Diabetes, obesity, and inactivity: A multi-linear regression of three interrelated health factors.

Co-authors

Amith Ramaswamy, Chiruvanur Ramesh Babu Sai Ruchitha Babu, Nikhil Premachandra rao, Prajwal Sreeram Vasanth Kumar

A multi-linear regression of three interrelated health factors.

THE ISSUES:

It is essential to examine these health indicators to understand the frequency and potential links between diseases and lifestyle choices. This dataset, which focuses on 2018 data from the Centers for Disease Control and Prevention concerning diabetes, obesity, and physical inactivity rates throughout the U.S., has a wide range of factors that are critical for understanding diabetes trends. Overall, this collection of CDC-derived information provides a solid foundation for an in-depth examination of health trends.

Based on the information gathered, we answer the following queries or issues.

- Is there any relation between obesity, inactivity, and diabetes?
- How to handle the skewness (inequality) in the data?
- Is there a link between state and other factors like diabetes, obesity, and physical inactivity?
- Can we use any other parameter other than obesity and inactivity to improve the model performance?
- How to measure the effect of combined variables on the target variable?
- How can we determine whether our model's predictions are consistently reliable across several scenarios?

THE FINDINGS:

After analyzing the diabetes data,

- In addressing all the quantitative variable columns like (“Diabetics”, “Obesity”, and “And inactivity” we have used some transformation techniques like the BOX-COX method to remove the skewness.
- In predicting if the person is diabetic we have taken into consideration all the columns which passed the statistical test for the target variable and removed the columns (“FIPS”, “COUNTY”,” YEAR”) which do not affect the target variable.
- The columns Inactivity and obesity when created in a linear regression model gave less accuracy while predicting diabetics, but the results were good when we used both variables to predict the diabetics column using Multi Linear regression (MLR).
- We Used the interaction term effect on predicting the diabetic's column, which allows the model to capture the joint effect of these variables on the target variable, beyond what each variable contributes individually.

DISCUSSION:

Initially, we took deep insight into the data and found that the columns (Diabetes, Obesity, Inactivity) had high levels of kurtosis, and the data was skewed, so we had to use transformation techniques to bring it into normal distribution.

For predicting the diabetes of a person we initially tried the Linear regression model by using Inactivity VS Diabetes and Obesity vs Diabetes and the model didn't turn out to be good.

So we would like to improve the model by using both the variables and see if the model does something better additionally we are using some more columns such as (FIPS, COUNTY, and STATE) to see if it has any effect on the predictor variable Diabetics and check these columns for effect using statistical techniques. We also used Interaction terms to see the effect of predictor variables on the target variable.

Author Contributions:

	Amith Ramaswamy	Sai Ruchitha Babu	Nikhil Premachandraro	Prajwal S V
Data cleaning	30%	25%	20%	25%
Analyzing	30%	25%	20%	25%
Coding	25%	30%	25%	20%
Visualizing	25%	20%	25%	30%
Report Writing	20%	25%	30%	25%

Appendix A: Appendix A: Method

Data collection:
This report delves into the 2018 information gathered by the Centers for Disease Control and Prevention concerning diabetes, obesity, and physical inactivity rates throughout U.S. counties. A detailed assessment of these health markers is critical for understanding the scope and possible links between lifestyle-related illnesses. The knowledge gained from this study not only improves our understanding of public health differences but also serves as a foundation for particular targets for addressing these serious health challenges. Data can be found at this link: https://data.cdc.gov/Heart-Disease-Stroke-Prevention/Heart-Disease-Mortality-Data-Among-US-Adults-35-by/s6p7-fvbw

Variable creation:
We identify and create variables that represent critical aspects of diabetes data. These variables include geographic identifiers (e.g., FIPS codes), county names, and state information. So at last we removed all the variable which was not contributing to our model and took only what we needed like FIPS, COUNTY, STATE, DIABETIC, OBESE, AND INACTIVITY

Data Cleaning

- **Data Separation**: We initially divide the available data into three separate sets, each focusing on different aspects that as diabetes, obesity, and inactivity.
- **Information Extraction**: For the diabetes dataset, we extract key information such as county, state, and year. Simultaneously, we remove unnecessary columns such as ('YEAR', 'FIPS', 'COUNTY', and 'STATE') repeated from the original dataset.
- **Combining Insights**: We merge the diabetes and obesity datasets based on a common identifier (FIPS) to combine insights into one comprehensive dataset, simplifying the overall analysis.
- **Final Integration**: The last step involves integrating the combined dataset with information on physical inactivity, resulting in a finalized dataset that offers a holistic view of these health-related factors.

Analytic method:

- **Data Visualization and Skewness:** We started by visualizing the data with bar graphs, and we noticed some "skewness" in the data, which means it wasn't evenly distributed. This could mess up our model's performance.
- **Data Normalization with Box-Cox:** To fix the skewness issue, we used a statistical technique called Box-Cox transformation. This helped normalize the data, making it more suitable for our modeling efforts.
- **ANOVA Test:** Following normalization, we ran an Analysis of Variance (ANOVA) test. This test determines if there are significant variations in the means of distinct groups in the data. It's like seeing whether there are genuine variances rather than simply random fluctuations.
- **Multilinear Regression Model:** We trained a multilinear regression model after the ANOVA. When predicting outcomes, we may use this form of model to take into account numerous elements at once. For us, our goal was most likely to predict all that based on a variety of characteristics.
- **Model Evaluation with Breusch-Pagan Test and Cross-Validation:** We used a Breusch-Pagan test to check for heteroskedasticity (residuals are unequal over a range of measured values) in our model to confirm its dependability. This test determines if the error variability is consistent across all levels of our predictors. In addition, we utilized cross-validation to verify our model's performance on different subsets of the data, ensuring that it functions well in a variety of settings.

Appendix B: Results

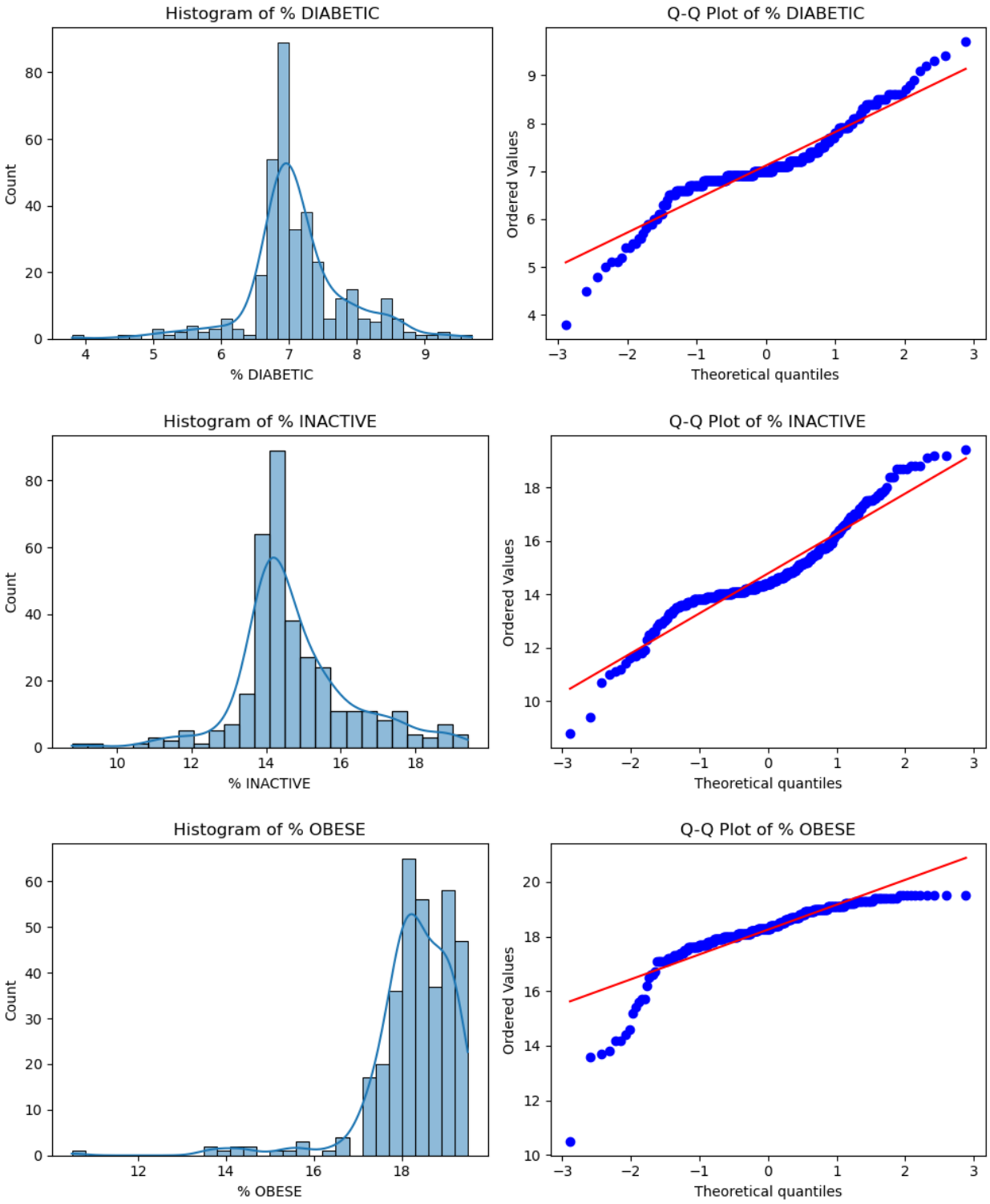
From the data set of 354 data points containing Diabetics, Obesity, and Inactivity
On applying descriptive statistics on all three data points we find the mean, median, standard deviation, skewness, and Kurtosis for all.
In the next part, we find the Correlation between all three variables, to make a call on deciding which all variables we need for computing the model

```
print("Kurtosis = ",round(stats.kurtosis(result_df['% OBESE'],fisher=False),5))
```

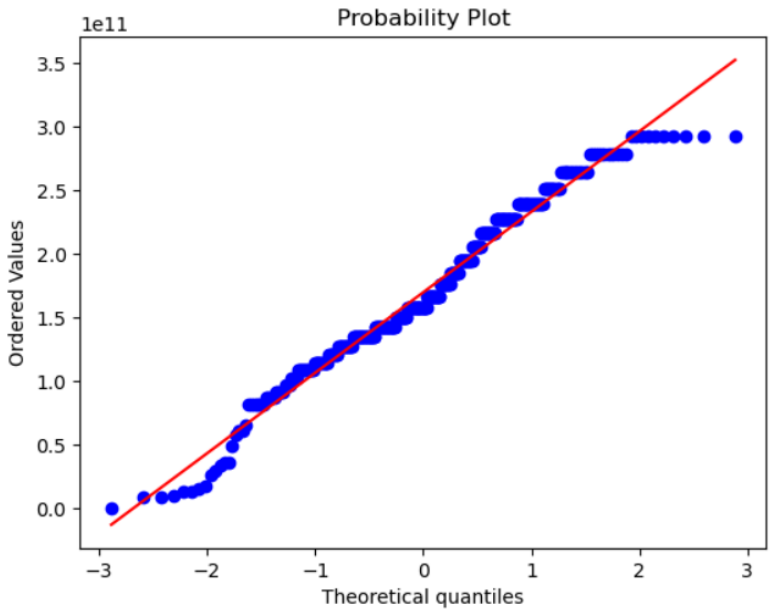
```
Descriptive Statistics
% DIABETIC
Median =  7.0
Mean =  7.11582
Stdev =  0.72741
Skewness =  -0.04881
Kurtosis =  5.78842
% INACTIVE
Median =  14.4
Mean =  14.77627
Stdev =  1.54236
Skewness =  0.42571
Kurtosis =  4.6136
% OBESE
Median =  18.3
Mean =  18.25254
Stdev =  1.02803
Skewness =  -2.75189
Kurtosis =  15.93152
```

```
result_df.iloc[:,-3:].corr()
```

	% DIABETIC	% OBESE	% INACTIVE
% DIABETIC	1.000000	0.389941	0.567104
% OBESE	0.389941	1.000000	0.472656
% INACTIVE	0.567104	0.472656	1.000000

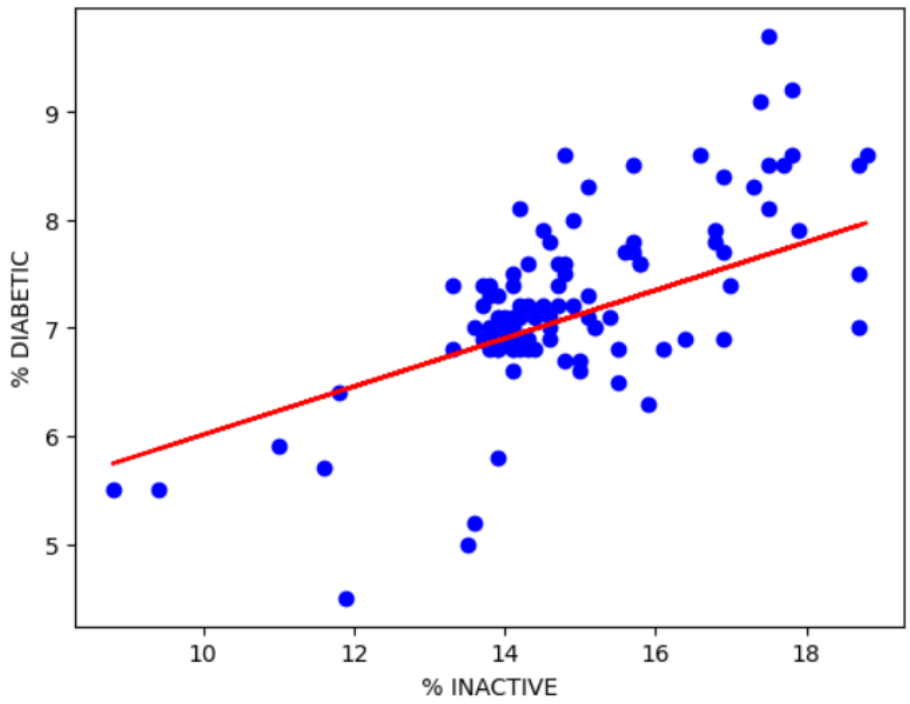


After this, we plotted the Histogram and the Q-Q plot distribution which shows the distribution of the data

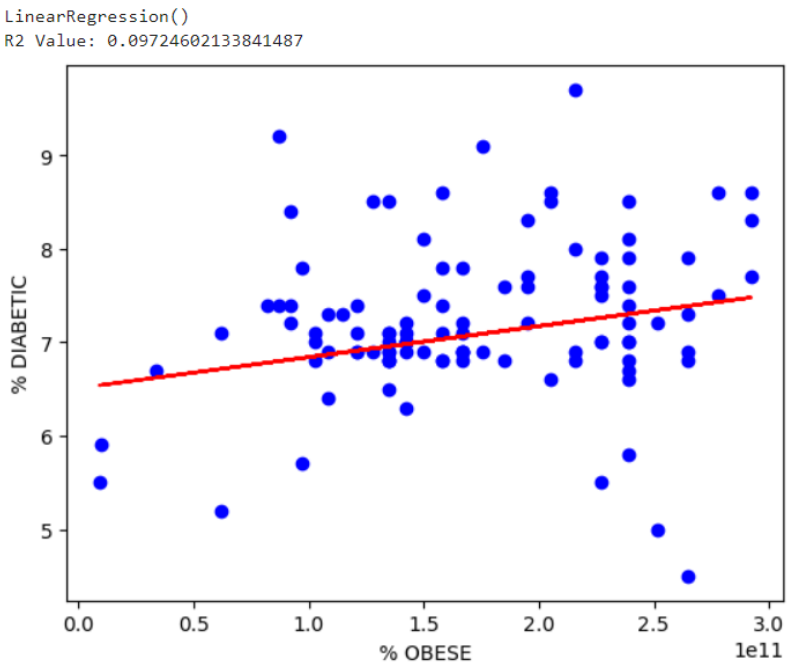


The Obesity from the above graph is left skewed so we tried to normalize the data using the Box-Cox method the below graph shows the probability graph after normalization

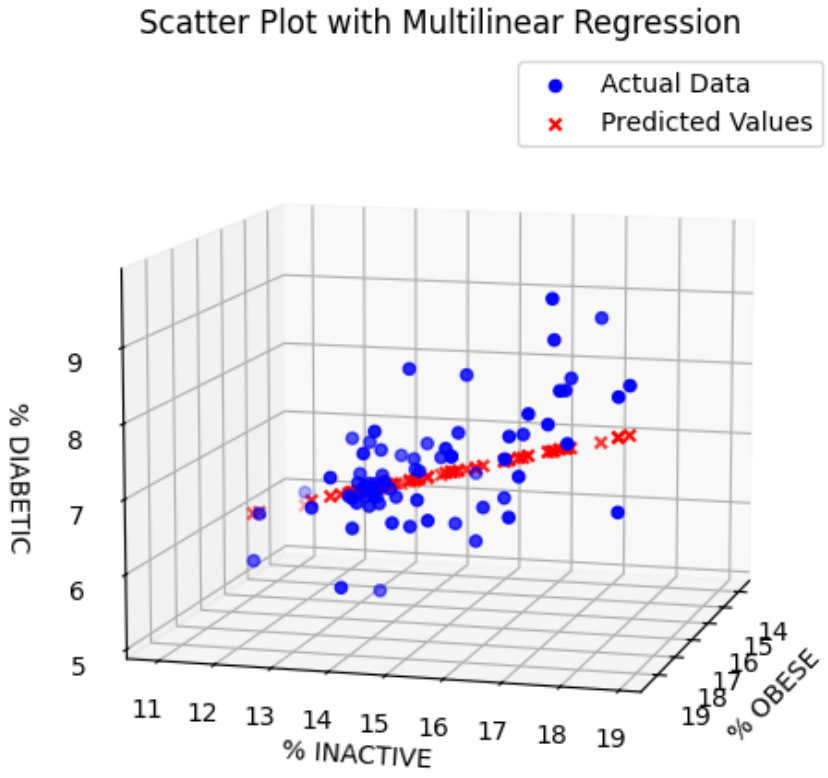
```
LinearRegression()
R2 Value: 0.2322780423451284
```



Scatter plot for the Linear regression model of Inactivity vs Diabetic



Scatter plot for the Linear regression model of Obesity vs Diabetic



Scatter plot for the Multi-linear regression model of Inactivity, Obesity vs Diabetic

```
Lagrange multiplier statistic: 89.14332976218978
Lagrange multiplier p-value:   2.116511912730936e-06
F-statistic:                   3.0579967042900766
P-value of F-statistic:        1.0238970068375836e-07
```

The final result of our model after 5 k-fold cross-validation

Concluded results

```
LinearRegression()
R2 Value: 0.6088867828206832

Accuracy values for 9-fold Cross Validation:
 [92.3975073  90.59163795 94.97238697 91.08393082 94.26022422 98.2084383
 97.40664731 98.15251719 90.51641866]

Final Average Accuracy of the model: 94.18
```

In this study, our findings suggest that inactivity emerges as a more significant indicator for predicting diabetes compared to obesity, as evidenced by the correlation matrix analysis. Following the training of the linear model using this dataset, we determined that our model exhibited a commendable R2 value of 0.688 and an accuracy of 94.18.

Appendix C: Python Code

Implementation of the Box-Cox concept for normalizing the data

```
COX
("Before    Box-Cox,    Kurtosis    value    =    ",round(stats.kurtosis(result_df['%
'],fisher=False),5))
t_df['% OBESE'],parameters=stats.boxcox(result_df['% OBESE'])
("After    Box-Cox,    Kurtosis    value    =    ",round(stats.kurtosis(result_df['%
'],fisher=False),5))
(parameters)
.probplot(result_df['% OBESE'], dist="norm", plot=plt)
```

Implementation of ANOVA function for all categorical columns

```
unctionAnova(inpData, TargetVariable, CategoricalPredictorList):
 Creating an empty list of final selected predictors
electedPredictors=[]

or predictor in CategoricalPredictorList:
   CategoryGroupLists=inpData.groupby(predictor)[TargetVariable].apply(list)
   AnovaResults = f_oneway(*CategoryGroupLists)
   # If the ANOVA P-Value is <0.05, that means we reject H0
   if (AnovaResults[1] < 0.05):
            print(predictor, 'is correlated with', TargetVariable, '| P-Value:',
Results[1])
       SelectedPredictors.append(predictor)
   else:
          print(predictor, 'is NOT correlated with', TargetVariable, '| P-Value:',
Results[1])

eturn(SelectedPredictors)

ling the function to check which categorical variables are correlated with target
oricalPredictorList=['FIPS', 'COUNTY', 'STATE']
ionAnova(inpData=result_df,              TargetVariable='%              DIABETIC',
oricalPredictorList=CategoricalPredictorList)

t_df.drop(['FIPS','COUNTY'],axis=1, inplace=True)
t_df.head(5)
```

Implementing the Multi-linear Regression

```
d_df2=result_df[['% INACTIVE','% DIABETIC','% OBESE']]
arate Target Variable and Predictor Variables
tVariable='% DIABETIC'
ctors=['% OBESE', '% INACTIVE']

ged_df2[Predictors].values
ged_df2[TargetVariable].values

it the data into training and testing set
in,   X_test,   y_train,   y_test   =   train_test_split(X,   y,   test_size=0.3,
m_state=42)

# Linear Regression in Python #######
del = LinearRegression()

ting all the parameters of Linear regression
(RegModel)

ting the model on Training Data
RegModel.fit(X_train,y_train)
ction=LREG.predict(X_test)

uring Goodness of fit in Training data
('R2 Value:',metrics.r2_score(y_train, LREG.predict(X_train)))
```

Implementation of Multi linear regression using interaction terms and Cross-validation

```
with state, interaction term with Cross-Validation

tVariable='% DIABETIC'
ctors=['% OBESE', '% INACTIVE', 'STATE_Alabama'......'Interaction_Term']

ged_df2[Predictors].values
ged_df2[TargetVariable].values

## K-fold cross-validation #######
ining a custom function to calculate accuracy
e sure there are no zeros in the Target variable if you are using MAPE
ccuracy_Score(orig,pred):
APE = np.mean(100 * (np.abs(orig-pred)/orig))
eturn(100-MAPE)

tom Scoring MAPE calculation
m_Scoring=make_scorer(Accuracy_Score, greater_is_better=True)

in,   X_test,   y_train,   y_test   =   train_test_split(X,   y,   test_size=0.3,
m_state=428)

# Linear Regression in Python #######
del = LinearRegression()

ting the model on Training Data
RegModel.fit(X_train,y_train)
ction=LREG.predict(X_test)

uring Goodness of fit in Training data
('R2 Value:',metrics.r2_score(y_train, LREG.predict(X_train)))

ning 10-fold cross-validation on a given algorithm
acy_Values=cross_val_score(RegModel, X , y, cv=9, scoring=custom_Scoring)
('\nAccuracy values for 9-fold Cross Validation:\n',Accuracy_Values)
('\nFinal Average Accuracy of the model:', round(Accuracy_Values.mean(),2))
```