- (●) 3.45 bits
- ( ) 2.45 bits
- ( ) -2.45 bits
- ( ) 0.04 bits

- ( ) It learns effectively with a little amount of data.
- ( ) It learns based on raw data, i.e., input data with different format and size.
- ( ) It learns based on unlabeled data.
- (●) It learns based on the distribution of data.

- (●)

```
import pandas as pd

file = pd.read_csv('sales.csv')

file['Total_Sales'] = file['Sale Price Per Unit'] * file['Quantity']
file['Gross Margin'] = file['Total_Sales'] - (file['Cost Per Unit'])
file['Day'] = pd.to_datetime(file['Date']).dt.day_name()

r = file.groupby('Day')['Gross Margin'].mean().sort_values(ascending =
True).reset_index()
print(r['Day'][0])
```

```
import seaborn as sns

sns.set_style('whitegrid')
%matplotlib inline
from sklearn.datasets import load_boston
boston = load_boston()
print (boston.DESCR)
boston_df = DataFrame(boston.data)
boston_df.columns = boston.feature_names
boston_df['Price']= boston.target
boston_df.head()
sns.lmplot('RM','Price',data=boston_df)
```

```
file = pd.read_csv('housing.csv')
X = file.drop('medv', axis = 1)
y = file['medv']

x_train,x_test,y_train,y_test = train_test_split(X,y,test_size = 0.3,
random_state = 9)

model = LinearRegression()
model.fit(x_train,y_train)

y_pred = model.predict(x_train)
mse = mean_squared_error(y_train,y_test)
```

- (●) Only D
- ( ) Both A and D
- ( ) Both B and D
- ( ) A, C and D

```
data = pd.read_csv('housing.csv')
X= data.drop('medv', axis = 1)
y = data['medv']

model = Ridge(normalize = True, alpha = 0.01)
model.fit(X,y)

print( round(model.score(X,y),2))
```

- a and b
- b and c
- b and d
- a and d