1. Use the given link Data Set

Answer the below questions:

1. Find out top 5 attributes having highest correlation (select only Numeric features).

|  |  |
| --- | --- |
|  | # Use the library cluster generation to make a positive definite matrix of 15 features    library(clusterGeneration)    S = genPositiveDefMat("unifcorrmat",dim=15)    # create 15 features using multivariate normal distribution for 5000 datapoints    library(mnormt)    n = 5000  X = rmnorm(n,varcov=S$Sigma) |

# Create a two class dependent variable using binomial distribution

Y = rbinom(n,size=1,prob=0.3)

data = data.frame(Y,X)

# Create a correlation table for Y versus all features

cor(data,data$Y)

library(mlbench)

data(PimaIndiansDiabetes)

data\_lm = as.data.frame(PimaIndiansDiabetes)

# Fit a logistic regression model

fit\_glm = glm(diabetes~.,data\_lm,family = "binomial")

# generate summary

summary(fit\_glm)

Call:

glm(formula = diabetes ~ ., family = "binomial", data = data\_lm)

Deviance Residuals:

Min        1Q      Median   3Q          Max

-2.5566  -0.7274  -0.4159   0.7267   2.9297

Coefficients:

                 Estimate          Std. Error        z value            Pr(>|z|)

(Intercept)       -8.4046964      0.7166359       -11.728            < 2e-16           \*\*\*

pregnant          0.1231823       0.0320776       3.840               0.000123         \*\*\*

glucose           0.0351637       0.0037087       9.481               < 2e-16           \*\*\*

pressure          -0.0132955      0.0052336       -2.540              0.011072         \*

triceps           0.0006190       0.0068994       0.090               0.928515

insulin           -0.0011917      0.0009012       -1.322              0.186065

mass              0.0897010       0.0150876       5.945               2.76e-09          \*\*\*

pedigree          0.9451797       0.2991475       3.160               0.001580         \*\*

age               0.0148690       0.0093348       1.593               0.111192

---

Signif. codes:  0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 993.48  on 767  degrees of freedom

Residual deviance: 723.45  on 759  degrees of freedom

AIC: 741.45

library(caret)

varImp(fit\_glm)

library(clusterGeneration)

S = genPositiveDefMat("unifcorrmat",dim=15)

#create 15 features using multivariate normal distribution for 5000 datapoints

library(mnormt)

n = 5000

X = rmnorm(n,varcov=S$Sigma)

# Create a two class dependent variable using binomial distribution

Y = rbinom(n,size=1,prob=0.3)

data = data.frame(Y,X)

# Create a correlation table for Y versus all features

cor(data,data$Y)

# Using the mlbench library to load diabetes data

library(mlbench)

data(PimaIndiansDiabetes)

data\_lm=as.data.frame(PimaIndiansDiabetes)

# Fit a logistic regression model

fit\_glm=glm(diabetes~.,data\_lm,family = "binomial")

# generate summary

summary(fit\_glm)

# Using varImp() function

library(caret)

varImp(fit\_glm)

#Import the random forest library and fit a model

library(randomForest)

fit\_rf=randomForest(diabetes~., data=data\_lm)

# Create an importance based on mean decreasing gini

importance(fit\_rf)

# compare the feature importance with varImp() function

varImp(fit\_rf)

# Create a plot of importance scores by random forest

varImpPlot(fit\_rf)

1. Find out top 3 reasons for having more crime in a city.
2. *#group Violent crime and nonViolent crime by state*
3. crimedata\_state = crimedata.groupby('state').agg({'ViolentCrimesPerPop':'mean','nonViolPerPop':'mean'})[['ViolentCrimesPerPop','nonViolPerPop']].reset\_index()
4. *##Aggregate view of Non-Violent Crimes by State*
5. data1 = dict(type='choropleth',
6. colorscale = 'Viridis',
7. autocolorscale = False,
8. locations = crimedata\_state['state'],
9. locationmode = 'USA-states',
10. z = crimedata\_state['nonViolPerPop'].astype(float),
11. colorbar = {'title':'non-Violent Crimes(Per-100K-Pop)'}
12. )
13. layout1 = dict(
14. title = 'Aggregate view of non-Violent Crimes Per 100K Population',
15. geo = dict(
16. scope='usa',
17. projection=dict( type='albers usa' ),
18. showlakes = True,
19. lakecolor='rgb(85,173,240)'),
20. )
22. fig1 = go.Figure(data = [data1],layout = layout1)
23. iplot(fig1,validate=False)
24. *##Aggregate view of Violent Crimes by State*
25. data2 = dict(type='choropleth',
26. autocolorscale = False,
27. colorscale = "Earth",
28. locations = crimedata\_state['state'],
29. locationmode = 'USA-states',
30. z = crimedata\_state['ViolentCrimesPerPop'].astype('float'),
31. colorbar = {'title':'Violent Crimes(Per-100K-Pop)'}
32. )
33. layout2 = dict(
34. title = 'Aggregate view of Violent Crimes Per 100K Population across US',
35. geo = dict(
36. scope='usa',
37. projection=dict( type='albers usa' ),
38. showlakes = True,
39. lakecolor = 'rgb(85,173,240)'),
40. fig2 = go.Figure(data = [data2],layout = layout2)
41. iplot(fig2,validate=False)
42. *## Check if there is multicollinearity between any of the explanatory variables under study*
43. cols = ['HousVacant','PctHousOccup','PctHousOwnOcc','PctVacantBoarded','PctVacMore6Mos','PctUnemployed','PctEmploy','murdPerPop','rapesPerPop','robbbPerPop','assaultPerPop','ViolentCrimesPerPop','burglPerPop','larcPerPop','autoTheftPerPop','arsonsPerPop','nonViolPerPop']
44. crimedata\_study = crimedata.filter(cols, axis=1)
45. corr\_crimedata\_study = crimedata\_study.corr()
46. iv\_corr = corr\_crimedata\_study.iloc[:-10,:-10]
47. multicol\_limit = 0.3
48. iv\_corr = (iv\_corr[abs(iv\_corr) > multicol\_limit][iv\_corr != 1.0]).unstack().to\_dict()
49. iv\_multicoll\_corr = pd.DataFrame(list(set([(tuple(sorted(key)), iv\_corr[key]) for key **in** iv\_corr])),
50. columns=['Independent Variables', 'Correlation Coefficient'])
51. print (iv\_multicoll\_corr[iv\_multicoll\_corr.notnull().all(axis=1)])

c. Which all attributes have correlation with crime rate?

> crime.data <- read.csv(url.data, na.strings¼ ‘’)

str(crime.data)

> summary(crime.data)

> crime.data <- subset(crime.data, !duplicated(crime.data$CASE.)) > summary(crime.data)

> crime.data <- subset(crime.data, !is.na(crime.data$LATITUDE)) > crime.data <- subset(crime.data, !is.na(crime.data$WARD))

> which(is.na(crime.data$LOCATION)) > crime.data <- crime.data[-which(is.na(crime.data$LOCATION)), ]

> crime.data <- subset(crime.data, !is.na(crime.data$LATITUDE)) > crime.data <- subset(crime.data, !is.na(crime.data$WARD))

> crime.data <- crime.data[crime.data$CASE. !¼ ‘CASE#’,]

> head(crime.data$DATE..OF.OCCURRENCE)

> crime.data$date <- as.POSIXlt(crime.data$DATE..OF.OCCURRENCE, format¼ “%m/%d/%Y %H:%M”) > head(crime.data$date)

> library(chron) > crime.data$time <- times(format(crime.data$date, “%H:%M:%S”)) > head(crime.data$time)

crime.data$date <- as.POSIXlt(crime.data$DATE..OF.OCCURRENCE, format¼ “%m/%d/%Y %I:%M %p”).