2021

DATA SCIENCE

MANISH KUMAR

[FRAMINGHAM HEART STUDY]

[The Framingham Heart Study was a turning point in identifying the risk factors of heart disease, and is one of the most important epidemiological studies conducted. A lot of our present understanding of cardiovascular disease can be attributed to this study]

CONTENTS

1. Introduction

The Framingham Heart Study was a turning point in identifying the risk factors of heart disease, and is one of the most important epidemiological studies conducted.
A lot of our present understanding of cardiovascular disease can be attributed to this study.

The Framingham Heart Study: Origin

The origin of the FHS can be attributed to the premature death of US President Franklin D. Roosevelt in the year 1945.
The President had extremely high blood pressure, which at that time was not considered a big deal.
Before his presidency, his blood pressure was 140/100mmHg, which is considered high according to today's standards. One year before his death, the president's blood pressure had shot up to 210/120mmHg, which today is considered a hypertensive crisis.
At that time, his personal physician was a specialist in EENT (Eyes, Ears, Nose, and Throat), and was not a cardiologist. He insisted that the President was healthy, and said that his blood pressure was "no more than a man of his age.
The President died of a massive cerebral hemorrhage in the year 1945, and his blood pressure was 300/190mmHg on that day. The death of President Roosevelt paints a picture of our understanding of heart disease in the mid 20th century.
There was a huge increase in deaths from Coronary Heart Disease, or CHD in the 1930's,1940's, and 1950's. In the 1940's, heart disease was the number one cause of death among Americans.

A Solution ?

To better understand heart disease and the measures that could be taken to combat it, the Framingham Heart Study (FHS) was established in the late 1940's.
It was a joint project of Boston University and the National Heart, Lung, and Blood Institute (NHLBI).
A large cohort of initially healthy patients between the age group 30 and 59 in the city of Framingham, Massachusetts were tracked for a period of 20 years, to better

understand cardiovascular disease. The study was conducted with an initial cohort of 5209 patients.
The aim of the study was to enroll people free of the disease, and see who developed the disease in the next 20 years.
How was this done?
Every two years, the participants would have to report to a testing center, where an examination was conducted. The patients were examined and their health information was updated.
They were also given questionnaires to fill up, in which they updated behavioral information, such as exercise or smoking habits.
The data collected from this study allowed for a better understanding of the risk factors of heart disease. Medical interventions then took place based on the findings of the FHS.

Important Milestones

Some important milestones of the Framingham Heart Study:

- Smoking was found to increase the risk of CHD (1960)
- Cholesterol and high BP increased the risk of CHD (1961)
- Physical activity decreased the risk of CHD (1967)
- High levels of HDL cholesterol was found to increase the risk of CHD (1988)
- The lifetime risk of developing CHD was higher in men than in women (1999)
- Obesity is a risk factor for heart failure (2002)

Impact and Further Studies

There were many revolutionary breakthroughs in our knowledge of cardio vascular disease due to the FHS, and many medical interventions have taken place since then to prevent and decrease the risk of heart disease.
Around 20 years after the original cohort, a second study was started. This study involved the offspring of the first cohort and their spouses, and took place in 1971.
In the year 2002, the third generation cohort started, who were the grandchildren of the original cohort. The study is ongoing, and has expanded to take in various other risk factors such as family history, social network analysis, and genetic information.

Overall, there have been around 2400 studies written using data from the FHS, and this number continues to grow each year.
The FHS has contributed greatly in reducing mortality rates associated with CHD, and has corrected many clinical misconceptions about heart disease. We know so much more about reducing the risk of heart disease, treatment, and improving quality of life due to the FHS.

2. DATA

The data frame consists of 16 variables; 15 independent variables, or risk factors, and 1dependent variable.

Risk Factors/Independent Variables

1. Demographic:

• Male : A value of 1 indicates that the participant is male, and 0 indicates they are female.
• Age: The age of the participant
• Education Level: 1-High School, 2-High School Diploma/GED, 3-College, 4-Degree

2. Behavioral:

• Current Smoker: 1- The participant is a current smoker, 0- participant does not smoke currently
• cigsPerDay: Number of cigarettes smoked per day

3. Medical History:

• BPMeds: Amount of BP medication the participant is on
• prevalentStroke: 0- no prevalence of stroke, 1-has had occurrences of stroke
• prevalentHyp: 0-no prevalence of hypertension, 1-prevalence of hypertension

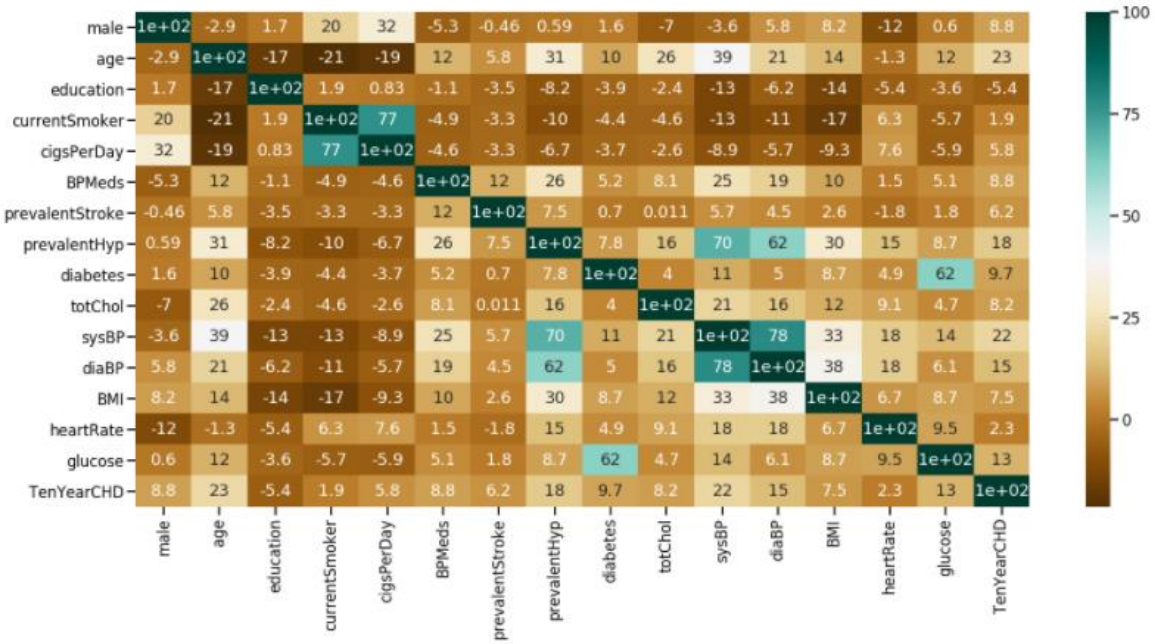
• diabetes: 0-no diabetes, 1-has diabetes

4. Risk factors from first physical examination:

• totChol: Total cholesterol
• sysBP: Systolic blood pressure
• diaBP: Diastolic blood pressure
• BMI: Body Mass Index
• Heart Rate: Heart rate in bpm
• Glucose: Glucose level (mg/dL)

3. Exploratory Data Analysis (EDA)

Before creating the model, I will perform some exploratory data visualization, to get some insights on the data.
As mentioned above, there are many risk factors such as smoking and high cholesterol levels that were found by the Framingham Heart Study to increase 10 year risk of CHD.
I will take a look at some of these risk factors, and see if I can find these relationships in this dataset. This will be done using the Seaborn library. Here is the heat map of the data.



Observations :

Correlation plot gives us valuable information regarding Relation within Attributes. It can Either be Negative or Positive or Nill. We need to always keep 1 feature from 2 Strongly Correlated ones but since we want to perform EDA we'll keep all and drop them before modelling.

- **currentSmoker & cigsPerDay** has strong Correlation of 77 (Scaled for better Observations)
- **prevalentHyp vs sysBP / diaBP** are having Positive Correlation of 70 and 62.
- While, **glucose & diabetes** are postively Correlated.
- **sysBP & diaBP** are also having Positive Correlation.

Usually we fill Null Values with Measures of Central Tendency (Mean / Median / Mode) or we've techniques like Forward / Backward fill but in this case we can observe the Correlation plot and consider it to Fill missing values. E.g., We have Positive Correlation between currentSmoker & cigsPerDay, we know that currentSmoker has values either 1 (is a Smoker) or 0 (is not a Smoker), we can groupby currentSmoker and Impute Missing values based on Median. We can do the same for BMI based on male (Gender) & Age.

EDA

- We'll explore various features in this section and perform Univariate, Bivariate & Multivariate Analysis.
- We'll observe descriptive statistics which will give us brief idea about spread of individual features.
- Visualizing Target attribute will shows us if we've imbalanced dataset.

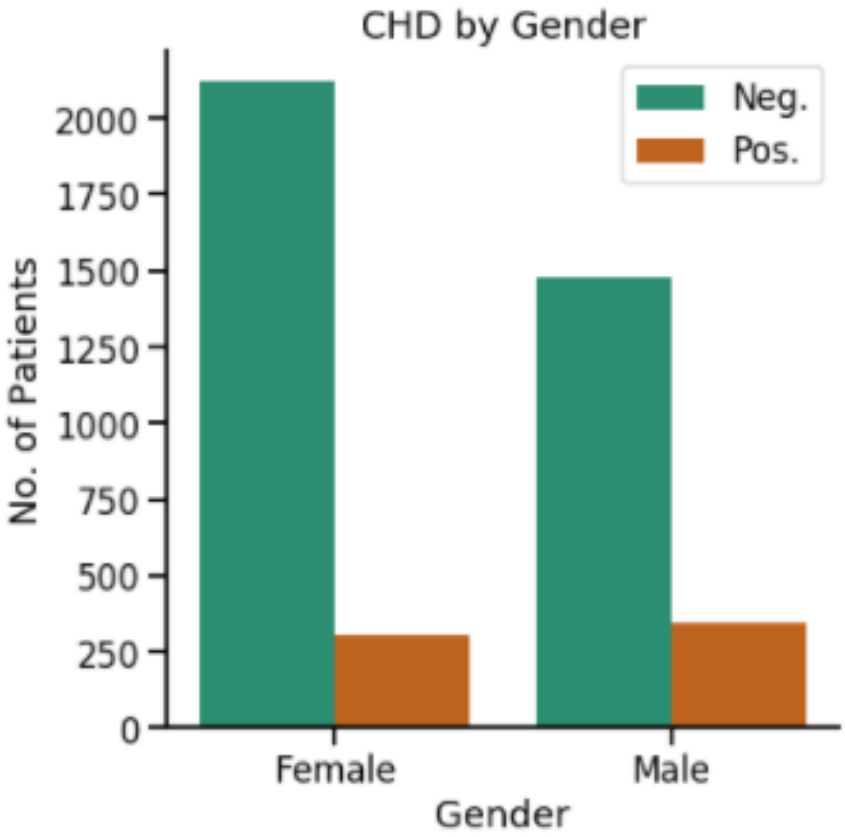


Fig: CHD by Gender

Observations :

- Above Bivariate Analysis plot depicts Gender wise absence / presence of Chronic Heart Disease (CHD).

- Observations tells us that we've Excessive number of people who are not suffering from CHD.
    - **Negative** : Approx. 80 to 90% of Females are falling in Negative Category while Approx. 60 to 70% of Males are in Negative Slot.
    - **Positive** : While Approx. 10% of Females & Males are suffering from CHD.

- By this we can say that our Dataset is Imbalanced where Approx. 80 to 90% are Negative Classifications and Approx. 10 to 15% are Positive Classes.

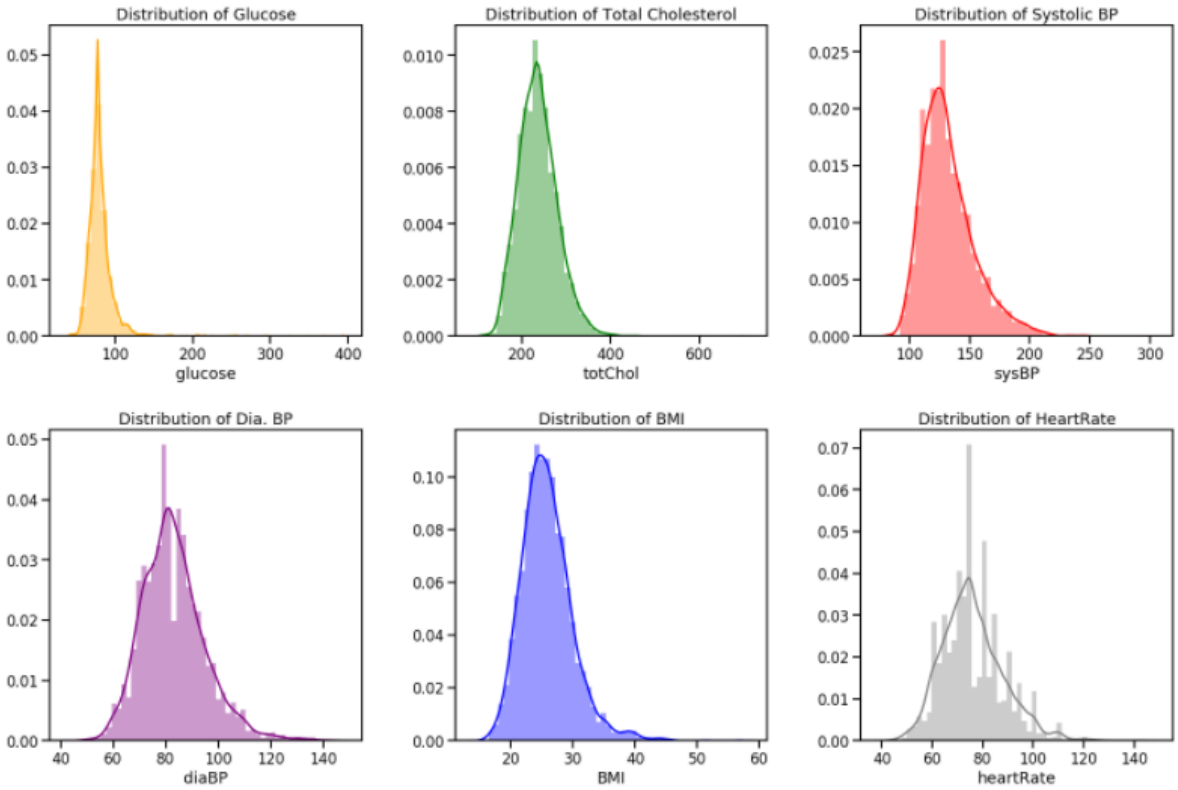


Fig: Distribution of continuous variables

Observations :

- We can see **Glucose, Total Cholesterol, Systolic BP & BMI** is **Right Skewed.**
- While **Diastolic BP & Heart Rate** are close to **Normal / Gaussian Distribution**.

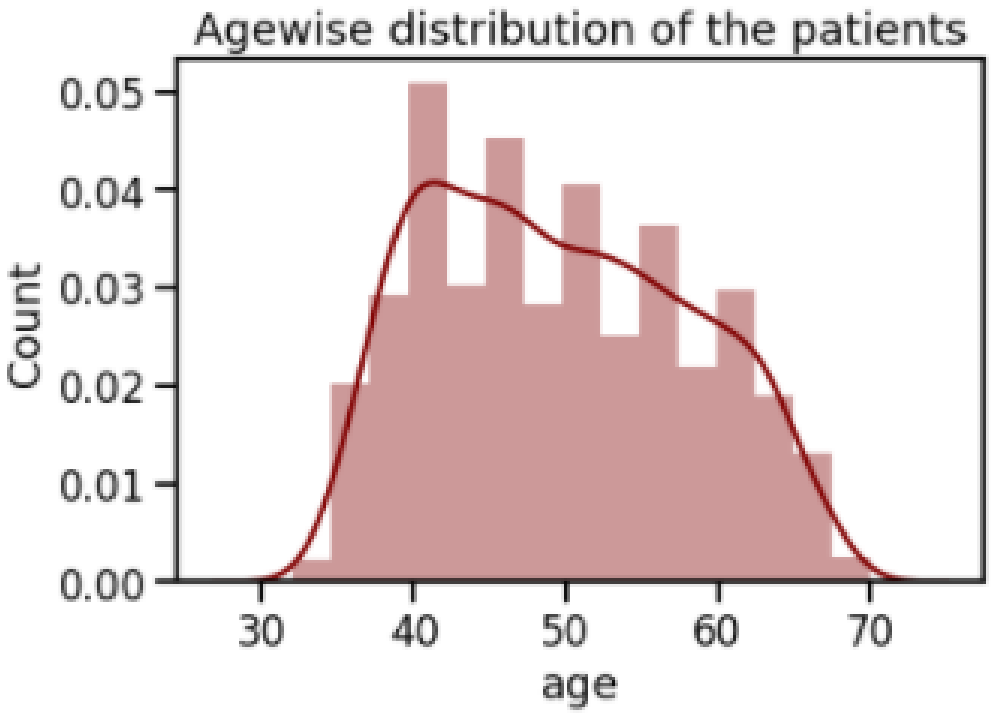
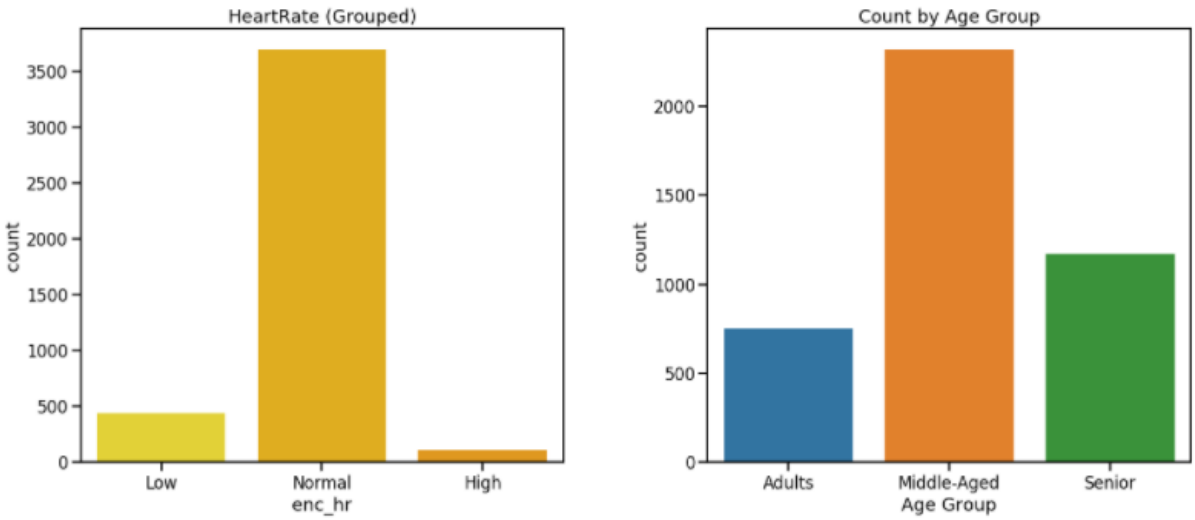


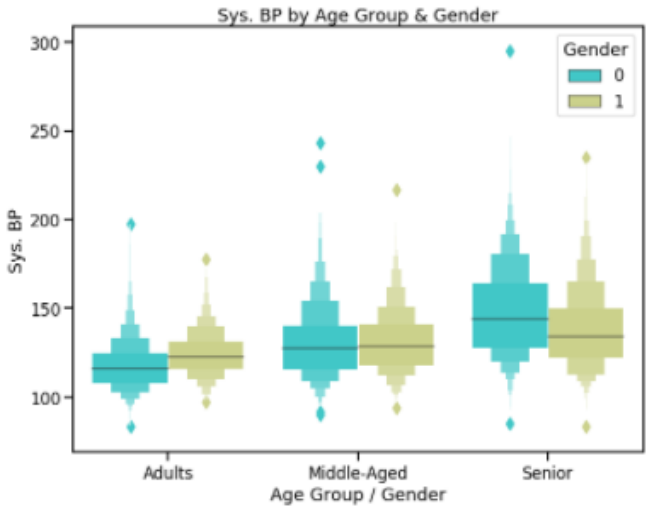
Fig: Age wise Distribution of patients

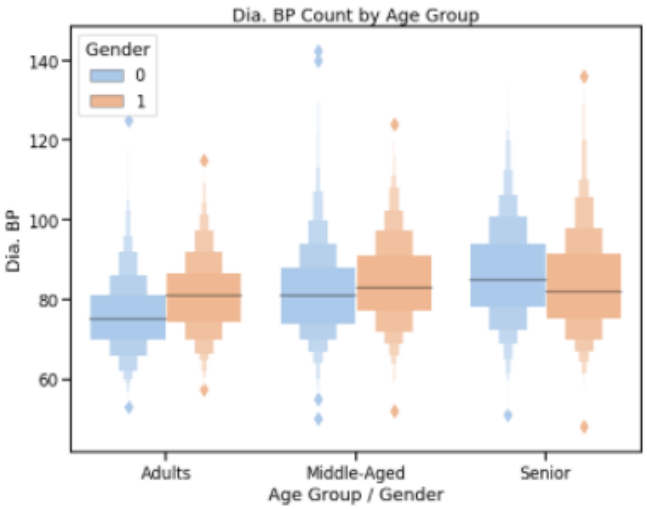
Observation :

- Subjects ranging from Age 40 to 50 are in Majority followed by 50 to 70.
- Let us define a user-defined Function to encode Age.



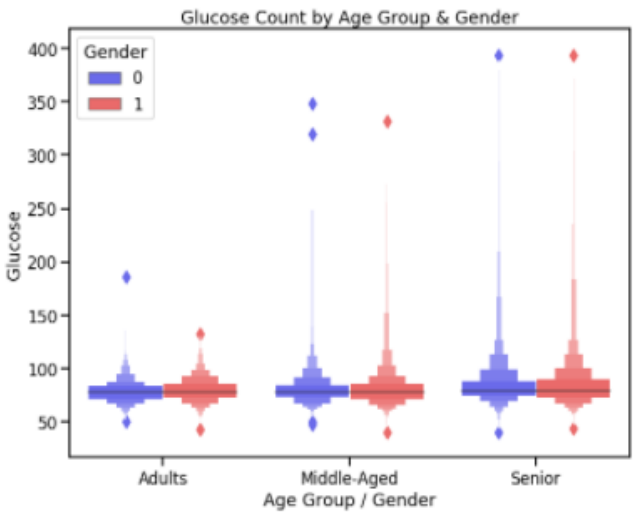
- We can observe that Subject with Normal HeartRate are in Majority followed by Resting / Low HeartRate and High HeartRate.
- We've more number of Middle-Aged Adults in our Dataset followed by Seniors And then Adults.

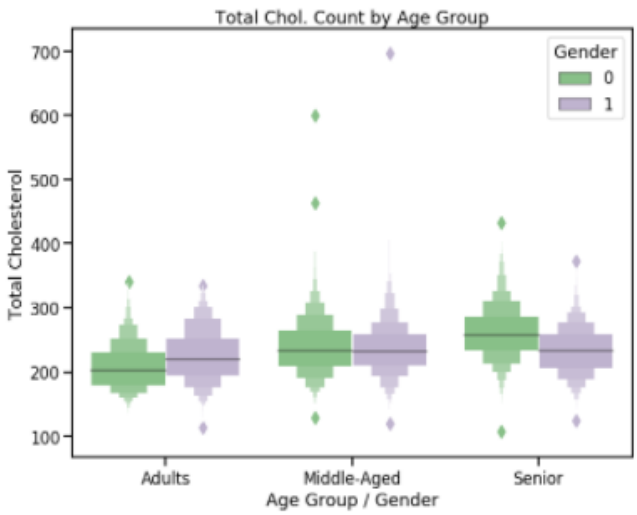




Observations :

- **Sys. BP by Age Group & Gender** : Sys. BP is Increasing by Age Group and Gender.
- **Dia. BP by Age Group & Gender** : Similar to Sys. BP , the Dia. BP is seen Increasing by Age Group & Gender

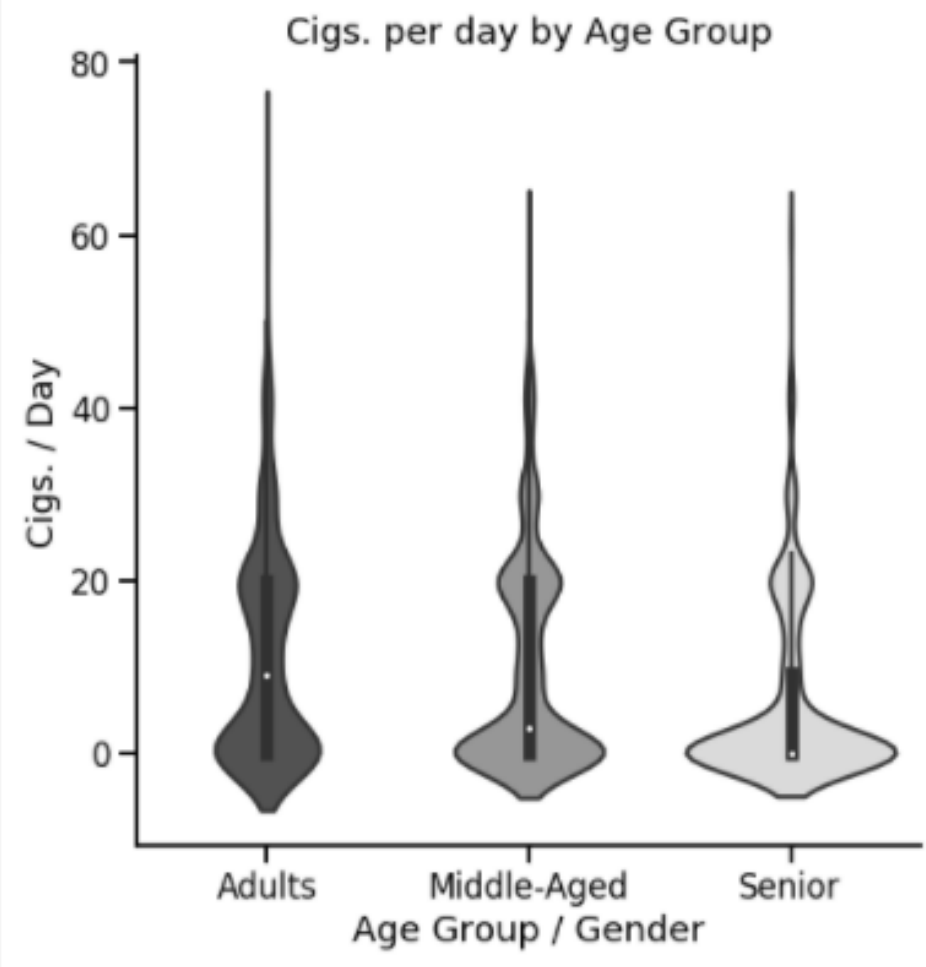




Observations :

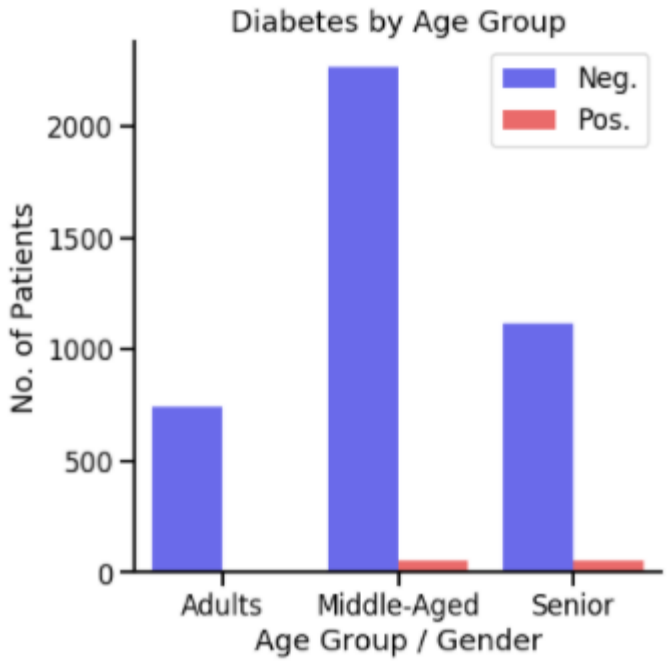
- **Glucose Count by Age Group & Gender** : We can clearly observe that as Age increases the count of Glucose increases too. While Gender wise Glucose Count has almost similiar Median with Few outliers in each.

- **Total Cholesterol by Age Group & Gender** : Excluding Outliers, Observation make us Clear that for females Cholesterol level is Increasing by Age considering the Quantile (25%, 50%, 75%) values into account. While, for Males the Cholesterol level Quantile is Approx. Similar for each Age Group.



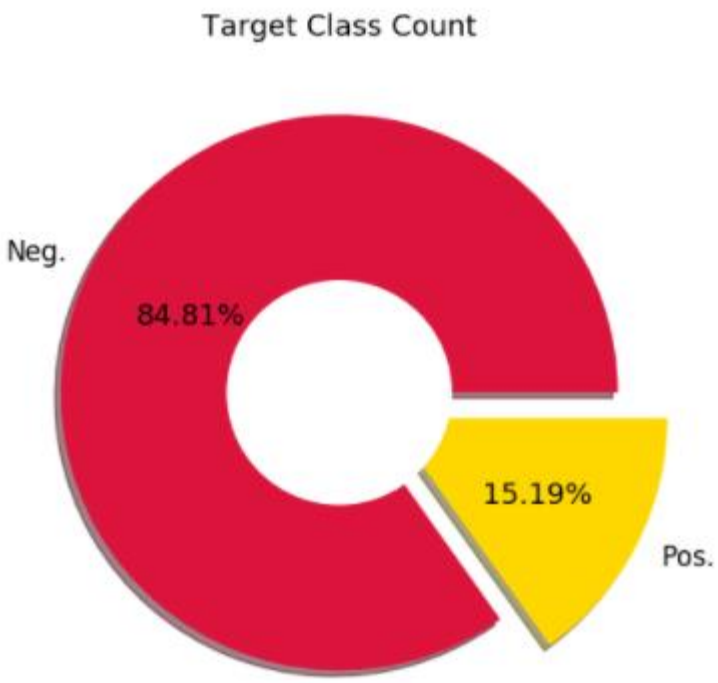
Observation :

- **Adults** : In Adults we can observe that Median values has Lower Kernel Density followed by 75% IQR's Density. While, 25% IQR marks the Higher Kernel Density.
- **Middle-Aged** : In Middle-Aged Group we can observe that 25% IQR & Median has Higher Kernel Density while 75% IQR has a quite Lower Kernel Density.
- **Senior** : In Seniority section we can observe that Median and 25% IQR are Closely Intact to each other having Higher Kernel Density, while 75% IQR got Lower Kernel Density.



Observation :

- **Adults** : Subject with Negative Diabetes Diagnosis are approx. 800 count while Positive Diabetes Diagnosis is Almost Nil.
- **Middle-Aged** : Subject with Negative Diabetes Diagnosis are reaching the Peak of Approx. 2500 Count while Positive Count is Under 100.
- **Senior** : Subject diagnosed Negative are Approx. 1000 while Positive Count is Under 100.



Observations :

- We can see that we've Imbalanced Dataset here having ratio of 85:15 where Positive Class is Minor.
- We'll need to Over-sample the Dataset in this case to get the best out of it.
- But before we proceed with Over-Sampling we'll First try Basic Logistic Regression Model on Data we had processed.

4. Model building:

- In this section we'll split dataset into Training & Validation set.
- We'll build a basic Logistic Regression model on data as is.

In [26]:
```
#train-test split
from sklearn.model_selection import train_test_split, cross_val_score, GridSearchCV
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report, roc_auc_score, roc_curve, auc

x = norm_df
y = df_copy['TenYearCHD']

x_train, x_test, y_train, y_test = train_test_split(x, y, random_state=23)
x_train.shape, x_test.shape, y_train.shape, y_test.shape
```

Out[26]: ((3180, 12), (1060, 12), (3180,), (1060,))

- Now we've 3180 Records For Training and 1060 Records for Evaluation / Validation.
- Ahead, we'll put forth Logistic Regression as our Estimator.

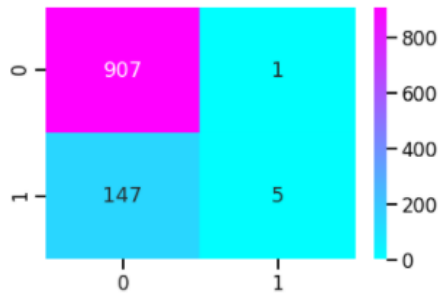
Logistic Regression

- Logistic Regression is always a best approach before moving ahead to complex Algorithms.
- Most of the times if we have done good Feature Engineering then algorithms as simple as Logistics Regression can give us fairly acceptable results.
- We will choose our solver as "liblinear" because our dataset isn't big to try other solvers so we'll go ahead with "liblinear"

In [28]:
```
#Metrics Evaluation

print ('Accuracy Score :', accuracy_score(y_test, log_pred))
print ('Cross Validation Score : ', cross_val_score(log_reg, x_train, y_train, cv=5).mean())
print (classification_report(y_test, log_pred))

sns.heatmap(confusion_matrix(y_test, log_pred), annot=True, cmap='cool', fmt='d')
```

```
Accuracy Score : 0.8603773584905661
Cross Validation Score :  0.8481173535663423
              precision    recall  f1-score   support

           0       0.86      1.00      0.92       908
           1       0.83      0.03      0.06       152

    accuracy                           0.86      1060
   macro avg       0.85      0.52      0.49      1060
weighted avg       0.86      0.86      0.80      1060
```

Out[28]: <matplotlib.axes._subplots.AxesSubplot at 0x7fcaba402d30>



Understanding the Metrics :

- **Accuracy Score** : Accuracy Score in Imbalanced Dataset can be a Trap.
  - **Why Accuracy is not our Evaluation Metrics ?**
  - **Ans.:** We get illusion of High Accuracy because our Estimator Learns well from Majority Class and is able to Predict well on Majority Class but not Minority Class leaving us in an illusion of High Accuracy. 0.8603 seems like a good accuracy but it has no value Since we can observe (Refer Confusion Matrix) above that we've Misclassification happening here for Minority Class.

- **Cross-Val Score** : Cross-Validation Scores uses the average of the output, which will be affected by the number of folds. Cross-Validation Scores Help us Identify if our Model is Over / Under-fitting.

- **Classification Report** : In Classification Report our Important metrics is Precision (TP/TP + FP) & Recall (TP/TP + FN). We can see our Recall Scores is good only for Majority Class but Positive Class has Bad Recall Score.

- **Confusion Matrix** : Diagonal Values of Confusion Matrix are correct. So we can see that In Negative Diagnosis out of 908 the 907 are correctly Classified while only 1 is misclassified, we can also call it as **Type 1 Error**. While, In case of Positive Diagnosis out of 152 examples, 5 are Classified correctly while rest 147 are misclassified as Negative Class. (Below is Interpretation of Confusion Matrix).

		Predicted class	
Actual Class		Class = Yes	Class = No
	Class = Yes	True Positive	False Negative
	Class = No	False Positive	True Negative

- **Let us add Class Weight Parameter to our Logistic Regression Estimator and see if it makes any Difference**

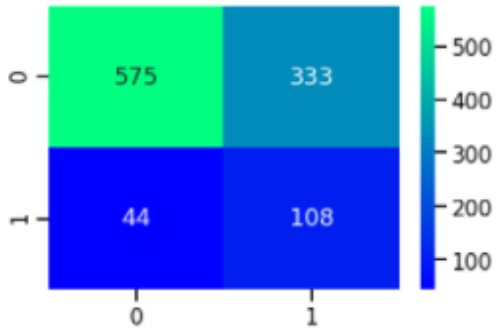
In [29]:
```
# train a logistic regression model on the training set
from sklearn.linear_model import LogisticRegression

log_reg_cw = LogisticRegression(solver='liblinear', class_weight='balanced')
log_reg_cw.fit(x_train, y_train)

log_cw_pred = log_reg_cw.predict(x_test)
log_cw_pred
```

Out[29]:
```
array([1, 1, 0, ..., 0, 1, 0])
```

In [30]:
```
#Metrics Evaluation

print ('Accuracy Score :', accuracy_score(y_test, log_cw_pred))
print ('Cross Validation Score : ', cross_val_score(log_reg_cw, x_train, y_train, cv=5).mean())
print (classification_report(y_test, log_cw_pred))

sns.heatmap(confusion_matrix(y_test, log_cw_pred), annot=True, cmap='winter', fmt='d')
```

```
Accuracy Score : 0.6443396226415095
Cross Validation Score :  0.6493659233918587
              precision    recall  f1-score   support

           0       0.93      0.63      0.75       908
           1       0.24      0.71      0.36       152

    accuracy                           0.64      1060
   macro avg       0.59      0.67      0.56      1060
weighted avg       0.83      0.64      0.70      1060
```

Out[30]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7fcabd4cc6d8>
```



Understanding the Metrics :

- By adding Class Weight Parameter to our Estimator we've got good Recall Score.
- Also, we can Interpret Confusion Matrix above, Its observable that for Negative Class 575 are Classified Correct & 333 are misclassified. While for Positive Class 108 are Classified correct and 44 are misclassified, It is a better result than our previous prediction. **Type 2 Error** has reduced upto some extent. Type 2 Errors in our case can be threatening. Let me explain how.
- **If Positive Class , i.e, One who actually has CHD is Classified as Negative Class then its a real threat as no actions will be taken on the Subject. It is termed as Type 2 Error. So such Model cannot be deployed in Production.**
- Let us Proceed ahead with Over-Sampling and examine our Metrics.

Over-Sampling using SMOTE

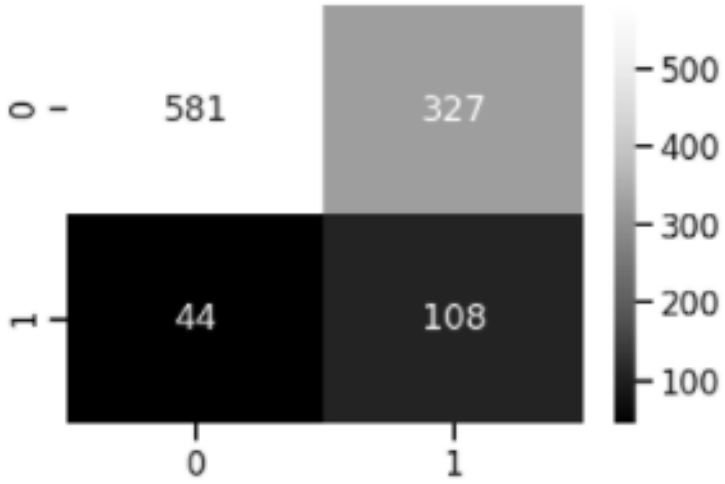
- SMOTE creates synthetic observations based upon the existing minority observations.
- SMOTE is widely used by Data Science Practioners.

Logistic Regression Post Over-Sampling

- Let's implement Logistic Regression Again and Interpret our Results

```
Accuracy Score : 0.65
Cross Validation Score :  0.6770818189999515
              precision    recall  f1-score   support

           0       0.93      0.64      0.76       908
           1       0.25      0.71      0.37       152

    accuracy                           0.65      1060
   macro avg       0.59      0.68      0.56      1060
weighted avg       0.83      0.65      0.70      1060
```

Out[33]: <matplotlib.axes._subplots.AxesSubplot at 0x7fcab901a2b0>



Understanding the Metrics :

- From Confusion Matrix above, Its observable that for Negative Class 581 are Classified Correct & 327 are misclassified. While for Positive Class 108 are Classified correct and 44 are misclassified, It is a better result than our previous prediction in case of **Type 1 Error** has reduced upto some extent. Type 2 Errors remains similar.

- **Type 2 Error** : If Positive Class , i.e, One who actually has CHD is Classified as Negative Class then its a real threat as no actions will be taken on the Subject. It is termed as Type 2 Error. So such Model cannot be deployed in Production.

ROC-AUC (Receiver Operating Characteristics - Area Under Curve)

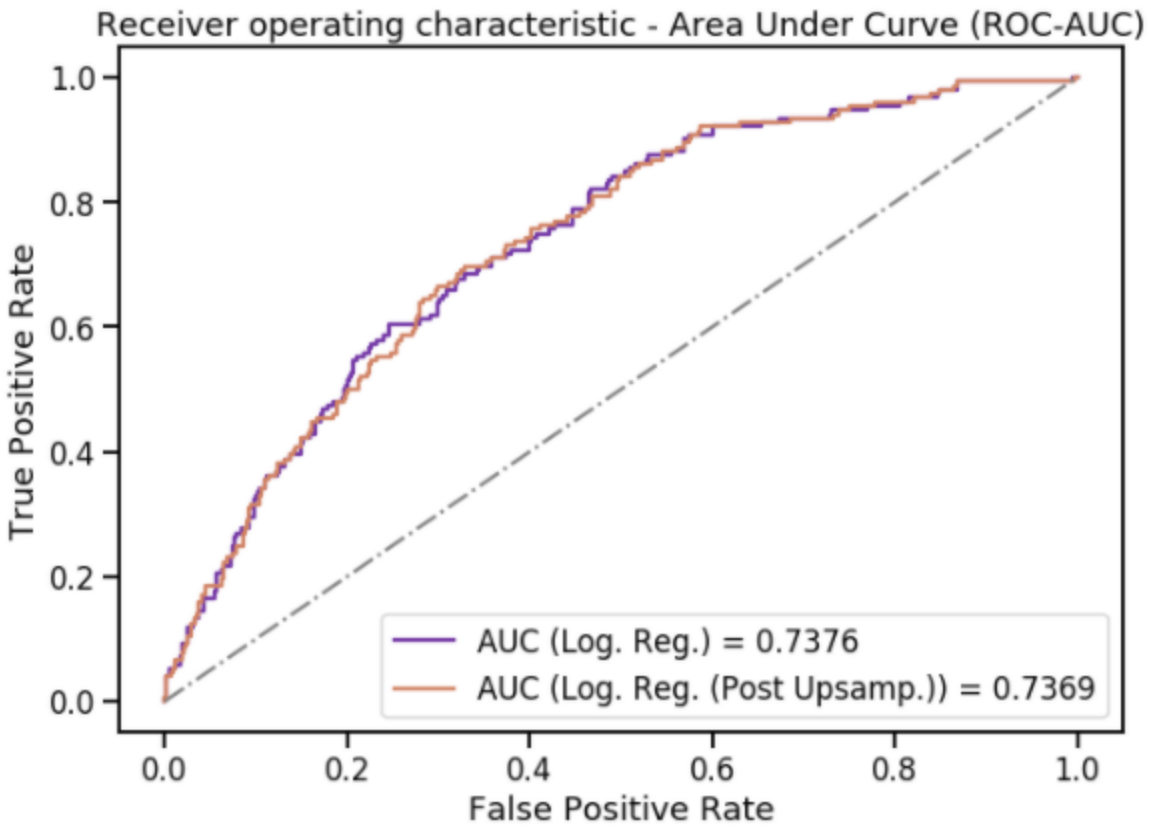
- It is a performance measurement for classification problem.

- ROC is a probability curve and AUC represents degree or measure of separability.

- It tells how much model is capable of distinguishing between classes. Higher the AUC, better the model is at predicting 0s as 0s and 1s as 1s.

- By analogy, Higher the AUC, better the model is at distinguishing between patients with disease and no disease.

- The ROC curve is plotted with TPR against the FPR where TPR is on y-axis and FPR is on the x-axis.



Observations :

- Above we've plotted ROC-AUC for 2 models.
- As mentioned above, the more the ROC-AUC Score the better is the model.
- We can evaluate Models performance based on this. It's clear that Logistic Regression model with Class Weight as "balanced" is giving us decent Score of 0.7376 followed by Logistic Regression post Over-Sampling which has given us Score of 0.7369.

5. Conclusion

- We figured out how our dataset was suffering from Class imbalance & so We handled imbalanced dataset with the help of SMOTE.
- From business perspective , it can help doctors to take necessary measures for patients carrying high risk of heart disease.
- This can also benefit Insurance Companies to some extent, If effective measures are taken then the number of claims can be reduced.

6. What's next?

- We can also try to **add more Parameters** for **Tuning the model.**
- One can also try to fill **missing values** by **forward / backward** fill.
- One can try also **implement Ensemble method, Tree Algorithm and / or Deep Neural Network** Modeling.