

Name : Kshitij V Darwhekar

Roll No: TETB19

Sub: Soft Computitng

Batch: B2

## Experiment 2: Liner Regression and Logistic Regression Model Implementation on Given Dataset.

```
import matplotlib.pyplot as plt
import seaborn as sns
import pandas as pd
import numpy as np
```

```
from google.colab import drive
drive.mount('/content/drive/')
```

```
Mounted at /content/drive/
```

```
pf = pd.read_csv("/content/drive/MyDrive/ML/heart.csv")
```

```
pf.head()
```

	sbp	tobacco	ldl	adiposity	famhist	typea	obesity	alcohol	age	chd
0	160	12.00	5.73	23.11	Present	49	25.30	97.20	52	1
1	144	0.01	4.41	28.61	Absent	55	28.87	2.06	63	1
2	118	0.08	3.48	32.28	Present	52	29.14	3.81	46	0
3	170	7.50	6.41	38.03	Present	51	31.99	24.26	58	1
4	134	13.60	3.50	27.78	Present	60	25.99	57.34	49	1

```
history_mapping = {'Absent': 0, 'Present': 1}
pf["famhist"] = pf["famhist"].map(history_mapping)
pf.head()
```

	sbp	tobacco	ldl	adiposity	famhist	typea	obesity	alcohol	age	chd
0	160	12.00	5.73	23.11	1	49	25.30	97.20	52	1
1	144	0.01	4.41	28.61	0	55	28.87	2.06	63	1
2	118	0.08	3.48	32.28	1	52	29.14	3.81	46	0



```
sns.set(style='whitegrid', context='notebook')
cols = ['sbp','tobacco','ldl','adiposity','famhist','typea','obesity', 'alcohol','age', 'c
f, ax = plt.subplots(figsize=(15, 10))
cm = np.corrcoef(pf[cols].values.T)
sns.set(font_scale=1.5)
hm = sns.heatmap(cm,
                  cbar=True,
                  annot=True,
                  square=True,
                  fmt='.2f',
                  annot_kws={'size': 15},
                  yticklabels=cols,
                  xticklabels=cols)

plt.show()
```



```
X = pf[['tobacco','ldl','adiposity','famhist','typea','obesity','alcohol','age']].values
y = pf[['chd']].values
```

```
from sklearn.model_selection import train_test_split
```

```
X_train , X_test , y_train,y_test = train_test_split(X,y,train_size = 0.9)
```

```
# Apply logistic regression
```

```
from sklearn.linear_model import LogisticRegression
```

```
model = LogisticRegression(C=1,penalty='l2')
model.fit(X_train,y_train)
y_pred=model.predict(X_test)
```

```
/usr/local/lib/python3.7/dist-packages/sklearn/utils/validation.py:993: DataConversionWarning:
  y = column_or_1d(y, warn=True)
/usr/local/lib/python3.7/dist-packages/sklearn/linear_model/_logistic.py:818: ConvergenceWarning:
  STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.
```

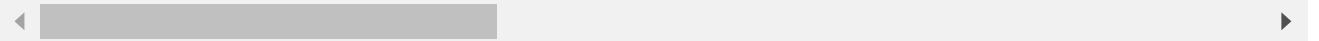
Increase the number of iterations (max\_iter) or scale the data as shown in:

<https://scikit-learn.org/stable/modules/preprocessing.html>

Please also refer to the documentation for alternative solver options:

[https://scikit-learn.org/stable/modules/linear\\_model.html#logistic-regression](https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression)

extra\_warning\_msg=\_LOGISTIC\_SOLVER\_CONVERGENCE\_MSG,



```
print ('Training Accuracy: %.2f' % model.score(X_train,y_train))
```

```
print ('Test Accuracy: %.2f' % model.score(X_test,y_test))
```

```
Training Accuracy: 0.74
```

```
Test Accuracy: 0.66
```

```
import seaborn as sns
from sklearn.tree import plot_tree
from sklearn import tree
from sklearn.metrics import confusion_matrix
```

```
cm = confusion_matrix(y_test,y_pred)
plt.figure(figsize=(5,5))
sns.heatmap(data=cm,linewidths=.5, annot=True,square =True, cmap ='Blues')
plt.ylabel('Actual label')
plt.xlabel('Predicted label')
```

Text(0.5, 37.79999999999998, 'Predicted label')

