# FINAL PROJECT REPORT

# MACHINE LEARNING

## Objective:

To build a suitable Machine Learning model for various data sets.

## Problem Statement 1:

| CPI | discounts | offers | Sales |
|---|---|---|---|
| 2600 | 3 | 20 | 550000 |
| 3000 | 4 | 15 | 565000 |
| 3200 | 5 | 18 | 610000 |
| 3600 | 3 | 30 | 595000 |
| 4000 | 5 | 8 | 760000 |
| 4100 | 6 | 8 | 810000 |

Given below information find out the Sales that has

5000 cpi , 3 percentage discounts, 20 rewards offers

4000 cpi , 8 percentage discounts, 19 rewards offers

## Solution:

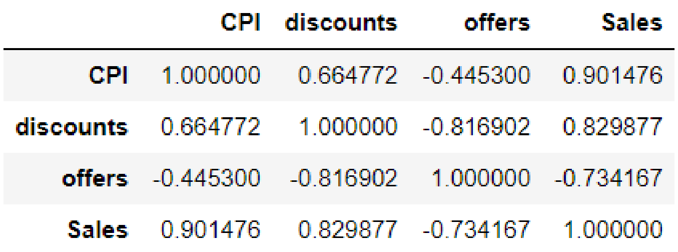
For the above problem and data set the Linear Regression model is suitable. Because these data points have a good linear relationship between variables and this data set have an one dependent variable and three independent variables.

Regression: Relationship between dependent variables and independent variables

Regression model equation

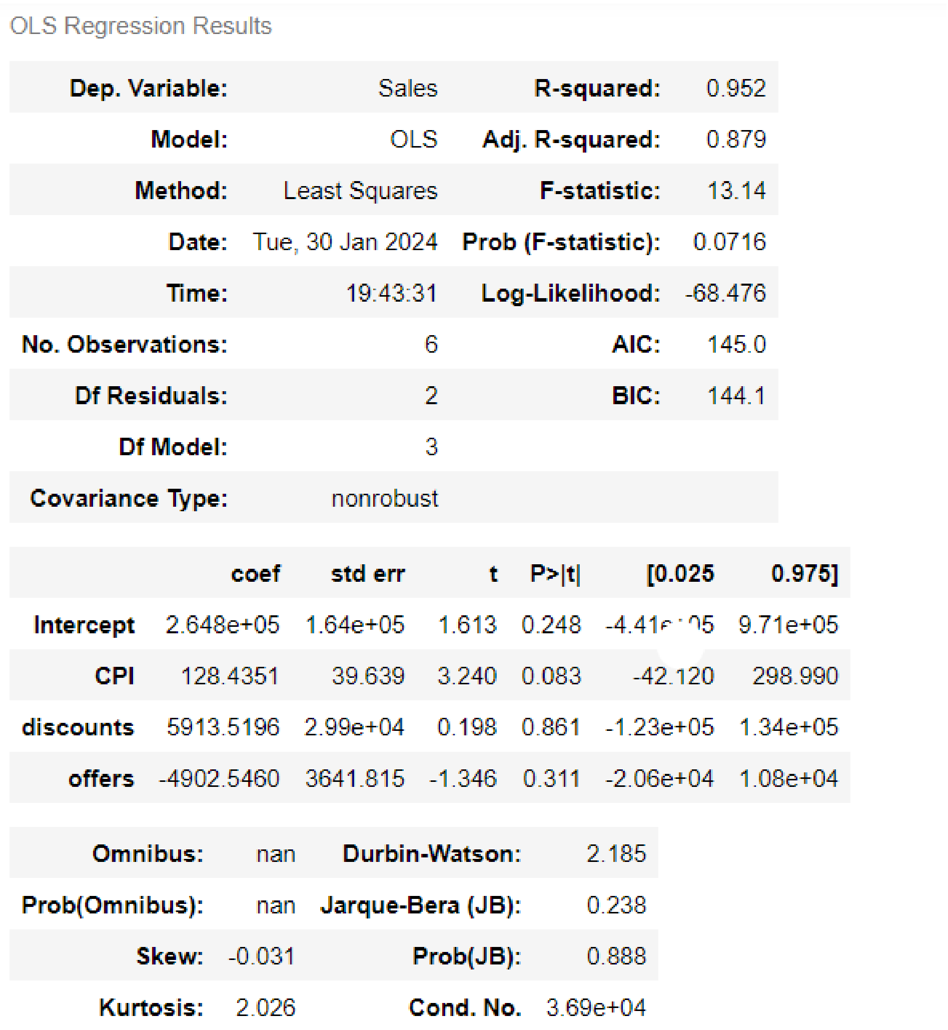
$$Y=mx+c$$

## Correlation Checking:



|  | CPI | discounts | offers | Sales |
|---|---|---|---|---|
| **CPI** | 1.000000 | 0.664772 | -0.445300 | 0.901476 |
| **discounts** | 0.664772 | 1.000000 | -0.816902 | 0.829877 |
| **offers** | -0.445300 | -0.816902 | 1.000000 | -0.734167 |
| **Sales** | 0.901476 | 0.829877 | -0.734167 | 1.000000 |

## Linear Regression model Summary:



OLS Regression Results

| | | | |
|---|---|---|---|
| **Dep. Variable:** | Sales | **R-squared:** | 0.952 |
| **Model:** | OLS | **Adj. R-squared:** | 0.879 |
| **Method:** | Least Squares | **F-statistic:** | 13.14 |
| **Date:** | Tue, 30 Jan 2024 | **Prob (F-statistic):** | 0.0716 |
| **Time:** | 19:43:31 | **Log-Likelihood:** | -68.476 |
| **No. Observations:** | 6 | **AIC:** | 145.0 |
| **Df Residuals:** | 2 | **BIC:** | 144.1 |
| **Df Model:** | 3 | | |
| **Covariance Type:** | nonrobust | | |

| | **coef** | **std err** | **t** | **P>\|t\|** | **[0.025** | **0.975]** |
|---|---|---|---|---|---|---|
| **Intercept** | 2.648e+05 | 1.64e+05 | 1.613 | 0.248 | -4.41e+05 | 9.71e+05 |
| **CPI** | 128.4351 | 39.639 | 3.240 | 0.083 | -42.120 | 298.990 |
| **discounts** | 5913.5196 | 2.99e+04 | 0.198 | 0.861 | -1.23e+05 | 1.34e+05 |
| **offers** | -4902.5460 | 3641.815 | -1.346 | 0.311 | -2.06e+04 | 1.08e+04 |

| | | | |
|---|---|---|---|
| **Omnibus:** | nan | **Durbin-Watson:** | 2.185 |
| **Prob(Omnibus):** | nan | **Jarque-Bera (JB):** | 0.238 |
| **Skew:** | -0.031 | **Prob(JB):** | 0.888 |
| **Kurtosis:** | 2.026 | **Cond. No.** | 3.69e+04 |

From above result we can know about the R-Square, Adjacent R-Square, correlation values.

### Linear Regression Model Building:

1. Collect the data
2. Preprocessing the data
3. Analyze the data
4. Select and split the dependent variable and independent variables
5. Make the data to X_train,X_test,y_train,y_test split
6. Fit the model.
7. Train and test the model
8. Evaluate and predict the model

**New Data Points:**

5000 cpi , 3 percentage discounts, 20 rewards offers

Solution: `array ([826645.34838222])`

4000 cpi , 8 percentage discounts, 19 rewards offers

Solution: `array([732680.36486005])`

By providing the 5000 cpi, 3 percentage discounts and 20 rewards offers is a good Result in Sales, but it slightly reduces the sales while providing 4000 cpi, 8 percentage And 19 rewards offers.

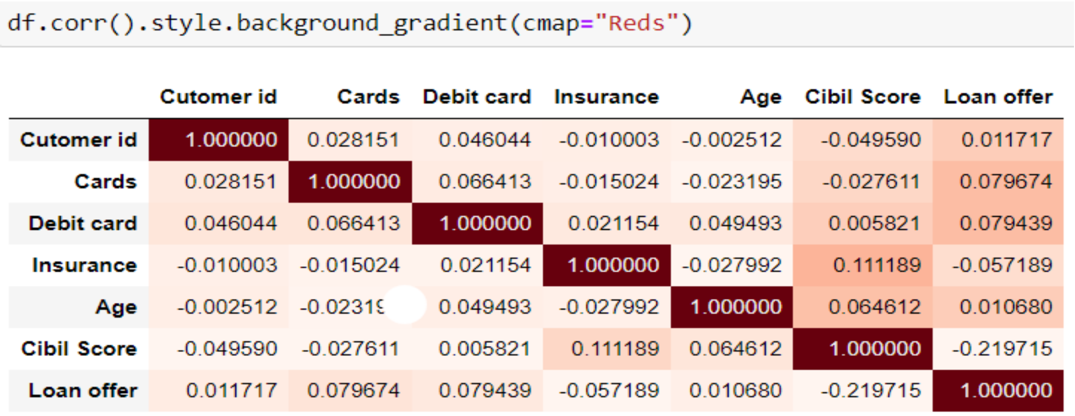
## Problem Statement 2:

| Cutomer id | Cards | Debit card | Insurance | Age | Cybill Score | Loan offer |
|---:|---:|---:|---:|---:|---:|---:|
| 5 | 0 | 1 | 0 | 50 | 34.94 | 0 |
| 3 | 1 | 0 | 0 | 18 | 0.891 | 1 |
| 66 | 0 | 1 | 0 | 5 | 0.33 | 1 |
| 70 | 0 | 1 | 1 | 31 | 0.037 | 0 |
| 96 | 0 | 1 | 0 | 30 | 0.038 | 1 |

## Solution:

For the above problem and data set the Logistic Regression model is suitable. Because dependent variables have an BINOMIAL data 0 and 1. This type of data or Categorical like yes or no predictions is suitable for Logistic Regression.

## Correlation Checking:



```
df.corr().style.background_gradient(cmap="Reds")
```

|  | Cutomer id | Cards | Debit card | Insurance | Age | Cibil Score | Loan offer |
|---|---:|---:|---:|---:|---:|---:|---:|
| **Cutomer id** | 1.000000 | 0.028151 | 0.046044 | -0.010003 | -0.002512 | -0.049590 | 0.011717 |
| **Cards** | 0.028151 | 1.000000 | 0.066413 | -0.015024 | -0.023195 | -0.027611 | 0.079674 |
| **Debit card** | 0.046044 | 0.066413 | 1.000000 | 0.021154 | 0.049493 | 0.005821 | 0.079439 |
| **Insurance** | -0.010003 | -0.015024 | 0.021154 | 1.000000 | -0.027992 | 0.111189 | -0.057189 |
| **Age** | -0.002512 | -0.02319 | 0.049493 | -0.027992 | 1.000000 | 0.064612 | 0.010680 |
| **Cibil Score** | -0.049590 | -0.027611 | 0.005821 | 0.111189 | 0.064612 | 1.000000 | -0.219715 |
| **Loan offer** | 0.011717 | 0.079674 | 0.079439 | -0.057189 | 0.010680 | -0.219715 | 1.000000 |

## Logistic Regression Model building:

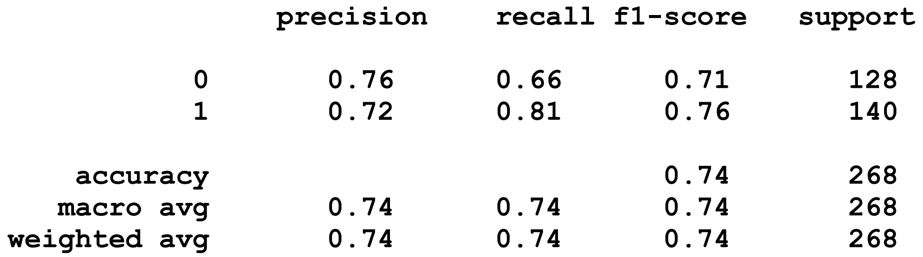
- Analyze the problem
- Collect the data
- Preprocessing the data
- Feature Selecting
- Train, test data splitting
- Fit, Train, Test and Evaluate the model4

## Logistic Regression Model Summary:

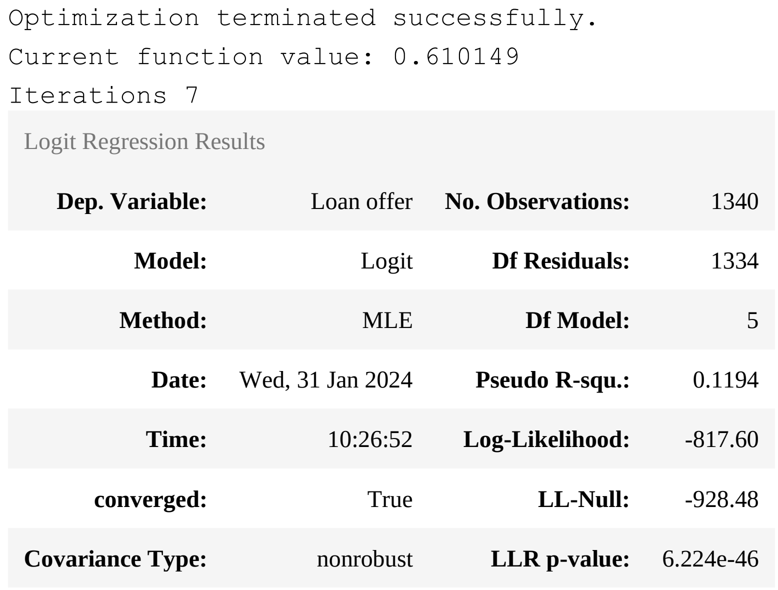
**Accuracy of Model:** 0.7388059

**Confusion Matrix:** array ([[ 85,  43],
[ 27, 113]], dtype=int64)

## Classification Report:



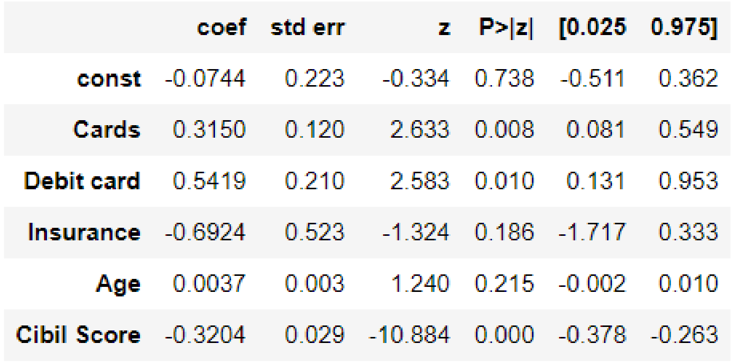
```
              precision    recall  f1-score   support

           0       0.76      0.66      0.71       128
           1       0.72      0.81      0.76       140

    accuracy                           0.74       268
   macro avg       0.74      0.74      0.74       268
weighted avg       0.74      0.74      0.74       268
```



```
Optimization terminated successfully.
Current function value: 0.610149
Iterations 7
```

Logit Regression Results

| | | | |
|---|---|---|---|
| **Dep. Variable:** | Loan offer | **No. Observations:** | 1340 |
| **Model:** | Logit | **Df Residuals:** | 1334 |
| **Method:** | MLE | **Df Model:** | 5 |
| **Date:** | Wed, 31 Jan 2024 | **Pseudo R-squ.:** | 0.1194 |
| **Time:** | 10:26:52 | **Log-Likelihood:** | -817.60 |
| **converged:** | True | **LL-Null:** | -928.48 |
| **Covariance Type:** | nonrobust | **LLR p-value:** | 6.224e-46 |



| | coef | std err | z | P>\|z\| | [0.025 | 0.975] |
|---|---|---|---|---|---|---|
| **const** | -0.0744 | 0.223 | -0.334 | 0.738 | -0.511 | 0.362 |
| **Cards** | 0.3150 | 0.120 | 2.633 | 0.008 | 0.081 | 0.549 |
| **Debit card** | 0.5419 | 0.210 | 2.583 | 0.010 | 0.131 | 0.953 |
| **Insurance** | -0.6924 | 0.523 | -1.324 | 0.186 | -1.717 | 0.333 |
| **Age** | 0.0037 | 0.003 | 1.240 | 0.215 | -0.002 | 0.010 |
| **Cibil Score** | -0.3204 | 0.029 | -10.884 | 0.000 | -0.378 | -0.263 |

## Problem Statement 3:

| age | work class | fnlwgt | education | education-num | marital status | occupation | relationship | race | sex |
|---|---|---|---|---|---|---|---|---|---|
| 39 | State-gov | 77516 | Bachelors | 13 | Never married | Adm-clerical | Not-in-family | White | Male |
| 50 | Self-emp-not-inc | 83311 | Bachelors | 13 | Married-civ-spouse | Exec-managerial | Husband | White | Male |
| 38 | Private | 215646 | HS-grad | 9 | Divorced | Handlers-cleaners | Not-in-family | White | Male |
| 53 | Private | 234721 | 11th | 7 | Married-civ-spouse | Handlers-cleaners | Husband | Black | Male |

**Note:** This is some sample from data set, the data set contains 48842 rows and 15 columns.

## Solution:

For the above data sets I used the KNN, K-Means, Decision Tree, Random Forest and SVM machine learning model. When the relationship between features and the target variable is complex and non-linear, the above-mentioned algorithms are used over Logistic Regression. So, we train and test the above data by each Machine Learning algorithm above mentioned.

Then the next important thing in this data sets are data preprocessing because these Data are non-linear and complex one, so we use some complex techniques to fill NAN values, Encoding the values in the datasets, I used **bfill** technique due to the NAN values in the categorical columns and encoding techniques **One-Hot Encoding and Label Encoding** are used.

## KNN Model

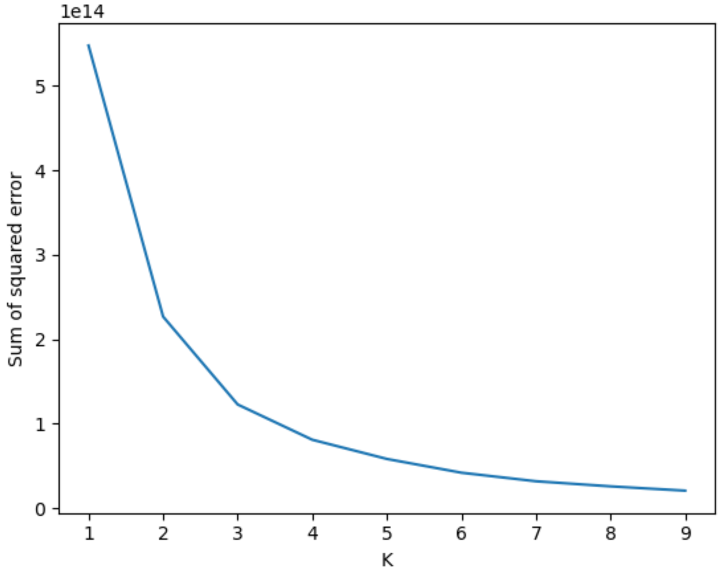
**Accuracy Score:** 0.7920974511 **(79%)**

**Confusion Matrix:**

```
array ([[7155, 206],
        [1825, 583]], dtype=int64)
```

## K-Means

**Accuracy Score:** 0.404467466 **(40%)**



## Support Vector Machine (SVM)

**Accuracy Score:** 0.79363292 **(79%)**

**Confusion Matrix:**

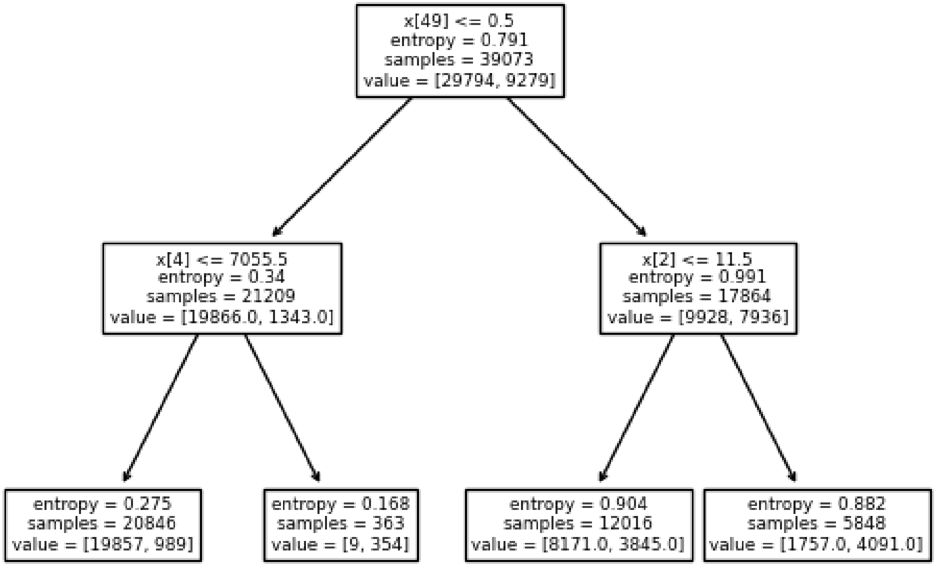
```
array ([[7341, 20],
        [1996, 412]], dtype=int64)
```

## Decision Tree

**Accuracy Score:** 0.82239737 **(82%)**

**Confusion Matrix:**

```
array ([[6892,  469],
        [1266, 1142]], dtype=int64)
```



## Random Forest

**Accuracy Score:** 0.848397993 **(84%)**

**Confusion Matrix:**

```
array ([[6795,  566],
        [ 915, 1493]], dtype=int64)
```