Health Insuarance Cross Sell Prediction using Multiple Machine Learning Algorithms

Adwait Dathan R
CB.EN.P2AID20007
radathan1@gmail.com

Siddharth M
CB.EN.P2AID20037
siddharth.m98@gmail.com

***Abstract*—Cross Selling is a business strategy used to enhance the profit by making a customer invest on an item related to the selling product. Machine learning techniques could be incorporated with the available data for predicting patterns and making better decisions for sales. This paper uses the idea of using machine learning techniques to predict if the customers currently using Vehicle insurance provided by the company would also be interested to buy Health Insurance provided by them. Different Classification and Regression techniques are used across in this paper. Best classification model is built that can predict the purchase of health insurance with the best accuracy.**

Keywords—machine learning, cross-selling, classifiers, regression, clustering, dimensionality reduction

I. INTRODUCTION

Cross Selling is the idea of selling an extra related product to the customer while purchase of a particular product. This helps increase the value of the sales produced. One of the Cross Selling example includes selling of headset with the purchase of mobile phones to the customer. It is the process of recommending the customer a product or service which is closely related to the current service bought by them [4].

In this paper we have data provided by a company that aims to enhance their sales business by cross selling health insurance with the vehicle insurance provided by them. They aim to find customers who may be interested to buy their insurance in future based on the already existing data of customers using their health insurance.

Cross Selling strategies can't be applied directly on a go as it wouldn't be much effective. There are many challenges through which we can make a cross sell prediction to a customer and it is important to properly understand how to make a customer like the company's products and services. Most of the clients would only buy the cross-sell product if they trust the products they already use and have good experience using them.[5] A lot of other constraints such as client and company relationships, service experience, usage time and many more factors come into play. This often leads to building patterns from the data and study how a customer behaves in understanding the company's products and about their interest in buying the health insurance [4].

To understand this, different machine learning techniques such as classification, regression, ensemble methods, bagging, boosting, dimensionality reduction and clustering have been performed on this dataset such that a proper analysis can be carried out on understanding the customer behavior. Multiple classifiers are used for carrying out classification by tuning the different models. Finally, all the tuned models are compared and the best classification model that can predict the cross selling of health insurance with vehicle insurance is produced.

II. MACHINE LEARNING ALGORITHMS

Machine learning algorithms are used to perform supervised and unsupervised learning through operations such as classification, regression, clustering and dimensionality reductions in the dataset. Dataset are split into two parts called the training data and testing data. Training data is used to train our model with this data, so we can later test the data with new datapoints from test data sets to test how good our model works. Classification is the process of identifying a particular class or target in our dataset using all the features associated with them. Here, the aim is to predict how good or how accurate our model can classify the targets. If we have only two targets, we have a binary classification and for more than two we call multi-class classification. Evaluation metrics here include confusion matrix that provides us the accuracy of the model and ROC-AUC curve, that provides us data about how good our model can predict the classes [2]. Regression is used to fit a line into our model such that when new data points arrive it can predict the data with respect to the boundary calculated by the regression model. The evaluation metrics here used are RMSE value, which tells us about the error in the predicted data with respect to the boundary fitted by the model trained using training data. Dimensionality reduction is performed to reduce the dimension of the dataset used for training and visualize the data. Clustering is used to automatically cluster the data based on features associated with themselves by finding similar patterns among them [1],[3].

A. Logistic Regression

It is one of the basic and popular machine learning algorithm used for predicting the probability of a given target variable. Some of the common examples of Logistic regression are checking whether an email is spam or not spam, whether the online transaction is fraud or not fraud [3].

The logistic Regression transforms its output to a probabilistic value using the logistic sigmoid function. The sigmoid function returns a probability score which is between zero and one. In order to map this to a discrete class we select a threshold point(p) above which we will classify as class zero and below which it is classified as class one. Usually we take the p-value as 0.5 [2].

The different types of logistic regression are Binary, Multinomial and Ordinal Logistic Regression. In binary logistic regression there are only two classes for the logistic model to predict either '0' or '1' class, while Multi-nominal

logistic regression has to predict a data containing more than two classes.. Ordinal regression is used to predict for a class among the ordered multiple categories.

B. Linear Regression

It is the most basic and easiest learning algorithms used to predict continuous variables. The algorithm finds the linear relationship of the target variable with the input features. The linear regression model tries to generalize a hypothetical line so that it is best fits the training data and produces low error for test data. Basically, the error is difference of the predicted value and observed value. The hypothetic function can be modelled using the equation [3].

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots \beta_n x_n$$

C. Stochastic Gradient Descent

Stochastic Gradient Descent is an algorithm that aims to enhance or optimize the parameters such that it minimizes the error. Here we use the concept of linear classifiers which has a convex loss function [1]. We basically update the coefficients such that it gives us an optimized boundary that has the least error.

Stochastic Gradient Descent classifier uses a plain SGD learning method with help of various loss functions and penalties for classification purpose. Similarly, Stochastic Gradient Descent regressor is used for regression purpose, where it tries to reduce the cost function associated to minimum [1].

D. Naïve Bayes(NB)

Naive Bayes is an important classification algorithm that uses the probabilistic classifiers. The classifier works on the basis of Bayes theorem. Using the theorem, we can find the probability of the event A happening under the assumption that the event B has already occurred. Here B is called as the evidence while A is the hypothesis [6].

$$P(A/B) = \frac{P(B/A) \cdot P(A)}{P(B)}$$

The main assumption is that the features are independent, that is the presence of a particular feature won't affect another. The main application area of naive bayes algorithms includes sentimental analysis, spam filtering, recommendation system. The different types of Naive bayes classifiers are Multinomial Naive Bayes, Bernoulli Naive Bayes, Gaussian Naive Bayes [1],[6].

E. Support Vector Machine(SVM)

Support Vector Machines are supervised algorithms that are used both in classification and regression. The idea behind SVM is to draw a hyperplane between different classes such that we can classify each of the target variables. The element that is nearest to the boundary of both classes is called the support vectors. And the distance between the hyperplane and them is called margins. SVM tries to find such a hyperplane such that it can create the best margin between the classifiers [2].

Linear SVM, where the hyperplane is a straight line that can classify the classes while Non-Linear SVM do nit have a straight hyperplane. The SVM algorithm uses the Kernel trick method here. The SVM kernel is a function that takes low dimensional input space and transforms it to a higher dimensional space. So, we end up with a separable problem instead of earlier being non-separable. It is mostly useful in non-linear separation problem [1].

F. Decision Tree

Decision tree is a supervised technique used for both classification and regression. This algorithm works on the basis of using decision rules to predict the classes to where they belong to. These decisions are taken ahead of testing data during the training phase.

The have a root node represents the entire population and this gets divided into further nodes based on the condition specified to make the decision. The process of division of nodes is called splitting. At end we get the nodes that don't divide anymore these are known as leaf. Some algorithm selected on which the decision tree works are ID3, C4.5, CART. Some of the attributes involved here are Entropy, Information gain and Gini index [1], [2].

G. K-Nearest Neighbour(KNN)

K-Nearest Neighbor model also known as lazy learner is one of the simplest supervised machine learning model which is used to classify unknown data to the known classes. The main advantages of this model are that it takes very less time for the calculation. The application field of KNN includes recommendation systems, decision making models and in image recognition systems [7].

The KNN model trains by grouping the training data based on their similarity. Further, when an unknown data is given the model classifies them according to its similarity to the respective classes.

The KNN algorithm conducts voting of the nearest neighbors of unknown data point so that the class with major vote will be the class of the unknown data point [1].

H. Random Forest

Random Forest is a supervised machine learning model that used the bagging technique that uses combination of learning models to enhance the overall result. Simply put, it is an ensemble of decision trees together, which are merged together. This algorithm can be used for both classification and regression. Random Forest has capability of making random subsets of the features and this can help us with the overfitting issue encountered in Decision tree [8].

It is much powerful because even with the default parameters it tends to provide really good results. In random forest it uses an Out of bag approach that is in the current tree which is taken by sampling with replacement, almost one-third of cases are left out. This helps run an unbiased estimation for the classification error. Gini values are used for the trees in as the attribute for decision. It has the ability of fixing up the missing value fast [1],[8].

I. Boosting

AdaBoost, focuses on classification problems and aims to convert a set of weak classifiers into a strong one. The final equation,

$$F(x) = sign(\sum_{m=1}^{M} \theta_m f_m(x)),$$

This equation is used for classification, where f_m stands for the mth weak classifier and theta_m is the corresponding weight. It is exactly the weighted combination of M weak classifiers. This is done by building a model from the training data, then creating a second model that attempts to correct the errors from the first model. Models are added until the training set is predicted perfectly or a maximum number of models are added [1],[9].

J. Principal Component Analysis(PCA)

Principal Component Analysis is powerful algorithm that helps us reduce the dimension of dataset in such a way that we don't loose too much of data and the reduced dimension can give near to same accuracy on a model as if all features were included on it. The basic idea behind PCA is to take less features so that it retains maximum information of the dataset [2].

Very large datasets with huge number of features often fall into curse of dimensionality and often takes a lot of time for training data. This is why we go for PCA, where the features can be reduced to a lower dimension and trained. Often cases are PCA helps in eliminating most of the outliers gathered in the data. This is why even though we are reducing the data sometimes much of the noises gets removed and lead to better prediction. Information loss do happen to an extent and in many cases, we may have a very small amount of accuracy loss which is preferred over wasting a lot of time in training all the features [1].

Another powerful feature of PCA is the power of visualization and as said earlier it can be used to spot the outliers in the dataset. It works under the principle of Single Value Decomposition, where the data is going through a rotation, scaling and rotation to get the required data which can be of much lower dimensions. The covariance matrix used in PCA is not more than a table that summaries the correlations between all the possible pairs of variables [1],[2].

K. K-Means Clustering

It belongs to the class of unsupervised machine learning where the unlabeled data is clustered. K-Means algorithm uses an iterative approach to cluster the data and the iteration stops when the clusters formed are stable. The number of clusters to be formed can be found out by conducting an elbow plot analysis which gives the optimum value of clusters or else the K-means algorithm takes eight as the default value. The application area of K-means algorithm includes Segmentation of images, analysis of the different groups of customers for a given company based on the customer data [1].

L. Density-based spatial clustering of applications with noise

Density-based spatial clustering of applications with noise (DBSCAN) is used for clustering of datasets.

DBSCAN uses the distance metrics such as Euclidean distances and takes in the minimum number of datapoints to form a cluster and run it over to cluster all the datapoints. It is much powerful in finding the association in the data which cannot be easily identified. Similarity between the data associated with multiple variables in the dataset can be found and this can help provide patterns in data efficiently [10].

DBSCAN can find arbitrarily-shaped clusters in our dataset. It has the ability to find a cluster which in fact is completely surrounded by another cluster. It is very robust to outliers. It requires just two parameters and is mostly insensitive to the ordering of the points in the database [1],[10].

M. Gaussian Mixture

Gaussian Mixture is one of the probabilistic models which operates under the assumption that the individual samples are generated from a mixture of Gaussian distributions. Here each Gaussian mixture represents individual clusters. Unlike K-means algorithm which uses hard clustering, the GMM model uses soft clustering approach., So, for each data point present in the distribution, a likelihood to occur in all the clusters are calculated and assigned [1].

N. Near-Miss

Near Miss is an under-sampling technique which helps solving the problem of imbalanced datasets for machine learning. It is an efficient way to balance the data. The algorithm does this by looking at the class distribution and randomly eliminating samples from the larger class. When two points belonging to different classes are very close to each other in the distribution, this algorithm eliminates the datapoint of the larger class thereby trying to balance the distribution. Three methods are thus used in Near Miss approach [11].

NearMiss-1 selects examples from the majority class that have the smallest average distance to the three closest examples from the minority class. NearMiss-2 selects examples from the majority class that have the smallest average distance to the three furthest examples from the minority class. NearMiss-3 involves selecting a given number of majority class examples for each example in the minority class that are closest. The NearMiss-3 seems desirable, given that it will only keep those majority class examples that are on the decision boundary. We have used this approach for balancing the dataset [11].

O. Root Mean Square Error (RMSE)

RMSE is metrics used in regression analysis that tells how far our points are from the line fitted. We calculate the residual here and RMSE tells us exactly what is the standard deviation of residuals. If we have N number of data and we fit a line through them such that yi is the datapoint and y^i is the predicted point then [1],

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N}(\hat{y}_i - y_i)}{N}}$$

P. Receiver operating characteristic(ROC)

ROC is a curve used in binary classification that provides us visual about the ability of our model to perform when the threshold is varied across in our model. It has the true positive rate on the Y axis and False positive rate on the X axis. When threshold is varied across the model their respective TPR and FPR are calculated and their points are plotted across the graph. On joining we get this ROC graph. Below the curve is the Area under curve (AUC) and is used to compare the models. An AUC of 0.5 tells that the model

is not able to make a proper prediction. Score of 1 is perfect and predicts everything correctly. TPR gives the classifier's capability to detect the positive class members. FPR, out of all the negative class members, gives the ratio of Negative class members that are being falsely classified as Positive ones [1],[2].

$$TPR = \frac{TP}{TP + FN}$$

$$FPR = \frac{FP}{FP + TN}$$

III. Experimental Analysis

Our aim of this project is to predict the customers that would accept the cross-sell insurance proposed to them. The data on which the experiment have been performed is a Kaggle competition data in which the company has provided data regarding sales of the customers who brought products after cross-selling and the ones who didn't. In total we had around 381109 customers data. Most of the data were associated with customer not buying the product. Around 46000 customers had brought the cross-selling product [4]. The dataset had 10 main features that were useful for our analysis and one target variable which was binary. We have performed encoding on some of the categorical features and cleaned the dataset. After doing those process, we got our features increased to 14. We found that there was a huge imbalance on the binary class of the target variable. We performed Near-Miss technique and balanced the classes of the dataset. We have performed all the machine learning techniques on this balanced dataset.

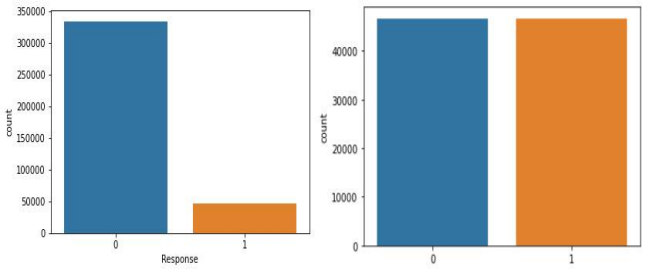

Figure 1. Class imbalance displayed and balanced after under sampling

After balancing our dataset got reduced to 96,000 customer data with 50% of each of the classes as in Figure 1. We performed both regression, classification, dimensionality reduction and clustering on this data to study the data and find the complete analysis and the end the compared models of classification to find the best model. For all the experiments we have performed a test-train split with 30% of test data. Once we tuned the hyperparameters of a specific model with respect to the training data we tested the model against the test data and this was used later for comparison of accuracy of the model.

A. Regression

We used Regression technique to analyze how our features were used to predict the Vintage of the customer. Initially we performed Linear Regression to get a RMSE of around 86.82. We regularized the model using hyperparameter tuning by performing GridSeachCV and found that the Lasso model was giving a better result over the Ridge model and we got the coefficients of each variable as in Figure 2, with our target variable. An alpha value of 10 contributed to the least RMSE value.

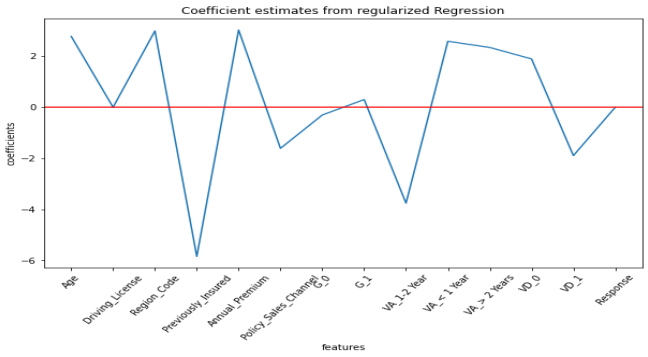

Figure 2. Coefficient estimates from regularized linear regression.

Next, we performed polynomial regression and found that we could reduce the RMSE to 81.69. Regularized model showed a degree 2 polynomial performed well on our dataset. Polynomial regression was performed on a sample of data to visualize fitting of curve for 1 to 10 degrees of polynomial as in figure 3.

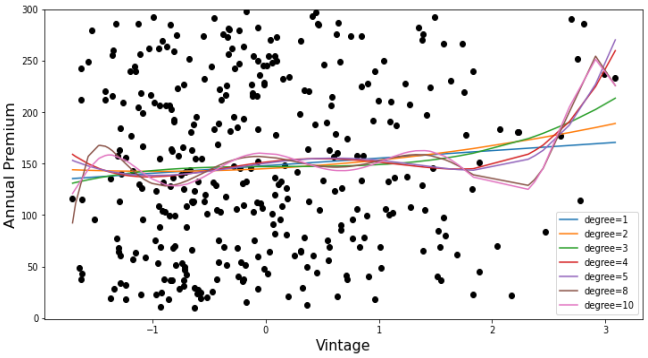

Figure 3. Polynomial regression on sample data performed on Annual Premium and Vintage for degree 1 to 10.

We performed SVR and found that RMSE to be around 84.30 when we used a linear kernel, 81.44 on a poly kernel and 81.66 on a rbf kernel. After performing hyperparameter tuning we found that RMSE of 81.66 was the best score it could produce with rbf kernel and C value of 1.

On a decision tree regressor, we found it to give us a RMSE of 85.38. When the depth wasn't specified the decision, tree seems to overfit the data. So, on analysis we found that with R2 value a depth of 2 gave good result and as the depth increases, the tree seems to overfit the data.

B. Classification

We used Classification technique to check how accurate our model was to predict if the customers would buy the cross-selling product. Logistic Regression on regularization gives us an accuracy of 91%. The regularized model had a l2 penalty with C value as 0.001. We tried Stochastic Gradient Descend, which is a linear classifier. An accuracy of 90.96% was found. On plotting the effects of score with respect to different loss function, we found hinge loss to do well on our dataset as shown in figure 5. It was also

observed that across n iterations, SGD model as in figure 4, seems to generalize over a 1000 customer data and the score is always in range of 0.905 to 0.906.

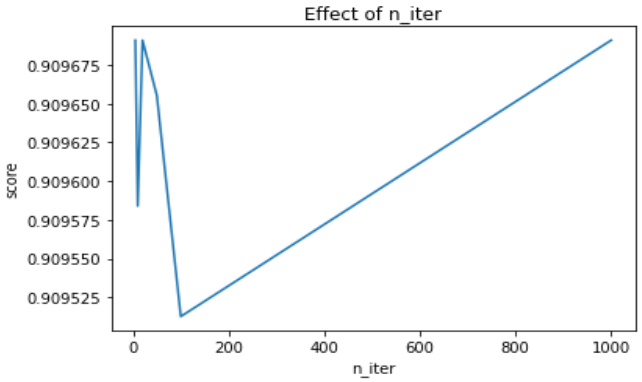

Figure 4. 1000 iterations carried out and their respective scores in SGD model

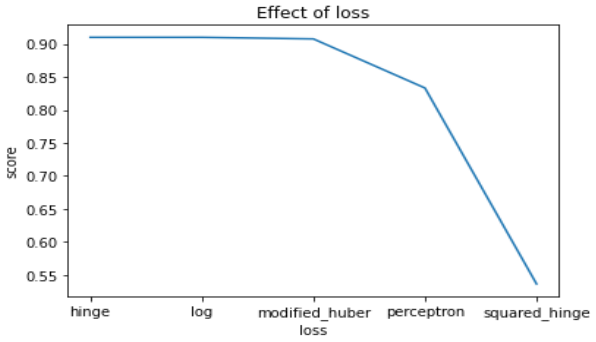

Figure 5. Effect of loss in SGD with their scores.

Gaussian Naïve Bayes gave us a 91% accuracy. SVC was performed across annual premium and vintage with respect to the user response. This was visualized after regularization in figure 6.

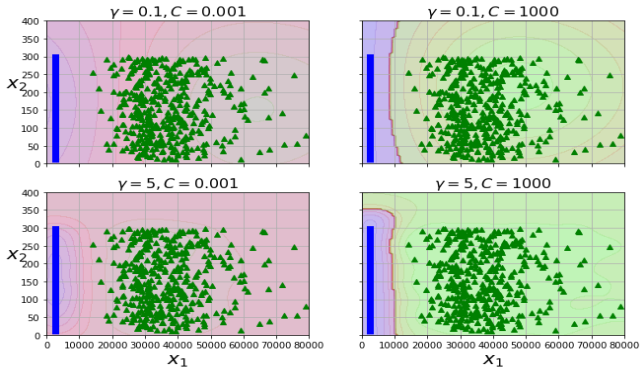

Figure 6. Regularization using SVC on Vintage and Annual Premium data.

It was observed that SVC provided a 90.9% accuracy with all features against response. Hyperparameter tuning suggest that C value to be 0.1, gamma around 0.001, kernel of rbf.

KNN Classifier gave us an accuracy of 90.4%. We regularized the model by checking the K-value against the mean error and observed that k value of 27 corresponded to least mean error. The mean error came around 0.094 as in figure 7.

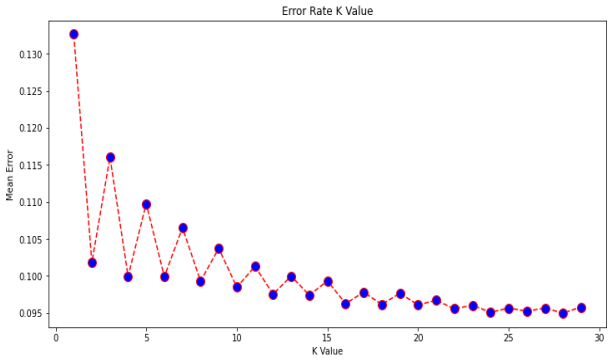

Figure 7. Error rate variation with respect to K value in KNN Classifier.

Decision tree gave us on default and accuracy rate of 86%. But on regularization after hyperparameter tunning we could tune max depth of 2 with Gini criteria to get a score of around 90.8%. Random forest model with Regularization gave us accuracy of 91%.

We performed voting to see if we could end up with some better accuracy model. For this we used Logistic Regression, Random Forest and SVC. We still got an accuracy of 91% on voting. We performed soft voting to get this result. We performed XG boost on this to get and accuracy of 91.01% which was only a very slight improvement of 0.01%. Adaboost was performed but didn't show any improvement in classifying the dataset.

C. Dimmensionality Reduction

For this we performed Principal Component Analysis. Here, we found the components needed to pass in our PCA. The cumulative explained variance is varied from 0 to 1. If we can get the features in association wit 0.9 of the variance. Then our dataset only required those much components for classification or regression. This is 90% of the data within those particular variables. We can see in this figure that around 8 components are enough. Thus, we are reducing the components from 14 to 8 as shown in figure 8.

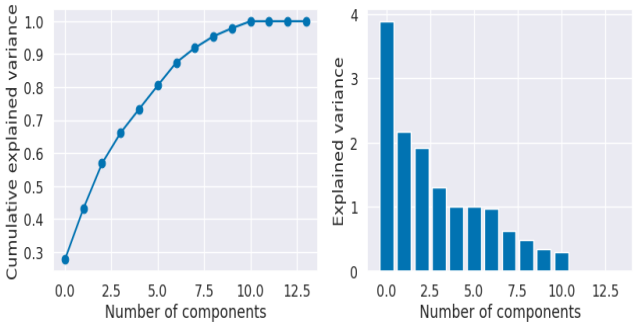

Figure 8. Graph indicating how number of components varies across explained variance in PCA.

We checked how those 8 components were related with the features. So, a correlation was observed among them in figure 9. The yellow color indicates the correlation to be positive with respect to scale and violet to have the negative correlation with the scale. We can see that Vintage and the 6th component has a highly positive relation and Driving license and 5th component are highly negative in correlation.

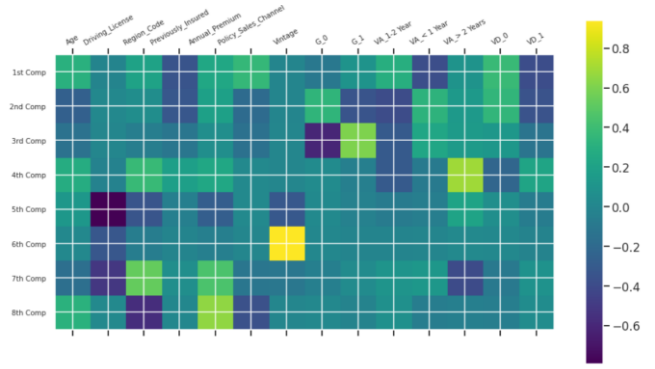

Figure 9. All 8 components generated via PCA relationship with each features of our dataset.

The eight components shared an explained variance in the ratio 0.27686159, 0.15527205, 0.13659068, 0.09268575, 0.07209146, 0.0713745 ,0.06989988, 0.04438569. A new data frame was build using the values associated with these components and further machine learning techniques could be performed on it.

We plotted the relationship between 1st and 2nd PCA component associated with the dataset in figure 10.

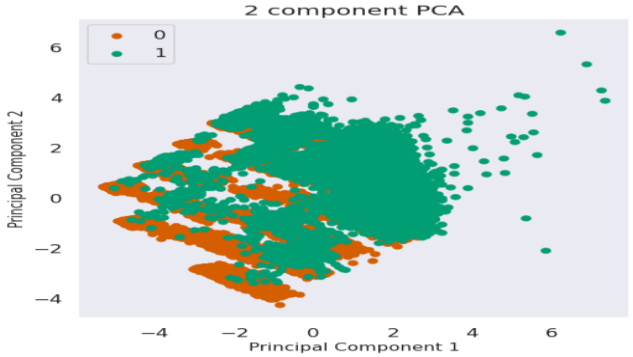

Figure 10. PCA component 1st and 2nd visualized.

We ran Logistic regression on this dataset to give us and accuracy of around 88%. Which is almost near to the accuracy we got when we didn't reduce any features.

D. Clustering

We performed K-means clustering on the dimension reduced by PCA. To get the optimal value of k for our algorithm we use the elbow method. By observing the curve of distortion with k we choose 6 as the cluster value as in figure 11.

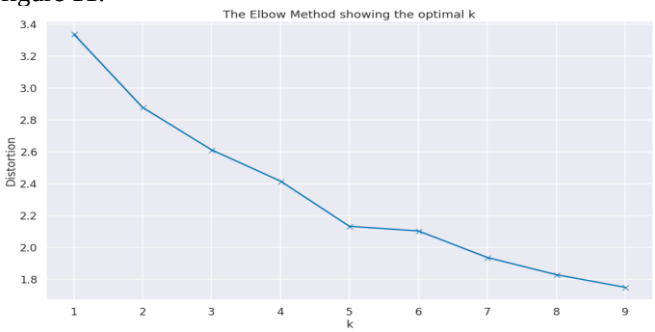

Figure 11. Elbow method to detect the optimal value of k for k-means.

Next, we performed K-means using clusters as 6 and we got all our dataset clustered into 6 different groups. We plot the figure 12, for this for the first two components of our dataset as shown below.

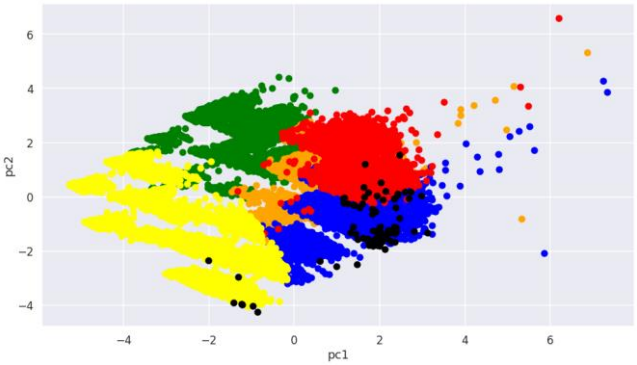

Figure 12. k-mean clustering visualized across all 6 clusters.

When we plot it for 3 components, we have a clustering like this. We can see the different colors associated with these clusters. We plotted the inertia associated with each of the clusters as shown in figure 13.

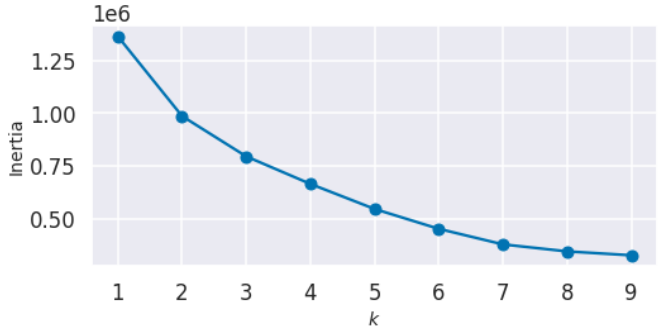

Figure 13. Inertia across 6 clusters in K-means.

We used Silhouette score to determine the performance of each clusters on our dataset as shown in figure 14.

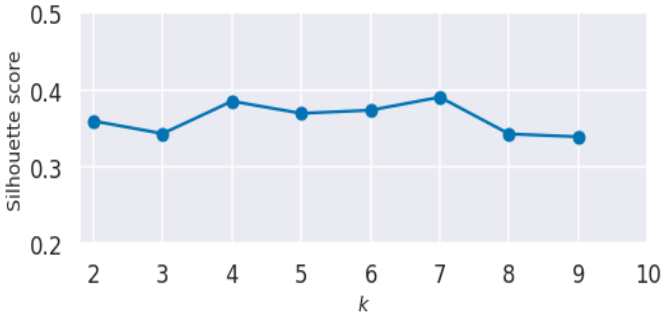

Figure 14. Silhouette score across k clusters in K-means.

We used a pipeline to perform clustering with logistic regression and we regularized it to accuracy score of 89%. We performed DBSCAN on the dataset to get around 16 clusters. Figure 15 has a visualization of 7 clusters.

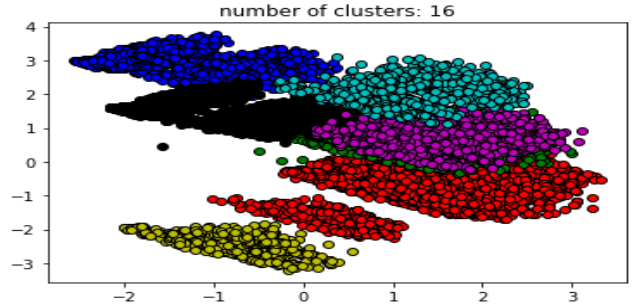

Figure 15. Clusters formed in DBSCAN.

For gaussian mixture we got around 6 gaussian components and figure 16 is plot for clustering on first two components of dataset. The converged log likelihood value is around 0.67 and n iterations is around 5.

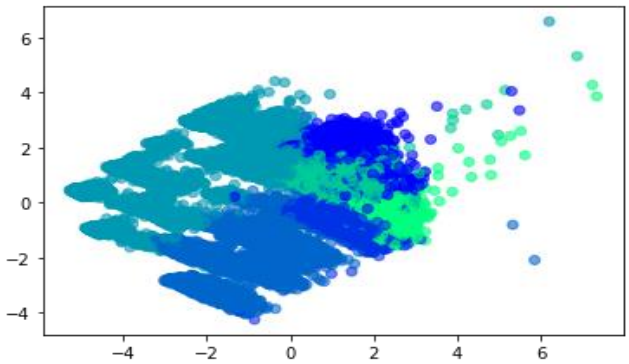

Figure 16. Components of gaussian mixture.

IV. Experimental Results

Experimental results are published below. All results performed are on the test data we have divided initially, which is 30% of the dataset. We applied regression techniques on dataset against Vintage as feature and compared them. We also clustered the complete dataset after applying PCA using different clustering algorithm and basically, we tested how the k varied across k-means algorithm which gave the best silhouette score. Finally, we classified all the models for the complete dataset and compared them using ROC AUC score and found the best model that predicted the customers who would buy the cross-selling product. All the models compared for both regression and classifications were regularized models.

A. Regression

The below table display the regression results. We can see that SVR model with 'rbf' as the kernel displayed the best results with the least RMSE score of 81.66 compared to other models as displayed in table I.

TABLE I. COMPARISON OF REGRESSION MODEL

SI.no	***Regression Models***	***RMSE score***
1	Multiple Linear Regression	86.82
2	Polynomial Regression	81.69
3	SVR	81.66
4	Decision Tree Regressor	85.38

B. Classification

All the classification algorithms results were compared by plotting ROC AUC curve and analyzing the best AUC score. Random forest classifier proved to show the best score of 0.96 and hence we go for this model. The ROC AUC plot is shown in figure 17.

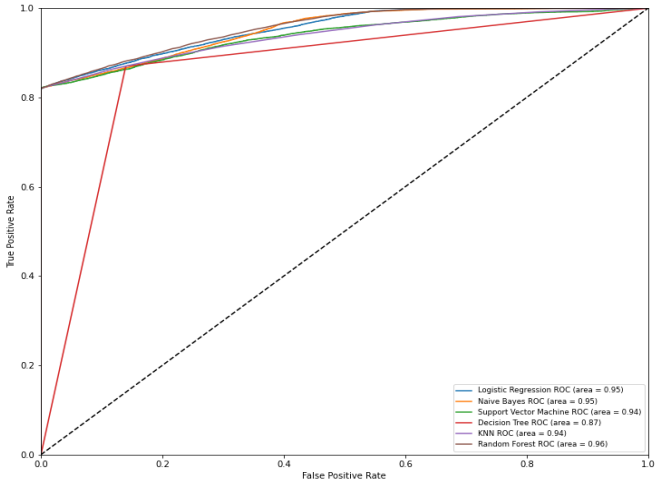

Figure 17. ROC AUC curve comparing different classification model and their AUC Score.

All the compared data is displayed in table II.

TABLE II. COMPARISON OF CLASSIFICATION MODEL

SI.no	***Classifiers***	***AUC score***
1	Logistic Regression	0.95
2	Naïve Bayes	0.95
3	SVM	0.94
4	Decision Tree	0.87
5	KNN	0.94
6	Random Forest	0.96

C. Clustering

We performed K-means clustering on our dataset and we concluded that 6 clusters were a good number to group out dataset. Further we analyzed the Silhouette score and found it to have highest score of 0.39. The complete comparison of all the scores with each cluster number is given in table III.

TABLE III. COMPARISON OF SILHOUETTE SCORES

K-value	***Silhouette Coefficient***
2	0.3429949234233916
3	0.3850471419238876
4	0.3693305799824815
5	0.37337312083467467
6	0.3903452467295772
7	0.34253532403371895
8	0.33891526262299676

V. Conclusion

The major contribution of this paper was to identify which of our classifier was the best model that could predict if our customer will buy the cross-selling product. The Kaggle dataset uploaded by a company for contest was used for this experiment [4]. From the above experiment we conclude that Random forest classifier is the best model which classifies the dataset and we can go ahead to build a model for deployment with this classifier. Even though most of the classifiers gave 89%+ accuracy rate, Random forest proved to outperform all by providing 91.01% accuracy and an AUC score of 0.96.

References

[1] Géron, A., 2019. Hands-on machine learning with Scikit-Learn and TensorFlow. 2nd ed. O'REILLY.

[2] Starmer, J., 2018. StatQuest: Machine Learning. [online] Youtube. Available at: <https://www.youtube.com/watch?v=Gv9_4yMHFhI&list=PLblh5JKOoLUICTaGLRoHQDuF_7q2GfuJF>.

[3] Andrew, N., 2011. Machine Learning. [online] Coursera. Available at: <https://www.coursera.org/learn/machine-learning>.

[4] Kumar, A., 2020. Health Insurance Cross Sell Prediction. [online] Kaggle.com. Available at: <https://www.kaggle.com/anmolkumar/health-insurance-cross-sell-prediction>.

[5] S. Cody , *What is Cross-Selling? Plus 3 Tips, 5 Methods & Examples.* [Online]. Available: https://instapage.com/blog/cross-selling.

[6] K. Nikhil , Ed., "Naive Bayes Classifiers," GeeksforGeeks, 15-May-2020. [Online]. Available: https://www.geeksforgeeks.org/naive-bayes-classifiers/.

[7] "K-Nearest Neighbor(KNN) Algorithm for Machine Learning - Javatpoint," *www.javatpoint.com*. [Online]. Available: https://www.javatpoint.com/k-nearest-neighbor-algorithm-for-machine-learning.

[8] N. Donges, "A complete guide to the random forest algorithm," Built In. [Online]. Available: https://builtin.com/data-science/random-forest-algorithm.

[9] SauceCat, "Boosting algorithm: AdaBoost," Medium, 30-Apr-2017. [Online]. Available: https://towardsdatascience.com/boosting-algorithm-adaboost-b6737a9ee60c.

[10] S. Nagesh, "DBSCAN Clustering Algorithm in Machine Learning," KDnuggets. [Online]. Available: https://www.kdnuggets.com/2020/04/dbscan-clustering-algorithm-machine-learning.html.

[11] J. Brownlee, "Undersampling Algorithms for Imbalanced Classification," Machine Learning Mastery. [Online]. Available: https://machinelearningmastery.com/undersampling-algorithms-for-imbalanced-classification/.