Support Vector Machine

Ashish Toppo

Dataset Link: Social Network Ads Datasets
Using this data set I need to predict the whether a person will buy the product after seeing the advertisement.
Description:
Content
It includes age and estimated salary of the user. The purchased column indicates weather the particular user with age and estimated salary have bought the product or not by viewing the social ads of the product.

0 : No
1 : Yes

Reading of the dataset and importing necessary libraries

```
library(tidyverse)
library(reshape2)
library(caTools)
library(corrplot)
library(MLmetrics)
library(e1071)
data_0 <- read.csv("Social_Network_Ads.csv", stringsAsFactors = TRUE) #converting all strings to factor
```

```
#peeping into data
head(data_0)
```

```
##    User.ID Gender Age EstimatedSalary Purchased
## 1 15624510   Male  19           19000         0
## 2 15810944   Male  35           20000         0
## 3 15668575 Female  26           43000         0
## 4 15603246 Female  27           57000         0
## 5 15804002   Male  19           76000         0
## 6 15728773   Male  27           58000         0
```

```
# dimension of the dataset
dim(data_0)
```

```
## [1] 400   5
```

```
# structure of the dataset
str(data_0)
```

```
## 'data.frame':    400 obs. of  5 variables:
##  $ User.ID        : int  15624510 15810944 15668575 15603246 15804002 15728773 15598044 15694829 156
```

```
##  $ Gender         : Factor w/ 2 levels "Female","Male": 2 2 1 1 2 2 1 1 2 1 ...
##  $ Age            : int  19 35 26 27 19 27 27 32 25 35 ...
##  $ EstimatedSalary: int  19000 20000 43000 57000 76000 58000 84000 150000 33000 65000 ...
##  $ Purchased      : int  0 0 0 0 0 0 0 1 0 0 ...
```

Summary Statistics

```
summary(data_0)
```

```
##     User.ID            Gender         Age        EstimatedSalary
##  Min.   :15566689   Female:204   Min.   :18.00   Min.   : 15000
##  1st Qu.:15626764   Male  :196   1st Qu.:29.75   1st Qu.: 43000
##  Median :15694342                Median :37.00   Median : 70000
##  Mean   :15691540                Mean   :37.66   Mean   : 69743
##  3rd Qu.:15750363                3rd Qu.:46.00   3rd Qu.: 88000
##  Max.   :15815236                Max.   :60.00   Max.   :150000
##    Purchased
##  Min.   :0.0000
##  1st Qu.:0.0000
##  Median :0.0000
##  Mean   :0.3575
##  3rd Qu.:1.0000
##  Max.   :1.0000
```

Observations:
1. Variable "Purchased" needs to be converted into a categorical variable.
2. Variable "Gender" will be converted into binary.
3. Variable "EstimatedSalary" need to be normalized(min-max normalization) after doing EDA on this variable.
4. Dropping "User.ID" variable.

Missng Values and their proportions by columns

```
null_cols <- colSums(is.na(data_0))/nrow(data_0)*100
null_cols[null_cols!=0]
```

```
## named numeric(0)
```

There aren't any missing data in any of the columns.

Normaization

```
normalisation <- function(df,col){
  #`custom function for min-max normalization
  df[col] <- (df[col]-min(df[col]))/(max(df[col])-min(df[col]))
  df
}
```

Preparing the columns

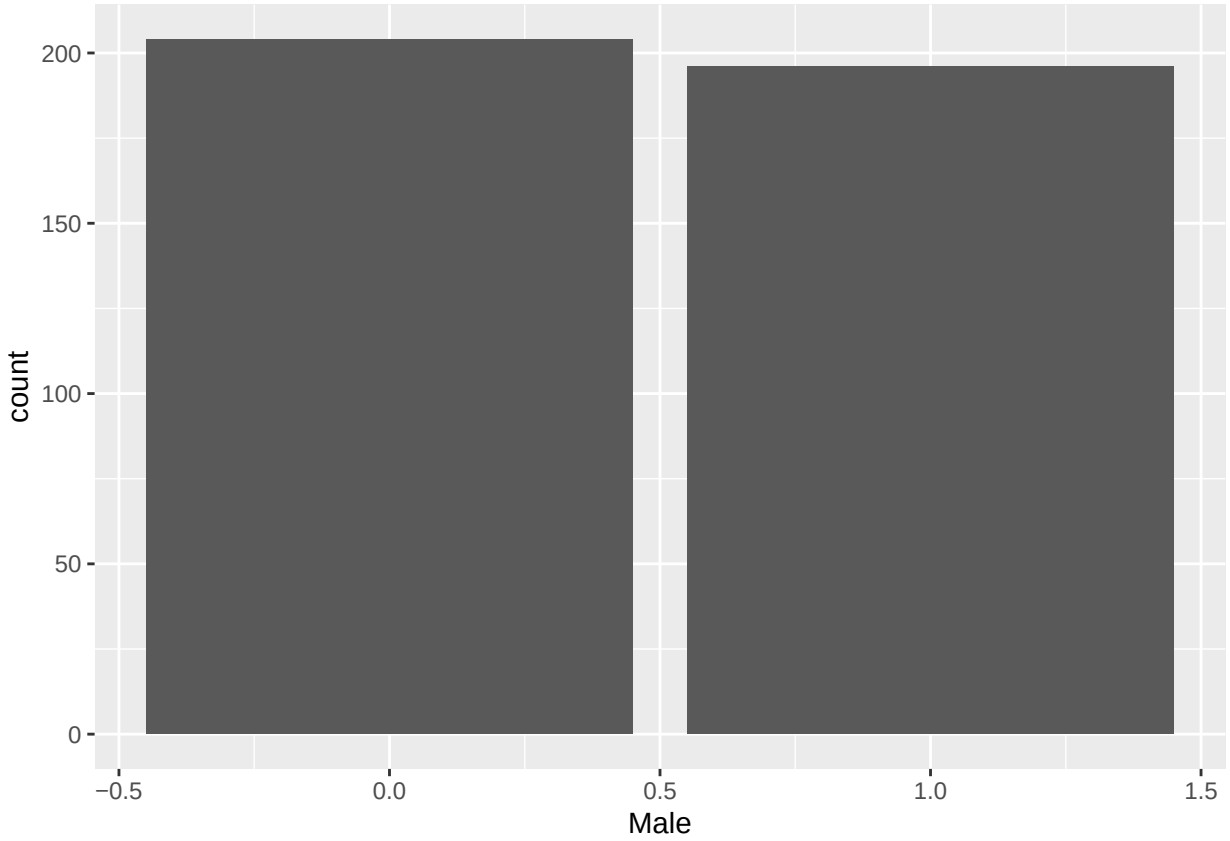
```
#normalizing EstimatedSalary
#data_1 <- normalisation(data_0,"EstimatedSalary")
#dropping User.ID
data_1 <- data_0 %>% select(!User.ID)
#Converting Gender to binary
data_1$Gender <- ifelse(data_1$Gender=="Male",1,0)
#renaming the column to Male
data_1 <- data_1 %>% rename(Male=Gender)
#converting Purchased into a factor
data_1$Purchased <- as.factor(data_1$Purchased)
#look into data
head(data_1)
```

```
##   Male Age EstimatedSalary Purchased
## 1    1  19           19000         0
## 2    1  35           20000         0
## 3    0  26           43000         0
## 4    0  27           57000         0
## 5    1  19           76000         0
## 6    1  27           58000         0
```

Looking into Columns for EDA

for Male

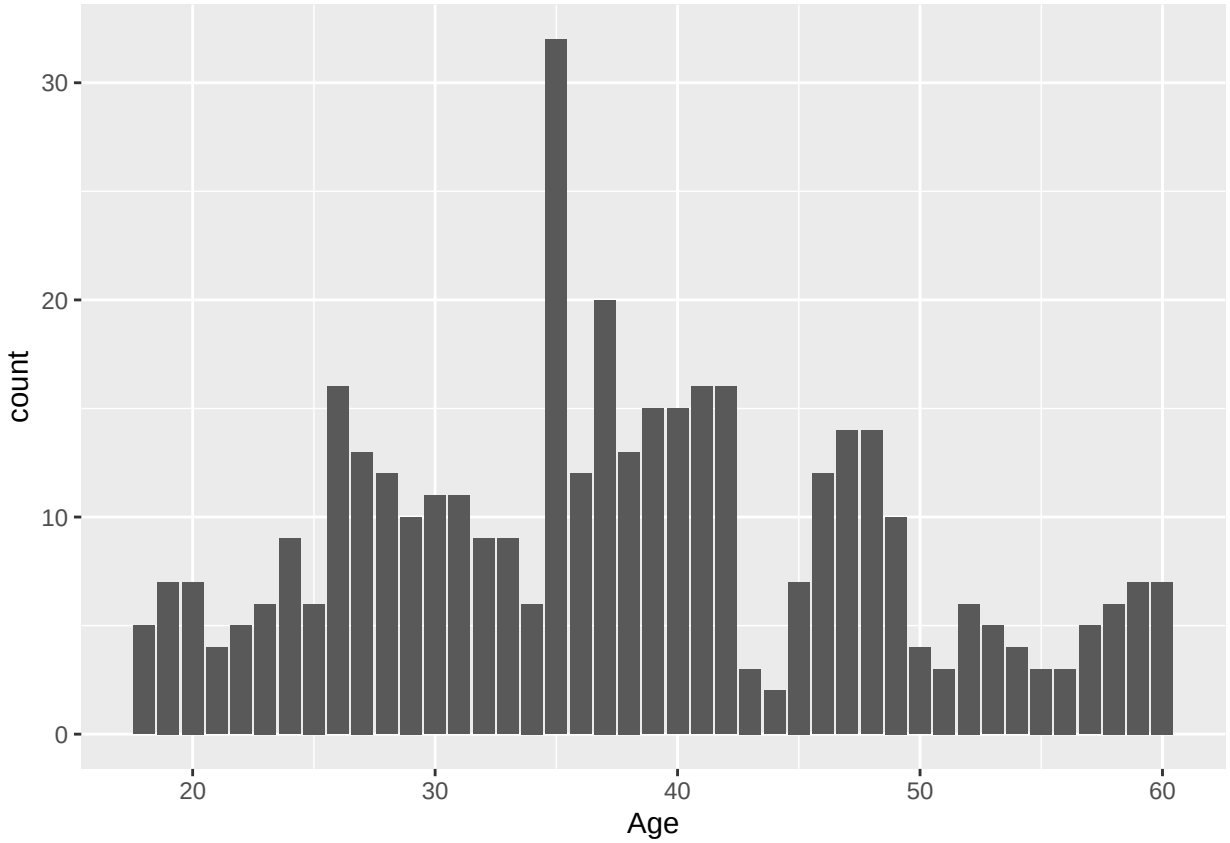
```
data_1 %>% ggplot(aes(Male))+geom_bar()
```



Observations:
1. Both male and female are almost in equal proportion.
2. More than half of data points(<200) are males.

for Age

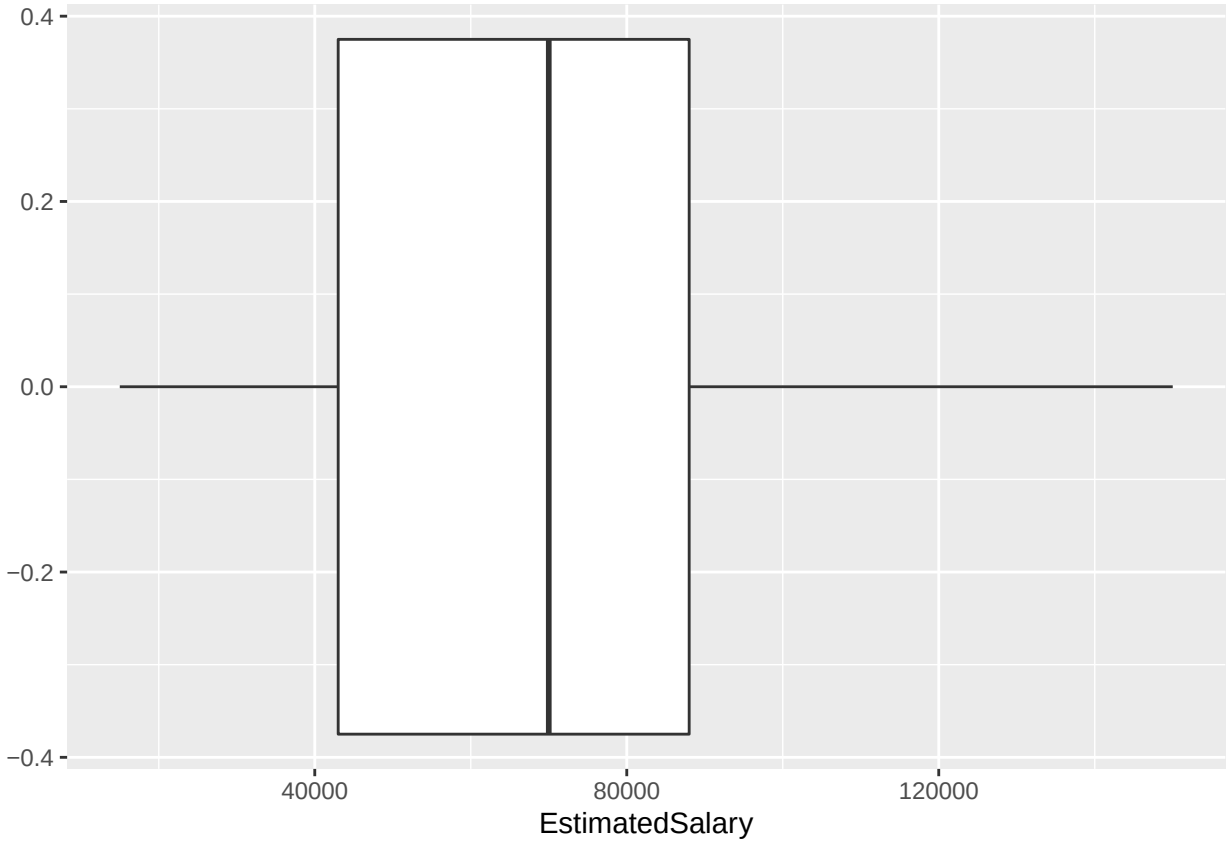
```
data_1 %>% ggplot(aes(Age))+geom_bar()
```



Observations:
1. There is a wavy pattern seen through the bar plot which decreases every age ending with 5(25,35,45,55). Some more depth to identifying this pattern is needed.
2. People of age 35 are in the dataset. This might be because most of the population is middle aged, as there are more no. of within age 25 and 45 who are exposed to ads.
3. It's worth checking which gender makes the purchases with age 35.

for EstimatedSalary

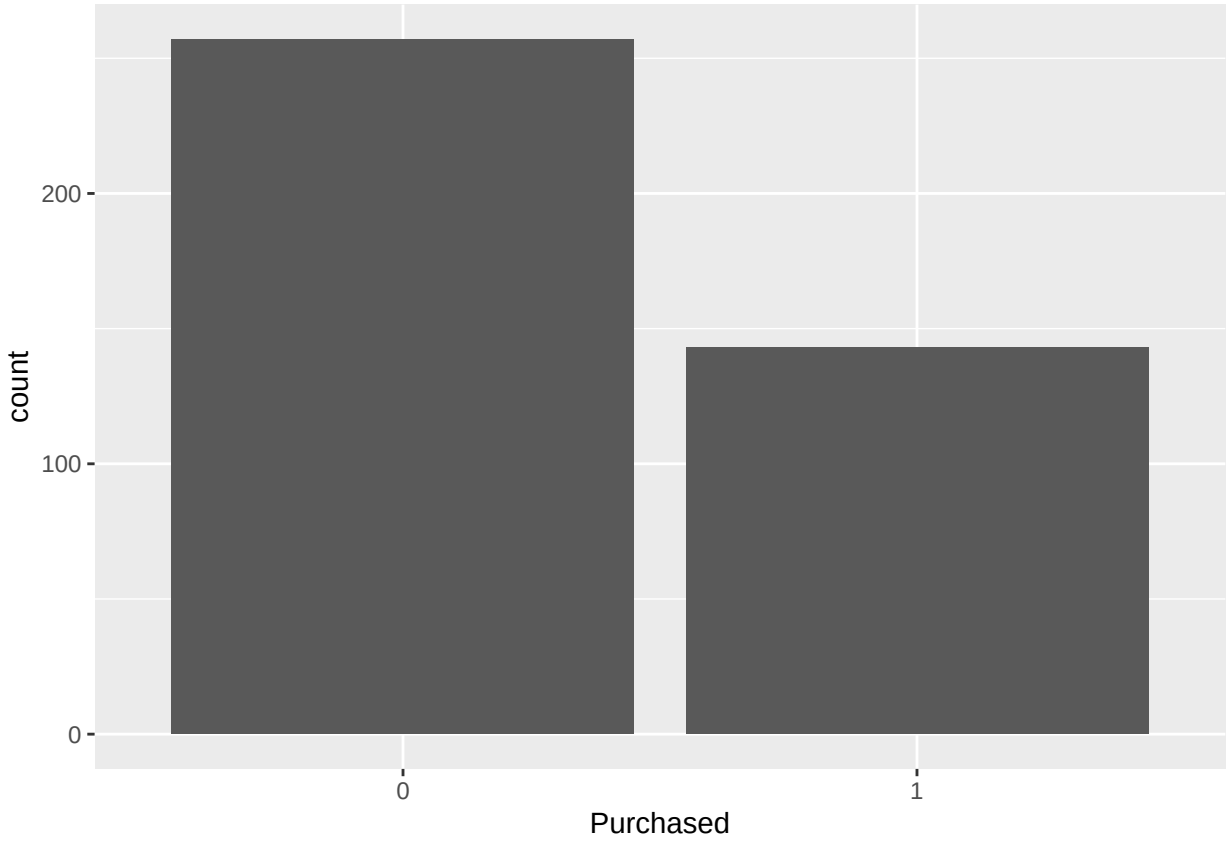
```
data_1 %>% ggplot(aes(EstimatedSalary))+geom_boxplot()
```



Observations:
1. There aren't any outliers.
2. The approx estimated salary of the user is 70000. 3. 50% of the users have a salary within 42000 to 90000.
4. Only 25% of the users have salary above 90000.

for Purchased

```
data_1 %>% ggplot(aes(Purchased))+geom_bar()
```

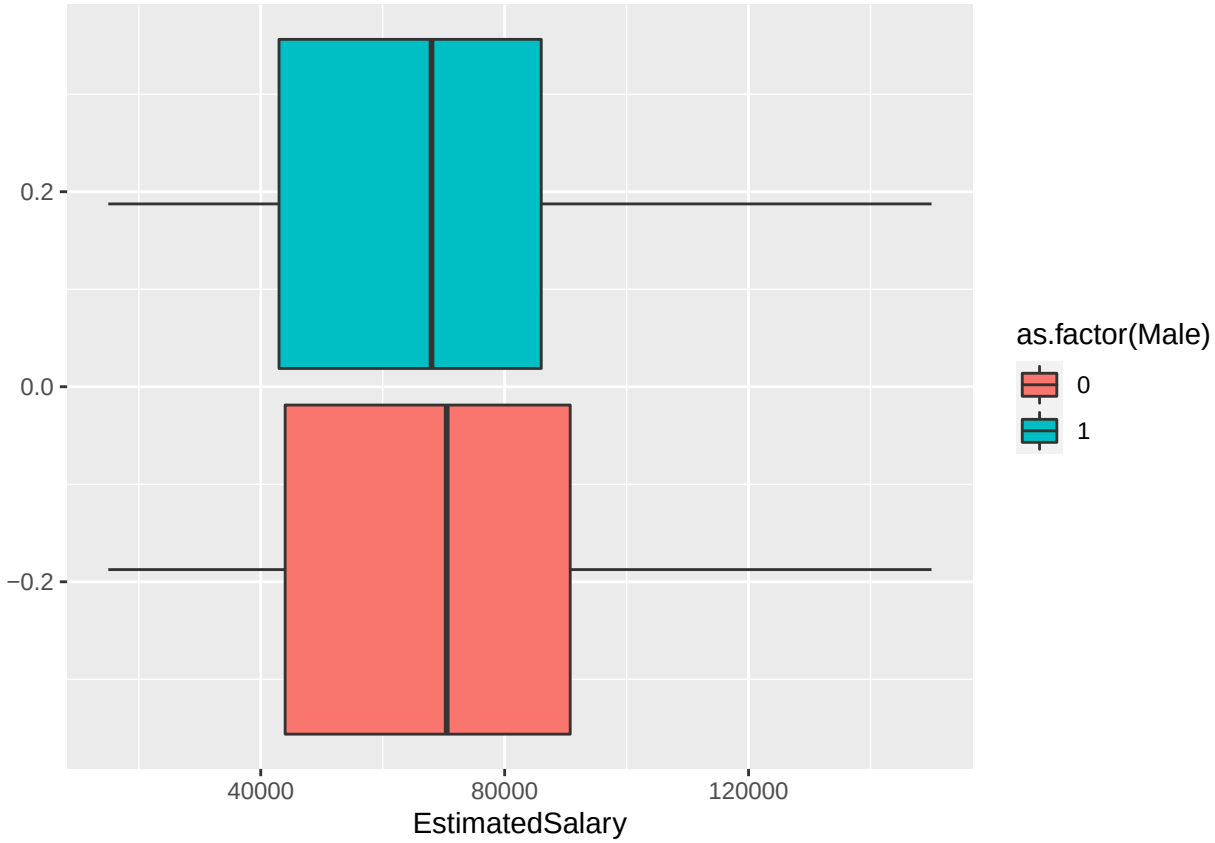


Observations:
1. The class labels are imbalanced it might effect the SVM model.
2. It can be seen that almost 35% users made a purchase after seeing the ads.
3. It can be because they spent more time on screen. This measurement can help to come up with new advertisement techniques.

Some Analysis

Q *How salary is distributed for male and female*

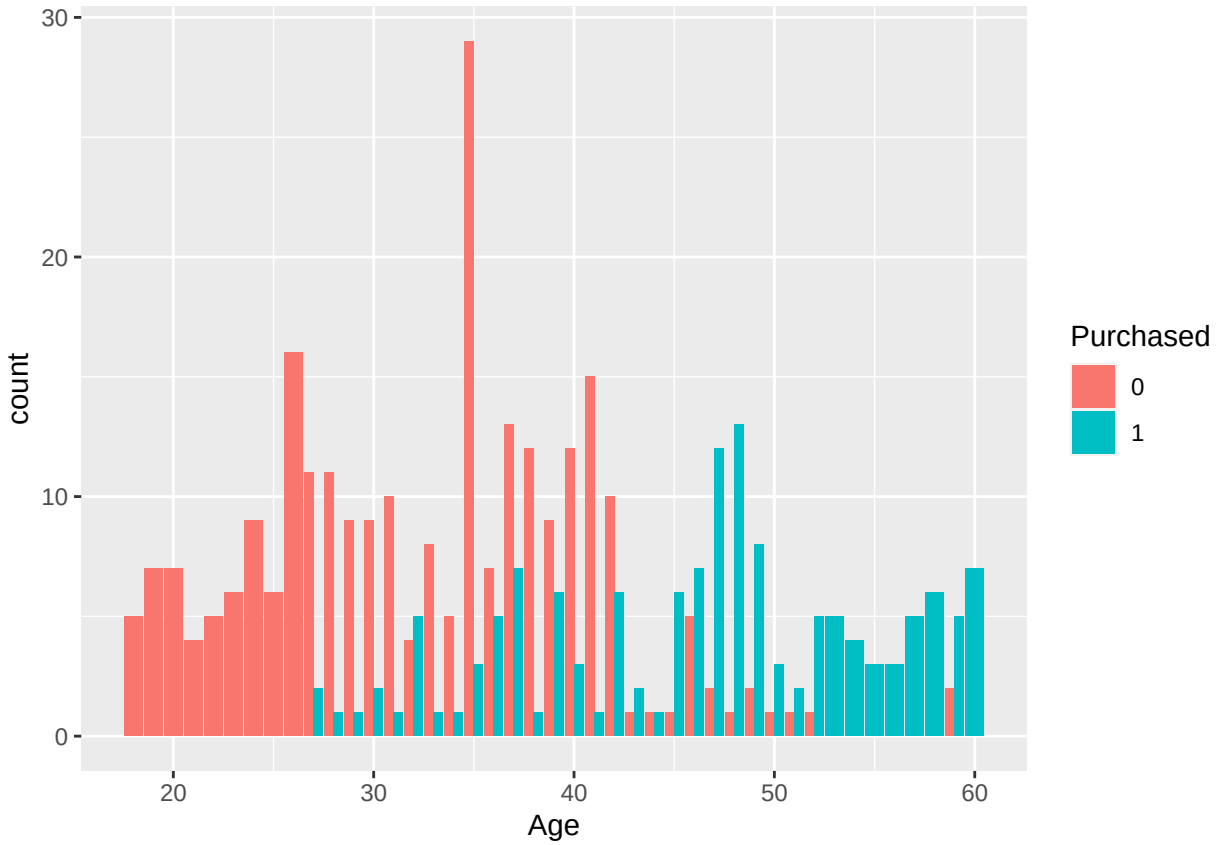
```
data_1 %>% ggplot(aes(EstimatedSalary))+geom_boxplot(aes(fill=as.factor(Male)))
```



Observations:
1. Females earn more compared to males.
2. The median salary for females is 70000 and for male is ~68000.
3. Females have a larger salary range, consider IQR ranging from 42000 to 90000.
4. It can be thought as mostly women tend to make purchases compared to men. So gender specific ads might help in increase in sales.

Q *Which age group make more no. of purchases*

```
data_1 %>% ggplot(aes(Age))+geom_bar(aes(fill=Purchased),position = "dodge")
```



Observations:
1. The problem with more no of people with age 35 is solved, almost 90% of them didn't made the purchase.
2. It's worth noting that early age group(age<40) are mostly exposed to the ads but they didn't made the purchase. So age targeted ads are required to influence young aged people.
3. People after the age 26 tend to make purchase as they are in their late twenties and early thirties. Marriage can be a potential factor.
4. People after the age 55 make purchases regularly. It might be because of entering the old age, so their needs tend to rise.

Q *Does gender influences purchase irrespective of the salary*

```
data_1 %>% ggplot(aes(EstimatedSalary))+geom_histogram(aes(fill=Purchased),position = "dodge")+facet_wra
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

Observations:

1. Compared to males who have a median salary of 68000, females who have a median salary of 70000 surprisingly didn't made more number of purchases.
2. Women who earn less and those who earn more made more purchases compared to the women in between the range 50000 to 90000.
3. People with salary $<$ 80000 make more purchases and most of them are women.

Q *How salary is distributed for various aged users who made the purchase*

```
data_1 %>% ggplot(aes(EstimatedSalary,Age))+geom_point(aes(color=Purchased))
```

1. A good insight is found which shows that people after the age 45 made purchases and if they earn more than 90000 even at younger age(>25) tend to buy after seeing the ads. *So the ads need to be age specific(age<45) and salary specific(salary<90000) to increase more no. of purchases.*

Checking of collinearity

```
#we have only two numeric columns
cor(data_1$Age,data_1$EstimatedSalary)
```

```
## [1] 0.155238
```

Observations:
Weak correlation so *linear kernel* can also be used.

Normalizing EstimatedSalry

```
data_1 <- normalisation(data_1,"EstimatedSalary")
head(data_1)
```

```
##   Male Age EstimatedSalary Purchased
## 1    1  19      0.02962963         0
## 2    1  35      0.03703704         0
```

```
## 3    0  26      0.20740741         0
## 4    0  27      0.31111111         0
## 5    1  19      0.45185185         0
## 6    1  27      0.31851852         0
```

Splitting and Modelling

```
set.seed(123) #to prevent the sampling of test ste differently every time we run the markdown
sample <- sample.split(data_1$Purchased,.7)
train <- data_1[sample,]
test <- data_1[!sample,]
dim(train)
```

```
## [1] 280   4
```

```
dim(test)
```

```
## [1] 120   4
```

Modelling

with default parameters

```
model1 <- svm(Purchased~.,data = train,type="C")
predict1 <- predict(model1,test)
summary(model1)
```

```
##
## Call:
## svm(formula = Purchased ~ ., data = train, type = "C")
##
##
## Parameters:
##    SVM-Type:  C-classification
##  SVM-Kernel:  radial
##        cost:  1
##
## Number of Support Vectors:  85
##
##  ( 43 42 )
##
##
## Number of Classes:  2
##
## Levels:
##  0 1
```

```
ConfusionMatrix(predict1,test$Purchased)
```

```
##       y_pred
## y_true  0  1
##      0 70  7
##      1  6 37
```

```
paste("Accuracy:",Accuracy(predict1,test$Purchased))
```

```
## [1] "Accuracy: 0.891666666666667"
```

```
paste("AUC",AUC(predict1,test$Purchased))
```

```
## [1] "AUC 0.884778012684989"
```

Observations:
Default parameters are giving good accuracy, but lets see if we can improve this accuracy with hyperparameter tuning.

Hyperparameter tuning

```
tune.model <- tune(svm,as.factor(Purchased)~.,data = train,type="C",ranges = list(cost=10^(-2:3),
                                                                                  kernel=c("linear","radial","poly
print(tune.model$best.model)
```

```
##
## Call:
## best.tune(method = svm, train.x = as.factor(Purchased) ~ ., data = train,
##     ranges = list(cost = 10^(-2:3), kernel = c("linear", "radial",
##         "polynomial")), type = "C")
##
##
## Parameters:
##    SVM-Type:  C-classification
##  SVM-Kernel:  radial
##        cost:  1
##
## Number of Support Vectors:  85
```

```
print(tune.model$best.parameters)
```

```
##   cost kernel
## 9    1 radial
```

```
print(tune.model$best.performance)
```

```
## [1] 0.09642857
```

Observations: The best kernel found is rbf, so lets check for various values of c and gamma.

```
tune.model1 <- tune(svm,as.factor(Purchased)~.,data = train,type="C",kernel="radial",ranges = list(cost=
                                                                                    gamma=seq(0,1,.1)
print(tune.model1$best.model)
```

```
##
## Call:
## best.tune(method = svm, train.x = as.factor(Purchased) ~ ., data = train,
##     ranges = list(cost = 10^(-3:5), gamma = seq(0, 1, 0.1)), type = "C",
##     kernel = "radial")
##
##
## Parameters:
##    SVM-Type:  C-classification
##  SVM-Kernel:  radial
##        cost:  1
##
## Number of Support Vectors:  95
```

```
print(tune.model1$best.parameters)
```

```
##    cost gamma
## 76    1   0.8
```

```
print(tune.model1$best.performance)
```

```
## [1] 0.08214286
```

Using best parameters

```
model2 <- svm(Purchased~.,data = train,type="C",cost=tune.model1$best.parameters$cost,kernel="radial",ga
predict2 <- predict(model2,test)
summary(model2)
```

```
##
## Call:
## svm(formula = Purchased ~ ., data = train, type = "C", cost = tune.model1$best.parameters$cost,
##     kernel = "radial", gamma = tune.model1$best.parameters$gamma)
##
##
## Parameters:
##    SVM-Type:  C-classification
##  SVM-Kernel:  radial
##        cost:  1
##
## Number of Support Vectors:  95
##
##  ( 44 51 )
##
##
```

```
## Number of Classes:  2
##
## Levels:
##  0 1
```

```
ConfusionMatrix(predict2,test$Purchased)
```

```
##       y_pred
## y_true  0  1
##      0 71  6
##      1  6 37
```

```
paste("Accuracy ",Accuracy(predict2,test$Purchased))
```

```
## [1] "Accuracy  0.9"
```

```
paste("AUC",AUC(predict2,test$Purchased))
```

```
## [1] "AUC 0.891271519178496"
```

Observations:
1. The model is performing better after hyper parameter tuning.
2. I used set.seed() to keep the sampling same so to match the results in the pdf, but in practice i don't use it.