

```
# Mid-Sem Exam
# Machine Learning Lab
# .....
# Harsh Chaturvedi
# Section: ML
# Roll No: 2014669
# .....

#Code:
# Imports
import numpy as np
import pandas as pd

import matplotlib.pyplot as plt

from sklearn.linear_model import LogisticRegression

from sklearn.model_selection import train_test_split

# Importing Dataset
dataset = pd.read_csv('Adult Income.csv')
dataset = dataset.dropna()

# Converting strings into numbers
incomeGroup = {'<=50K':0, '>50K': 1}
dataset['income'] = [incomeGroup[item] for item in
dataset['income']]

nativeCountry = dict(zip(dataset['native-country'].unique(),
range(dataset['native-country'].unique().shape[0]+1)))
dataset['native-country'] = [nativeCountry[item] for item in
dataset['native-country']]

gender = {'Male':0, 'Female': 1}
dataset['gender'] = [gender[item] for item in dataset['gender']]

race = dict(zip(dataset['race'].unique(),
range(dataset['race'].unique().shape[0]+1)))
dataset['race'] = [race[item] for item in dataset['race']]
```

```

relationship = dict(zip(dataset['relationship'].unique(),
range(dataset['relationship'].unique().shape[0]+1)))
dataset['relationship'] = [relationship[item] for item in
dataset['relationship']]

occupation = dict(zip(dataset['occupation'].unique(),
range(dataset['occupation'].unique().shape[0]+1)))
dataset['occupation'] = [occupation[item] for item in
dataset['occupation']]

maritalstatus = dict(zip(dataset['marital-status'].unique(),
range(dataset['marital-status'].unique().shape[0]+1)))
dataset['marital-status'] = [maritalstatus[item] for item in
dataset['marital-status']]

education = dict(zip(dataset['education'].unique(),
range(dataset['education'].unique().shape[0]+1)))
dataset['education'] = [education[item] for item in
dataset['education']]

workclass = dict(zip(dataset['workclass'].unique(),
range(dataset['workclass'].unique().shape[0]+1)))
dataset['workclass'] = [workclass[item] for item in
dataset['workclass']]

# Seperating Results for Clarity
y = dataset['income']
dataset.drop('income', axis='columns', inplace=True)

# Splitting Data for testing and training
X_train, X_test, y_train, y_test = train_test_split(
    dataset, y,
    stratify=y, random_state=64
)

# Model Selection and Training

```

```
lrModel = LogisticRegression(max_iter=48000,  
C=0.0001).fit(X_train, y_train)  
  
# Model Evaluation  
print('Training set score: ', lrModel.score(X_train, y_train))  
print('Test set score: ', lrModel.score(X_test, y_test))  
  
# .....  
# Result:  
# Training set score: 0.7963746553465644  
# Test set score: 0.8034559004176562  
# .....
```