# A)

In [1]:

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

A1)

In [2]:

```
ds = pd.read_csv('student-mat.csv')
```

In [64]:

```
pd.set_option('display.max_columns',500)
ds.head()
```

Out[64]:

| | school | sex | age | address | famsize | Pstatus | Medu | Fedu | Mjob | Fjob | reason | guardi |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | GP | F | 18 | U | GT3 | A | 4 | 4 | at_home | teacher | course | motl |
| **1** | GP | F | 17 | U | GT3 | T | 1 | 1 | at_home | other | course | fatl |
| **2** | GP | F | 15 | U | LE3 | T | 1 | 1 | at_home | other | other | motl |
| **3** | GP | F | 15 | U | GT3 | T | 4 | 2 | health | services | home | motl |
| **4** | GP | F | 16 | U | GT3 | T | 3 | 3 | other | other | home | fatl |

A2)

In [62]:

```
attributes = len(ds.columns)
attributes
```

Out[62]:

```
33
```

In [19]:

```
dimension = np.shape(ds)
dimension
```

Out[19]:

```
(395, 33)
```

A3)

In [55]:

```
from statistics import mean, mode, median
df1 = ds['Dalc']
average_dalc = round(mean(df1),2)
average_dalc
```

Out[55]:

```
1.48
```

In [28]:

```
df2 = ds['Walc']
average_walc = round(mean(df2),2)
average_walc
```

Out[28]:

```
2.29
```

In [30]:

```
df3 = ds['absences']
average_absence = round(mean(df3),2)
average_absence
```

Out[30]:

```
5.71
```

A4)

In [32]:

```
max_absence = max(df3)
max_absence
```

Out[32]:

```
75
```

In [33]:

```
min_absence = min(df3)
min_absence
```

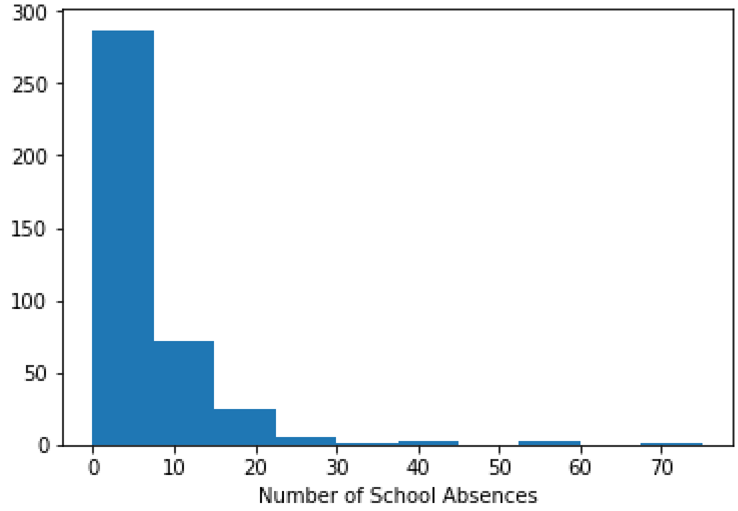
Out[33]:

```
0
```

A5)

In [59]:

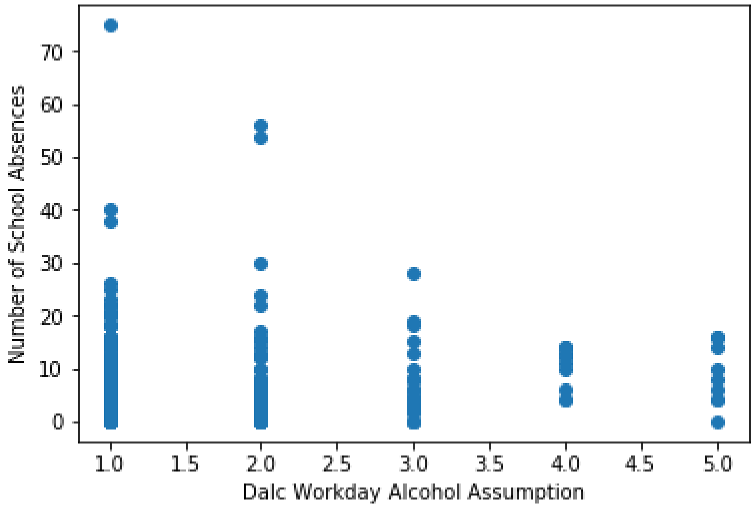
```
plt.hist(df3)
plt.xlabel('Number of School Absences')
plt.show()
```



A6)

In [60]:

```
plt.scatter(df1,df3)
plt.xlabel('Dalc Workday Alcohol Assumption')
plt.ylabel('Number of School Absences')
plt.show()
```



A7)

In [44]:

```
df1.corr(df3)
```

Out[44]:

```
0.11190802615038616
```

In [46]:

```
range_absence = max_absence - min_absence
range_absence
```

Out[46]:

```
75
```

In [53]:

```
mode_absence = mode(df3)
mode_absence
```

Out[53]:

```
0
```

# B)

In [124]:

```
from sklearn.naive_bayes import GaussianNB
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder, OneHotEncoder, StandardScaler
```

STEP 4

In [125]:

```
x = ds.iloc[:,[2,19,26,27]].values
y = ds.iloc[:,20].values
```

STEP 5

In [126]:

```
labelencoder_X = LabelEncoder()
x[:, 1] = labelencoder_X.fit_transform(x[:, 1])
```

STEP 6

In [127]:

```
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size = 0.25, random_s
```

STEP 7

In [133]:

```
sc_x = StandardScaler()
x_train = sc_x.fit_transform(x_train)
x_test = sc_x.fit_transform(x_test)
```

```
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\utils\validation.p
y:475: DataConversionWarning: Data with input dtype object was convert
ed to float64 by StandardScaler.
  warnings.warn(msg, DataConversionWarning)
```

STEP 8

In [134]:

```
model = GaussianNB()
model.fit(x_train,y_train)
```

Out[134]:

```
GaussianNB(priors=None)
```

STEP 9

In [135]:

```
y_predict = model.predict(x_test)
```

STEP 10

In [136]:

```
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test, y_predict)
print(cm)
```

```
[[ 0  7]
 [ 2 90]]
```

STEP 11

In [141]:

```
accuracy = 90/99
print(round(accuracy*100,2), '%')
```

```
90.91 %
```

# C

In [150]:

```
x1 = ds.iloc[:,[30,31]].values
y1 = ds.iloc[:, -1].values
```

In [151]:

```
x1_train, x1_test, y1_train, y1_test = train_test_split(x1, y1, test_size = 0.25, ra
```

In [160]:

```
from sklearn.linear_model import LinearRegression
```

In [163]:

```
model1 = LinearRegression()
model1.fit(x1_train,y1_train)
```

Out[163]:

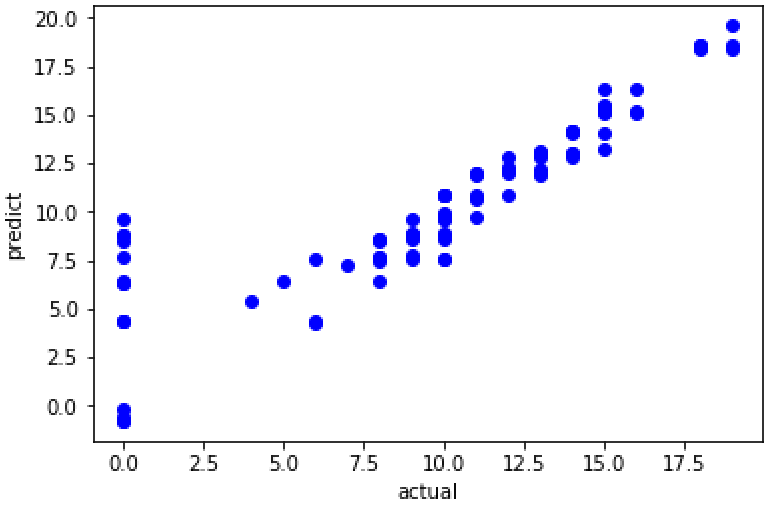
```
LinearRegression(copy_X=True, fit_intercept=True, n_jobs=1, normalize=
False)
```

In [172]:

```
y1_predict = model1.predict(x1_test)
```

In [177]:

```
plt.scatter(y1_test,y1_predict, color = 'blue')
plt.xlabel('actual')
plt.ylabel('predict')
plt.show()
```


predict
actual