LIFE INSURANCE SALES

BUSINESS REPORT- PROJECT NOTES 2

Philjoy Dsilva

21-July-2024

PGPDSBA

Contents

List of Tables

List of Figures

Life Insurance Sales

1) Model building and interpretation.

The dataset is divided into independent and dependent variables. The independent variables are scaled using StandardScaler. After scaling, the independent and dependent variables are merged, and the data is split into training and testing sets with a 70:30 ratio. The following models are trained to identify the best model for predicting agent bonuses.

a) Build various models (You can choose to build models for either or all of descriptive, predictive or prescriptive purposes)

b) Test your predictive model against the test set using various appropriate performance metrics

c) Interpretation of the model(s)

Linear Regression

The linear regression model is built using the Statsmodels package. For this regression, both the independent and dependent variables are combined.

Model 1:

AgentBonus =

CustTenure + NumberOfPolicy + MonthlyIncome + Complaint + ExistingPolicyTenure + *SumAssured + LastMonthCalls + CustCareScore + Channel_Online + Channel_Third_Party_Partner + Occupation_Large_Business + Occupation_Salaried + Occupation_Small_Business + EducationField_PG + EducationField_UG + Gender_Male + ExistingProdType_2 + ExistingProdType_3 + ExistingProdType_4 + ExistingProdType_5 + ExistingProdType_6 + Designation_Executive + Designation_Manager + Designation_Senior_Manager + Designation_VP + MaritalStatus_Married + MaritalStatus_Unmarried + Zone_North + Zone_South + Zone_West + PaymentMethod_Monthly + PaymentMethod_Quarterly + PaymentMethod_Yearly +

Most of the variables are not significant, and there is a correlation between the independent variables (as inferred from EDA). To address this, correlated variables are removed using the Variance Inflation Factor (VIF) method. Variables with a VIF greater than 5 are removed one by one, and the VIF for the remaining variables is recalculated. The table below displays the VIF values for the 29 variables that are not correlated.

	variables	VIF
2	MonthlyIncome	2.43
28	PaymentMethod_Yearly	2.42
23	MaritalStatus_Unmarried	1.95
12	EducationField_PG	1.94
22	MaritalStatus_Married	1.94
13	EducationField_UG	1.88
26	PaymentMethod_Monthly	1.86
17	ExistingProdType_5	1.85
15	ExistingProdType_2	1.85
21	Designation_VP	1.79
16	ExistingProdType_3	1.63
5	SumAssured	1.55
20	Designation_Senior_Manager	1.53
11	Occupation_Small_Business	1.36
0	CustTenure	1.28
19	Designation_Manager	1.27
27	PaymentMethod_Quarterly	1.19
6	LastMonthCalls	1.19
18	ExistingProdType_6	1.19
10	Occupation_Large_Business	1.12
4	ExistingPolicyTenure	1.12
1	NumberOfPolicy	1.10
8	Channel_Online	1.05
9	Channel_Third_Party_Partner	1.04
14	Gender_Male	1.02
7	CustCareScore	1.02
25	Zone_West	1.01
24	Zone_South	1.01
3	Complaint	1.01

Table 1 - VIF Tables

Model 2:

CustTenure + NumberOfPolicy + MonthlyIncome + Complaint + ExistingPolicyTenure + SumAssured + LastMonthCalls + CustCareScore + Channel_Online + Channel_Third_Party_Partner + Occupation_Large_Business + Occupation_Small_Business + EducationField_PG + EducationField_UG + Gender_Male + ExistingProdType_2 + ExistingProdType_3 + ExistingProdType_5 + ExistingProdType_6 + Designation_Manager + Designation_Senior_Manager + Designation_VP + MaritalStatus_Married + MaritalStatus_Unmarried + Zone_South + Zone_West + PaymentMethod_Monthly + PaymentMethod_Quarterly + PaymentMethod_Yearly

The majority of the independent variables in the model are not significant. Here are the coefficients:

- Intercept
- Customer Tenure
- Monthly Income
- Existing Policy Tenure
- Sum Assured
- Customer Care Score
- Designation (Manager)
- Designation (VP)

After removing the insignificant variables, the final model is as below

Model 3:

CustTenure + MonthlyIncome + ExistingPolicyTenure + SumAssured + CustCareScore + Designation_Manager + Designation_VP

OLS Regression Results

Dep. Variable:	AgentBonus	**R-squared:**	0.783
Model:	OLS	**Adj. R-squared:**	0.783
Method:	Least Squares	**F-statistic:**	1630.
Date:	Sat, 20 Jul 2024	**Prob (F-statistic):**	0.00
Time:	00:37:05	**Log-Likelihood:**	-25003.
No. Observations:	3164	**AIC:**	5.002e+04
Df Residuals:	3156	**BIC:**	5.007e+04
Df Model:	7		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	4073.5441	11.648	349.726	0.000	4050.706	4096.382
CustTenure	225.3250	13.152	17.132	0.000	199.537	251.113
MonthlyIncome	287.9973	14.856	19.386	0.000	258.869	317.125
ExistingPolicyTenure	121.6540	12.267	9.917	0.000	97.602	145.706
SumAssured	888.4446	14.341	61.951	0.000	860.326	916.563
CustCare Score	24.1449	11.652	2.072	0.038	1.298	46.992
Designation_Manager	-40.5992	11.875	-3.419	0.001	-63.882	-17.316
Designation_VP	51.2924	13.880	3.695	0.000	24.077	78.507

Omnibus:	128.913	**Durbin-Watson:**	1.971
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	149.440
Skew:	0.475	**Prob(JB):**	3.54e-33
Kurtosis:	3.482	**Cond. No.**	2.32

Table 2 - Model 3 regression results

The model strength is 78.3% (R-Squared).

> Omnibus Test: This test evaluates the skewness and kurtosis of the residuals. Ideally, the value should be as low as possible. The JB P-value also supports the normality of the data.
>
> Durbin-Watson Test: This test assesses the homoscedasticity of the residuals. The ideal value is close to 2, and in this case, it is nearly 2 (1.9), indicating no autocorrelation.
>
> Jarque-Bera Test: This confirmatory test checks the skewness and kurtosis of the residuals, supporting the Omnibus Test. The P-value is less than 0.05, indicating the dataset is normally distributed.
>
> Condition Number: A high condition number suggests the presence of multicollinearity. Here, the condition number is less than 10, indicating no multicollinearity.

The Final Equation for prediction of AgentBonus is

AgentBonus =
(4073.54) * Intercept + (225.32) * CustTenure + (288.0) * MonthlyIncome + (121.65) * ExistingPolicyTenure + (888.44) * SumAssured + (24.14) * CustCareScore + (-40.6) * Designation_Manager + (51.29) * Designation_VP

<u>Validation Against Test Data</u>
The final regression model was used for prediction, yielding the following results:

The R-squared value for the test data is 0.769.
The RMSE for the linear regression model is 671.51.
Both Lasso and Ridge regression are regularization methods. They aim to minimize the sum of squared residuals (RSS) while incorporating a penalty term to prevent overfitting.

Using Lasso regression and ridge regression to regularize and minimize the sum of squared residuals (RSS) along with some penalty term.

Ridge Regression

AgentBonus =
(220.88) * CustTenure + (10.19) * NumberOfPolicy + (167.12) * MonthlyIncome + (19.51) * Complaint + (121.17) * ExistingPolicyTenure + (877.51) * SumAssured + (-9.72) * LastMonthCalls + (22.61) * CustCareScore + (10.06) * Channel_Online + (-2.3) * Channel_Third_Party_Partner + (-42.61) * Occupation_Large_Business + (-53.32) * Occupation_Salaried + (-61.33) * Occupation_Small_Business + (-2.38) * EducationField_PG + (-3.27) * EducationField_UG + (13.93) * Gender_Male + (16.82) * ExistingProdType_2 + (-70.59) * ExistingProdType_3 + (-23.97) * ExistingProdType_4 + (0.49) * ExistingProdType_5 + (22.84) * ExistingProdType_6 + (-229.82) * Designation_Executive + (-210.53) * Designation_Manager + (-92.98) * Designation_Senior_Manager + (43.25) * Designation_VP + (-28.57) * MaritalStatus_Married + (6.05) * MaritalStatus_Unmarried + (14.56) * Zone_North + (6.08) * Zone_South + (18.79) * Zone_West + (-13.24) * PaymentMethod_Monthly + (9.64) * PaymentMethod_Quarterly + (-49.26) * PaymentMethod_Yearly +

Ridge regression includes an L2 penalty, which shrinks the coefficients towards zero but never reduces them to absolute zero. This regularization helps to manage multicollinearity and prevent overfitting by penalizing large coefficients, thus leading to a more generalizable model. However, unlike Lasso regression, Ridge regression does not produce sparse models where some coefficients are exactly zero; instead, it retains all predictors with reduced effect sizes.

Validation Against Test Data

The Ridge Regression model was used for prediction, yielding the following results:

- The R-squared value for the test data is 0.771.
- The RMSE for the Ridge Regression model is 668.43.

Ridge Regression performs slightly better than the earlier model.

Lasso Regression

Lasso regression, or L1 regularization, strikes a balance between simplicity and accuracy. It provides interpretable models while effectively addressing the risk of overfitting. Lasso regression is distinctive because it can reduce some coefficients to exactly zero, which facilitates model interpretation and feature selection.

AgentBonus =
(219.82) * CustTenure + (9.43) * NumberOfPolicy + (170.79) * MonthlyIncome + (18.45) * Complaint + (120.04) * ExistingPolicyTenure + (881.27) * SumAssured + (-7.8) * LastMonthCalls + (22.0) * CustCareScore + (9.01) * Channel_Online + (-1.47) * Channel_Third_Party_Partner + (-9.02) * Occupation_Large_Business + (3.36) * Occupation_Salaried + (-3.46) * Occupation_Small_Business + (-0.0) * EducationField_PG + (-0.81) * EducationField_UG + (13.04) * Gender_Male + (16.58) * ExistingProdType_2 + (-48.29) * ExistingProdType_3 + (-2.64) * ExistingProdType_4 + (11.7) * ExistingProdType_5 + (27.32) * ExistingProdType_6 + (-221.21) * Designation_Executive + (-203.7) * Designation_Manager + (-89.32) * Designation_Senior_Manager + (42.41) * Designation_VP + (-28.41) * MaritalStatus_Married + (4.91) * MaritalStatus_Unmarried + (-0.0) * Zone_North + (4.14) * Zone_South + (3.6) * Zone_West + (-0.0) * PaymentMethod_Monthly + (11.15) * PaymentMethod_Quarterly + (-43.4) * PaymentMethod_Yearly

Validation Against Test Data

The Lasso Regression model was employed for prediction, with the following results:

- The R-squared value for the test data is 0.771.
- The RMSE for the Lasso Regression model is 668.02.

In this case, Ridge and Lasso regression perform similarly.

CART

The CART model employs the Decision Tree Regressor and includes a Feature Importance Plot. Here is the Feature Importance chart for the CART model:

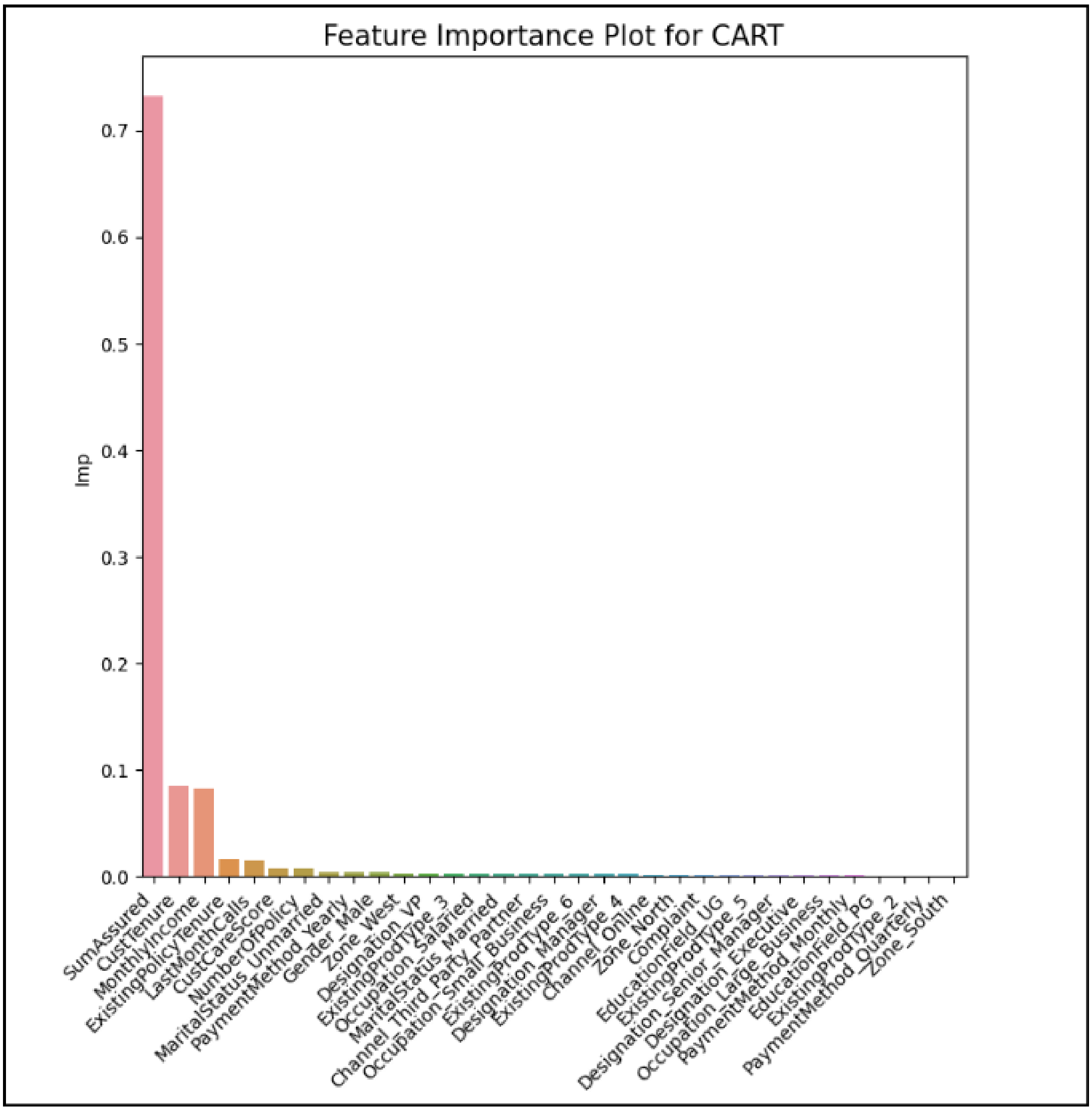

Figure 1 - Feature plot for CART

In the CART model, Sum Assured is identified as the most important feature. However, the model appears to be overfitting the training data, as indicated by an R-squared value of 1 and an RMSE of 0 for the training set.

Validation Against Test Data
For prediction using the CART model:
The R-squared value for the test data is 0.65.
The RMSE is 823.42
The CART model needs to be tuned to improve its performance.

2) Model Tuning and business implication

a) Ensemble modelling, wherever applicable

b) Any other model tuning measures(if applicable)

CART Tuned/Pruned

RandomizedSearchCV was utilized for tuning the model.

The tuned CART Model is:

DecisionTreeRegressor(criterion='friedman_mse', max_depth=89,

min_impurity_decrease=0.0081, min_samples_leaf=42,

min_samples_split=3)

The key variables for the tuned CART model are SumAssured, MonthlyIncome, and CustTenure.

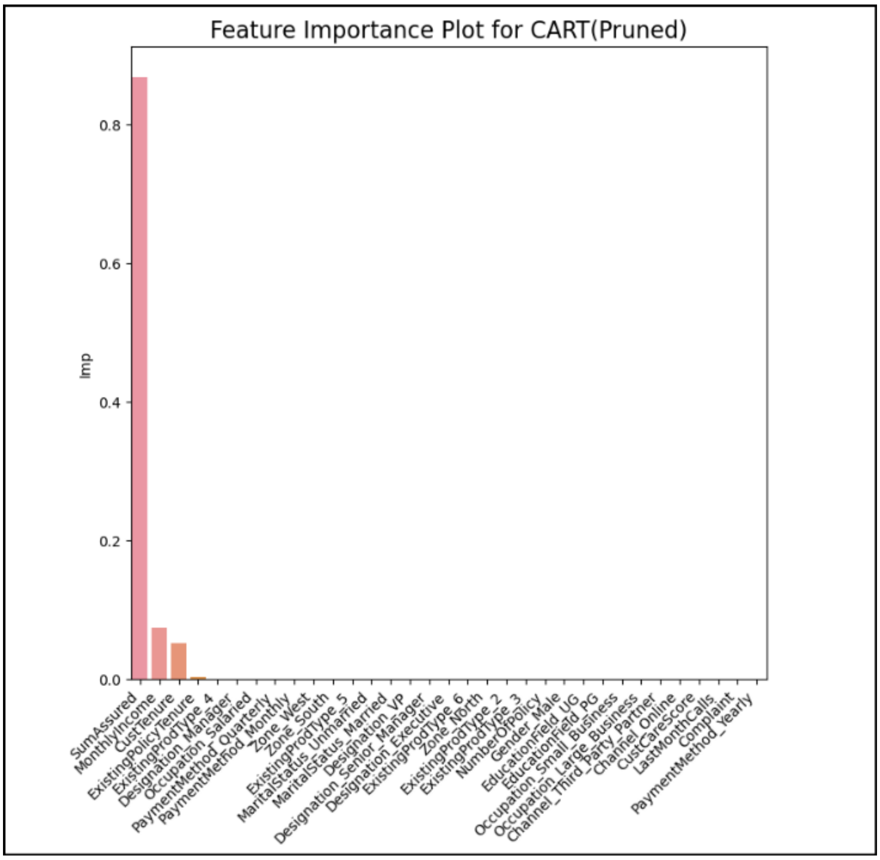

Figure 2 - Feature plot for CART Pruned

Validation Against Test:

- The CART (Pruned/Tuned) model was used for prediction.
- R-squared value for test data: 0.77
- RMSE for the tuned CART model: 665.89

- The RMSE has decreased compared to the base CART model.
- The tuned CART model provides better predictions than the base CART model.

K-Neighbor Regressor

The K-Neighbor Regressor is tuned with GridSearchCV.

Optimum model - KNeighborsRegressor(metric='manhattan', n_neighbors=17, weights='distance')

Despite tuning, the K-Neighbor Regressor is still overfitting the training data.
The R-squared value for the training data is 1, and the RMSE for the trained data model is 0.

Validation Against Test:

- The K-Neighbor Regressor was applied for prediction.
- R-squared value for test data: 0.54
- RMSE for K-Neighbor Regressor model: 940.38
- This model is the least preferred and performs worse than the other models.

Random Forest Regressor

The Random Forest Regressor is used to predict the agent bonus.

The Optimum model after RandomizedSearchCV is
(criterion='friedman_mse', max_samples=0.1,
n_estimators=436)

The Feature Importance Plot for RF is similar to the CART Model

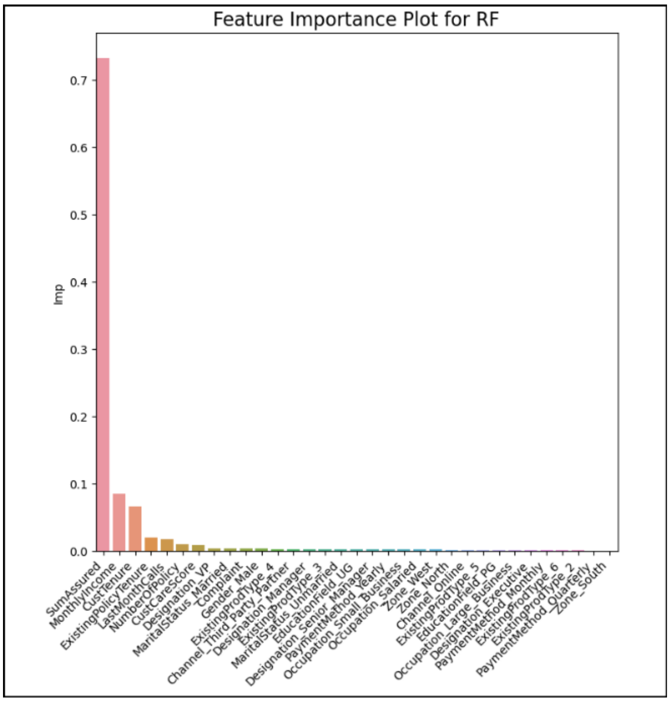

Figure 3 - Feature plot for RF

Validation Against Test:

- The Random Forest Regressor model was used for prediction.
- R-squared value for test data: 0.80
- RMSE for the Random Forest Regressor model: 617.10
- The Random Forest Regressor performs better than the tuned CART model in predicting the dependent variable.

Bagging

The CART model serves as the base for Bagging, resulting in the CART Bagging model.

The parameters for CART Bagging, tuned using RandomizedSearchCV, are:

- n_estimators: list(range(100, 500, 2))
- max_samples: list(np.arange(0.01, 1, 0.01))
- max_features: list(np.arange(0.01, 1, 0.01))

Validation Against Test:

- The CART Bagging model was used for prediction.
- R-squared value for the test data: 0.80
- RMSE for the CART Bagging Regressor model: 615
- The CART Bagging model enhances prediction accuracy.

ADA Boosting Regre

AdaBoostRegressor is used for ADA Boosting.

The Tuned Model parameter is - AdaBoostRegressor(learning_rate=0.12684999999998958, n_estimators=105)

Validation Against Test:

- The ADA Boosting model was used for prediction.
- R-squared value for the test data: 0.75
- RMSE for the ADA Boosting Regressor model: 684.48
- ADA Boosting does not perform as well as the other ensemble models.

Voting Regressor

The Voting Regressor is employed to create a heterogeneous ensemble model. The base models used in this ensemble include:

- Lasso Regression
- CART (Pruned/Tuned)
- Random Forest Regressor
- CART Bagging
- ADA Boosting

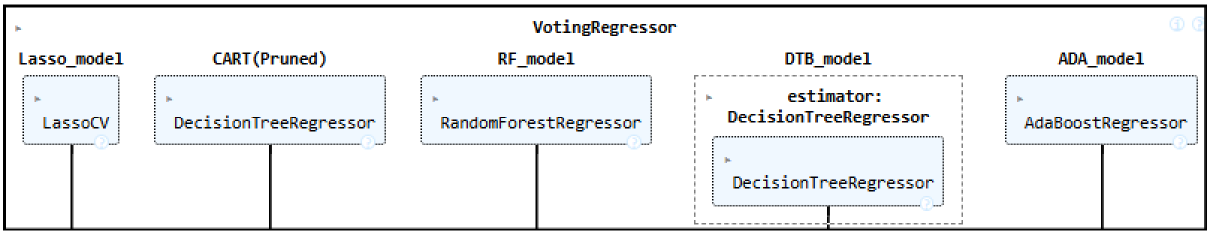

Figure 4 - Voting Reg

Below are the parameters - VotingRegressor(estimators=[('Lasso_model', LassoCV(alphas=[0.0001, 0.001, 0.01, 0.1, 1, 10])), ('CART(Pruned)', DecisionTreeRegressor(criterion='friedman_mse', max_depth=89, min_impurity_decrease=0.0081, min_samples_leaf=42, min_samples_split=3)), ('RF_model', RandomForestRegressor(criterion='friedman_mse', max_samples=0.1, n_estimators=786)), ('DTB_model', BaggingRegressor(estimator=DecisionTreeRegressor(), max_features=0.78, max_samples=0.6, n_estimators=238)), ('ADA_model', AdaBoostRegressor(learning_rate=0.12685, n_estimators=105))], n_jobs=-1)

Validation Against Test:

- The Voting Regressor model was applied for prediction.
- R-squared value for the test data: 0.80
- RMSE for the Voting Regressor model: 622.93
- The Weighted Voting Regressor is utilized to enhance model performance.

Weighted Voting Regressor

The base models used for this are:

- Lasso Regression
- CART (Pruned/Tuned)
- Random Forest Regressor
- CART Bagging
- ADA Boosting

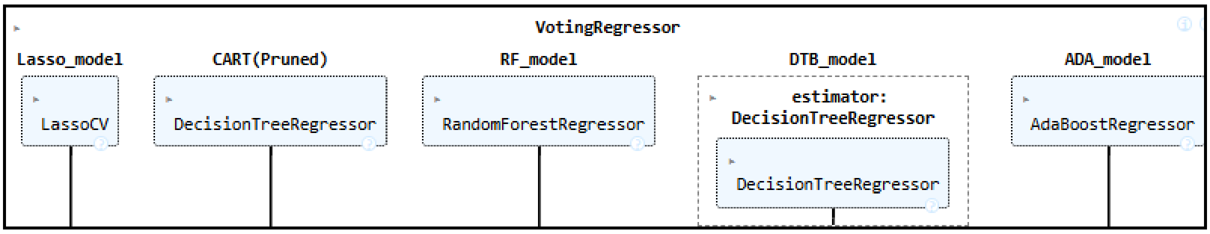

Figure 5 - Weighted voting reg

Validation Against Test:

- The Weighted Voting Regressor model was used for prediction.
- R-squared value for the test data: 0.80
- RMSE for the Weighted Voting Regressor model: 619.20
- While the Weighted Voting Regressor enhances model performance

Stacking Regressor

The Stacking Regressor is used to get better ensemble model.

The base models utilized are:

- Lasso Regression
- CART (Pruned/Tuned)
- Random Forest Regressor
- CART Bagging
- ADA Boosting

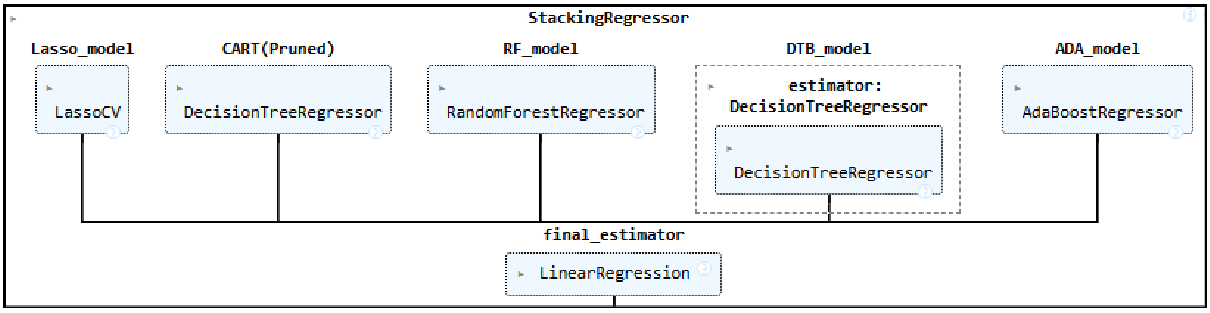

Figure 6 - Stacking reg

The Final Estimator model is Linear Regression.

Below are the parameters-
StackingRegressor(estimators=[('Lasso_model', LassoCV(alphas=[0.0001, 0.001, 0.01, 0.1, 1, 10])), ('CART(Pruned)', DecisionTreeRegressor(criterion='friedman_mse', max_depth=89, min_impurity_decrease=0.0081, min_samples_leaf=42, min_samples_split=3)), ('RF_model', RandomForestRegressor(criterion='friedman_mse', max_samples=0.1, n_estimators=786)), ('DTB_model', BaggingRegressor(estimator=DecisionTreeRegressor(), max_features=0.78, max_samples=0.6, n_estimators=238)), ('ADA_model', AdaBoostRegressor(learning_rate=0.12685, n_estimators=105))], final_estimator=LinearRegression(), n_jobs=-1)

Validation Against Test:

- The Stacking Regressor model was used for prediction.
- R-squared value for the test data: 0.81
- RMSE for the Stacking Regressor model: 598.53

c) Interpretation of the most optimum model and its implication on the business

R-Squared and RMSE are compared across models to identify the most optimal one.

	Model	R-Squared(%)	MAE	MSE	MAPE	RMSE	Max Error
0	Linear Regression	76.90	524.81	450935.45	0.13	671.52	2610.78
1	Lasso Regression	77.14	523.55	446259.25	0.14	668.03	2642.07
2	Ridge Regression	77.11	523.89	446805.61	0.14	668.44	2640.16
3	CART	65.53	601.16	672987.68	0.16	820.36	4313.00
4	CART(Pruned)	77.29	518.05	443287.36	0.13	665.80	2662.24
5	K-Neighbors Regressor	54.70	763.75	884328.55	0.20	940.39	3691.89
6	Random Forest Regressor	80.49	486.06	380821.59	0.13	617.11	2637.72
7	CART-Bagging	80.57	484.46	379305.27	0.13	615.88	2851.71
8	ADA Boosting	76.00	560.53	468525.46	0.15	684.49	2225.15
9	Voting Regressor	80.12	492.03	388054.16	0.13	622.94	2561.36
10	Voting Regressor(Weighted)	80.36	487.74	383412.31	0.13	619.20	2675.05
11	Stacking Regressor	81.65	466.25	358245.78	0.12	598.54	2566.55

Figure 7 - Optimal model

- The Stacking Regressor includes 5 base models: Lasso Regression, CART (Pruned/Tuned), Random Forest Regressor, CART Bagging, and ADA Boosting, with Linear Regression as the final estimator.
- It has a low RMSE and a high R-squared value.
- CART, the second-best model with an R-squared value of 80.57%, is simpler and does not require the complexity of base models used in the stacking regressor.
- The most important features identified are SumAssured, MonthlyIncome, and CustTenure.

Implication

- The Agent Bonus is influenced by the SumAssured of the policy, the customer's Monthly Income, and the Customer Tenure with their organization.
- Agents sell policies, and the payments made by customers are a primary source of income for the company. A higher SumAssured leads to a higher bonus and greater profit for the company, which can be invested until the policy matures. This arrangement benefits both the agents and the company.
- To encourage agents to sell higher Sum Assured policies, a table of Sum Assured values and corresponding Agent Bonuses can be created from the models. This can serve as an incentive for agents to target higher Sum Assured policies.
- By predicting agent bonuses, the insurance company can categorize agents into different bonus levels such as High, Medium, and Low, which can be useful for training purposes.
- Identifying key agents for company development is possible if agent details are provided, which can be explored as part of future studies.
- The company can also use this information to develop strategies for optimizing employee bonuses and reducing costs to maximize profits.

The end.