January 2022

This Business Report shall provide detailed explanation of how we approached each problem given in the assignment. It shall also provide relative resolution and explanation with regards to the problems

Life Insurance Sales - Capstone

Business Report Notes -2

Thakur Arun Singh

Contents

[Problem 1: Model building and interpretation 1](#_Toc93254905)

[Problem 1.a 2](#_Toc93254906)

[Problem 1.b 16](#_Toc93254907)

[Problem 1.c 31](#_Toc93254908)

[Problem 2: Model Tuning and business implication 33](#_Toc93254909)

[Problem 2.a 33](#_Toc93254910)

[Problem 2.b 35](#_Toc93254911)

[Problem 2.c 39](#_Toc93254912)

## Problem 1: Model building and interpretation

The dataset belongs to a leading life insurance company. The company wants to predict the bonus for its agents so that it may design appropriate engagement activity for their high performing agents and up skill programs for low performing agents

### Problem 1.a

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

**Resolution:**

**Project Approach**

The work that we have completed:

We have created multiple models and applied them on different sets of data as required. All the different models which were created were then evaluated using the AUC / F1 score at the end for the testing data set. Based on this an optimal model was chosen. Eventually we also found the feature importance for the most optimal model.

**Models Built**

Various tree based as well as distance based models were built as part of this exercise using the different data sets as elaborated earlier. These models were built using sklearn and statsmodel libraries. There were various constraints, biggest one being the Type 2 error, which we had to minimize, as bonus variable was the main objective of this exercise. This will discuss in detail later in the report.

**Model Tuning method**

Various model tuning approaches were followed. Primarily we made use of GridSearchCV function with cv = 3 for model hyper parameter tuning. Also we had to tweak the threshold values to maximize the recall values. Threshold tweaking was required as we had a typical problem of recall precision trade off.

Various different approaches were followed to create multiple models. As mentioned earlier we had created multiple data sets like tree, tree\_scaled, tree\_smote, tree\_smote\_scaled, linear, linear\_scaled,linear\_smote, linear\_smote\_scaled etc.

We have also creating two generic functions which will be used to evaluate various models and also to tweak their threshold to maximize the recall.

Usage 1 – APPLY\_EVAL

This is used to train the model, apply the model on test set and then output all the performance metrics like confusion matrix, Classification report, AUC curve etc.

Usage 1 – TWEAK\_THRESHOLD

This is used to tweak the threshold, once the best model has been selected after hyper parameter tuning. Threshold is tweaked to maximize the recall.

Logic 2 – APPLY\_EVAL

X\_train, X\_test,y\_train & y\_test are input to the function along with the model and param grid for GCV. Model is trained, tuned then validated against the test set and performance metrics are generated.

Logic 2 – TWEAK\_THRESHOLD

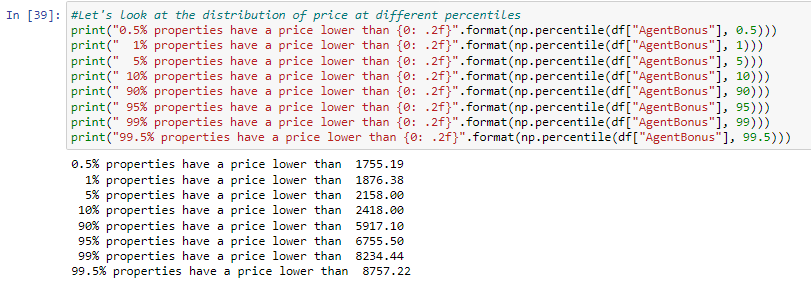
Threshold tweaking is done by calculating performance metrics like recall for all the values of probabilities between 0 and 1, and a step size of 0.1. Threshold with best AUC score is selected.

|  |  |  |
| --- | --- | --- |
| **TREE / LINEAR** | **ENSEMBLE MODELLING** | **SCALED / UNSCALED** |
| 1. We used two data sets.   * Tree * Linear   2.Tree - For Tree based models like CART, Random Forest etc.  3. Linear - For distance based models like Kmeans, LDA etc. | 1. Various ensemble models were also used apart from regular models.  2. Both Bagging and Boosting approaches were tried, evaluated and compared to determine the best model for our purpose. | 1. Some of the models were sensitive to scaling e.g. SVM, KMeans etc.  2. On the other hand we had models like Logit and other tree based models which are scaling agnostic, we used unscaled data set there. |

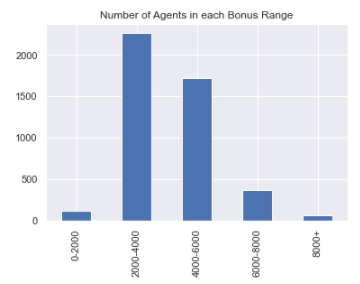
We have created multiple models as part of the Agent bonus prediction. The models include descriptive models like KMeans where we try to segment the gain insights and also predictive classification models like Random Forest, Gradient Boosting model, Logistic regression in order to predict bonus. Combined they can provide prescriptive analysis to the life insurance company and help them with the strategies.

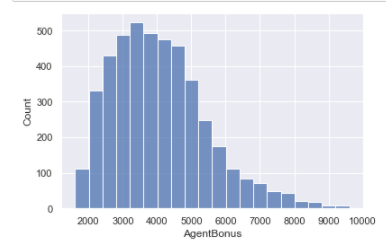
Various permutation and combinations were tried for various models.

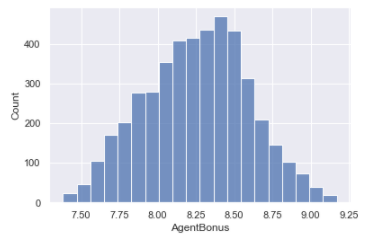
* We have included the distribution of price at different percentiles



* Let's create a range variable to understand how many records we have in different slabs





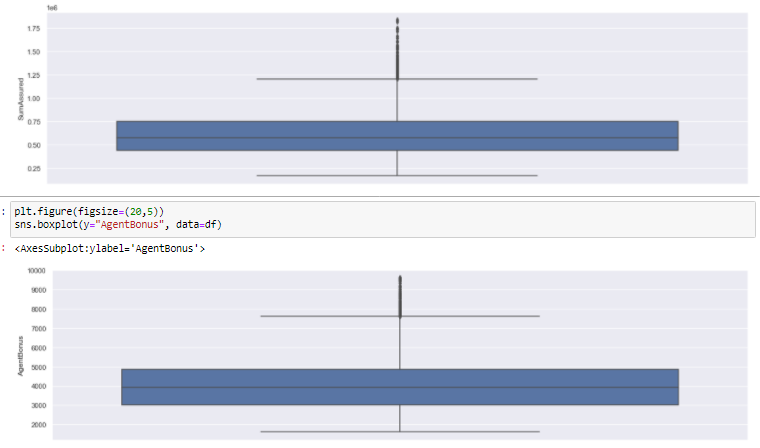


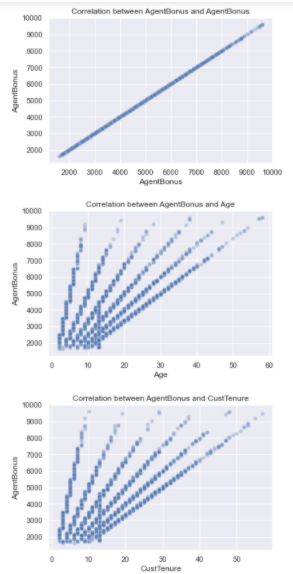
Log transformation of the AgentBonus variable looks to be slightly more symmetrically distributed. We can use a log of the AgentBonus variable as our target variable in the regression model, to check if performance is better than the AgentBonus feature used without any transformation.

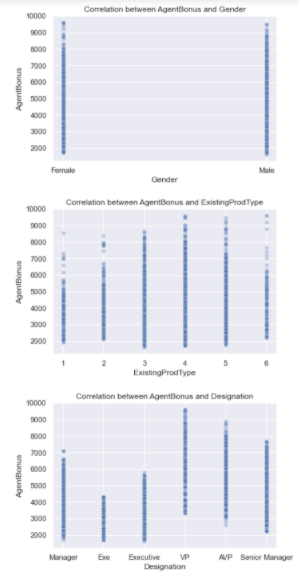
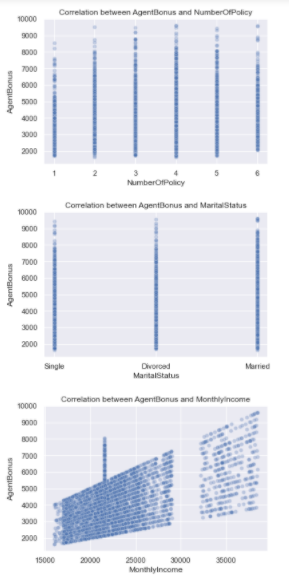
Sum assured is highly correlated to Agent Bonus - we can see it in the below table.

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **AgentBonus** | **Age** | **CustTenure** | **ExistingProdType** | **NumberOfPolicy** | **MonthlyIncome** | **Complaint** | **ExistingPolicyTenure** | **SumAssured** | **LastMonthCalls** | **CustCareScore** |
| **AgentBonus** | 1 | 0.5523 | 0.5558 | 0.113 | 0.0793 | 0.5667 | 0.0143 | 0.3491 | 0.8449 | 0.1997 | 0.0232 |
| **Age** | 0.5523 | 1 | 0.3235 | 0.0735 | 0.0468 | 0.328 | 0.0203 | 0.1915 | 0.4662 | 0.1169 | 0.0343 |
| **CustTenure** | 0.5558 | 0.3235 | 1 | 0.0828 | 0.0487 | 0.3184 | 0.0043 | 0.1928 | 0.4682 | 0.1177 | 0.0115 |
| **ExistingProdType** | 0.113 | 0.0735 | 0.0828 | 1 | 0.1499 | 0.1906 | -0.003 | 0.0593 | 0.1037 | 0.0332 | 0.0041 |
| **NumberOfPolicy** | 0.0793 | 0.0468 | 0.0487 | 0.1499 | 1 | 0.1335 | -0.016 | 0.0505 | 0.0638 | 0.0751 | -0.001 |
| **MonthlyIncome** | 0.5667 | 0.328 | 0.3184 | 0.1906 | 0.1335 | 1 | -0.005 | 0.1425 | 0.4607 | 0.3374 | 0.0356 |
| **Complaint** | 0.0143 | 0.0203 | 0.0043 | -0.003 | -0.016 | -0.005 | 1 | 0.0027 | -2E-04 | -0.026 | -0.004 |
| **ExistingPolicyTenure** | 0.3491 | 0.1915 | 0.1928 | 0.0593 | 0.0505 | 0.1425 | 0.0027 | 1 | 0.3018 | 0.0965 | -0.007 |
| **SumAssured** | 0.8449 | 0.4662 | 0.4682 | 0.1037 | 0.0638 | 0.4607 | -2E-04 | 0.3018 | 1 | 0.158 | 0.0033 |
| **LastMonthCalls** | 0.1997 | 0.1169 | 0.1177 | 0.0332 | 0.0751 | 0.3374 | -0.026 | 0.0965 | 0.158 | 1 | 0.0064 |
| **CustCareScore** | 0.0232 | 0.0343 | 0.0115 | 0.0041 | -0.001 | 0.0356 | -0.004 | -0.007 | 0.0033 | 0.0064 | 1 |

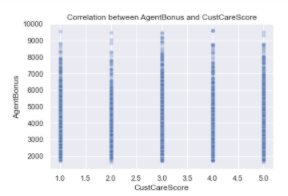
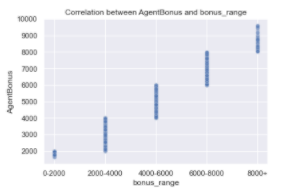
Let's check if being a SumAssured has any bonus impact

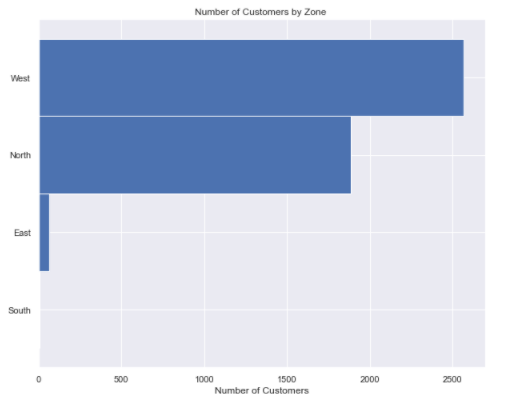


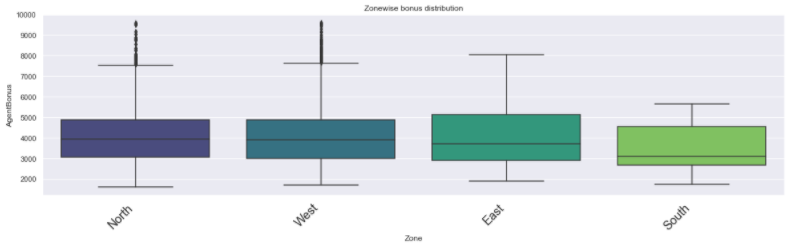
 



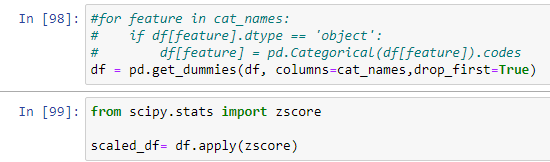


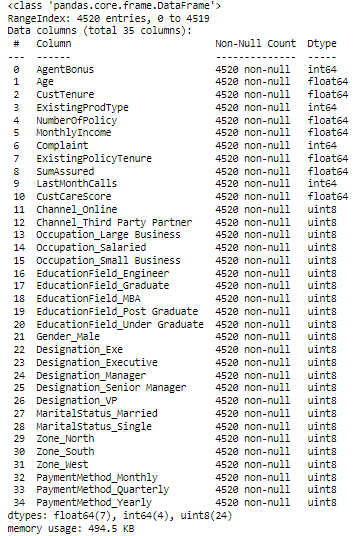
Above box plots gives us the zone wise distribution

Below table shows StandardScaler

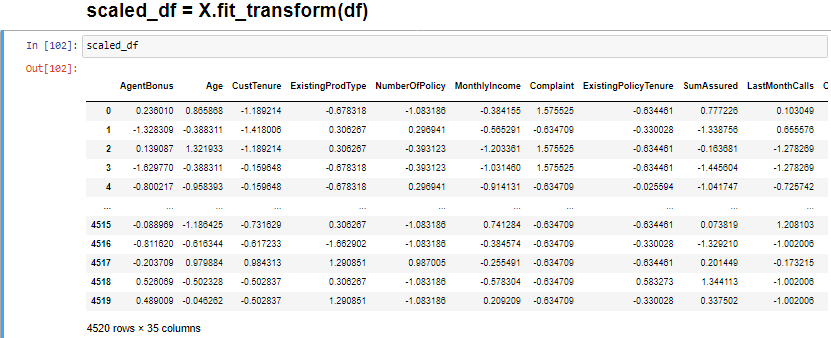


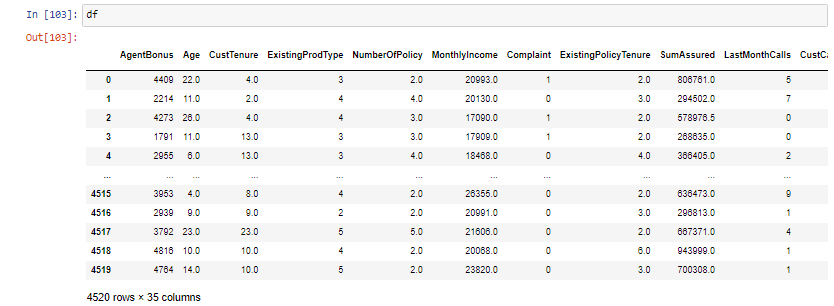
Then we apply Zscore



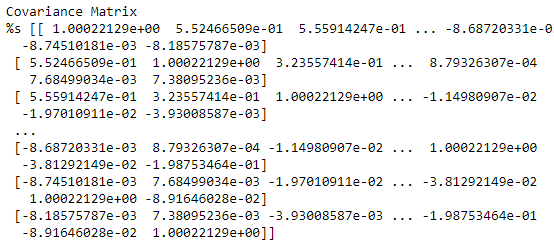


New data set - scaled\_df = X.fit\_transform(df)

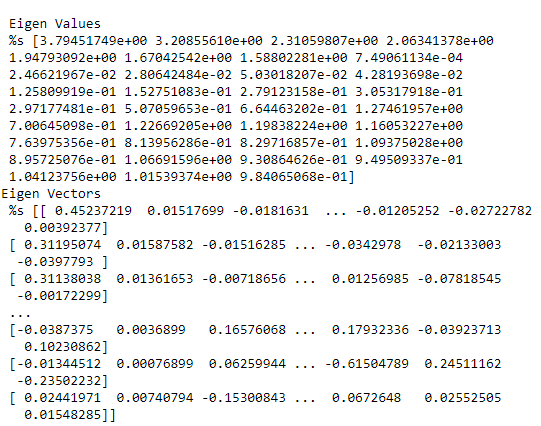
****

****

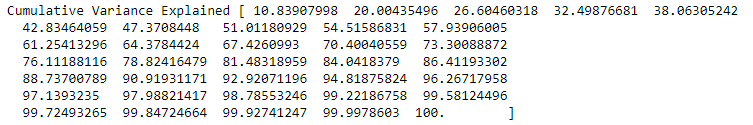
**We create the Covariance Matrix**

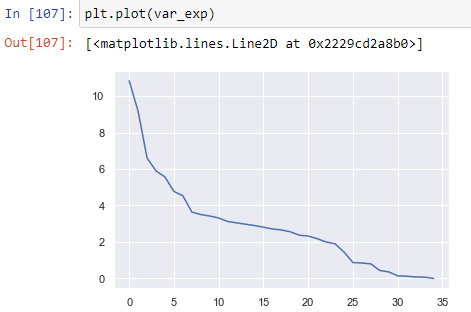


**Step 2- Get eigen values and eigen vector**

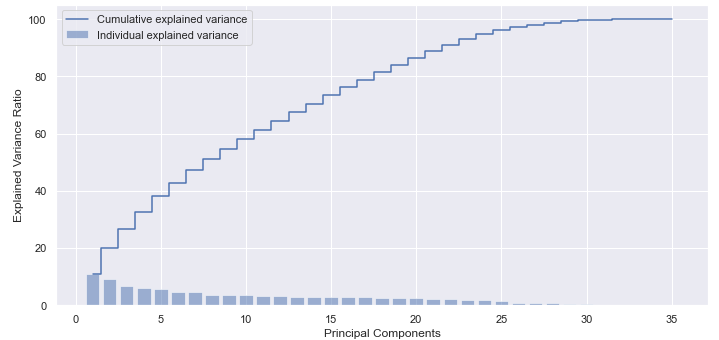


We also performed Cumulative Variance



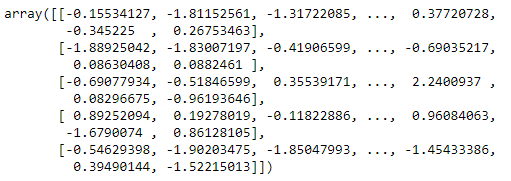


The below figure shows Individual explained variance and Cumulative explained variance plotted against Explained Variance Ratio Principal Components



Then using scikit learn PCA. It does all the above steps and maps data to PCA dimensions in one shot

**NOTE** - we are generating only 4 PCA dimensions (dimensionality reduction from 18 to 4)





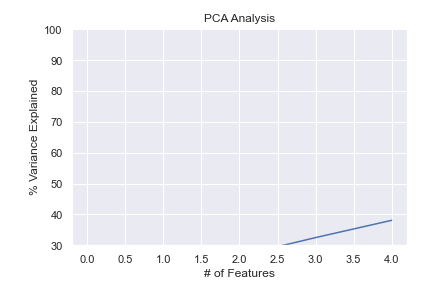
Visually we can observe that there is steep drop in variance explained with increase in number of PC's.

We will proceed with 5 components here. But depending on requirement 90% variation or 5 components will also do well.

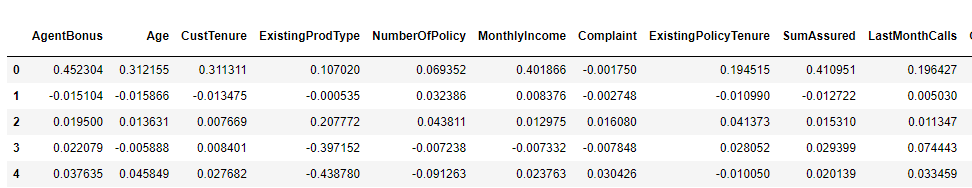
Cumulative sum of variance explained with [n] features

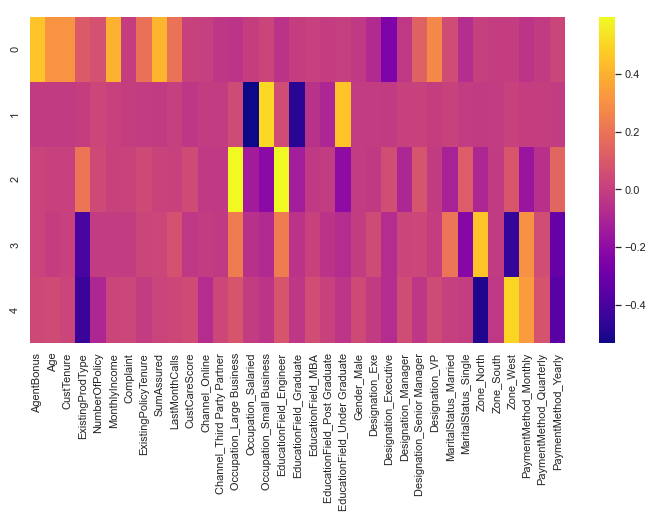


Below graph shows the PCA Analysis.



Below table gives the snapshot of scaled data frame





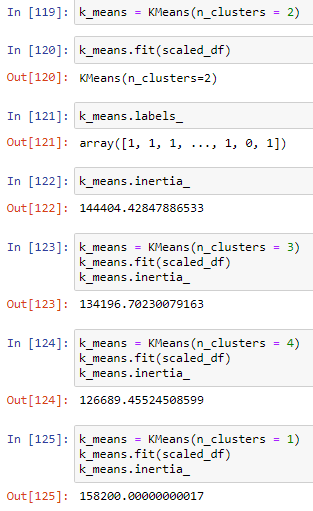
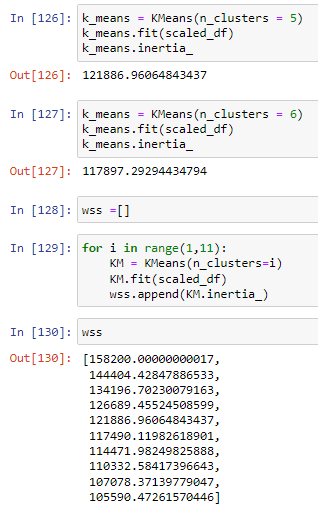
The above heat map and the color bar basically represent the correlation between the various features and the principal component itself. Component 2 looks more related to aspect - We can label it as aspect property. Depending on relations ship, we could go ahead and label relationship with features.

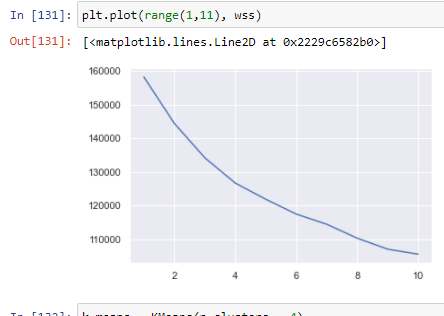
### Problem 1.b

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

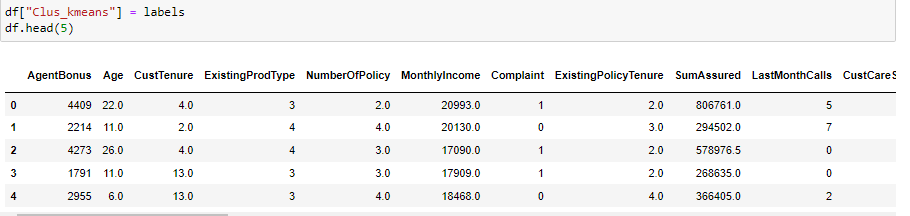
**Resolution:**

Here we start with KMeans clustering

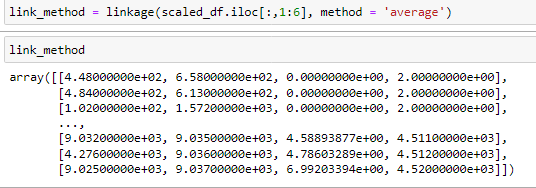


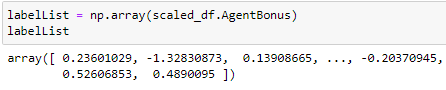
The above graph shows the WSS



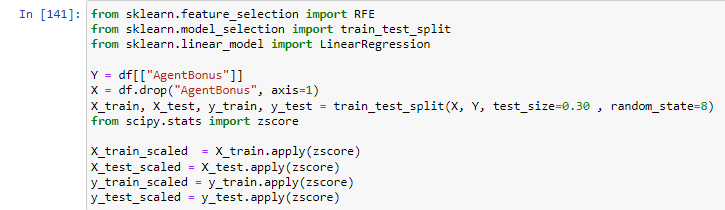
After the clustering we prepare

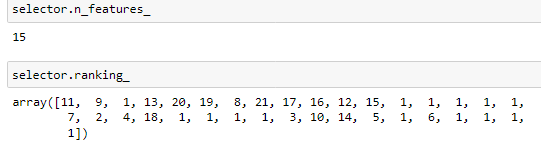




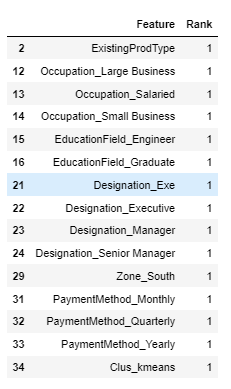


Now we create Regression Model





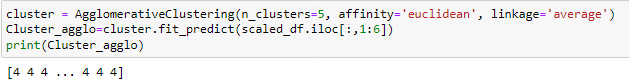
Below table gives us the feature and the rank



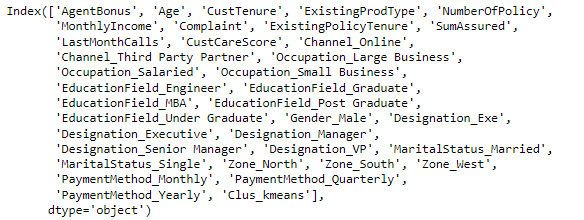
Most important features are Designation, Occupation, South zone, Payment Method, Existing Prod Type, Education Field,

Zone west and North, Existing Policy Tenure, Cust Tenure, Age

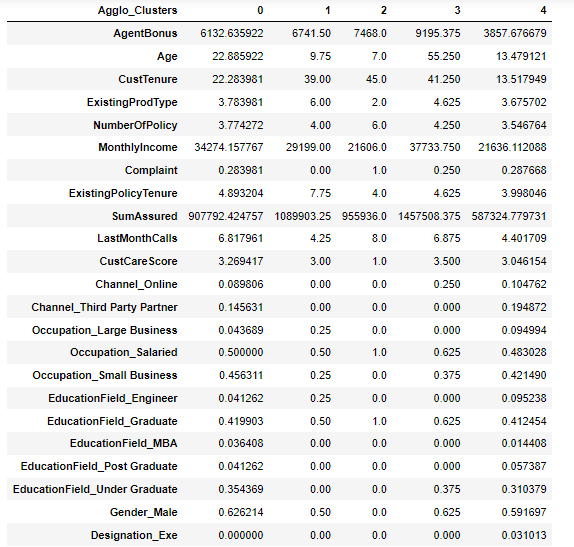
We have also created Agglomerative Clustering

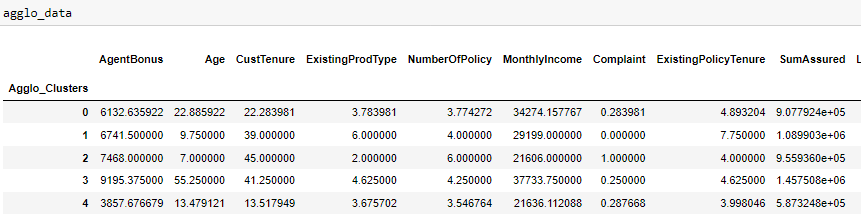


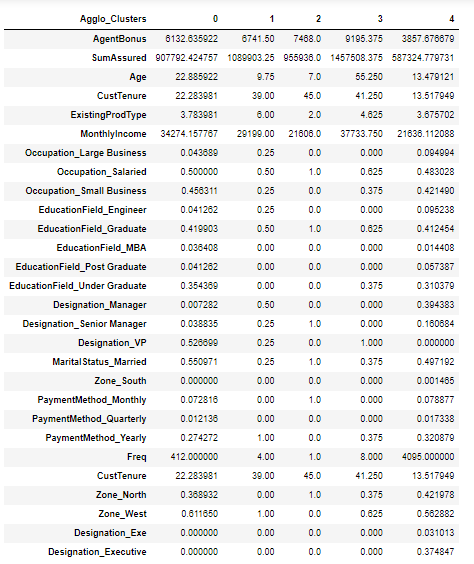
We then drop df.drop(columns=['Agglo\_CLusters'],inplace=True)



Below table shows the grouping by Agglo\_Clusters



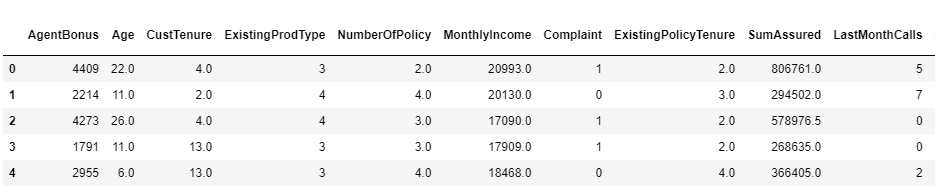




Now we get the silhouette score

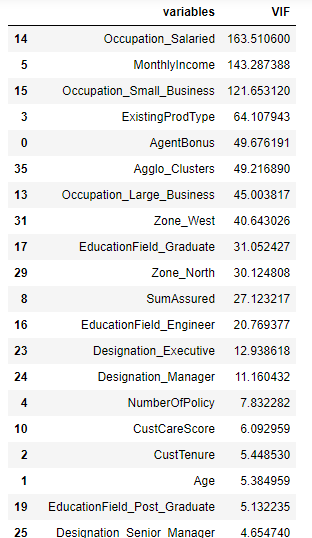
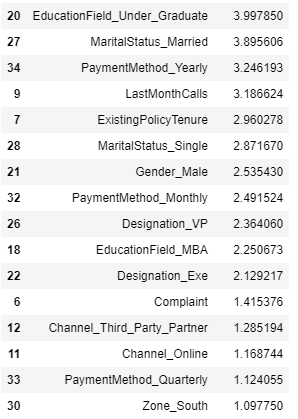


Below table shows the data with Sil\_width



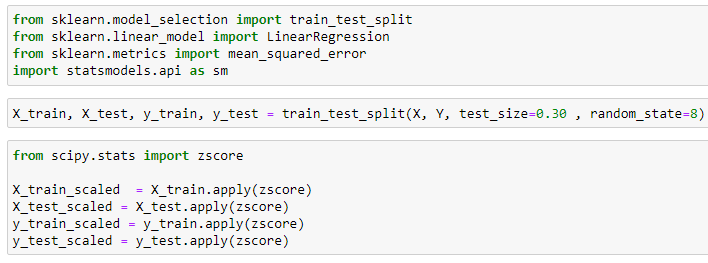


Next we calculate variance inflation factor

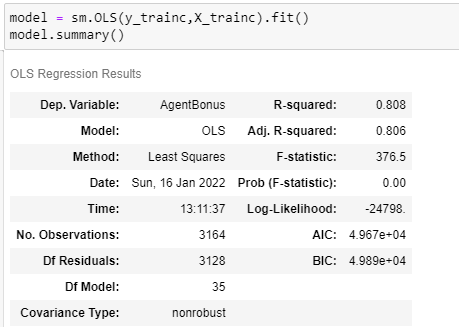
 

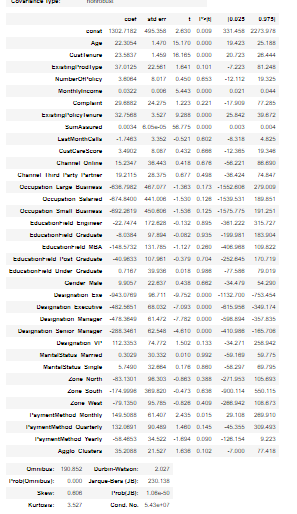
Here, we see that the value of VIF is high for many variables. Here, we may drop variables with VIF more than 5 (very high correlation) & build our model

Now we split the data



**Using Statsmodels OLS**





Notes:

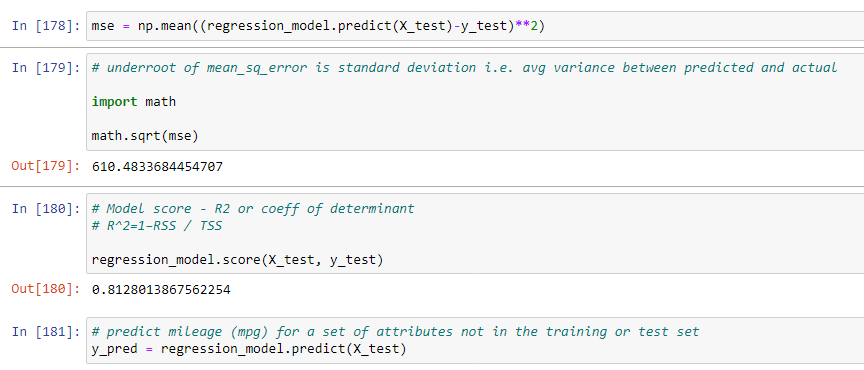
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

[2] The condition number is large, 5.43e+07. This might indicate that there are

strong multicollinearity or other numerical problems.

Then we create another Regression Model

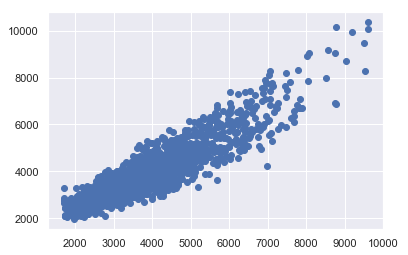




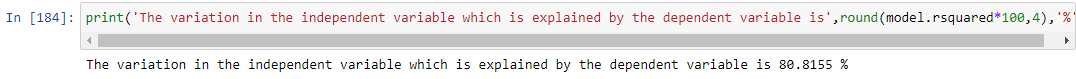
Predict mileage (mpg) for a set of attributes not in the training or test set

Since this is regression, plot the predicted y value vs actual y values for the test data

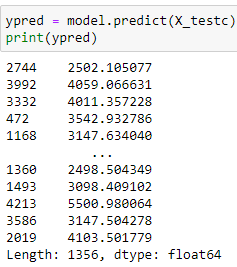
A good model's prediction will be close to actual leading to high R



Now we get the value of coefficient of determination

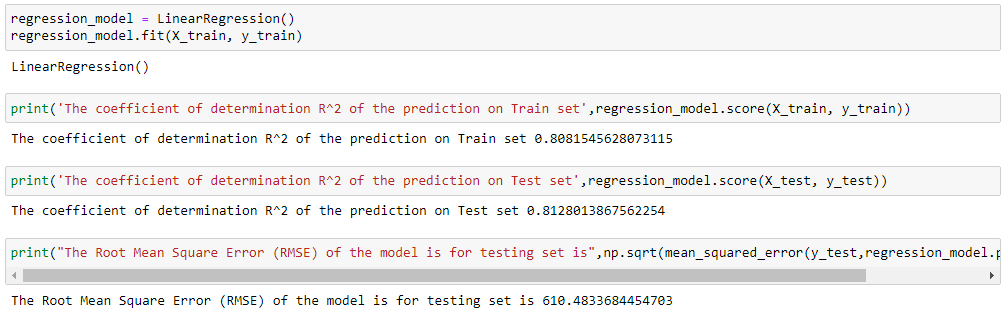


Then we get the Predictions on test set

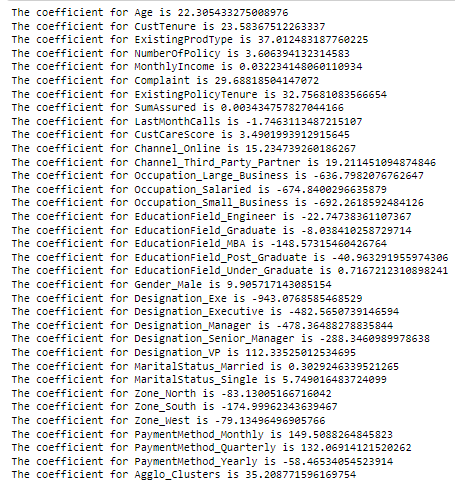


The Root Mean Square Error

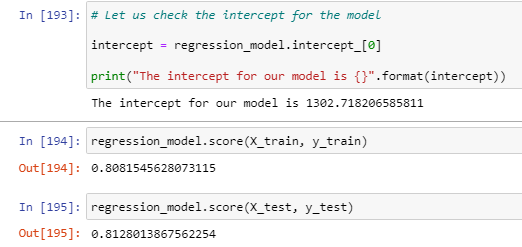


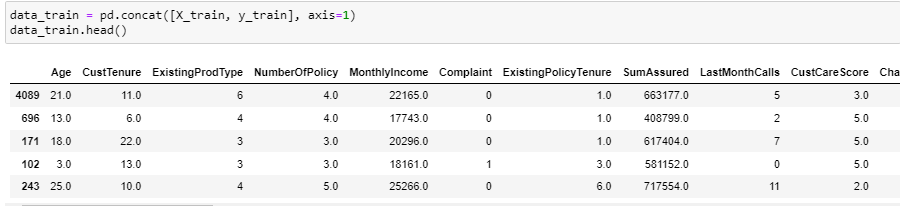


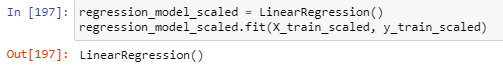
Let us explore the coefficients for each of the independent attributes



Now let us check the intercept for the model

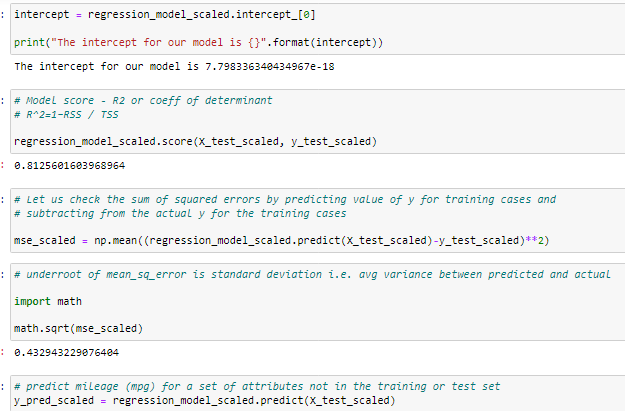






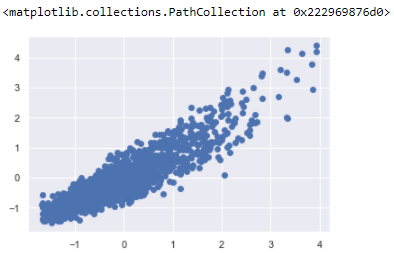
Let us explore the coefficients for each of the independent attributes



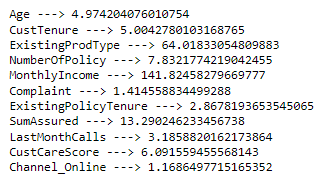


Predict mileage (mpg) for a set of attributes not in the training or test set

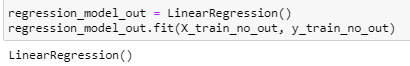
Since this is regression, plot the predicted y value vs actual y values for the test data. A good model's prediction will be close to actual leading to high R and R2 values



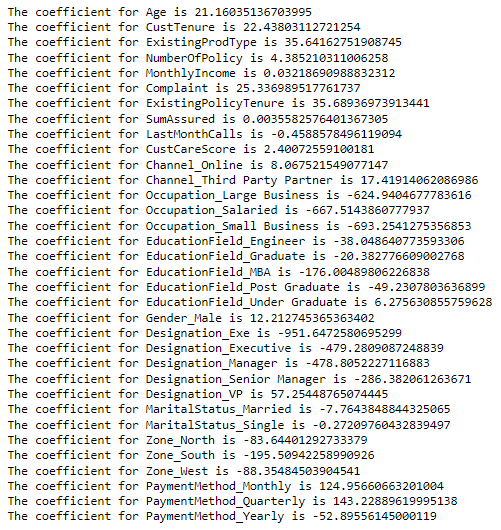
We again check the variance inflation factor

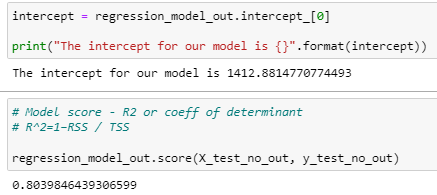


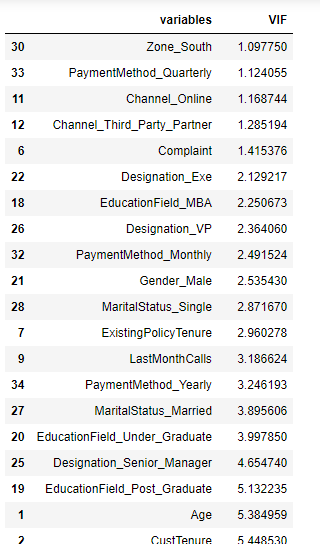
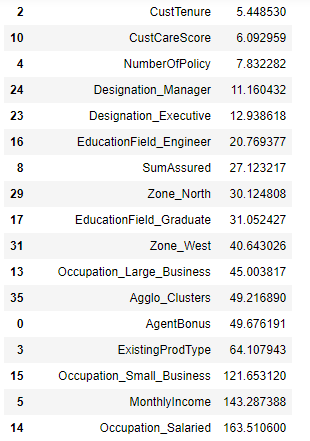
Let’s invoke the LinearRegression function and find the best fit model on training data.



Now let us explore the coefficients for each of the independent attributes





### Problem 1.c

Interpretation of the model(s)

**Resolution:**

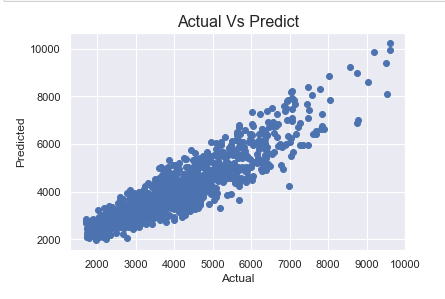
The final Linear Regression equation is

AgentBonus = b0 \* Intercept + b1 \* Zone\_South + b2 \* PaymentMethod\_Quarterly + b3 \* Channel\_Online + b4 \* Channel\_Third\_Party\_Partner + b5 \* Complaint + b6 \* Designation\_Exe + b7 \* Designation\_VP + b8 \* EducationField\_MBA + b9 \* PaymentMethod\_Monthly + b10 \* Gender\_Male + b11 \* MaritalStatus\_Single + b12 \* ExistingPolicyTenure + b13 \* LastMonthCalls + b14 \* PaymentMethod\_Yearly + b15 \* Designation\_Senior\_Manager + b16 \* MaritalStatus\_Married + b17 \* EducationField\_Under\_Graduate + b18 \* EducationField\_Post\_Graduate + b19 \* Age + b20 \* CustTenure + b21 \* CustCareScore + b22 \* NumberOfPolicy + b23 \* Designation\_Manager + b24 \* Designation\_Executive + b25 \* EducationField\_Engineer + b26 \* SumAssured + b27 \* Zone\_North + b28 \* EducationField\_Graduate + b29 \* Occupation\_Large\_Business + b30 \* Zone\_West + b31 \* ExistingProdType + b32 \* MonthlyIncome + b33 \* Occupation\_Small\_Business

AgentBonus = (843.69) \* Intercept + (-0.74) \* Zone\_South + (121.99) \* PaymentMethod\_Quarterly + (16.45) \* Channel\_Online + (8.64) \* Channel\_Third\_Party\_Partner + (31.43) \* Complaint + (-897.0) \* Designation\_Exe + (19.69) \* Designation\_VP + (-134.73) \* EducationField\_MBA + (179.77) \* PaymentMethod\_Monthly + (3.74) \* Gender\_Male + (13.12) \* MaritalStatus\_Single + (35.55) \* ExistingPolicyTenure + (-2.62) \* LastMonthCalls + (-71.13) \* PaymentMethod\_Yearly + (-261.65) \* Designation\_Senior\_Manager + (-16.47) \* MaritalStatus\_Married + (15.54) \* EducationField\_Under\_Graduate + (-76.12) \* EducationField\_Post\_Graduate + (23.04) \* Age + (24.0) \* CustTenure + (6.01) \* CustCareScore + (1.52) \* NumberOfPolicy + (-447.69) \* Designation\_Manager + (-469.53) \* Designation\_Executive + (-39.86) \* EducationField\_Engineer + (0.0) \* SumAssured + (-44.05) \* Zone\_North + (-63.73) \* EducationField\_Graduate + (-39.31) \* Occupation\_Large\_Business + (-44.97) \* Zone\_West + (54.11) \* ExistingProdType + (0.03) \* MonthlyIncome + (-92.01) \* Occupation\_Small\_Business

When Age increases by 1 unit, AgentBonus increases by 23.04 units, keeping all other predictors constant. similarly, when MonthlyIncome increases by 1 unit, AgentBonus increases by 0.03 units, keeping all other predictors constant.

There are also some negative co-efficient values. Occupation\_Large\_Business has its corresponding co-efficient as -39.31. This implies, when the Occupation is Large business, the AgentBonus decreases by 39.31 units, keeping all other predictors constant.



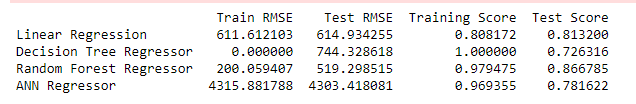
## Problem 2: Model Tuning and business implication

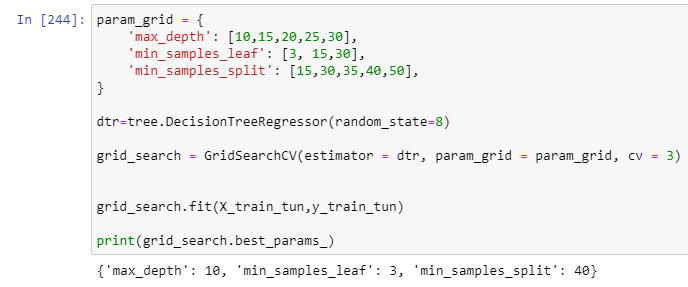
### Problem 2.a

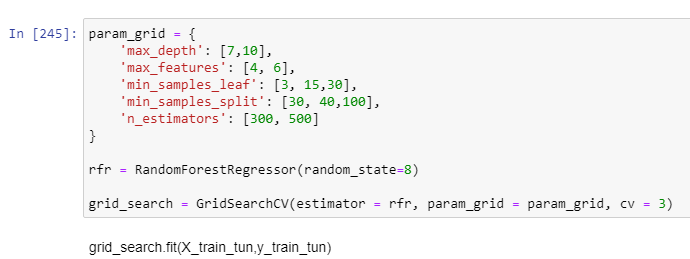
Ensemble modelling (if necessary)

**Resolution:**

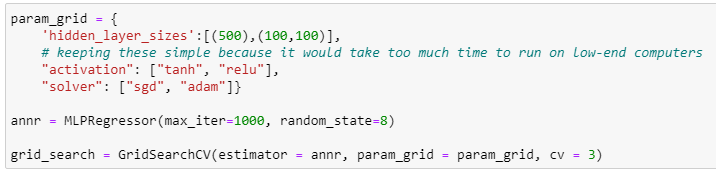
We are scaling the data for ANN. Without scaling it will give very poor results. Computations becomes easier.

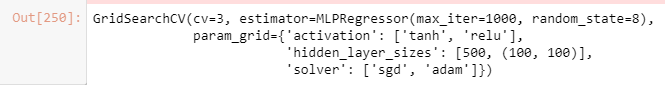


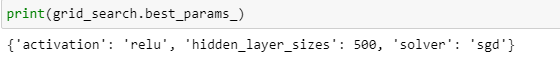




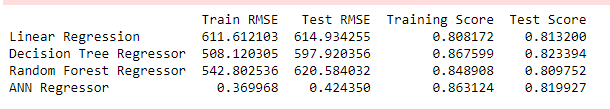




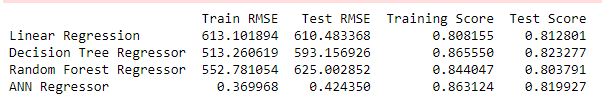




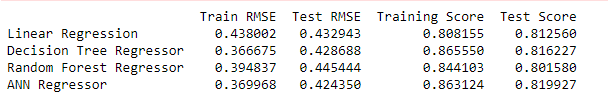
**best\_params\_annr={'activation': 'relu', 'hidden\_layer\_sizes': 500, 'solver': 'sgd'}**



**Without tuning**



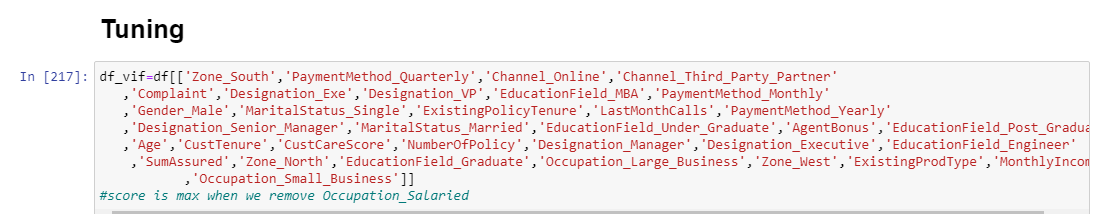
**Final Output**

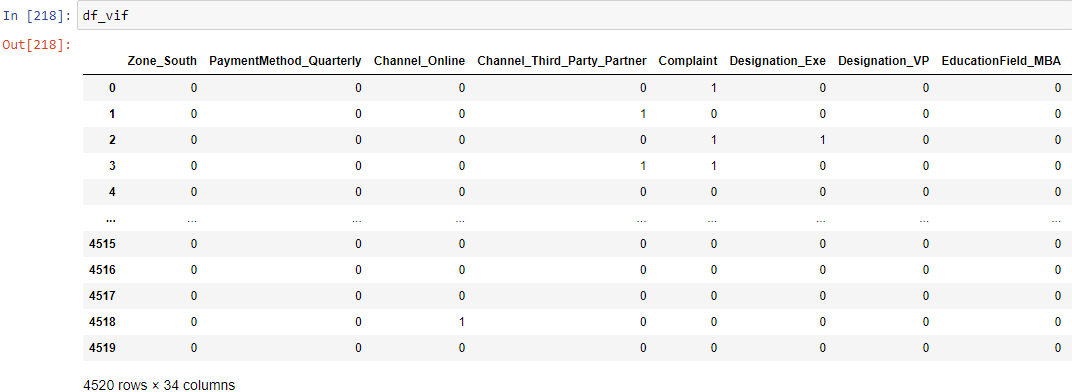


### Problem 2.b

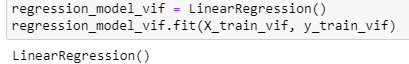
Any other model tuning measures (if applicable)

**Resolution:**

****

****

We now invoke the LinearRegression function and find the best fit model on training data



Let us explore the coefficients for each of the independent attributes



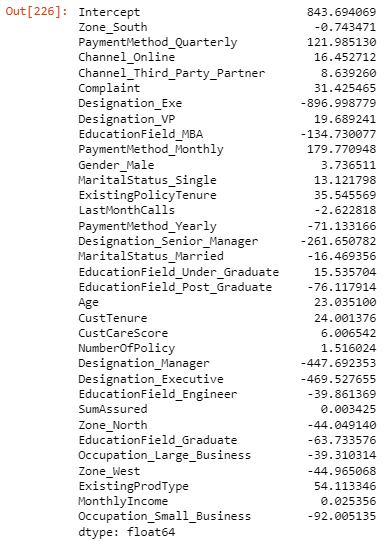
Model score - R2 or coeff of determinant

R^2=1–RSS / TSS

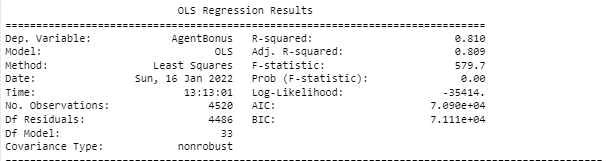
0.8131119381670872

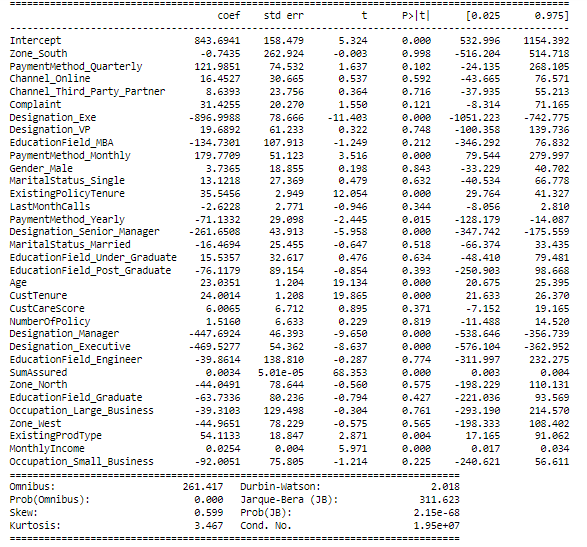
**We can see that the scaled output has a better score.**

Now we see stats model formula



lm1.summary



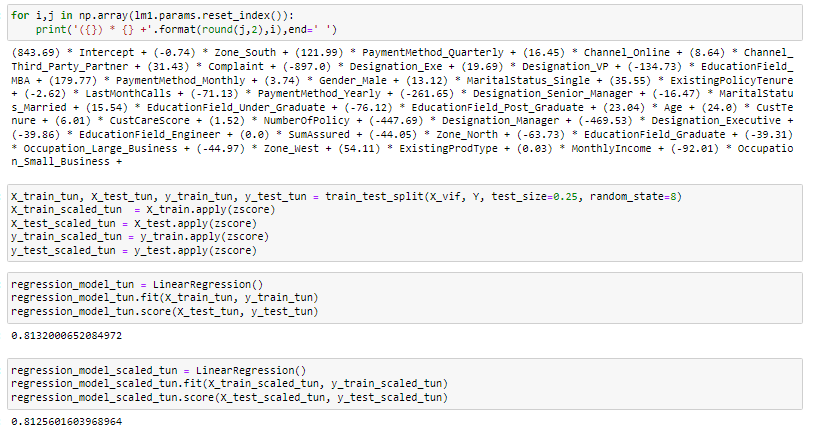


Notes:

[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

[2] The condition number is large, 1.95e+07. This might indicate that there are

strong multicollinearity or other numerical problems.



The overall P value is less than alpha, so rejecting H0 and accepting Ha that at least 1 regression co-efficient is not 0. Here all regression co-efficients are not 0

### Problem 2.c

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

**Resolution:**

When Age increases by 1 unit, AgentBonus increases by 23.04 units, keeping all other predictors constant. similarly, when MonthlyIncome increases by 1 unit, AgentBonus increases by 0.03 units, keeping all other predictors constant.

There are also some negative co-efficient values. Occupation\_Large\_Business has its corresponding co-efficient as -39.31. This implies, when the Occupation is large business, the AgentBonus decreases by 39.31 units, keeping all other predictors constant.

The End

Thakur Arun Singh

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*