Report on Analysis of supermarket sales.

Registration Number: 2010727

Word Count: 1332

Summary of the experiments performed on "**Tosco and Spency", "SUNSBORY'S"**

The below mentioned machine learning classifiers are implemented to classify new customer based on the given data sets and required analysis is performed on **"Tosco and Spency"**

1) **Decision Tree Classifier**
2) **Random Forest Classifier**
3) **SVM Classifier**

Data Pre-processing steps for above mentioned classifiers.

Initially to build our classifier models all the required libraries must be imported (NumPy, pandas, sklearn). After importing the packages, load the give data.csv file using python command into the data frame. Once csv has loaded the file needs to be checked if there are any nan or missing values and need to fill them using mean. The data types of the variables are examined to see whether any category features exist as there are no categorical variables in the data, it is examined for missing values. There are 750 missing values in the column "F15.", For the sake of efficiency, the column had removed from the dataset. The input and output variable are separated in order to predict the new customer class. Similarly, the same steps had performed on the test dataset as well. Fig 1 shows the information regarding class and count. Need to separate the data attributes and labels into train and test datasets now that we have retrieved the data attributes and labels. We will use the 'train test split' function from scikit-learn for this, which takes the attributes and labels as inputs and outputs the train and test sets.

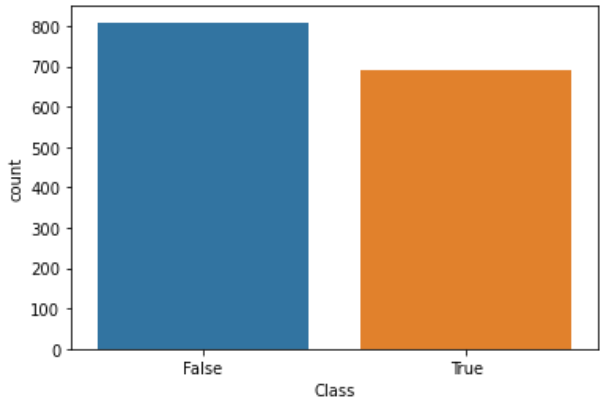

Fig 1

Tosco and Spency

Decision Tree Classifier implementation.

To implement the Decision tree classifier, as it is a classification problem, we will use the sklearn library's Decision Tree Classifier function. After that, we will change the criterion to 'entropy,' which will change the measure used to split the attribute to information gain. The classifier will next be fitted to the train attributes and labels. It is observed that the test accuracy obtained is 78.22%. To improve the accuracy, gird search cv can be used to find the optimal parameter. Next, we will set up our classifier and GridSearchCv, the major component that will assist us in finding the optimum hyperparameters. The classifier and list of params are then passed to gridsearchcv as parameters and perform the predictions on test prediction to obtain the data. After applying the grid search the accuracy has increased from 78.22% to 79.77%.

Random Forest Classifier Implementation.

To implement the Random Forest classifier, we need to import the dataset library from sklearn. Initially the classifier is implemented on the x train and y train values and the prediction had checked on y_test and accuracy obtained is 85.77%.

To improve the previous accuracy random search cv is implemented to find the optimal parameter. The obtained best parameters are fitted in to the Random Forest classifier and the prediction is applied on the feature variable and predicted accuracy obtained is 87.77%.

SVM Classifier Implementation.

SVM Classifier Implementation, Import the SVM module and use the SVC() function to generate a support vector classifier object by passing the kernel argument as the linear kernel. After that, fit your model on the train set using fit() and predict on the test set with predict (). By comparing actual test set values to expected values, accuracy can be calculated. The model accuracy obtained is 84.66%. To improve the accuracy, we can do hyper tunning parameter using SVM. GridSearchCV is a meta-estimator is one of its best features. It takes an estimator like SVC and turns it into a new estimator that operates identically in this case, as a classifier. Once it's found the optimal combination, it runs fit on all the data it's received (without cross-validation) to create a single new model with the best parameter settings. GridSearchCV's best parameters are listed in the best params_ attribute, while the best estimator is listed in the best estimator_ attribute and the accuracy obtained is 74.22 %.

Conclusion of TOSCO & SPENCY
On comparing the Accuracy of the three Classification techniques, Random Forest Classifier has more accuracy when comp ared to the other two. Therefore, it is suitable to classify the new customers. The predictions of Random Forest are copied to the csv file.

The below mentioned machine learning regressors are implemented to predict how much a new customer spends based on the given data sets of **SUNSBORY'S**

1) **Linear Regressor**
2) **Random Forest Regressor**
3) **Decision Tree Regressor**

Data Pre-processing steps for above mentioned regressor.

Initially to build our classifier models all the required libraries must be imported (NumPy, pandas, sklearn). After importing the packages, load the give data.csv file using python command into the data frame. The data types of the variables are examined to see whether any category features exist as there are two categorical variables in the data, it is examined for missing values. There are no missing values in the input data. The categorical features are encoded using one hot encoding to convert them into binary form. The input and output variable are separated in order to predict the new customer expenditure. Similarly, the same steps had performed on the test dataset as well. Need to separate the data attributes and labels into train and test datasets now that we have retrieved the data attributes and labels. We will use the 'train test split' function from scikit-learn for this, which takes the attributes and labels as inputs and outputs the train and test sets.

SUNSBORY'S

Linear Regression Implementation

After splitting the data, the model is fitted over the training dataset using LinearRegression from sklearn.model selection. T o do so, we must import the LinearRegression class, instantiate it, and pass our training data to the fit() method. Linear reg ression model finds the optimal value for the intercept and slope, resulting in the best-fitting line for the data. Now that th e algorithm has been trained the predictions are made on the test data and unseen data. To calculate the mean square err or the actual output value and the predicted output values are compared. Mean square error obtained in the linear regress ion is 219483.52569672914. The Fig 2 demonstrates the actual values, predicted values and the line of best fit.

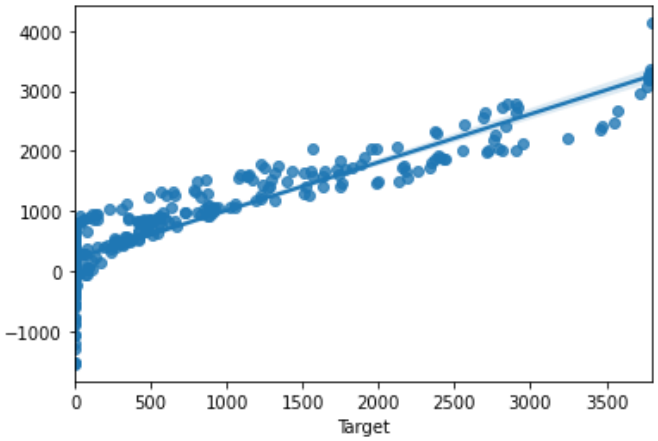

Fig 2

Random Forest Regressor Implementation

The Random Forest regressor class is imported and is set to the regressor variable to train the model. .fit() is used to fit the model on training data. In this scenario random forest is implemented using grid search cross validation which searches for the combination of best parameters to be implemented on the model for best accuracy. The results of grid search are pass ed into the random forest regressor which is in turn used to fit the training data. The same regressor model is used to make predictions on test data and unseen data. Mean squared error metric is used to evaluate the model and the value of mean squared error of the random forest regressor is 401333.85

Decision Tree Regressor

To implement decision tree Decision Tree Regressor is imported from sklearn.tree, The training data obtained in the previo us split is used to fit the model. An elbow graph(Fig 3) is plotted to determine the best parameters of the model.

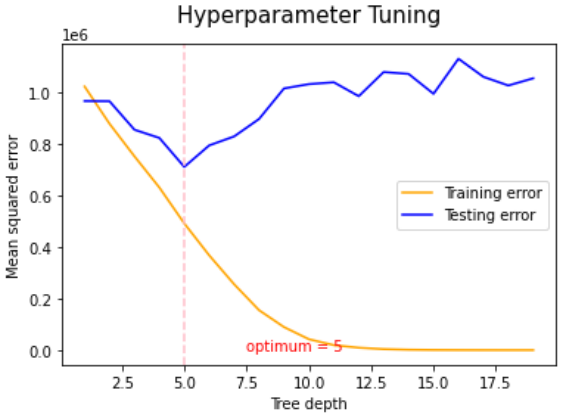

Fig 3

By examining the graph, it is evident that the optimum tree depth is 5. On passing these results to a grid search cross valida tion method there is a scope for further improvement of fit of the model. The best parameters according to the grid search are {'max_depth': 6, 'min_samples_split': 20}. Passing these parameters to the decision tree, the model is fit on the training data. The model is further used to make predictions on the test data and unseen data. Mean squared error obtained using the fitted model is 74143.136 which is high.

Conclusion on SUNSBORY'S

On comparing the Mean Squared Errors of the three Regression techniques, Linear Regression model has less Mean Square d Error when compared to the other two. Therefore, it is suitable to make predictions on new customers. The predictions a re copied to the csv file.