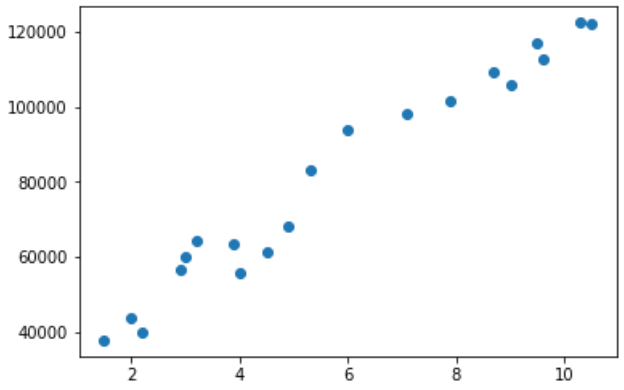
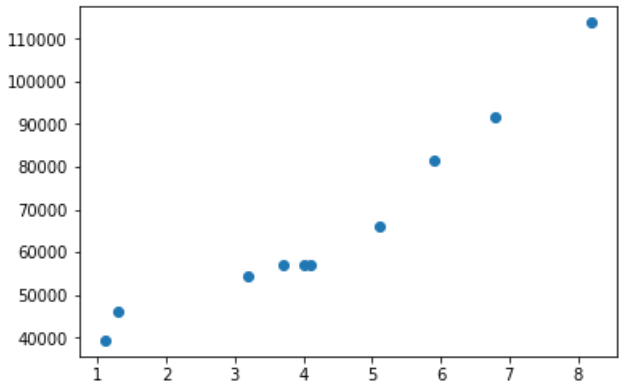
In [148]:
```
#Importing Libraries
import pandas as pd
import numpy as np
import sklearn.model_selection as md_select
import matplotlib.pyplot as plot
```

In [149]:
```
#Creating a dataframe with values and storing them
data = pd.read_csv (r'/home/lavahawk0123/Documents/Tasks-MRM/Tasks-ADMIN/project_1_dataset.csv')
msk = np.random.rand(len(data)) < 0.7
train = data[msk]
test = data[~msk]
x_train = train.iloc[:, 0]
y_train = train.iloc[:, 1]
x_test = test.iloc[:, 0]
y_test = test.iloc[:, 1]
```

In [150]:
```
plot.scatter(x_train,y_train)
plot.show()
```



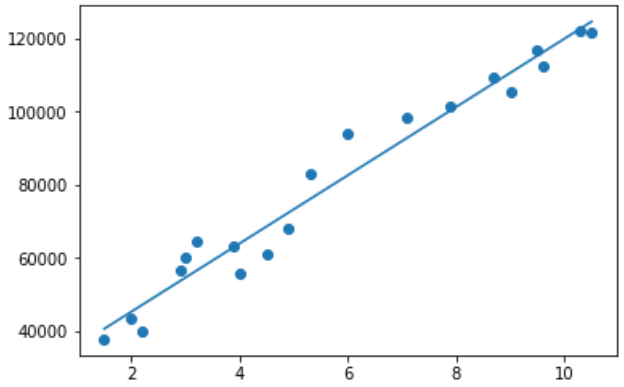
In [151]:
```
plot.scatter(x_test,y_test)
plot.show()
```



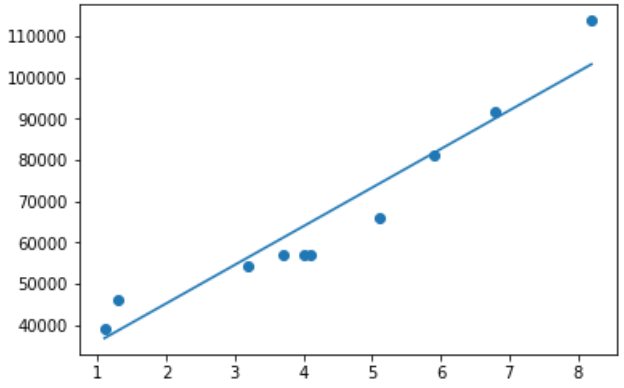
In [152]:
```
# Finding the cost function and performing gradient descent
teta_0 = 0
teta_1 = 0

L = 0.005  # The learning Rate
iteration = 10000  # The number of minimum iterations to perform gradient descent
i=0 # counter variable to plot cost function v/s iterations graph
n = float(len(x_train)) # To find the number of elements in dataset

# Performing Gradient Descent by continuouslt iterating
for i in range(iteration):
    pred=teta_0*x_train + teta_1  # The current predicted value of dependant variable
    Diff_t0=(-1/n)*sum(x_train * (y_train - pred))  # Derivative w.r.t teta_0
    Diff_t1=(-1/n)*sum(y_train - pred)  # Derivative w.r.t teta_1
    teta_0=teta_0-L*Diff_t0  # Update teta_0
    teta_1=teta_1-L*Diff_t1  # Update teta_m
    i+=1

print (teta_0, teta_1)
```

```
(9350.310361934035, 26564.389983037334)
```

In [153]:
```
# Making predictions for train set
pred = teta_0*x_train + teta_1

plot.scatter(x_train, y_train)
plot.plot([min(x_train), max(x_train)], [min(pred), max(pred)])  # regression line
plot.show()
```



In [154]:
```
# Making predictions for train set
pred = teta_0*x_test + teta_1

plot.scatter(x_test, y_test)
plot.plot([min(x_test), max(x_test)], [min(pred), max(pred)])  # regression line
plot.show()
```



In [155]:
```
Estimate =teta_0*15+teta_1
given_value= 167005.32889087
error =abs(((Estimate-given_value)/given_value)*100)
print("predictes salary of a person with 15 years of experience is "+str(Estimate))
print("Percentage Error: "+str(error)+" % which is within limits")
```

```
predictes salary of a person with 15 years of experience is 166819.045412
Percentage Error: 0.111543434009 % which is within limits
```

In [ ]:
```

```