max\_depth = 3 in my dataset.

A random forest is simply a collection of decision trees whose results are aggregated into one result. Their ability to limit overfitting without increasing error by too much due to bias is why they are such powerful models. Random Forests reduce variance is by training on different samples of the data.

1. **EXPLORATORY ANALYSIS**

This dataset has 1000 entries with 5 columns. There was equal entries of male and female. I did not really have to clean the data because there were no missing values. I didn’t have to stratify the ‘Purchased’ column either because it was not too big of a difference between the multiple answers of no and yes.

Chart, bar chart

Description automatically generated 0 = No; 1 = Yes

**Table 1: Data Types**

|  |  |
| --- | --- |
| *Variable Name* | *Data Type* |
| V1 – User ID | Integer |
| V2 – Gender | Object |
| V3 – Age | Integer |
| V4 – Annual Salary | Integer |
| V5 – Purchased | Integer |

1. **METHODS**

In this section, describe how you prepared the data for your model and performed multiple experiments using different parameters for the model.

* 1. *Data Preparation*

Too begin with, I did not have to do too much cleaning because the dataset was very clean to begin with. There were no missing data. I did have to normalize the dataset and scale the X. I tried an experiment using the raw data and the model scores were not as good. I also decided to drop the gender column because I deemed it not as important.

* 1. *Experimental Design*

You will run your model several times with different parameters to see what different results you get. In a table, describe your experimental parameters. Three or four experiments are sufficient. This is where you will describe how you divided your data into train, validate and test data sets. For example:

Table X: Experiment Parameters

|  |  |
| --- | --- |
| **Experiment Number** | **Parameters** |
| 1 | Normalized data with 80/20 split for train and test |
| 2 | Normalized data with 90/10 split for train and test |
| 3 | Raw data with 80/20 split for train and test (I did not include this in my colab notebook but I did test it) |

* 1. *Tools Used*

Describe all of the software tools you used to perform your data preparation and model implementation. For example:

Colab notebook in the google drive was used. The libraries Pandas, Matplotlib, Seaborn, and NumPy was used. NumPy is a leading scientific computing library in Python while Pandas is for data manipulation and analysis. Seaborn is used for making statistical graphics. Matplotlib is used for creating static, animated, and interactive visualizations in Python. These tools allow me to navigate through the dataset and analyze it. I also used Scikit learn to look up any questions I had and for parameters for the different regressions.

1. **RESULTS**
   1. *Mean square Error and R-Square calculation*

Graphical user interface

Description automatically generated

* 1. *Discussion of Results*

Decision Tree classifier gave me the best score and logistic regression gave me the worst. Logistic regression looks at the simultaneous effects of all the predictors, so can perform much better with a small sample size. The flip side of this is that often effects are sequential rather than simultaneous, in which case decision trees are much better.

* 1. *Problems Encountered*

I had trouble with logistic regression, but I changed some values, and it ended up working. For decision tree, I had to change the max\_depth so my score would be better. I also tried to find the ROC for the 90/10 split experiment, but I kept getting an error.

* 1. *Improvements/Future Work*

Honestly, I think I picked a dataset that was lacking that much information. It was an interesting dataset, but I would have liked to pick a dataset that was more intricate, where I could change and explore more. I would try to do more experiments.

1. **CONCLUSION**

In conclusion, this was a very simple dataset that didn’t need too much preprocessing or alterations. It was a very good dataset with a very high score of usability on Kaggle. I got very good scores with all 4 ML regressions. I did not run in to too many problems and my model was very good.