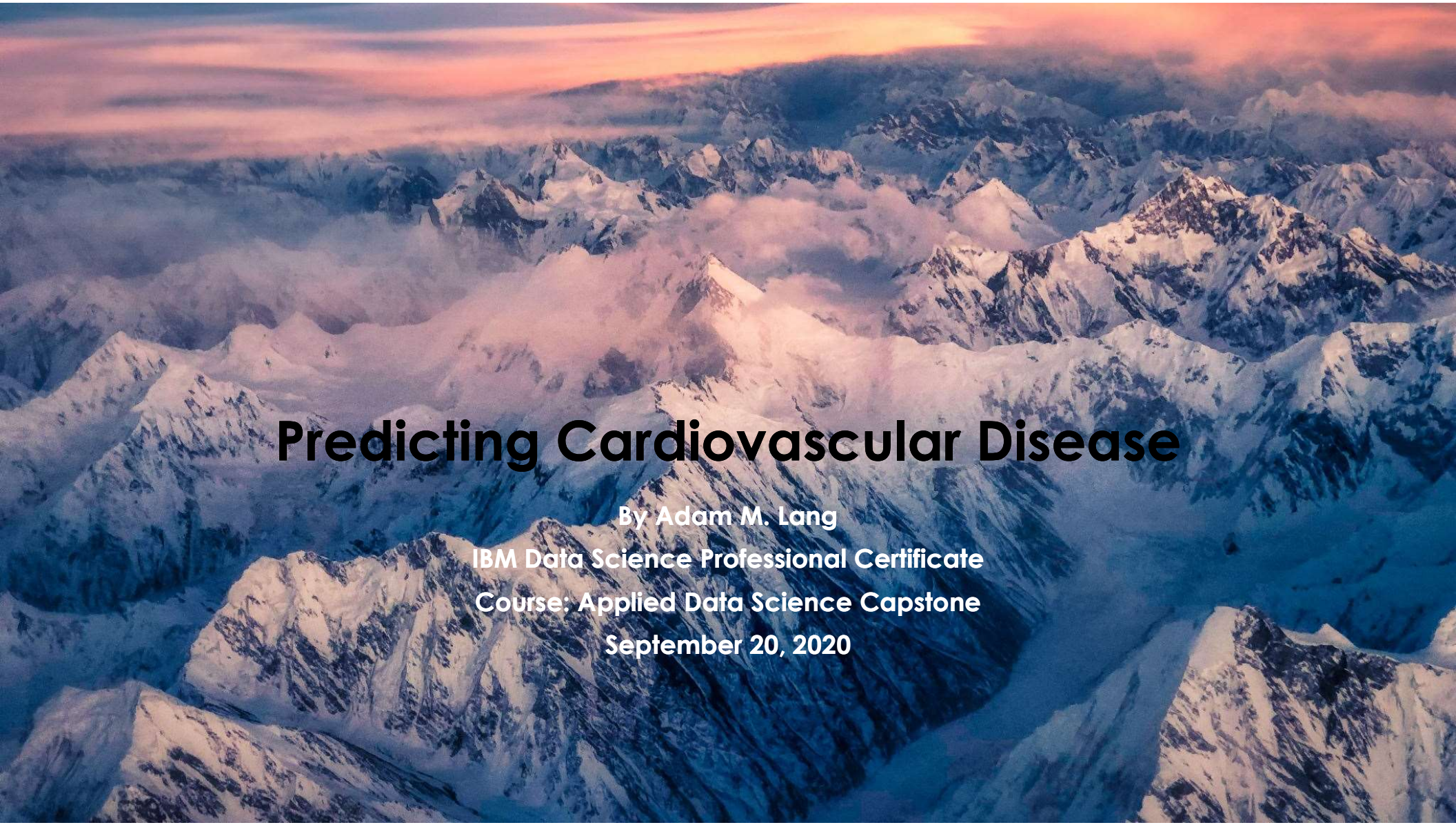


Predicting Cardiovascular Disease

By Adam M. Lang

IBM Data Science Professional Certificate

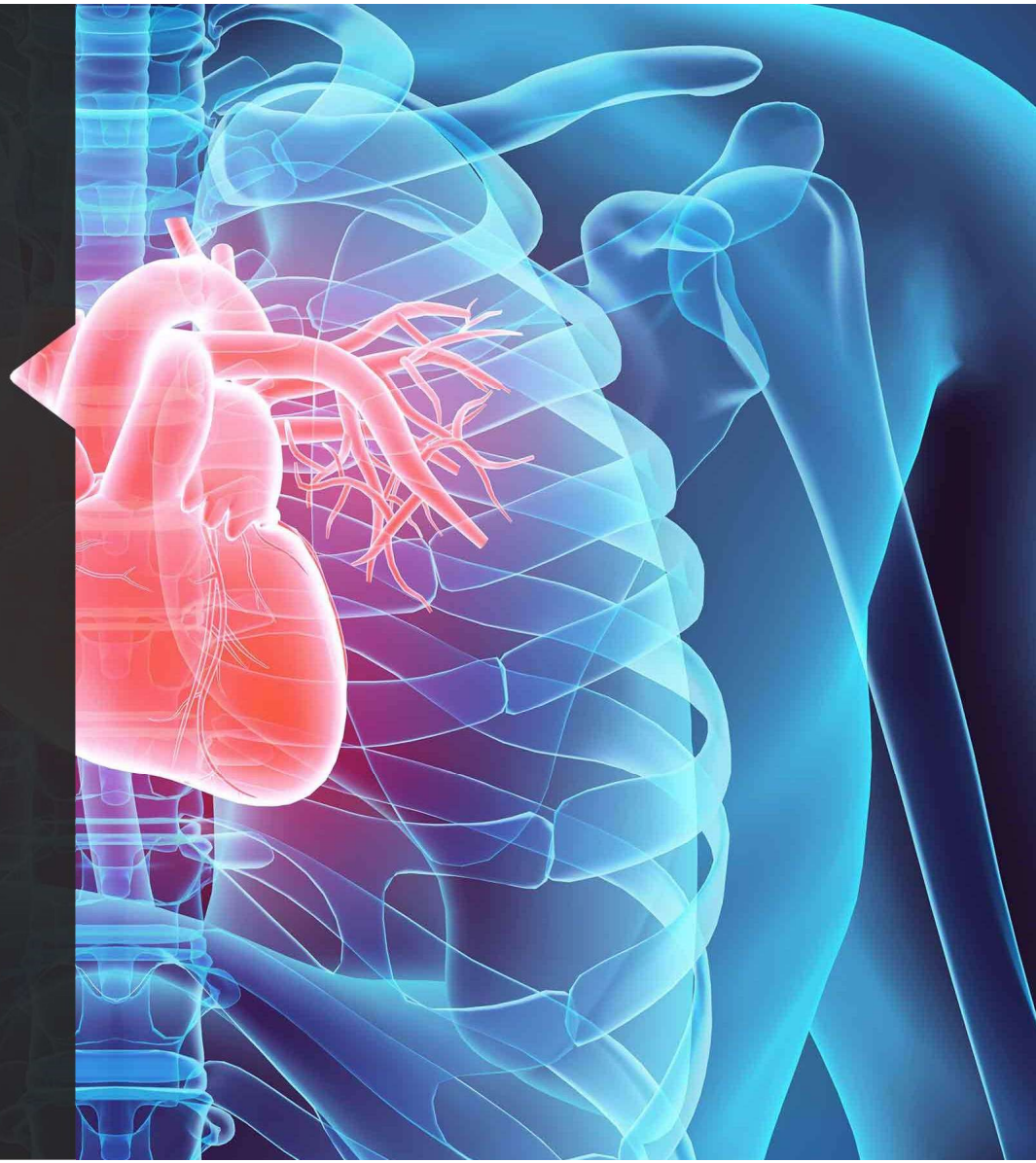
Course: Applied Data Science Capstone

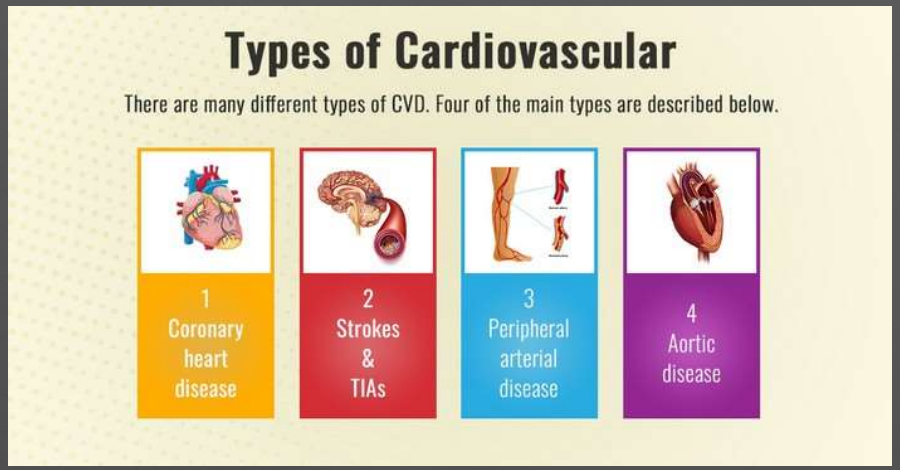
September 20, 2020

Introduction and Business Understanding

Introduction

- Cardiovascular Disease (CVD) is the leading cause of death in U.S. and for women in the U.S.
- 1 death every 37 seconds
- 17.5 million people die each year from cardiovascular diseases, 31% of all deaths worldwide.
- This statistic is expected to grow to more than 23.6 million by 2030 (Hajar, 2016).





What are CVDs?

- A group of disorders of the heart and blood vessels, including:
- Coronary heart disease,
- Cerebrovascular disease,
- Peripheral arterial disease
- Rheumatic heart disease,
- Congenital heart disease,
- Deep vein thrombosis, and Pulmonary embolism.

Risk factors

Modifiable

- ✓ Smoking or exposure to environmental tobacco
- ✓ Obesity
- ✓ Sedentary lifestyle
- ✓ Diabetes
- ✓ High cholesterol or abnormal blood lipids (fats)
- ✓ Hypertension
- ✓ Excessive Alcohol intake

Non-modifiable

- ✓ Gender: women develop 10 years later in life than men
- ✓ Age older than 50 years: 80% of people who die from CVD are 65 or older
- ✓ Family History of heart disease: primary risk is family member who developed heart disease before age 55

“Business Problem”

- CVD influences development of other acute and chronic diseases
- CVD influences how we approach preventative medicine and lifestyle improvements
- Working with CVD Risk Factor data will only help evolve our approach to predicting and managing CVD and its risk factors

Project Goals

1.How adequately can we predict cardiovascular disease based on certain risk factors?

2. Which risk factors are most important? Are some more important predictors of cardiovascular disease more important than others?

3. We will use machine learning algorithms to perform prediction including:

- Support Vector Machine (SVM)
- K-Nearest Neighbors (KNN)
- Random Forest (RF)
- Naive Bayes (NB)
- Which model will be the best predictor?

Data Understanding

Data Understanding

- Cardiovascular Disease Dataset from Kaggle
- 70,000 patient records
- 11 features
- CVD target variable

- 3 Types of input features:

1. Objective: factual information
2. Examination: results of medical exam
3. Subjective: information given by patient

Age | Objective Feature | age | int (days)
Height | Objective Feature | height | int (cm) |
Weight | Objective Feature | weight | float (kg) |
Gender | Objective Feature | gender | categorical code |
Systolic blood pressure | Examination Feature | ap_hi | int |
Diastolic blood pressure | Examination Feature | ap_lo | int |
Cholesterol | Examination Feature | cholesterol | 1: normal, 2: above normal, 3: well above normal |
Glucose | Examination Feature | gluc | 1: normal, 2: above normal, 3: well above normal |
Smoking | Subjective Feature | smoke | binary |
Alcohol intake | Subjective Feature | alco | binary |
Physical activity | Subjective Feature | active | binary |

Presence or absence of cardiovascular disease | Target Variable | cardio | binary |

Data Understanding: Features

Data Understanding: Features explained

5 Continuous(Numeric) Variables:

- Age (number of days)
- Height (cm)
- Weight (kg)
- Systolic Blood Pressure
- Diastolic Blood Pressure

5 Categorical Variables:

- Cholesterol (Levels 1-3)
- Glucose (Levels 1-3)
- Smoking (0 or 1)
- Alcohol intake (0 or 1)
- Activity (0 or 1)

Methodology

Exploratory Data Analysis

- Age: number of days
- Columns: ap_hi is Systolic BP, ap_lo is Diastolic BP will need to be changed
- Patient ID will have to be dropped as it is irrelevant to analysis

Exploratory Data Analysis

In [198]:
```
#inspect dataset first 5 rows
cardio.head(5)
```

Out[198]:

	id	age	gender	height	weight	ap_hi	ap_lo	cholesterol	gluc	smoke	alco	active	cardio
0	0	18393	2	168	62.0	110	80	1	1	0	0	1	0
1	1	20228	1	156	85.0	140	90	3	1	0	0	1	1
2	2	18857	1	165	64.0	130	70	3	1	0	0	0	1
3	3	17623	2	169	82.0	150	100	1	1	0	0	1	1
4	4	17474	1	156	56.0	100	60	1	1	0	0	0	0

Exploratory Data Analysis: Data Types

- All data types appear to be numeric form
- All data starting with Cholesterol and below are categorical but are label encoded

Examine Variables

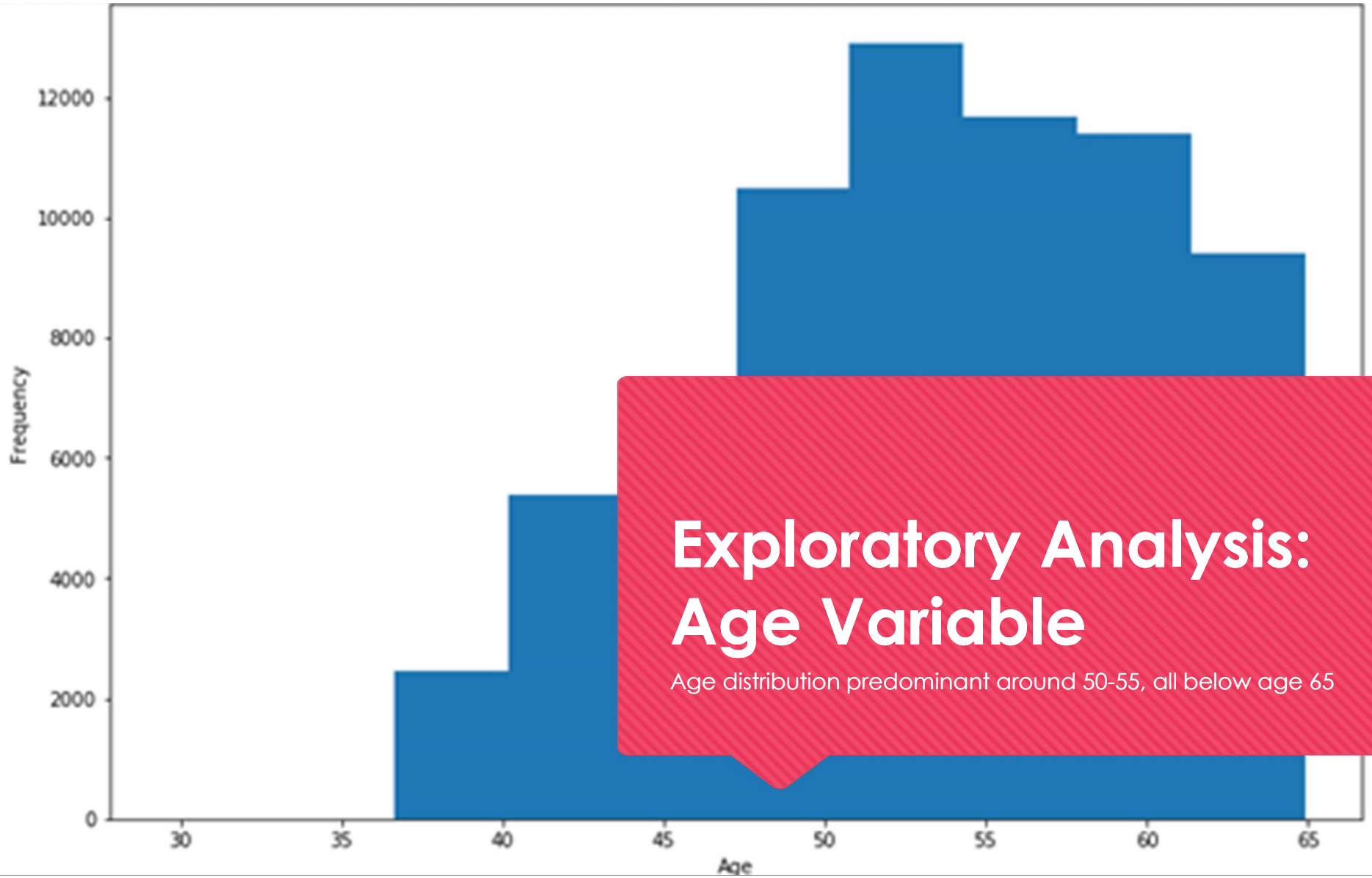
```
In [204]: cardio.dtypes

Out[204]: Age              int64
          Gender           int64
          Height           int64
          Weight         float64
          Systolic_BP      int64
          Diastolic_BP     int64
          Cholesterol      int64
          Glucose          int64
          Smoke            int64
          Alcohol          int64
          Active           int64
          Cardio           int64
          dtype: object
```

Exploratory Data Analysis: Data Integrity

- 24 duplicate records
- Drop all from dataset

```
e duplicate variables first

dio[cardio.duplicated(keep=False)]
licated.sort_values(by=['Age', "Gender", "Height"
alues to see duplication clearly

(24) # Show all duplicated values
```

Gender	Height	Weight	Systolic_BP	Diastolic_BP	Cholester
1	175	69.0	120	80	
1	175	69.0	120	80	
1	165	60.0	120	80	
1	165	60.0	120	80	
1	165	65.0	120	80	
1	165	65.0	120	80	
1	160	58.0	120	80	
1	160	58.0	120	80	
1	165	65.0	120	80	
1	165	65.0	120	80	



Exploratory Analysis: Age Variable

Age distribution predominant around 50-55, all below age 65

Now let's create a new column to describe the decade of age groups.

In [214]:
```
cardio['Age_By_Decade'] = pd.cut(x=cardio['Age'], bins=[30, 39, 49, 59,69], labels=['30s', '40s', '50s','60s'])
cardio.head(5)
```

Out[214]:

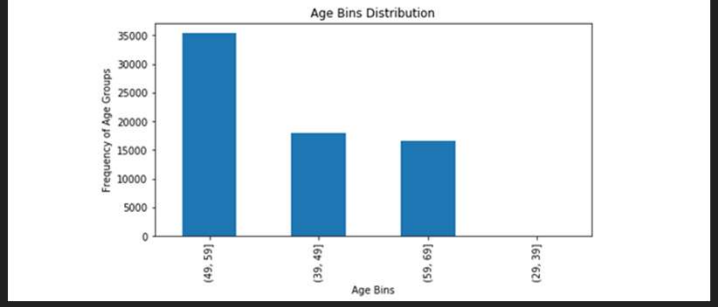
	Age	Gender	Height	Weight	Systolic_BP	Diastolic_BP	Cholesterol	Glucose	Smoke	Alcohol	Active	Cardi	Age_Bins	Age_By_Decade
0	50.358324	2	168	62.0	110	80	1	1	0	0	1		(49, 59]	50s
1	55.382383	1	156	85.0	140	90	3	1	0	0	1		(49, 59]	50s
2	51.628712	1	165	64.0	130	70	3	1	0	0	0		(49, 59]	50s
3	48.250135	2	169	82.0	150	100	1	1	0	0	1		(39, 49]	40s
4	47.842187	1	156	56.0	100	60	1	1	0	0	0		(39, 49]	40s

Transform Age into Bins

Evaluate Age Bins

```
Out[21]: (49, 59]    35370
         (39, 49]    17982
         (59, 69]    16620
         (29, 39]        4
         Name: Age_Bins, dtype: int64
```

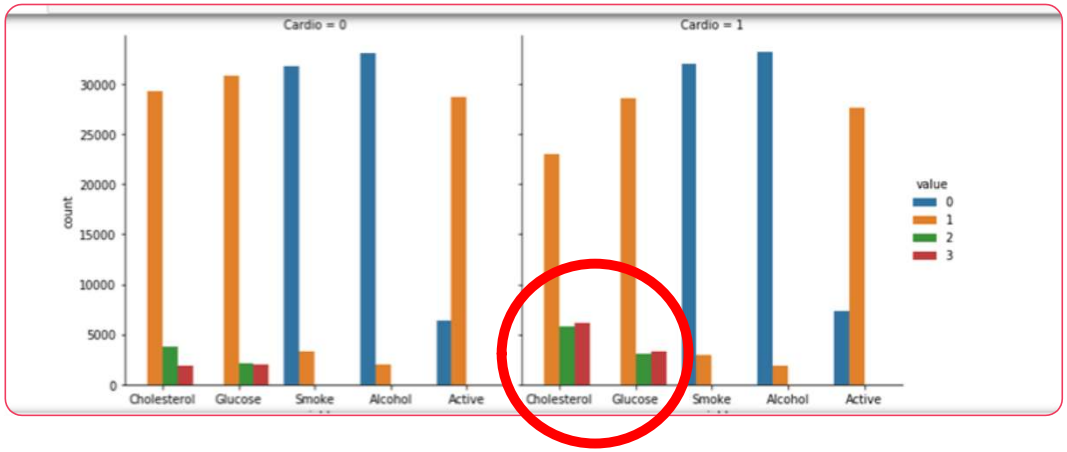
```
[24]: cardio['Age_Bins'].value_counts().plot(kind='bar',figsize=(8,4))
      plt.title("Age Bins Distribution")
      plt.xlabel("Age Bins")
      plt.ylabel("Frequency of Age Groups")
      plt.show();
```



- Sample sizes are not even
- Twice as many 50-59 year old's than 60-69 year old's
- Does this reflect a normal population of patients that could have heart disease?

Data Visualization: Categorical Variables

- More people with CVD that have higher Glucose and Cholesterol levels than those without CVD
- Smoke, Alcohol, Activity are similar and more normal for both groups with and without CVD



Outlier Detection

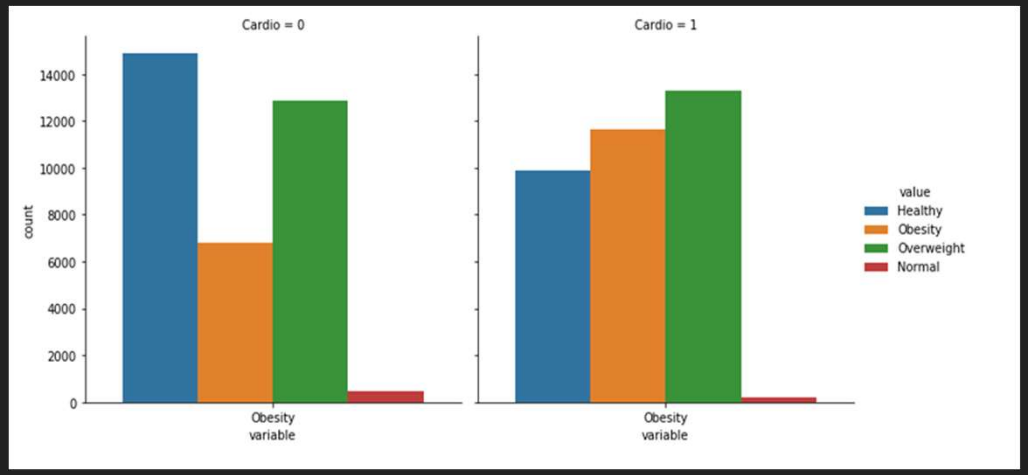
	Age	Gender	Height	Weight	Systolic_BP	Diastolic_BP	6
count	69976.000000	69976.000000	69976.000000	69976.000000	69976.000000	69976.000000	
mean	53.304175	1.349648	164.359152	74.208519	128.820453	96.636261	
std	6.755442	0.476862	8.211218	14.397211	154.037729	188.504581	
min	29.563920	1.000000	55.000000	10.000000	-150.000000	-70.000000	
25%	48.362389	1.000000	159.000000	65.000000	120.000000	80.000000	
50%	53.944982	1.000000	165.000000	72.000000	120.000000	80.000000	
75%	58.391343	2.000000	170.000000	82.000000	140.000000	90.000000	
max	64.923989	2.000000	250.000000	200.000000	16020.000000	11000.000000	

- Systolic BP highest is 1,6020, Diastolic BP highest is 11,000 = not possible!
- Highest recorded BP on planet earth: 370/360 (Narloch et al. 1995)
- Hypertensive crisis or emergency is BP of 200/120 mm/Hg (Unger et al. 2020)
- Weight is 200kg which is ~440lbs which is reasonable for CVD patients
- Height is 250cm which is over 8 feet tall
- **Solution: Drop outliers from BP and Height columns**

nder	Height	Weight	Systolic_BP	Diastolic_BP	Cholesterol	Glucose	Smoke	Alcohol	Active	Cardio	Age_Bins	Age_By_Decade	BMI	Obesity
2	168	62.0	110	80	1	1	0	0	1	0	(49, 59]	50s	21.967120	Healthy
1	156	85.0	140	90	3	1	0	0	1	1	(49, 59]	50s	34.927679	Obesity
1	165	64.0	130	70	3	1	0	0	0	1	(49, 59]	50s	23.507805	Healthy
2	169	82.0	150	100	1	1	0	0	1	1	(39, 49]	40s	28.710479	Overweight
1	156	56.0	100	60	1	1	0	0	0	0	(39, 49]	40s	23.011177	Healthy

Create BMI and Obesity Data Columns

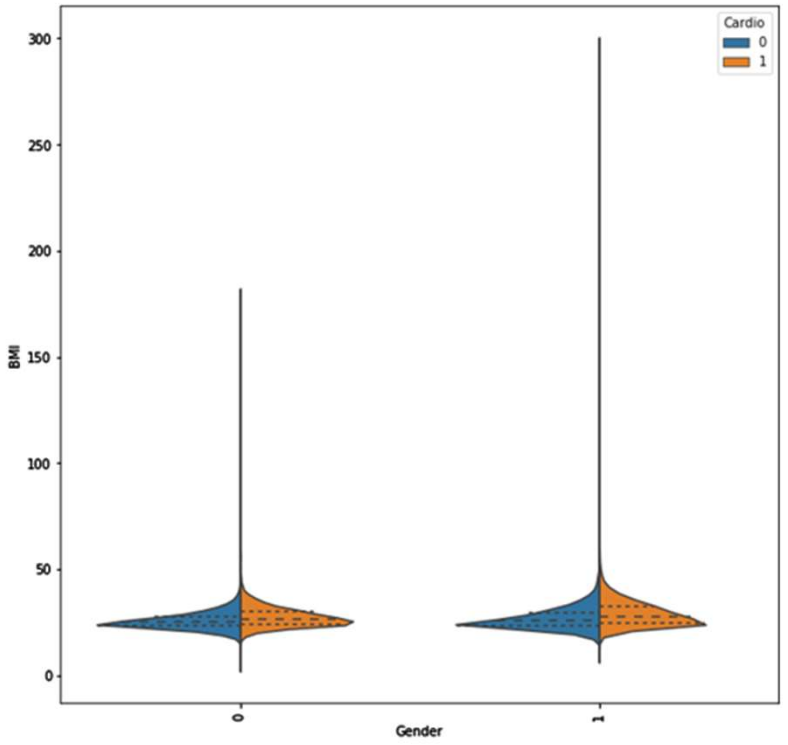
Evaluate Obesity based on CVD diagnosis



- Now we can see more patients with CVD are Overweight or Obese than non-CVD patients

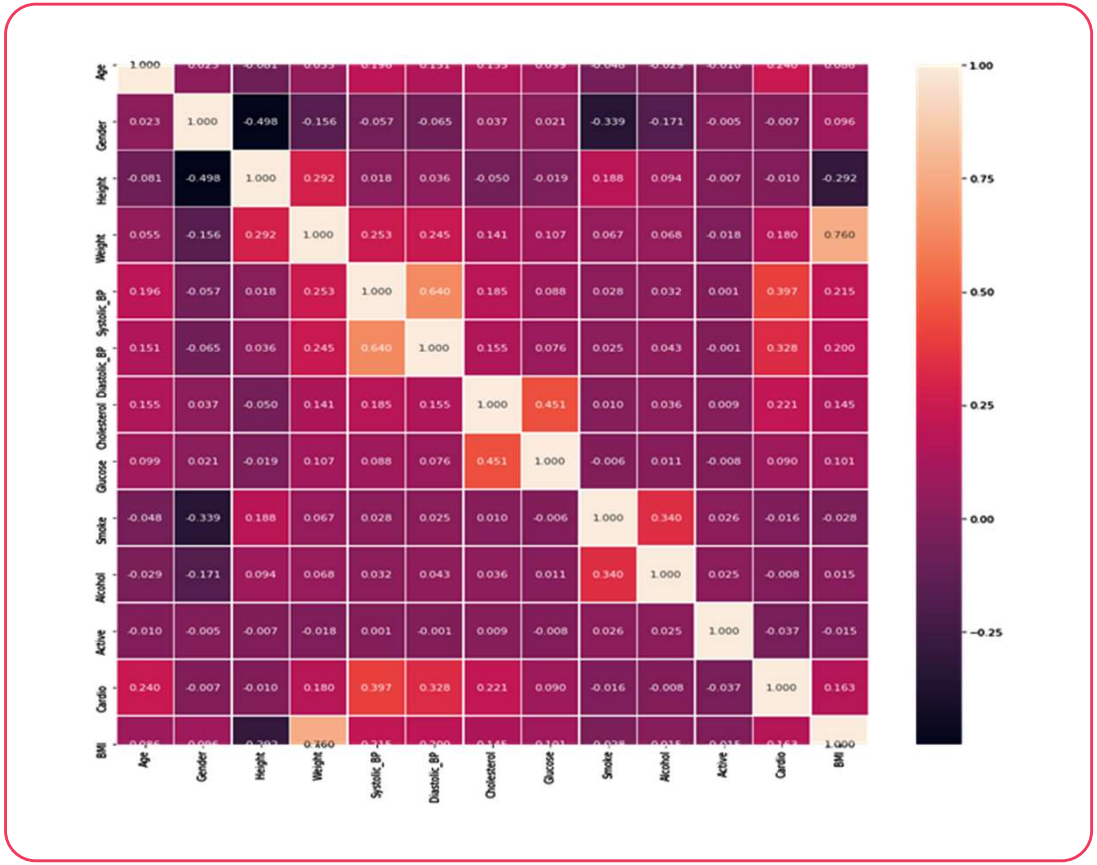
Descriptive/Inferential Statistics

- Violin Plot of quartiles for Male vs. Females and their BMI
- We can see the BMI median and quartiles are higher and more robust for those with CVD(1) and that are male(1)



Descriptive/Inferential Statistics

- Correlation heatmap
- Strongest correlations:
- ✓ Weight and BMI (0.760)
- ✓ Systolic BP and Diastolic BP (0.640)
- ✓ Glucose and Cholesterol (0.451)



Data Pre-Processing: Label Encoding

- Label Encoding of Newly Created Categorical Variables:
- Obesity: 1-Normal, 2-Healthy, 3-Overweight, 4-Obesity
- Age_By_Decade: 1-30s, 2-40s, 3-50s, 4-60s
- Age_Bins column was dropped as not necessary for machine learning

Age_By_Decade	BMI	Obesity
3	21.967120	2
3	34.927679	4
3	23.507805	2
2	28.710479	3
2	23.011177	2

Feature Selection

- All variables were selected for possible predictors
- "Cardio" was assigned as the predictor variable y

3. Feature Selection

In [65]:
```
Feature = x[['Age','Gender','Height','Weight','Systolic_BP','Diastolic_BP','Cholesterol','Glucose','Smoke','Alcohol',
             'Active','Age_By_Decade','BMI','Obesity']]
Feature.head()
```

Out[65]:

	Age	Gender	Height	Weight	Systolic_BP	Diastolic_BP	Cholesterol	Glucose	Smoke	Alcohol	Active	Age_By_Decade	BMI	Obesity
0	50.358324	0	168	62.0	110	80	1	1	0	0	1	3	21.967120	2
1	55.382383	1	156	85.0	140	90	3	1	0	0	1	3	34.927679	4
2	51.628712	1	165	64.0	130	70	3	1	0	0	0	3	23.507805	2
3	48.250135	0	169	82.0	150	100	1	1	0	0	1	2	28.710479	3
4	47.842187	1	156	56.0	100	60	1	1	0	0	0	2	23.011177	2

Model Training

4. Model Training

Prepare data by splitting into training and testing sets.

In [64]:
```
from sklearn.model_selection import train_test_split
X_train,X_test, y_train, y_test = train_test_split(X,y,test_size=0.3,random_state=42)
```

In [65]:
```
# your code
print(f"The shape of the X_train is:", X_train.shape)
print(f"The shape of the y_train is:", y_train.shape)
```

```
The shape of the X_train is: (48288, 14)
The shape of the y_train is: (48288,)
```

- 70% split for Training
- 30% split for Testing

Model Training: Feature Scaling

- Robust Scaler was used
- This removes the median and scales the data according to quantile range
- By default this scaler uses the Inter Quartile Range (IQR) which is range between 1st and 3rd quartiles (Van Dorpe, 2018)
- More robust for evenly mixed categorical and continuous data?

```
In [69]: X_train.head()
```

Out[69]:

	Age	Gender	Height	Weight	Systolic_BP	Diastolic_BP	Cholesterol	Glucose	Smoke	Alcohol	Active	Age_By_Decade	BMI	Obesity
0	1.051394	-1.0	0.818182	-0.294118	0.0	0.0	0.0	2.0	1.0	0.0	0.0	1.0	-0.675643	-0.5
1	0.394204	0.0	-0.636364	-0.588235	-1.0	-1.0	0.0	0.0	0.0	0.0	0.0	0.0	-0.242062	-0.5
2	-0.837343	0.0	-0.636364	-1.000000	0.0	0.0	0.0	0.0	1.0	1.0	0.0	-1.0	-0.691348	-0.5
3	-0.533898	0.0	0.454545	0.000000	-0.5	-1.0	0.0	0.0	0.0	0.0	0.0	-1.0	-0.229607	0.0
4	-0.435210	0.0	0.454545	-0.411765	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	-0.617703	-0.5

Machine Learning Model Selection

Supervised Learning, Classification

- "Supervised Learning" involves predetermined output attributes besides the use of input attributes. The algorithms attempt to predict and classify the predetermined attribute, and their accuracies and misclassification alongside other performance measures is dependent on the counts of the predetermined attribute correctly predicted or classified or otherwise. It is also important to note the learning process stops when the algorithm achieves an acceptable level of performance" (Kotsiantis, 2007).

- Classification: dichotomous dependent variable

0- Does not have CVD

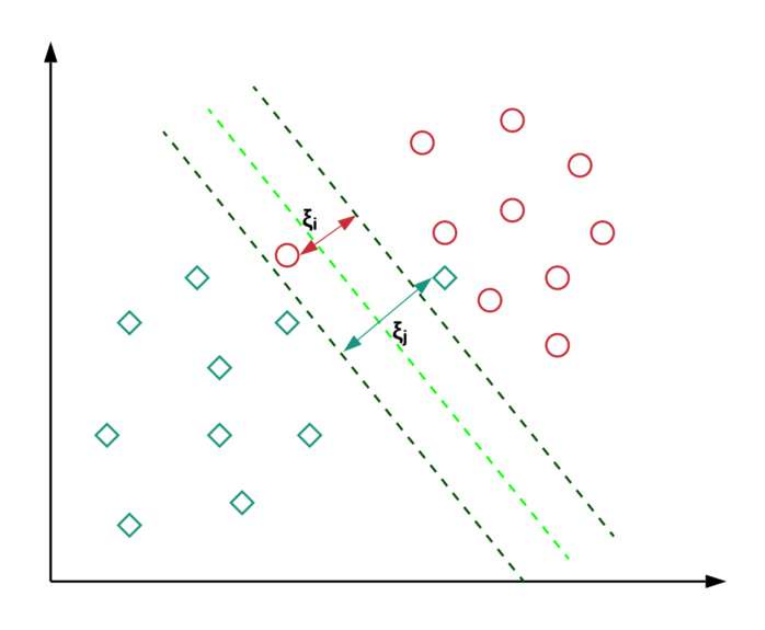
1- Has CVD

- Models:

1. Support Vector Machine (SVM)
2. K-Nearest Neighbors (KNN)
3. Random Forest (RF)
4. Naïve Bayes (NB)

Support Vector Machines

- Objective is to find a hyperplane in an N-dimensional space (N – the number of features) that distinctly classifies the patient has having CVD or not (Gandhi, 2018)
- Find the plane with maximal margin (maximum distance between both classes). By maximizing the margin distance there is re-enforcement, so all future data points are classified with confidence.
- Should work well with clearly defined classes
- Does not work well with exceptionally large datasets



Misra, 2019

K-Nearest Neighbors

Maklin, 2019

Strengths

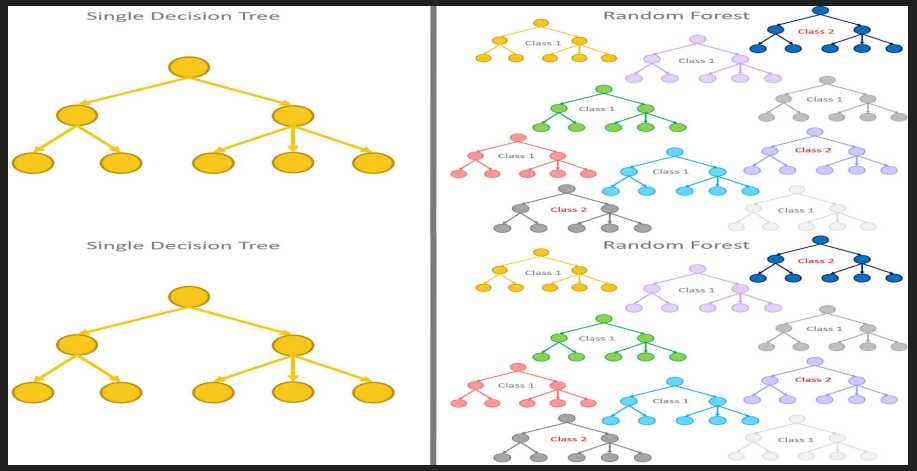
- Classifies data points based on points that are most similar
- Uses test data to make an "educated guess" on what each unclassified point should be classified as
- KNN does not make assumptions about the data which will be great for classifying CVD based on the number of features that we have selected.

Weaknesses

- Accuracy depends on data quality
- Optimal k value (# of nearest neighbors) is crucial
- If boundary is poor classification is poor

Random Forest

- Uses multiple decision trees and performs "bagging" or random sampling at each node of the decision trees to yield classification results
- We are also able to obtain "variable importance' with this algorithm which will tell us which variables are weighted more important than the others for predicting CVD



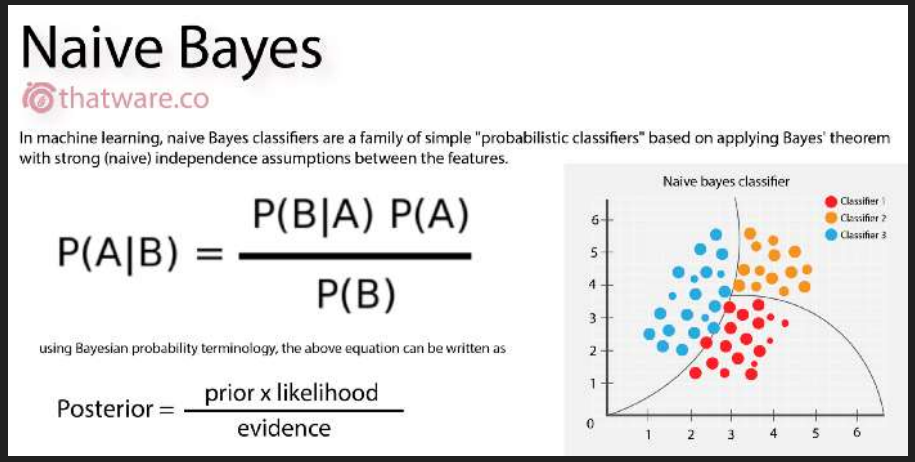
Silipo, 2019

Naïve Bayes

- Based on Bayes' Theorem
- There are two assumptions:

1. We consider all predictors to be independent of each other
2. All predictors have an equal effect on the outcome

- A Gaussian Naïve Bayes will be used assuming a normal distribution with the continuous data (Gandhi, 2018).



Yang, 2019

Weaknesses of Random Forest and Naïve Bayes

Random Forest Weaknesses

- Can overfit model if data is too "noisy"

Naïve Bayes Weaknesses

- We cannot always assume all predictors are independent of one another
- "Zero Frequency". This is when a categorical variable has a category in test data set, which was not observed in training data set, then model will assign a 0 (zero) probability and will be unable to make a prediction (Gandhi, 2018).

Validating the Models

Cross Validation

- 10 folds will be used
- Test models on data it has not seen before!
- Attempt to overcome "bias-variance-trade-off"

Grid Search

- Hyperparameter tuning of models
- "GridSearchCV" library in Python sklearn
- For example, the SVM model hyperparameters can be tested to find the optimal model:

- **C:** This is a regularization parameter
- **Kernel:** We can set the kernel parameter to linear, poly, rbf, sigmoid, precomputed or provide our own callable.
- **Degree:** We can pass in a custom degree to support the poly kernel parameter.
- **Gamma:** This is the coefficient for rbf, poly and sigmoid kernel parameter.
- **Max_Iter:** It is the maximum number of iterations for the solver.

Results

Accuracy Scores

	Accuracy Score
SVM	0.736120
KNN	0.731336
Random forest	0.715680
Naive bayes	0.705098

Jaccard Index and F1 Scores

	Jaccard Index	F1-score
SVM	0.74	0.74
KNN	0.73	0.73
Random Forest	0.72	0.72
Naive Bayes	0.71	0.70

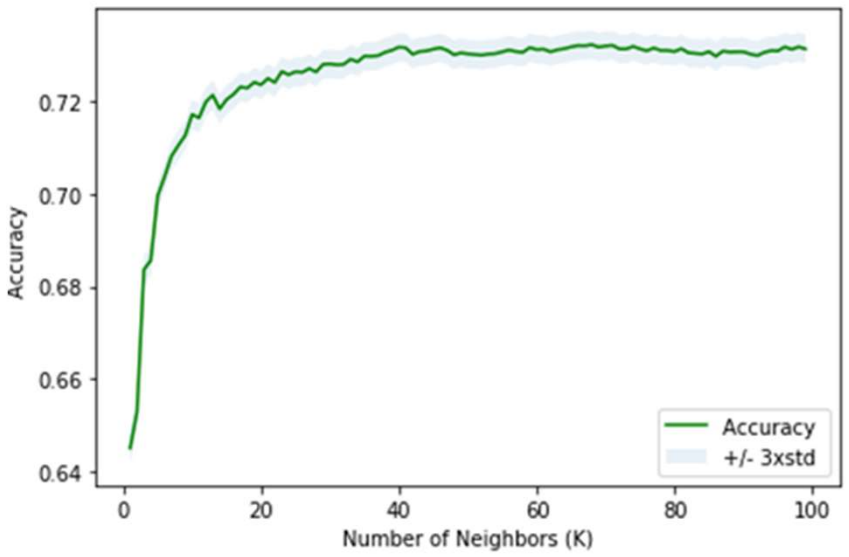
Random Forest Model

- Variable Importance
- Based on Gini Index, higher = better
- Best predictors of CVD are:
  - Height
  - Age_By_Decade
  - Cholesterol
  - Weight
  - Age
  - Gender

	Rank	Variable
0	9	Age
1	8	Gender
2	13	Height
3	10	Weight
4	1	Systolic_BP
5	7	Diastolic_BP
6	11	Cholesterol
7	6	Glucose
8	5	Smoke
9	2	Alcohol
10	3	Active
11	12	Age_By_Decade
12	4	BMI
13	0	Obesity

K-Nearest Neighbors

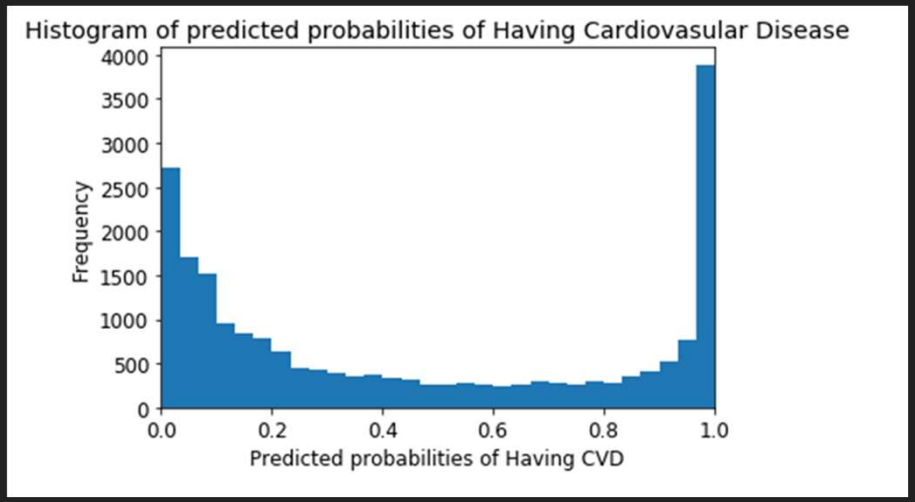
- 100 neighbors were used for model and accuracy was similar to SVM
- Actual number of neighbors should be 68 based on post model testing and this achieves improved accuracy score of 0.732 virtually similar to the SVM (best) model




```
rint( "The best accuracy was with", mean_acc.max(), "with k=",
he best accuracy was with 0.7321575259724571 with k= 68
```

Naïve Bayes – Predicted Probabilities

- We can see this histogram is highly positive skewed.
- The far-right column tells us that there are approximately 4500 observations with probability between 0.8 and 1.0 with probability of having CVD.
- There are relatively small number of observations with probability <0.5 and more with probability between 0.0 and 0.2.
- So, these small number of observations predict that this population will more than likely have CVD.
- **Majority of observations predict that the population will have CVD which corresponds with the best accuracy prediction of 74% by the SVM which says three-quarters of the population of the dataset having these risk factors have CVD.**



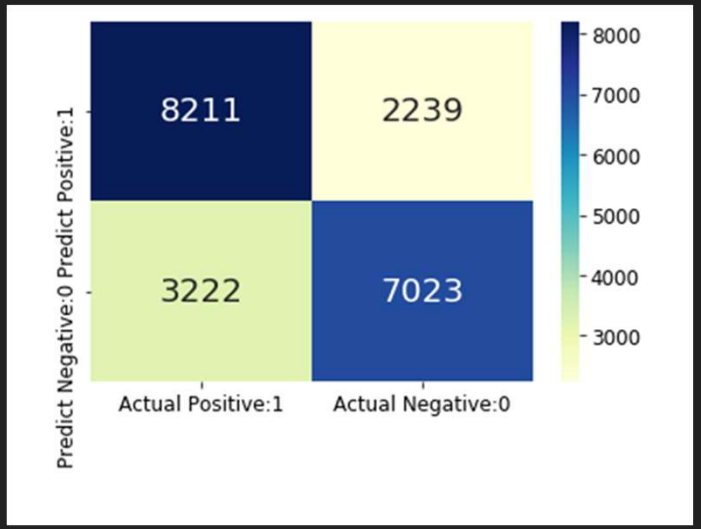
Support Vector Machine – Best Model

- We will Report the following for this model:

1. Confusion Matrix
2. AUC-ROC Curve
3. Precision-Recall

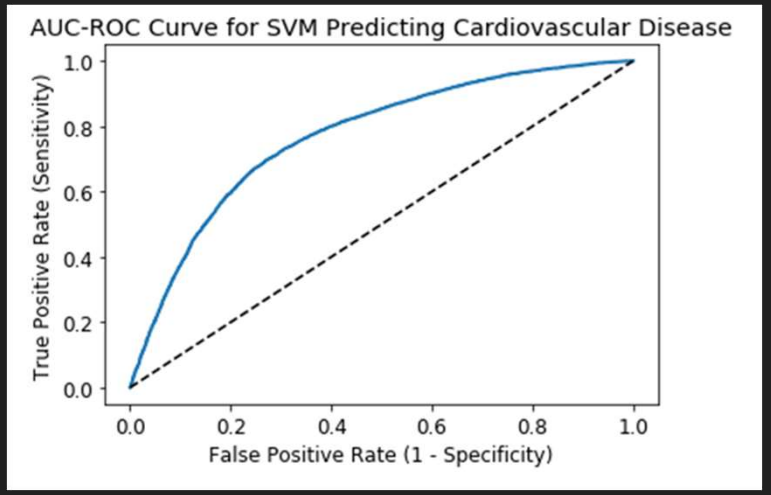
1. Confusion Matrix

- True Positives (Actual Positive:1 and Predict Positive:1) - 8211
- True Negatives (Actual Negative:0 and Predict Negative:0) - 7023
- False Positives (Actual Negative:0 but Predict Positive:1) - 2239 (Type I error)
- False Negatives (Actual Positive:1 but Predict Negative:0) - 3222 (Type II error)



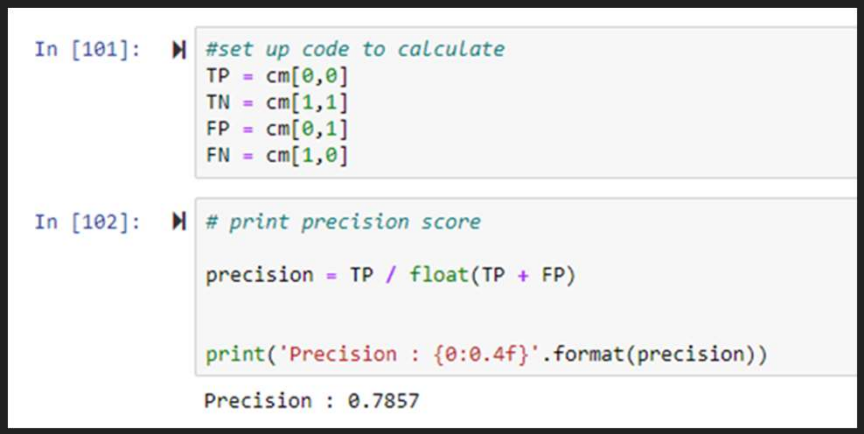
2. AUC-ROC Curve

- The AUC (area under the curve) value was 0.768 which shows the prediction rate is consistent and just below 80%.



3a. Precision

- **Precision**: can be defined as the percentage of correctly predicted positive outcomes out of all the predicted positive outcomes. It can be given as the ratio of true positives (TP) to the sum of true and false positives (TP + FP). **So, Precision identifies the proportion of correctly predicted positive outcomes.** It is more concerned with the positive class than the negative class. Mathematically, precision can be defined as the ratio of TP to (TP + FP).
- Precision is consistent with predictions of the model



```
In [101]:
#set up code to calculate
TP = cm[0,0]
TN = cm[1,1]
FP = cm[0,1]
FN = cm[1,0]

In [102]:
# print precision score

precision = TP / float(TP + FP)


print('Precision : {0:0.4f}'.format(precision))

Precision : 0.7857
```

3b. Recall

- **Recall**: can be defined as the percentage of correctly predicted positive outcomes out of all the actual positive outcomes. It can be given as the ratio of true positives (TP) to the sum of true positives and false negatives (TP + FN). Recall is also called Sensitivity.
- **Recall identifies the proportion of correctly predicted actual positives.**
- We can also see the false positive rate was low at 0.2417

```
In [103]: recall = TP / float(TP + FN)
          print('Recall or Sensitivity : {0:0.4f}'.format(recall))
          Recall or Sensitivity : 0.7182

          True Positive Rate is synonymous with Recall.

In [104]: true_positive_rate = TP / float(TP + FN)

          print('True Positive Rate : {0:0.4f}'.format(true_positive_rate))
          True Positive Rate : 0.7182

          False Positive Rate

In [105]: false_positive_rate = FP / float(FP + TN)

          print('False Positive Rate : {0:0.4f}'.format(false_positive_rate))
          False Positive Rate : 0.2417
```

Cross Validation Results

SVM Model

- Accuracy: 0.736 (virtually same as original model)
- Standard Deviation: 0.004
- Model consistent!

KNN Model

- Accuracy: 0.731 (virtually same as original model)
- Standard Deviation: 0.004
- Model consistent!

Grid Search Results – SVM (Best Model)

Parameters Tested for SVM

```
from sklearn.model_selection import GridSearchCV

parameters = [{
    'kernel': ['linear','poly','rbf','sigmoid'],
    'C': [1,2,3,300,500],
    'max_iter': [1000,100000]
}]
```

Hyperparameters Identified:

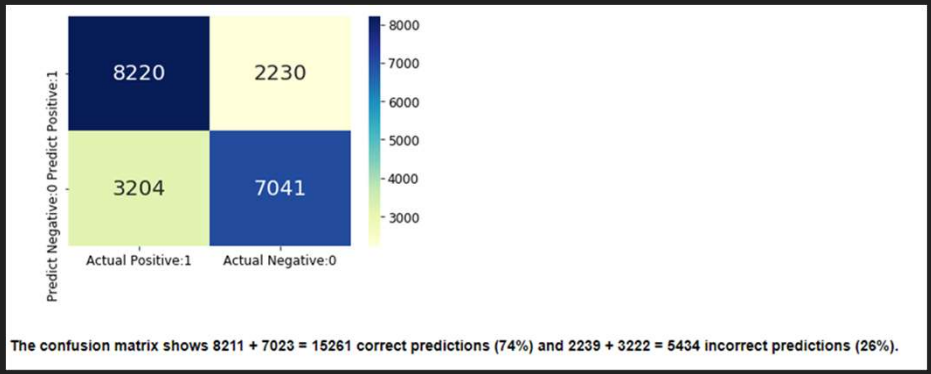
- C of 3
- Kernel: RBF (Radial Basis Function)
- Max_Iter: 100,000

```
In [112]:
#find best parameters
SVM_cv = GridSearchCV(svm, parameters, cv=3) # GridSearchCV
SVM_cv.fit(X_train,y_train)# Fit

# Print hyperparameter
print("Tuned hyperparameters: {}".format(SVM_cv.best_params_))
print("Best score: {}".format(SVM_cv.best_score_))

Tuned hyperparameters: {'C': 3, 'kernel': 'rbf', 'max_iter': 100000}
Best score: 0.7319831013916501

In [113]:
#run the model with best hyperparameters
SVM_best = SVC(C=3,kernel='rbf',max_iter=100000)
SVM_best.fit(X_train, y_train)
print("Test accuracy: ",SVM_best.score(X_test, y_test))

Test accuracy:  0.7374244986711767
```

Better prediction accuracy!!

SVM – Final Optimized Model

Confusion Matrix



- Precision, Recall, F1 Score

	Precision	Recall	F1 Score
Results	0.759465	0.687262	0.721562

SVM Final Model vs. Original Model

1. Precision decreased from 0.78 to 0.72
2. Recall decreased from 0.72 to 0.68
3. F1 score decreased from 0.74 to 0.72
4. The Jaccard Index remained the same at 0.74
   5. The number of True Positives went up from 8211 to 8220
   6. The number of True Negatives went up from 7023 to 7041
   7. The number of False Positives (Type 1 error) went down from 2239 to 2230
   8. The number of False Negatives (Type 2 error) went down from 3222 to 3204
   9. Accuracy increased from 0.7361 to 0.7374

- **In summary, the best SVM model overall showed a 74% prediction rate for CVD.**

Discussion

Overall

- SVM Best model
- Accuracy: 74%
- Ensemble Model Average: 72.2%
- Precision: 0.78
- Recall: 0.72
- F1 Score: 0.74
- Jaccard Index: 0.74
- True Positives: 8220

Why did Precision, Recall and F1 Score go down when model was optimized with Grid Search?

- Bias-Variance Trade off!!
- Bias is the difference between the average prediction of our model and the correct value which we are trying to predict. Models with high bias pay little attention to the training data and oversimplifies the model.
- Variance is the variability of model prediction for a given data point or a value which tells us spread of our data. Models with high variance pay a lot of attention to training data and does not generalize on the data which it has not seen before. As a result, such models perform very well on training data but has high error rates on test data.
- **Best model = Balance between bias and variance and minimize total error**
- **SVM Model had good balance!**

SVM Model Compared to other studies?

Other Studies Predicting CVD (Latha et al. 2019):

- Naïve Bayes: 69.11%
- Random Forests: 85%
- KNN: 99.65%
- SVM: 85%

What Factors play a role in prediction capacity?

- Size of dataset
- Random Forest better on larger dataset
- SVM better on smaller dataset
- "Every algorithm has own intrinsic capacity to outperform other algorithms depending upon situation (Khourdifi et al. 2019)."

Predictors of CVD based on RF Variable Importance?

Best Predictors of CVD for this project:

- Height
- Age_By_Decade
- Cholesterol
- Weight
- Age
- Gender

All Correspond with most common predictors of CVD!

Worst Predictors of CVD

- Obesity
- Systolic BP
- Alcohol
- Activity level
- BMI

**These are only from Random Forest and may vary by model and data type

Summary of Other Model Findings

- Naïve Bayes: predicted there are approximately 4500 observations with probability between 0.8 and 1.0 with probability of having CVD. There are relatively small number of observations with probability <0.5 and more with probability between 0.0 and 0.2. This is based on the probability threshold of 0.5 or greater having CVD.
- KNN: prediction as good as SVM with k=68

Cross Validation and Grid Search

SVM best accuracy: 74%

Best SVM Kernel: 'RBF' (Radial Basis Function)

The function of kernel is to take data as input and transform it into the required form. Different SVM algorithms use different types of kernel functions.

RBF most used kernel because it has localized and finite response along entire x-axis (Dataflair, 2018)

Weaknesses of Project

- Outlier removal – skewed results?
- 3 patients were 29 yrs. old and included in 30-39 age group – skewed results?
- Dataset: more common risk factors such as smoking, alcohol and activity level were more normal in those that had CVD which is usually not common in the general population

Age groups: all between 30 to 65!

- We know that CVD prevalence is: ~40% age 40-59, ~75% age 60-79, and ~86% age 80 and over (Rodgers et al. 2019).
- The dominant age group in this study was 50-59, followed by 40-49, and lastly 60-69.
- So, the sample size of one of the most dominant groups (60-69) was much smaller at almost half of the most dominant age group in this study.
- Age groups were not even in sample size which is not fair to the predictions we make.

Weaknesses of Project

Data Pre-Processing: Robust Scaler?

- Distance algorithms like KNN, K-Means, and SVM are most affected by the range of features. This is because behind the scenes **they are using distances between data points to determine their similarity.**
- **Therefore, scaling is used to reduce the chance that higher weight is given to features with higher magnitude. Tree based algorithms like Random Forests are insensitive to feature scaling** **(Bhandari, 2020).**

- Whether to standardize (scale) or normalize the data depends on if there is a Gaussian distribution.
- Normalization is used when there is not a Gaussian distribution
- Standardization is used when there IS a Gaussian distribution.
- In the end it is dependent upon the algorithms and the data. I do believe that feature scaling was appropriate to normalize the data as there were some high-level features and an even mix of continuous and categorical data points, so we had to "level the playing field".

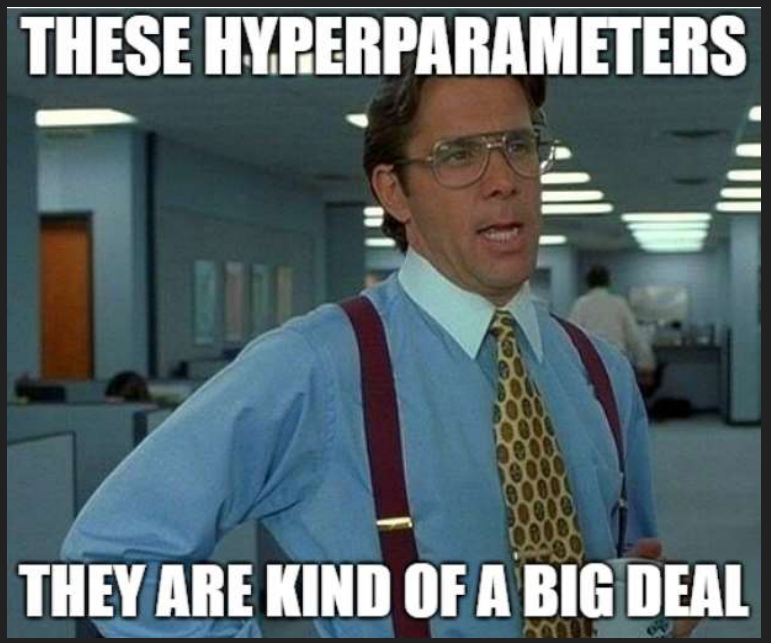
Future Directions

- Focus on more Ensemble Modeling
- More Robust Models: XGBoost, AdaBoost
- More Grid Search on other models
- Different data pre-processing techniques? Don't remove outliers?
- Bagging and Boosting improve accuracy of models for CVD prediction (Khourdifi et al. 2019)
- Further inferential statistical analysis?
- More even samples of each age group
- BMI vs. Weight had strong correlation in our exploratory analysis, but we did not look further into this --- linear or polynomial regression techniques?
- BMI and Obesity further evaluation

Conclusion

Conclusions

- SVM Model predicted CVD with accuracy of 74%
- Grid Search and Cross Validation helped optimize hyperparameters and prove model accuracy
- Random Forest provided variable importance: Height, Age_By_Decade, Cholesterol, Weight, Age, Gender = all most important predictors of CVD!
- Future modeling should use even random samples of each age group



References

- Bhandari, A. (2020) Feature Scaling for Machine Learning: Understanding the Difference Between Normalization vs. Standardization. Retrieved from: https://www.analyticsvidhya.com/blog/2020/04/feature-scaling-machine-learning-normalization-standardization/
- Cho et al. (2020) Summary of Updated Recommendations for Primary Prevention of Cardiovascular Disease in Women. JACC State of the Art Review. Vol. 75(20): DOI: 10.1016/j.jacc.2020.03.060
- Dataflair (2018) Kernel Functions – Introduction to SVM Kernel & Examples. Retrieved from: https://data-flair.training/blogs/svm-kernel-functions/
- Gandhi, R. (2018) Naïve Bayes Classifier. Retrieved from: https://towardsdatascience.com/naive-bayes-classifier-81d512f50a7c
- Gandhi, R. (2018) Support Vector Machine – Introduction to Machine Learning Algorithms. Retrieved from: https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47
- Hajar, R. (2016) Framingham Contribution to Cardiovascular Disease. Heart Views. Vol. 17(2):78-81.
- Khourdifi et al. (2019) Heart Disease Prediction and Classification Using Machine Learning Algorithms Optimized by Particle Swarm Optimization and Ant Colony Optimization. International Journal of Intelligent Engineering & Systems. Vol. 12(1):242-252.
- Kotsiantis, S (2007). Supervised machine learning: A review of classification techniques. Informatica, 31, 249–268.

References

- Latha et al. (2019) Improving the accuracy of prediction of heart disease risk based on ensemble classification techniques. Informatics in Medicine Unlocked. Vol. 16.
- Maklin, C. (2019) K Nearest Neighbor Algorithm In Python. Retrieved from: towardsdatascience.com
- Misra, R. (2019) Support Vector Machines – Soft Margin Formulation and Kernel Trick. Retrieved from: towardsdatascience.com
- Narloch et al. (1995) Influence of breathing technique on arterial blood pressure during heavy weightlifting. Arch Phys Med Rehabil. Vol. 76(5): 457-62.
- Rodgers et al. (2019) Cardiovascular Risks Associated with Gender and Aging. Journal of Cardiovascular Development and Disease. Vol. 6(19): 1-18.
- Silipo, R. (2019) From a Single Decision Tree to a Random Forest. Retrieved from: towardsdatascience.com
- Van Dorpe, S. (2018) Preprocessing with sklearn: a complete and comprehensive guide. Retrieved from: https://towardsdatascience.com/preprocessing-with-sklearn-a-complete-and-comprehensive-guide-670cb98fcfb9
- Yang, S. (2019) An Introduction to Naïve Bayes Classifier. Retrieved from: towardsdatascience.com