Problem Statement –

Vahan Bima is one of the leading insurance companies in India. It provides motor vehicle insurances at best prices with 24/7 claim settlement. It offers different types of policies for both personal and commercial vehicles. It has established its brand across different regions in India.

Around 90% of the businesses today use personalized services. The company wants to launch different personalized experience programs for customers of Vahan Bima. The personalized experience can be dedicated resources for claim settlement, different kinds of services at doorstep, etc. In order to do so, they would like to segment the customers into different tiers based on their customer lifetime value (CLTV).

In order to do it, they would like to predict the customer lifetime value based on the activity and interaction of the customer with the platform. So, as a part of this challenge, your task at hand is to build a high performance and interpretable machine learning model to predict the CLTV based on the user and policy data.

Data collection –

You are provided with the sample dataset of the company holding the information of customers and policy such as highest qualification of the user, total income earned by a customer in a year, employee status, policy opted by the user, type of policy and so on and the target variable indicating the total customer lifetime value.

You are provided with around 90K records containing the attributes of the user and policy and the target variable *cltv* indicating the total customer lifetime value.

EDA :-

- First we checked the information of dataset like data type of attributes , null values , total number of samples or rows
- We got total 11 independent attributes and a dependent attribute(Target Variable)
  - In that 11 independent attributes we got 7 object data types and 4 integer data type
  - A target variable with high cardinality or continuous data in integer data type
- So first we have to perform EDA like
  - Data type conversion
  - Checking null values > we checked it using .info() itself
  - Handling outliers
  - Visualizing data to check spread
- So the approach for each attribute is as follows -

Variable	Description	Approach
id	Unique identifier of a customer	Left as it is because all unique with no correlation to Target Variable They only act like an indexes
gender	Gender of the customer	Has two unique values **male and female** So we replace it with binary 0 and 1 simply using replace function in pandas To convert attribute dtype into int and also checked null values

area	Area of the customer	Has two unique values **Urban and Rural** So we replace it with binary 0 and 1 simply using replace function in pandas To convert attribute dtype into int and also checked null values
qualification	Highest Qualification of the customer	Has three unique values **high school, bachelors and others** So we used One Hot Encoding to create dummies and separate attributes of unique value with values 0 and 1 To convert attribute dtype into int
income	Income earned in a year (in rupees)	Has three unique values **high school, bachelors and others** So we used One Hot Encoding to create dummies and separate attributes of unique value with values 0 and 1 To convert attribute dtype into int
marital_status	Marital Status of the customer {0:Single, 1: Married}	Has two unique values **Single and Married** So we replace it with binary 0 and 1 simply using replace function in pandas To convert attribute dtype into int and also checked null values
vintage	No. of years since the first policy date	Has four unique values (**5L-10L) ,(2L-5L), (More than 10L) and (<=2L**) So we used One Hot Encoding to create dummies and separ ate attributes of unique value(prefix = income) with values 0 and 1 To convert attribute dtype into int *pd.get_dummies(x_train['income'],prefix='income')* **also** if we want to decrease the unique values count you can simply take 3 conditions like 1. income less than 5L 2. income between 5L and 10L 3. income more than 10L by replacing two unique value **(<=2L**) and **,(2L-5L)** with same v alue using *x_train['income'].replace({'<=2L':0,'2L-5L':0,'5L-10L':1,'More than 10L':2},inpl ace=True)*
claim_amount	Total Amount Claimed by the customer (in rupees)	This attribute has continuous values with zeros at **more than 5 percent** of all records and also had **outliers up to 5 percent of** of all records. So we checked the flow and plotted some graphs using seaborn library found out that there is pattern in zeros in claim amount to the Target variable **CLTV**. So we only replaced outliers using boxplot to detect outliers with are having value more than 10000 with median value of claim amount attribute because it has no impact of outlier
num_policies	Total no. of policies issued by the customer	Has two unique values **More than 1 and 1** So we replace it with binary 0 and 1 simply using replace function in pandas To convert attribute dtype into int and also checked null values *x_train['num_policies'].replace({'More than 1':1,'1':0},inplace=True*)
policy	Active policy of the customer	Has three unique values **A, B and C with** bad correlation with Target Variable So we simply replaced these unique values with 0,1 and 2 simply using replace function in pandas To convert attribute dtype into int and also checked null values *x_train['policy'].replace({'A':0,'B':1,'C':2},inplace=True)*
type_of_policy	Type of active policy	Has three unique values **Platinum,Silver,Gold with** bad correlation with Target Variable So we simply replaced these unique values with

		0,1 and 2 simply using replace function in pandas To convert attribute dtype into int and also checked null values x_train['type_of_policy'].replace({'Platinum':2,'Silver':0,'Gold':1},inplace=True)
cltv	*Customer life time value (Target Variable)*	*Our TARGET VARIABLE* had continuous values and no null values so kept it as it is

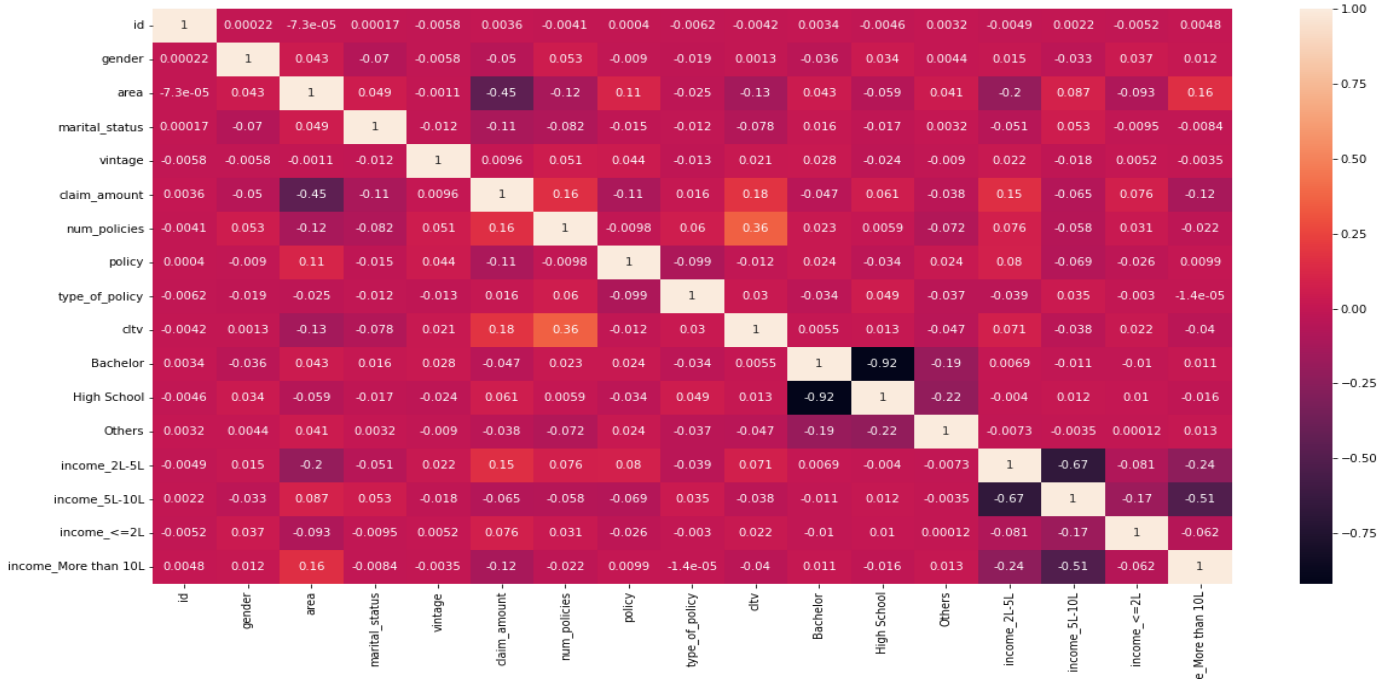
- model selection -

As our target variable has continuous type of data and our evaluation parameter is r2score we should using regression model so I used Linear Regression Model which obtained up to 14 percent accuracy

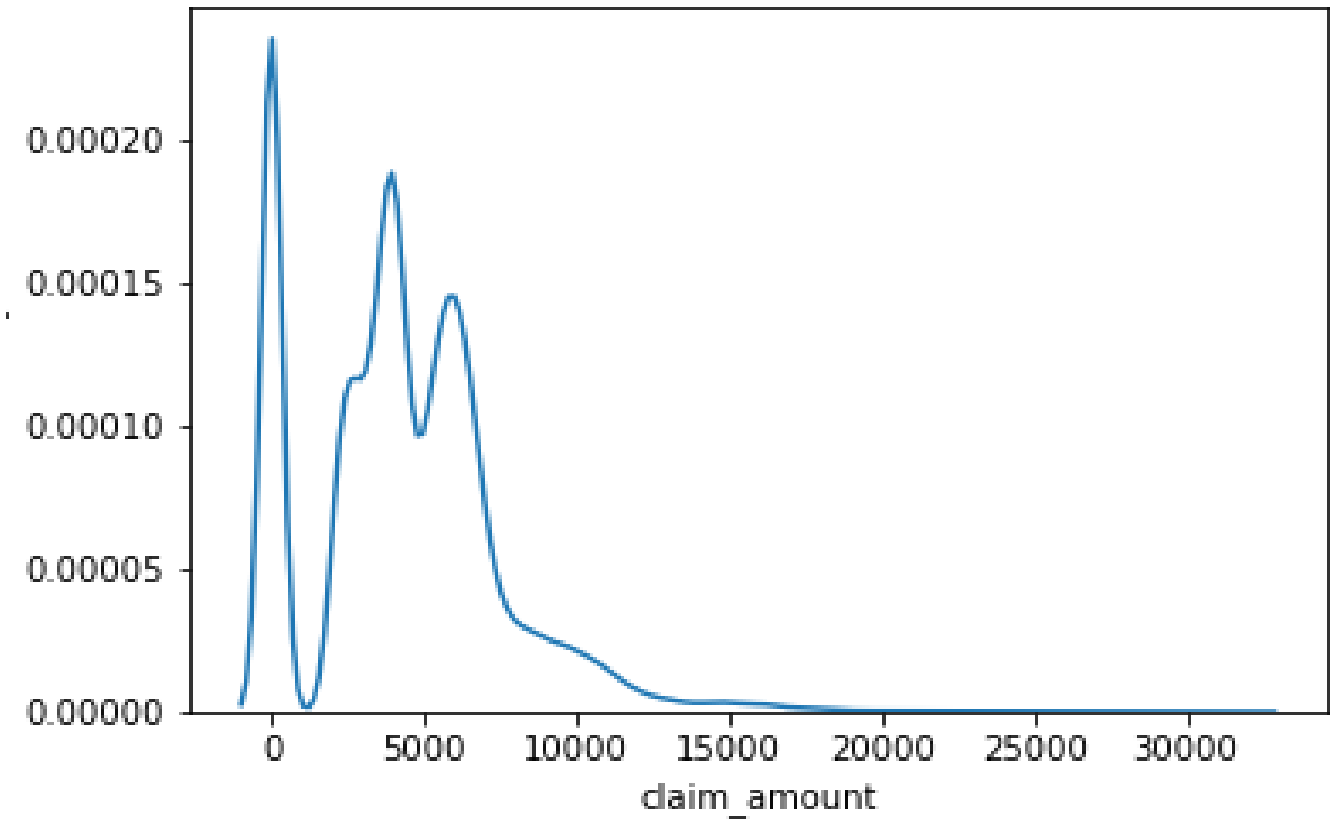
To improve the accuracy of our model I tried following ways –

1. I performed EDA on data and removed outliers from Target variable which is not a good idea because it cause over fitting
2. Then I replaced zeros in claim amount attribute to our median values
3. Then I replaced zeros in claim amount attribute to our mean values
4. Also I replaced outliers in claim amount attribute with median and then replaced zeros to our mean values
5. But all these the first model has given more accuracy of all

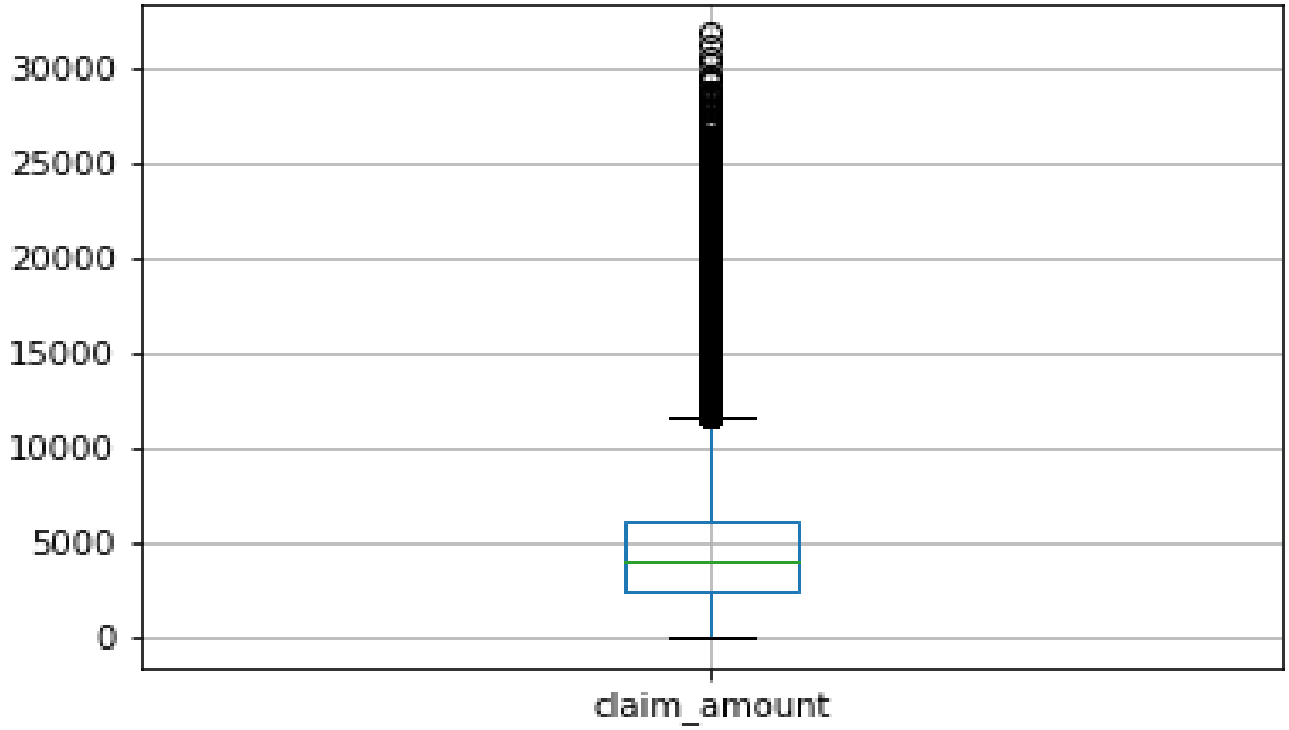
- **Plots for Visualization of Data -**
  - **correlation heatmap**



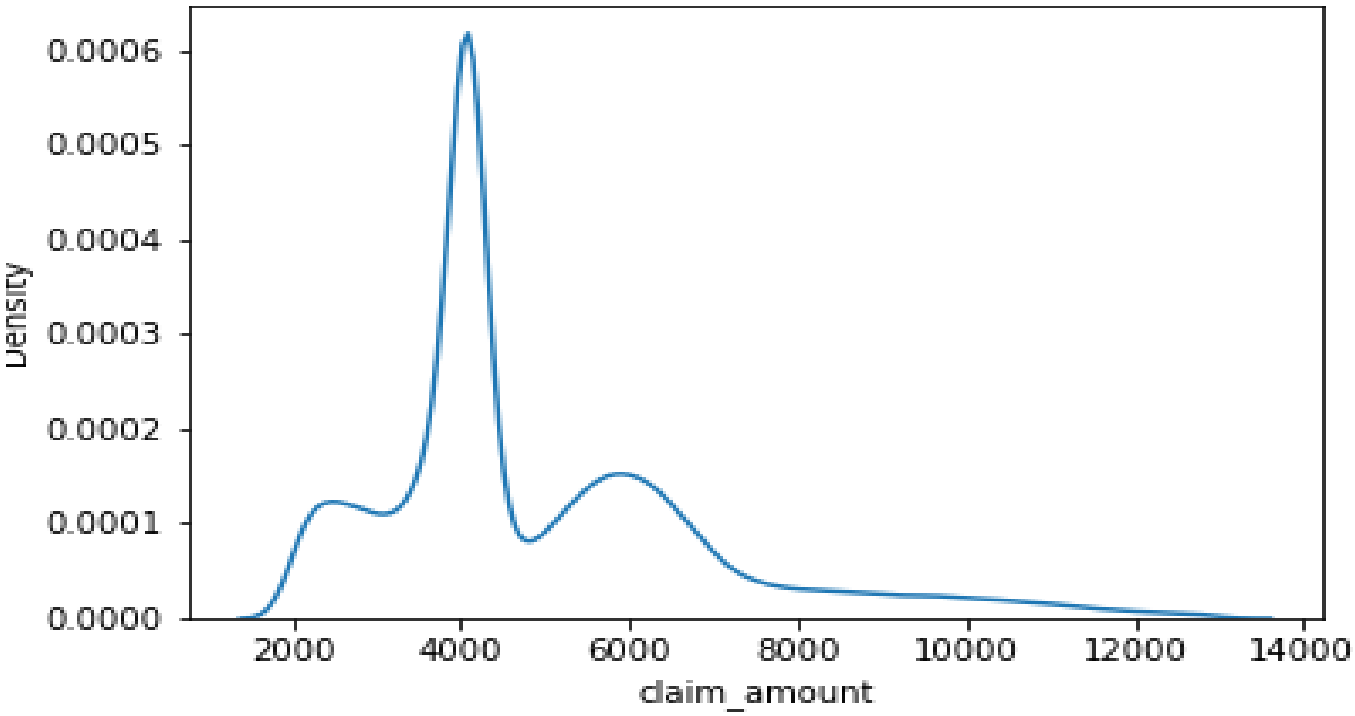
- **claim amount spread density**
  - **before**



- **claim amount outliers**



- **kde plot after replacing zeros with median in claim amount and removing outliers**



- **joint plot of claim amount and cltv after replacing zeros with median in claim amount and removing outliers**