**Machine Learning and Data Analysis**

**Lab 2: Introduction to NumPy**

**Lab 2.1: NumPy Warmup**

1. Replace the four data structures in the prior lab exercise with a numpy array containing the same data
2. Make required code changes to all the functions a, b, c, d, and e besides renaming the function labels.
3. Use numpy's mean function to compute average price of 'metallic' colored cars
4. Create a tutorial based on the new data structure and the modified code as a Jupyter notebook (function names changed and a new function added). Make sure in your tutorial you annotate the following:
   1. The NumPy data structure
   2. Each Function definition and each function call
   3. How you computed the average using numpy's statistical function call.

Note: Make sure when you submit your Jupyter notebook all code cells are executed and relevant outputs are available. Submit both ipynb and html outputs of the notebook.

**Appendix A:**import random

mycars = ("Honda Civic", "Toyota Corolla", "Ford Focus", "Audi A4", "Mercedes Benz","BMW")

mycolors = ["blue", "white", "blue", "black", "metallic", "metallic"]

def a(cars, colors):

return {car: {"color": color, "price": random.randint(10000, 30000)} for car, color in zip(cars, colors)}

car\_info = a(mycars, mycolors)

print(car\_info)

#-----------------------------------------------

def b(car\_info):

print("Car Information:")

for car, details in car\_info.items():

print(f"{car.title()}: Color = {details['color']}, Price = ${details['price']}")

b(car\_info)

def c(colors):

return set(colors)

colorset = c(mycolors)

print("\n Colors:")

for color in colorset:

print(color)

#----------------------------------------------

# Search and display details of cars of a specific color

# Compute the average price

#----------------------------------------------

**Appendix B:**

import numpy as np

def a(cars, colors, prices):

dtype = [('car', 'U20'), ('color', 'U10'), ('price', 'int32')]

data = np.zeros(len(cars), dtype=dtype)

data['car'] = cars

data['color'] = colors

data['price'] = prices

return data

#-----------------------------------------------

def b(car\_data):

for car in car\_data:

print(f"{car['car'].title()}: Color = {car['color']}, Price = ${car['price']}")

#-----------------------------------------------

def c(car\_data):

return np.unique(car\_data['color'])

#-----------------------------------------------

def d(car\_data, color):

return car\_data[car\_data['color'] == color]

#-----------------------------------------------

def e(car\_data, color):

colored\_cars = d(car\_data, color)

if colored\_cars.size == 0:

print(f"No cars found with the color {color}.")

return None

return np.mean(colored\_cars['price'])

#-----------------------------------------------

cars = np.array(["Honda Civic", "Toyota Corolla", "Ford Focus", "Audi A4","Mercedes Benz","BMW"])

colors = np.array(["blue", "white", "blue", "black", "metallic", "metallic"])

prices = np.random.randint(10000, 30000, size=len(cars))

car\_data = a(cars, colors, prices)

b(car\_data)

#-----------------------------------------------

unique\_colors = c(car\_data)

for color in unique\_colors:

print(color.title())

#-----------------------------------------------

color\_choice = "metallic"

average\_price = e(car\_data, color\_choice)

if average\_price is not None:

print(f"The average price of {color\_choice} cars is: ${average\_price:.2f}")