NPTEL-PYTHON FOR DATA SCIENCE

ASSIGNMENT-4-SOLUTION

1. **Answer: B:pandas.get_dummies():**
    - This function will encode dummy values for each categorical variable. Each category will be added as a new column in the dataframe.

2. **Answer:D**: Three key benefits of performing feature selection on your data are:
    - Reduces Overfitting: Less redundant data means fewer error due to noise
    - Improves Accuracy: Removing redundant data improves accuracy
    - Reduces Training Time: Less data means that algorithms train faster

3. **Answer:C: sklearn.model_selection.train_test_split()**
    - The dataset is usually split into training data and test data. The model learns from the training data. We use the test dataset in order to test our model's predictions.
4. **Answer:B**
    - **k** is the **number of nearest neighbours used to predict the class**

5. **Answer:C: sklearn.neighbors.KNeighborsClassifier()**
    - The sklearn library has provided a layer of abstraction on top of Python
    - Therefore, in order to make use of the KNN algorithm, it's sufficient to create an instance of KNeighborsClassifier.

6. **Answer:A**

    The standardized residuals of a model are plotted against the predicted values. This is called a residual plot. When the residuals' variance is not equal(constant) then it is called **Heteroscedasticity**.

7. **Answer:B:**

   R-squared is the percentage of the response variable variation that is explained by a linear model. R-squared is always between 0 and 1 where:

   - 0 indicates that the model explains none of the variability of the response variable is explained by the model.
   - 1 indicates that the model explains all the variability of the response variable is explained by the model.

8. **Answer:A**

   - The number of correct and incorrect predictions are summarized with count values
   - The number of participants that have been wrongly classified as female is 15

9. **Answer:D**

   - The Akaike information criterion (AIC) is an estimator of the relative quality of statistical models for a given set of data
   - Thus, AIC provides a means for model selection

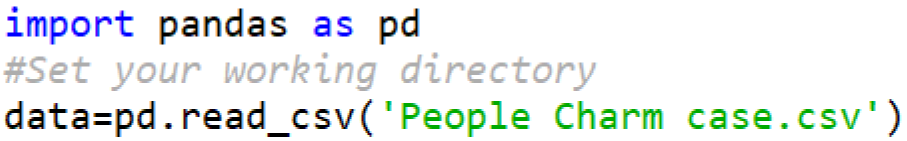
10. **Answer: D**

    - Maximum likelihood will provide values of β0 and β1 which maximize the probability of the occurrence of the dependent variable
    - We use the log-likelihood function to estimate the probability of observing the dependent variable, given the unknown parameters (β0 and β1)

11. **Answer: A**

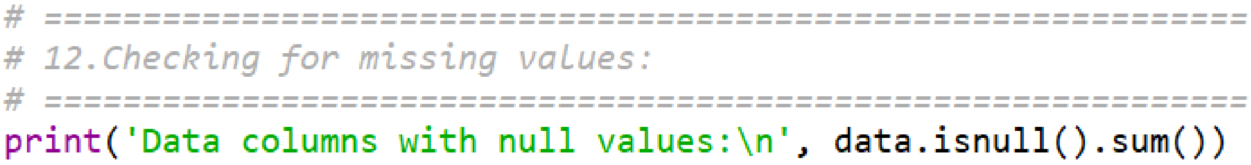
    - The degree of Gini index ranges between 0 and 1, where 0 denotes that all elements belong to one class and 1 denotes that the elements are randomly distributed across various classes

Use the following codes to import your data and then proceed with the questions:



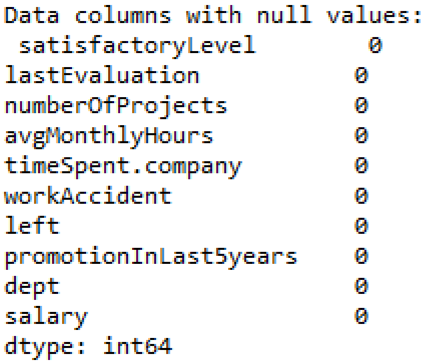
```
import pandas as pd
#Set your working directory
data=pd.read_csv('People Charm case.csv')
```

12. **INPUT**



```
# ===============================================================
# 12.Checking for missing values:
# ===============================================================
print('Data columns with null values:\n', data.isnull().sum())
```

OUTPUT

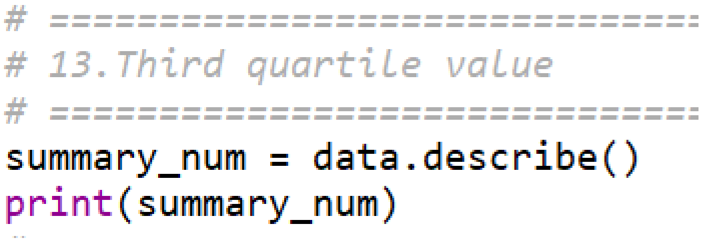


```
Data columns with null values:
 satisfactoryLevel        0
lastEvaluation           0
numberOfProjects         0
avgMonthlyHours          0
timeSpent.company        0
workAccident             0
left                     0
promotionInLast5years    0
dept                     0
salary                   0
dtype: int64
```

INFRENCE: Answer: D

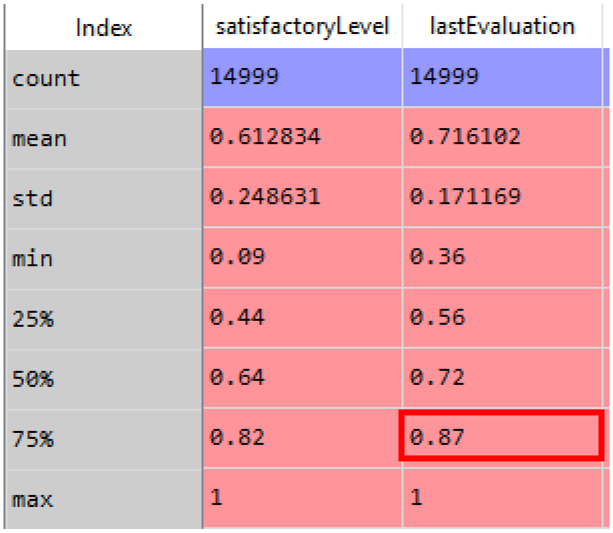
None of the variables in the data has missing values.

13. **INPUT:**



```
# =================================
# 13.Third quartile value
# =================================
summary_num = data.describe()
print(summary_num)
```

OUTPUT:

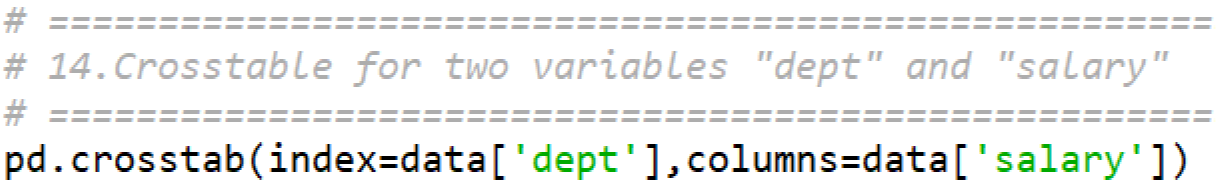


Index	satisfactoryLevel	lastEvaluation
count	14999	14999
mean	0.612834	0.716102
std	0.248631	0.171169
min	0.09	0.36
25%	0.44	0.56
50%	0.64	0.72
75%	0.82	0.87
max	1	1

INFRENCE: Answer: B

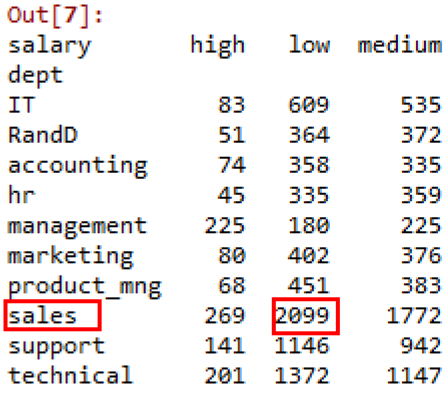
The third quartile for the variable **“lastEvaluation”** is **0.87.**

14. **INPUT:**



```
# ========================================================
# 14.Crosstable for two variables "dept" and "salary"
# ========================================================
pd.crosstab(index=data['dept'],columns=data['salary'])
```

OUTPUT:

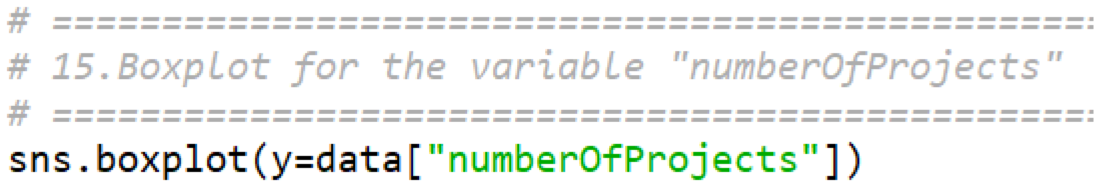


```
Out[7]:
salary        high   low  medium
dept
IT              83   609     535
RandD           51   364     372
accounting      74   358     335
hr              45   335     359
management     225   180     225
marketing       80   402     376
product_mng     68   451     383
sales          269  2099    1772
support        141  1146     942
technical      201  1372    1147
```

INFRENCE: Answer: C

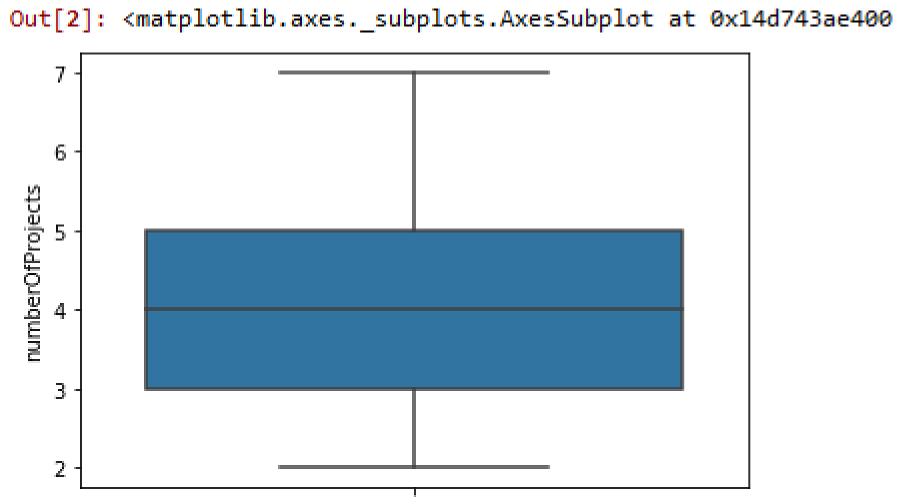
The **“SALES”** department has the highest frequency in low salary category

15. **INPUT:**



```
# ===========================================================
# 15.Boxplot for the variable "numberOfProjects"
# ===========================================================
sns.boxplot(y=data["numberOfProjects"])
```

OUTPUT:

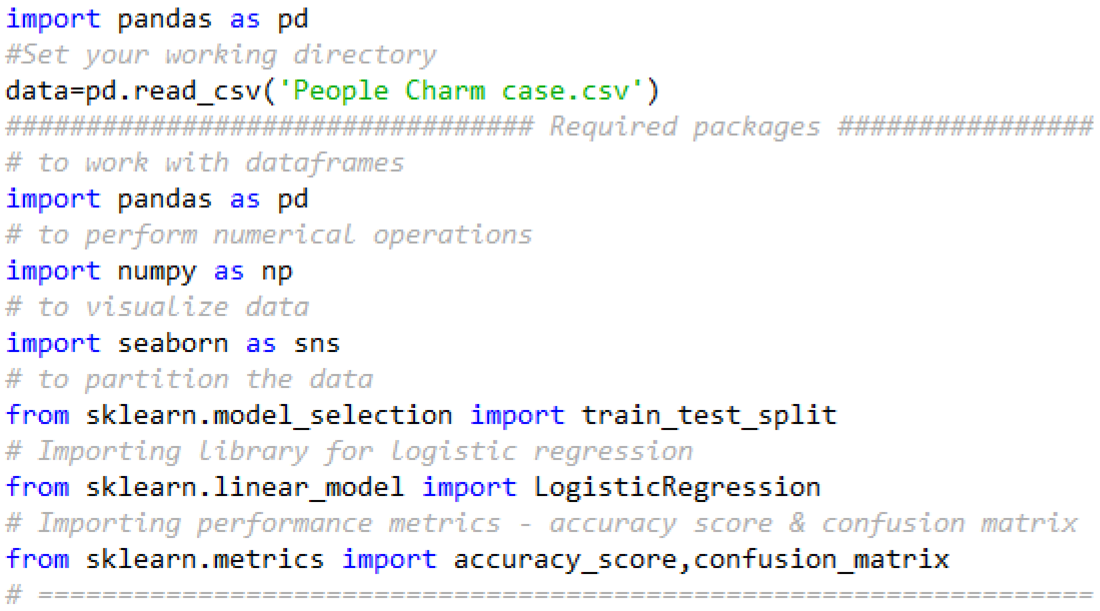


```
Out[2]: <matplotlib.axes._subplots.AxesSubplot at 0x14d743ae400
```

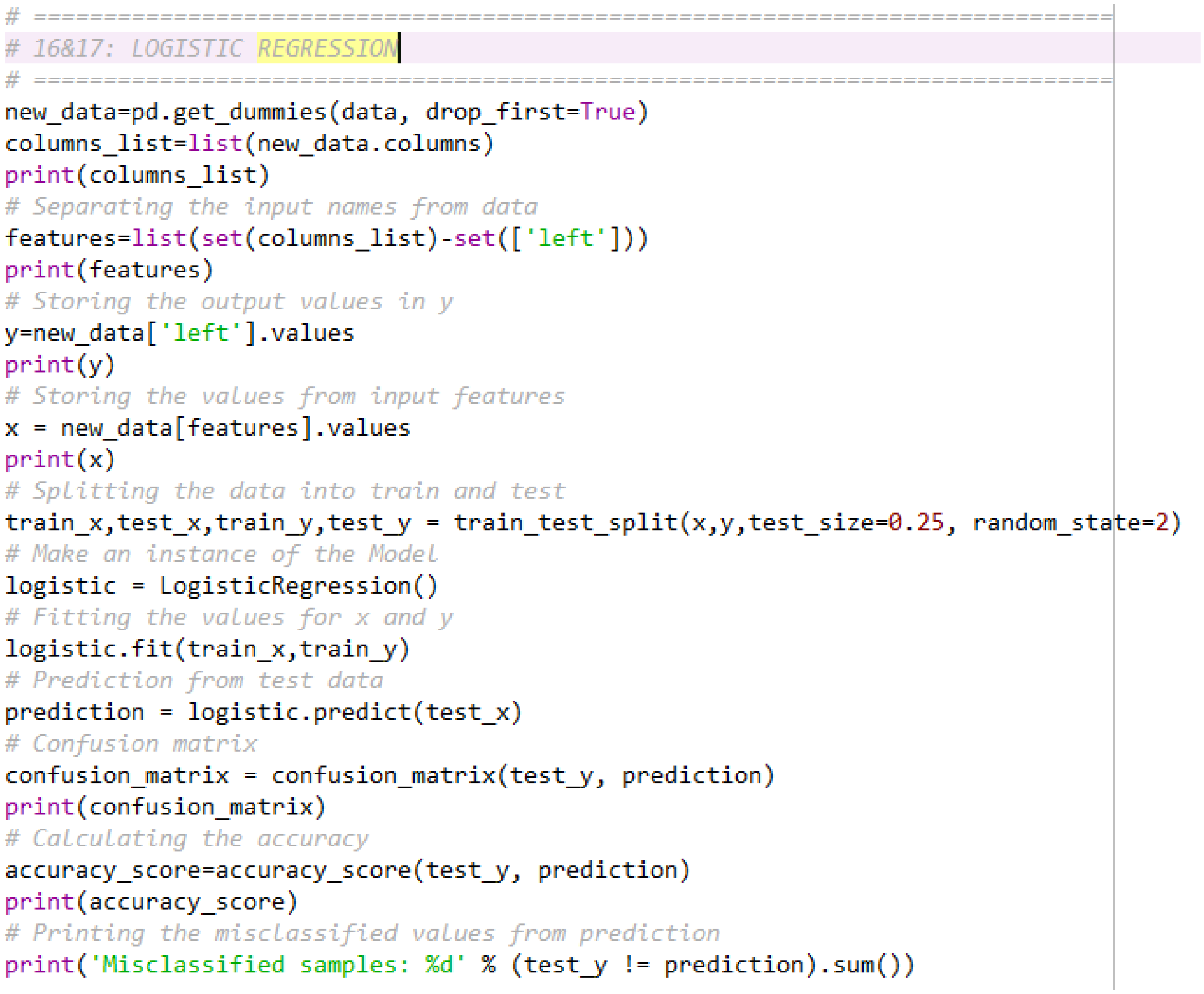
INFRENCE: Answer: B

From the above plot we can see that the median value for the **"numberOfProjects"** where the employees have worked on is **"4".**

16. & 17: INPUT:

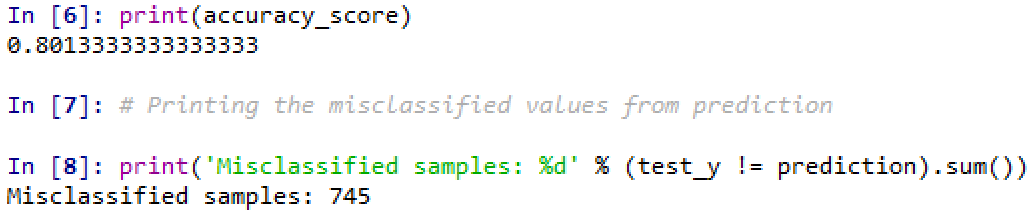


```
import pandas as pd
#Set your working directory
data=pd.read_csv('People Charm case.csv')
################################## Required packages ################
# to work with dataframes
import pandas as pd
# to perform numerical operations
import numpy as np
# to visualize data
import seaborn as sns
# to partition the data
from sklearn.model_selection import train_test_split
# Importing library for logistic regression
from sklearn.linear_model import LogisticRegression
# Importing performance metrics - accuracy score & confusion matrix
from sklearn.metrics import accuracy_score,confusion_matrix
# ===================================================================
```



```
# ==============================================================================
# 16&17: LOGISTIC REGRESSION
# ==============================================================================
new_data=pd.get_dummies(data, drop_first=True)
columns_list=list(new_data.columns)
print(columns_list)
# Separating the input names from data
features=list(set(columns_list)-set(['left']))
print(features)
# Storing the output values in y
y=new_data['left'].values
print(y)
# Storing the values from input features
x = new_data[features].values
print(x)
# Splitting the data into train and test
train_x,test_x,train_y,test_y = train_test_split(x,y,test_size=0.25, random_state=2)
# Make an instance of the Model
logistic = LogisticRegression()
# Fitting the values for x and y
logistic.fit(train_x,train_y)
# Prediction from test data
prediction = logistic.predict(test_x)
# Confusion matrix
confusion_matrix = confusion_matrix(test_y, prediction)
print(confusion_matrix)
# Calculating the accuracy
accuracy_score=accuracy_score(test_y, prediction)
print(accuracy_score)
# Printing the misclassified values from prediction
print('Misclassified samples: %d' % (test_y != prediction).sum())
```

OUTPUT:

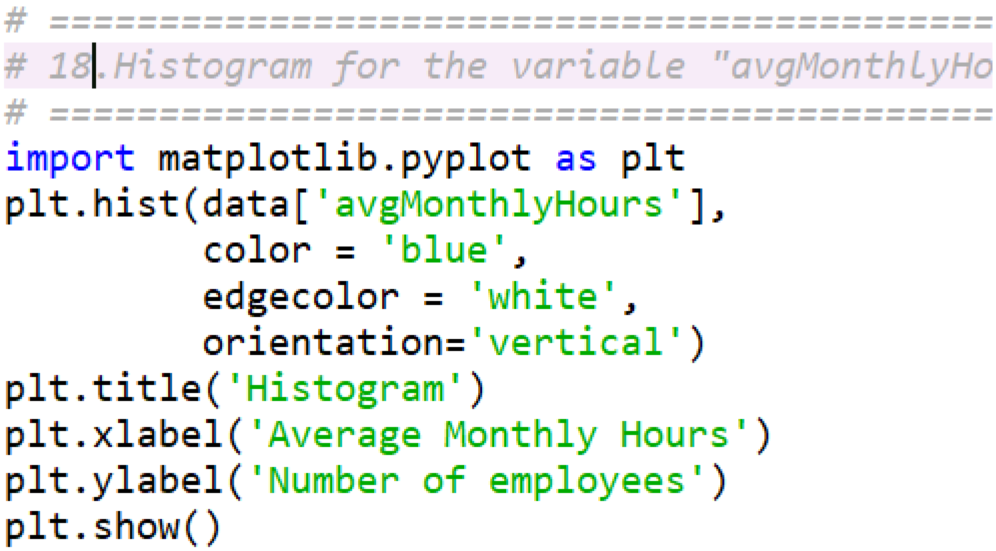


```
In [6]: print(accuracy_score)
0.8013333333333333

In [7]: # Printing the misclassified values from prediction

In [8]: print('Misclassified samples: %d' % (test_y != prediction).sum())
Misclassified samples: 745
```

INFRENCE: Answer for Q:16: A and Answer for Q:17: D

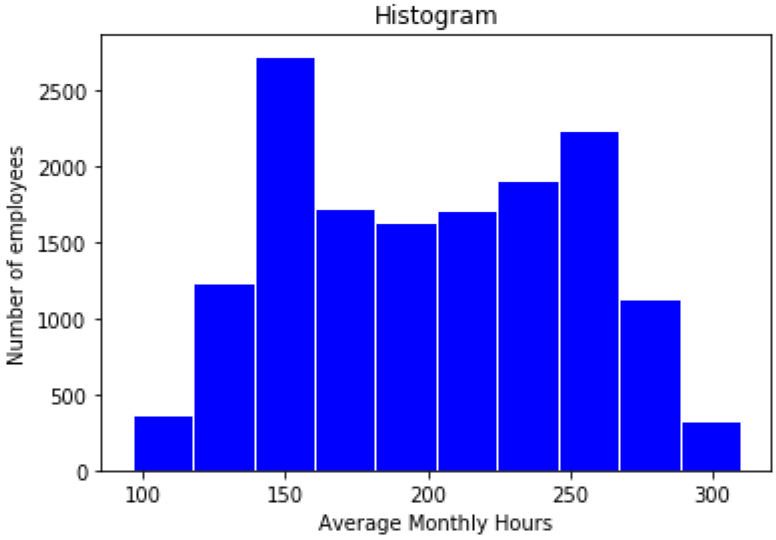
The Accuracy of our model is **"80%"** and the number of Misclassified samples are **"745".**

18. **INPUT:**



```
# ============================================
# 18.Histogram for the variable "avgMonthlyHo
# ============================================
import matplotlib.pyplot as plt
plt.hist(data['avgMonthlyHours'],
         color = 'blue',
         edgecolor = 'white',
         orientation='vertical')
plt.title('Histogram')
plt.xlabel('Average Monthly Hours')
plt.ylabel('Number of employees')
plt.show()
```

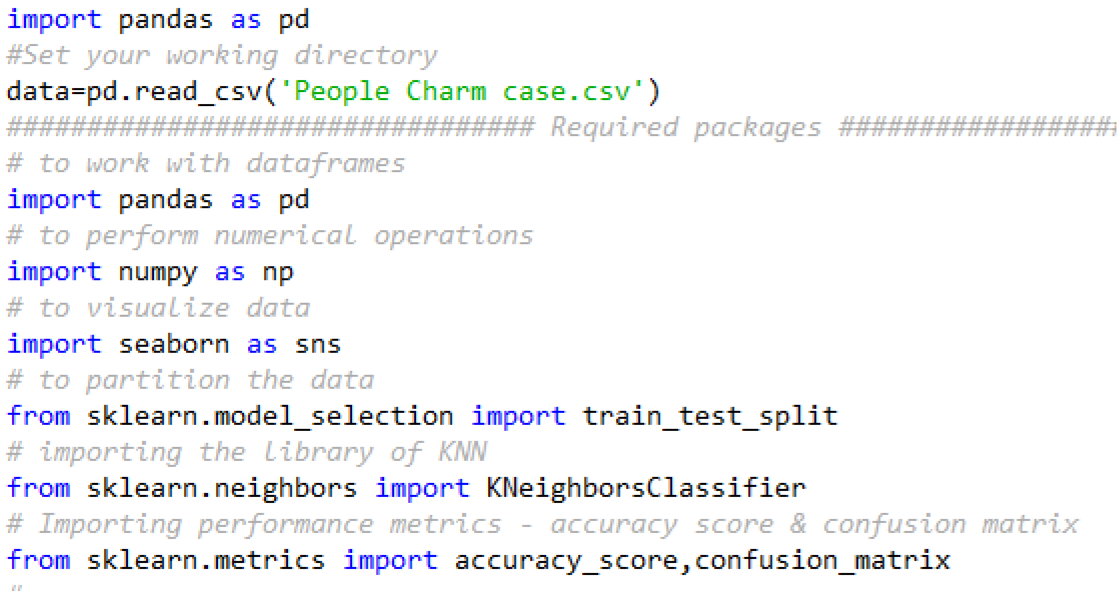
OUTPUT:



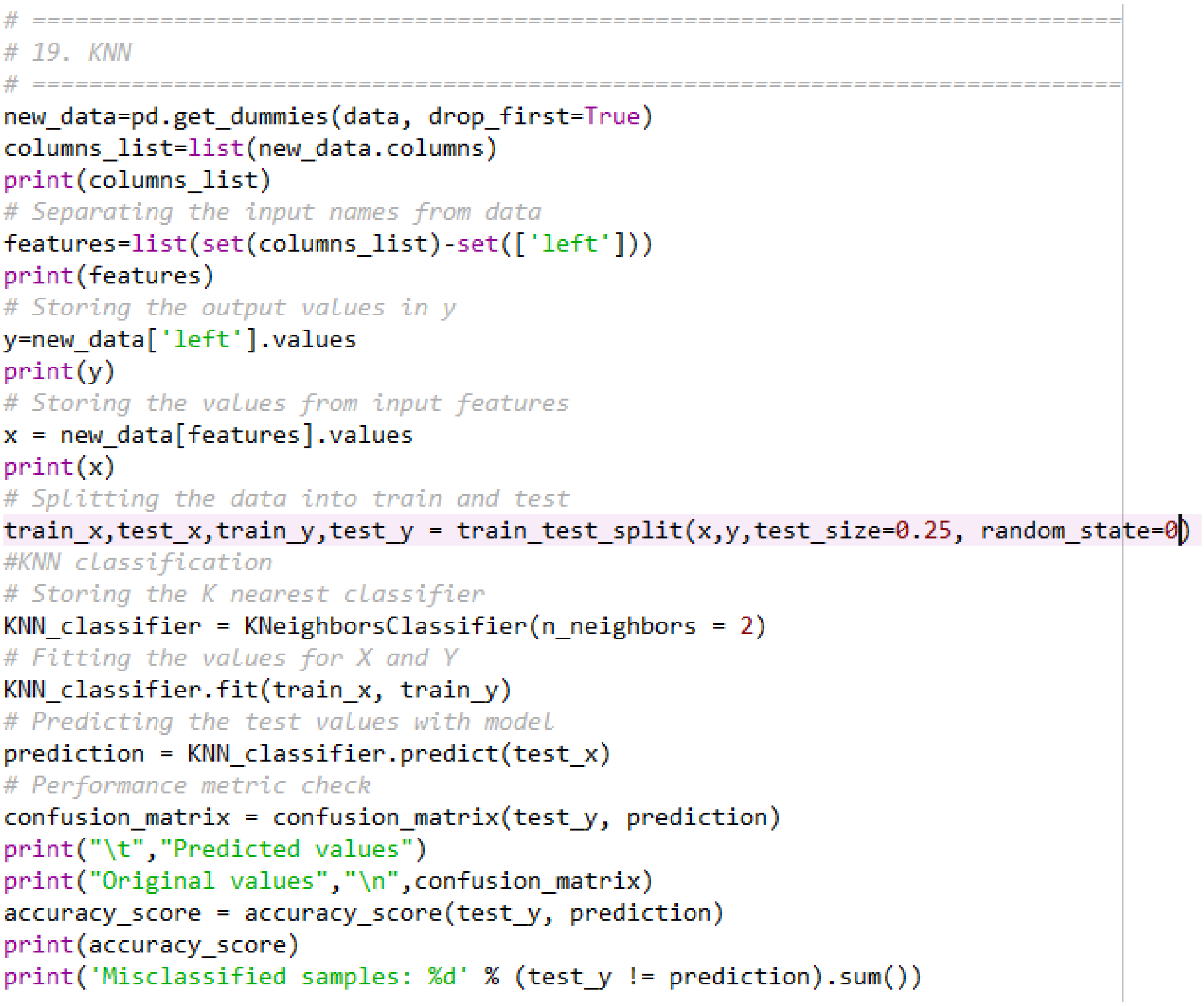
INFRENCE: Answer: C

From the plot we can see that the range in which the number of employees worked for 150 hours per month is **Above 2500.**

19. **INPUT:**

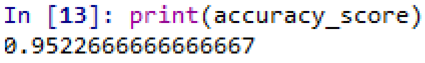


```
import pandas as pd
#Set your working directory
data=pd.read_csv('People Charm case.csv')
################################# Required packages #################
# to work with dataframes
import pandas as pd
# to perform numerical operations
import numpy as np
# to visualize data
import seaborn as sns
# to partition the data
from sklearn.model_selection import train_test_split
# importing the library of KNN
from sklearn.neighbors import KNeighborsClassifier
# Importing performance metrics - accuracy score & confusion matrix
from sklearn.metrics import accuracy_score,confusion_matrix
```



```
# =============================================================================
# 19. KNN
# =============================================================================
new_data=pd.get_dummies(data, drop_first=True)
columns_list=list(new_data.columns)
print(columns_list)
# Separating the input names from data
features=list(set(columns_list)-set(['left']))
print(features)
# Storing the output values in y
y=new_data['left'].values
print(y)
# Storing the values from input features
x = new_data[features].values
print(x)
# Splitting the data into train and test
train_x,test_x,train_y,test_y = train_test_split(x,y,test_size=0.25, random_state=0)
#KNN classification
# Storing the K nearest classifier
KNN_classifier = KNeighborsClassifier(n_neighbors = 2)
# Fitting the values for X and Y
KNN_classifier.fit(train_x, train_y)
# Predicting the test values with model
prediction = KNN_classifier.predict(test_x)
# Performance metric check
confusion_matrix = confusion_matrix(test_y, prediction)
print("\t","Predicted values")
print("Original values","\n",confusion_matrix)
accuracy_score = accuracy_score(test_y, prediction)
print(accuracy_score)
print('Misclassified samples: %d' % (test_y != prediction).sum())
```

OUTPUT:



```
In [13]: print(accuracy_score)
0.9522666666666667
```

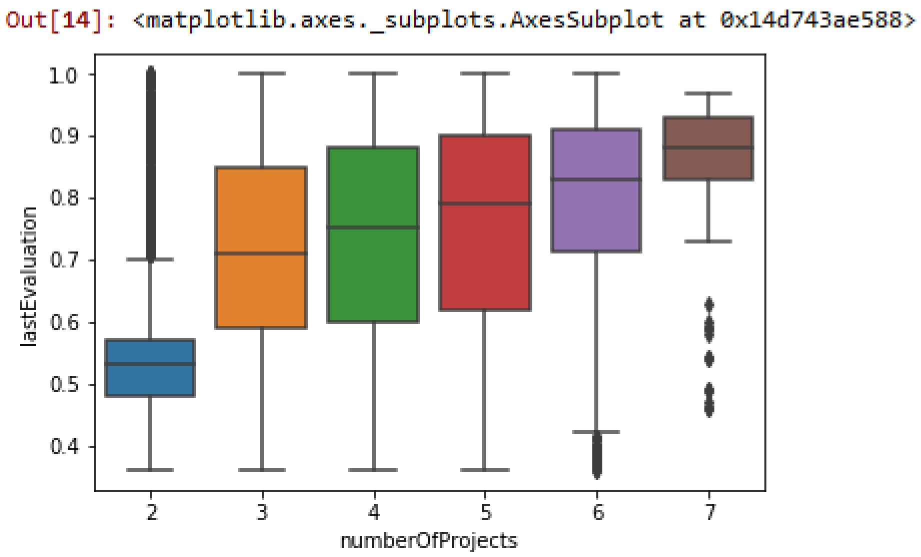
INFRENCE: Answer: A

The accuracy score of the predicted model is **95%.**

20. **INPUT:**

```
# =============================================================================
# 20.Boxplot for “lastEvaluation” and “numberOfProjects”
# =============================================================================
sns.boxplot(x=data["numberOfProjects"],  y = data["lastEvaluation"],
                data=data)
```

OUTPUT:



```
Out[14]: <matplotlib.axes._subplots.AxesSubplot at 0x14d743ae588>
```

INFRENCE: Answer: C

From the plot we can see that, the people who have worked in two projects performance level is low not high.