Last Name: First Name: Student ID:

AIDI 1000: AI Algorithms and Mathematics – Assignment - 1

Due Date : February 10, 2023, 11:59 PM

1. Simulate the Central Limit Theorem in any programming language. “The Central Limit Theorem states that the sampling distribution of the sample means approaches a normal distribution as the sample size gets larger — no matter what the shape of the population distribution". (25 points)

2. Construct the binomial distribution for the total number of heads in four flips of a balanced coin. Define the PMF(Probability Mass Function) of the following distribution. (15 points)

3. Suppose that 40% of the voters in a city are in favor of a ban of smoking in public buildings. Suppose 5 voters are to be randomly sampled. Find the probability that (10 points):

   – 2 favor the ban.

   – less than 4 favor the ban.

   – at least 1 favor the ban.

4. Most graduate schools of business require applicants for admission to take the SAT examination. Scores on the SAT are roughly normally distributed with a mean of 530 and a standard deviation of 110. What is the probability of an individual scoring above 500 on the SAT? (15 points)

5. The Edwards’s Theater chain has studied its movie customers to determine how much money they spend on concessions. The study revealed that the spending distribution is approximately normally distributed with a mean of 4.11 dollar and a standard deviation of 1.37 dollar. What percentage of customers will spend less than 3.00 dollar on concessions?(10 points)

6. A data scientist is testing a new model. She choose train and test sets at random from a large population of training data. She randomly choose 8 fold validation to get the accuracy for decision tree model, and choose 5 fold cross validation to get the accuracy for Logistic regression. The data are below: (25 points)

   – Decision Trees: 93,94,89,88,78,89,76,98

   – Logistic Regression: 78,90,89,76,89

      1. Are the two populations paired or independent? Explain your answer.

      2. Graph the data as you see fit. Why did you choose the graph(s) that you did and what does it (do they) tell you?

3. Choose a test appropriate for the hypothesis above, and justify your choice based on your answers to parts (a) and (b). Then perform the test by computing a p-value, and making a reject or not reject decision. Do use python or any programming language for this, and show your work. Finally, state your conclusion in the context of the problem.

Ans 1.

```python
import numpy as np
import matplotlib.pyplot as plt
from scipy.stats import norm
```

set the size of the population and the sample that we want to generate

```python
population_size = 500
sample_size = 20
```

The scale parameter is set to 1 for half of the population and 2 for the other half, so the population has a non-normal distribution then concatenates both the sample.

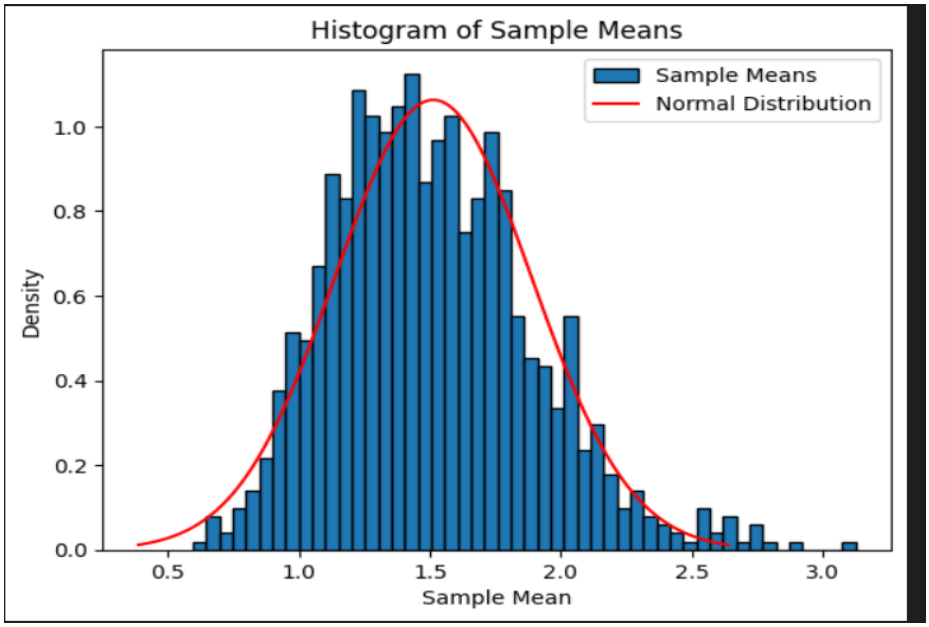
```python
# Generate a population with a non-normal distribution
population = np.concatenate((np.random.exponential(scale=1, size=population_size//2), np.random.exponential(scale=2, size=population_size//2)))
```

generates 500 samples from the population, each with a size of 20

```python
# Generate multiple samples from the population and calculate the mean of each sample
sample_means = [np.mean(np.random.choice(population, sample_size)) for i in range(1000)]
```

To create overlay a normal distribution curve we need calculated np.mean and np.std then calculate np.linspace to generate 100 evenly spaced values between the mean minus 3 standard deviations and the mean plus 3 standard deviations. The y variable is created using norm.pdf from scipy.stats to generate the values of the normal distribution with the mean and standard deviation.

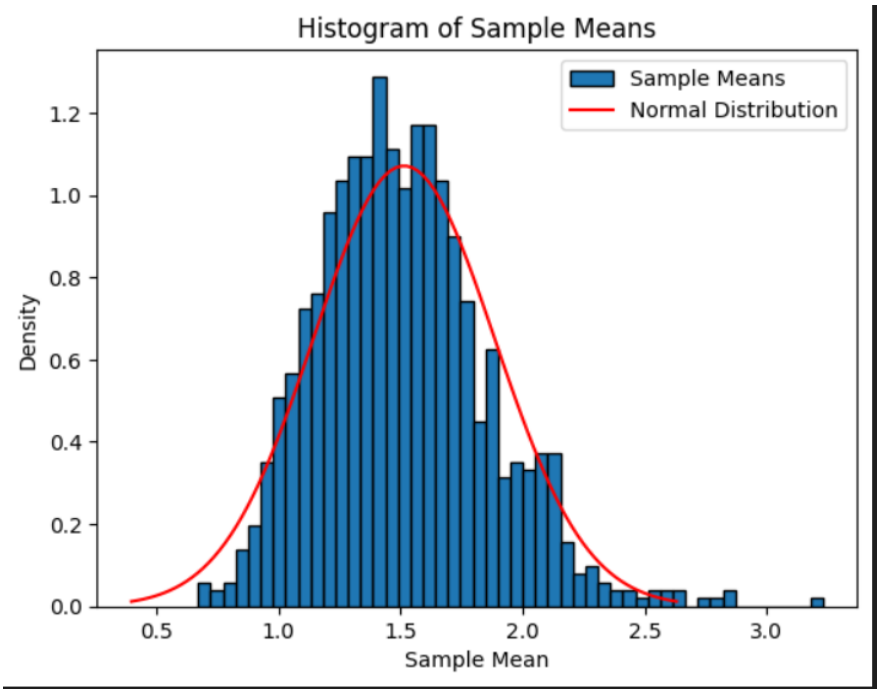
```python
# Plot the histogram of sample means and overlay a normal distribution curve
mu = np.mean(population)
sigma = np.std(population) / np.sqrt(sample_size)
x = np.linspace(mu - 3 * sigma, mu + 3 * sigma, 100)
y = norm.pdf(x, mu, sigma)
plt.hist(sample_means, bins=50, edgecolor='black', density=True, label='Sample Means')
plt.plot(x, y, 'r', label='Normal Distribution')
plt.title("Histogram of Sample Means")
plt.xlabel("Sample Mean")
plt.ylabel("Density")
plt.legend()
plt.show()
```



generates 500 samples from the population, each with a size of 20.

```
population_size = 1000
sample_size = 20
```

✓ 0.0s Python



As you can see, as the sample size increases, the distribution of the sample means approaches a normal distribution, as predicted by CLT.

Ans 2.

Ans-2) 4-times balanced coin flips getting head :

balanced coins getting head

$p(H) = 1/2 \quad q(T) = 1/2$

4-times $(N=4)$

$X=0 \quad X=4$

$X=1$

$X=2$

$X=3$

} 4-times coin flip to get Hing head

$\therefore P(X=0) = {}^4C_0 \left(\frac{1}{2}\right)^0 \left(\frac{1}{2}\right)^4$

$= (1)(1)\left(\frac{1}{16}\right) = \frac{1}{16}$ //

$P(X=1) = {}^4C_1 \left(\frac{1}{2}\right)^1 \left(\frac{1}{2}\right)^3 = \frac{4!}{1!(4-1)!} \left(\frac{1}{2}\right)\left(\frac{1}{2}\right)^3$

$= 4\left(\frac{1}{2}\right)\left(\frac{1}{8}\right) = \frac{1}{4}$ //

$P(X=2) = {}^4C_2 \left(\frac{1}{2}\right)^2 \left(\frac{1}{2}\right)^2$

$= \frac{4!}{2!(4-2)!} \left(\frac{1}{4}\right)\left(\frac{1}{4}\right)$

$= 6 \times \frac{1}{16} = \frac{3}{8}$ //

$$P(X=3) = {}^4C_3 \left(\frac{1}{2}\right)^3 \left(\frac{1}{2}\right)^1$$

$$= \frac{4!}{3!(4-1)!} \left(\frac{1}{8}\right)\left(\frac{1}{2}\right)$$

$$= \frac{1}{4}$$

$$P(X=4) = {}^4C_4 \left(\frac{1}{2}\right)^4 \left(\frac{1}{2}\right)^0$$

$$= \frac{4!}{4!(4-4)!} \left(\frac{1}{16}\right)$$

$$= \frac{1}{16} //$$

Probability distribution of getting head.

X	0	1	2	3	4
P(X)	1/16	1/4	3/8	1/4	1/16

Ans 3.

Ans-3

Voter

Given 40% favor in ban $\Rightarrow p = 0.40$

1. When 2 favor the ban from sample of 5

$$\Rightarrow P(X=2) = {}^5C_2 \cdot p^2 \cdot (1-q)^{5-2}$$

$$= \frac{5!}{2!(5-2)!} \cdot (0.4)^2 (0.6)^3$$

$$= 10 \cdot (0.16) \cdot (0.216) = 0.345$$

Voter

$\therefore$ 34.56 % favor ~~the~~ in ban

2. Less than 4 favor the ban

$$\Rightarrow P(X<4) = P(0) + P(X=1) + P(X=2) + P(X=3)$$

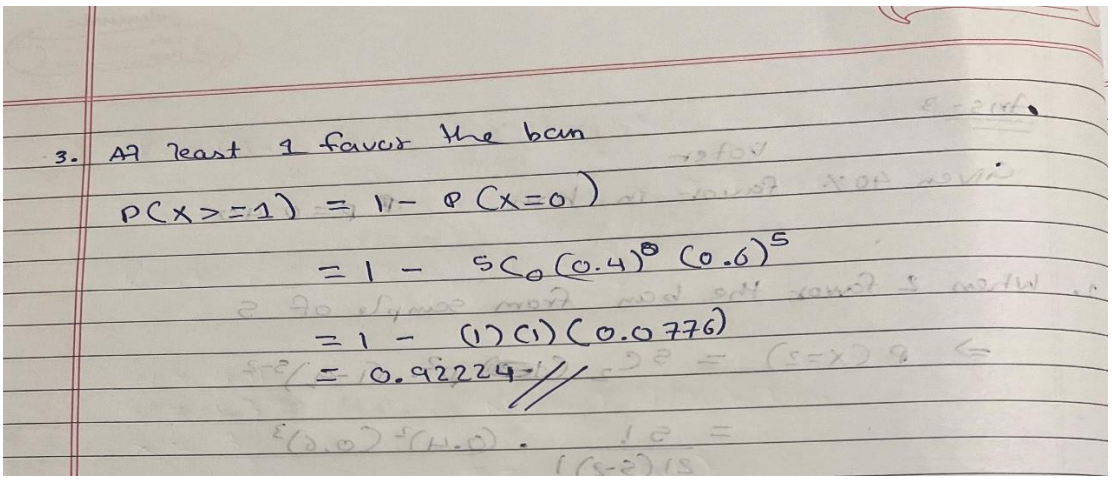
$$= {}^5C_0 (0.4)^0 (0.6)^5 + {}^5C_1 (0.4)^1 (0.6)^4 + {}^5C_2 (0.4)^2 (0.6)^3 + {}^5C_3 (0.4)^3 (0.6)^2$$

$$= (1)(1)(0.0776) + 5(0.4)(0.1296) + 0.345 + 10(0.027)$$ (0.3)

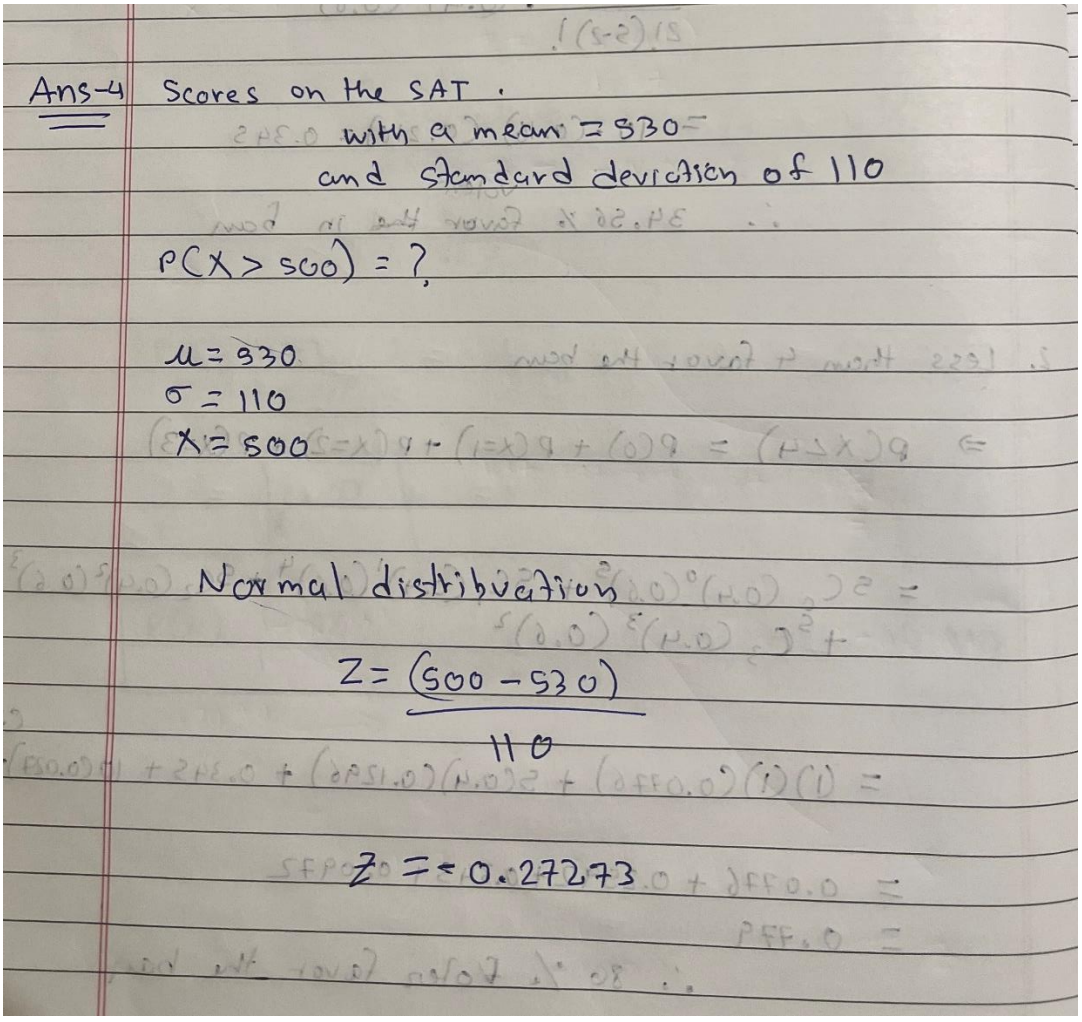
$$= 0.0776 + 0.2592 + 0.345 + 0.0972$$

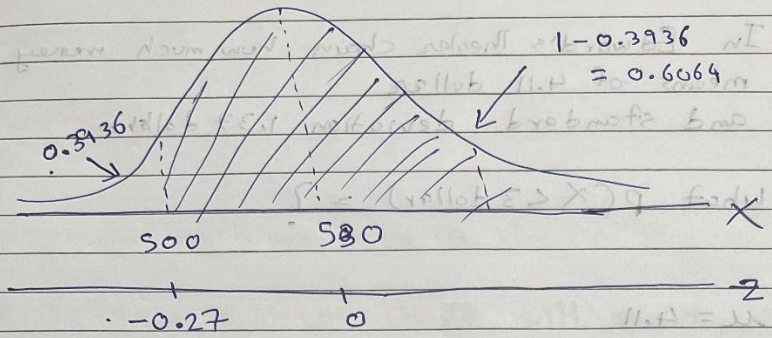
$$= 0.779$$

$\therefore$ 80 % Voter favor the ban


3. At least 1 favor the ban
P(X>=1) = 1 - P(X=0)
= 1 - 5C0 (0.4)^0 (0.6)^5
= 1 - (1)(1)(0.0776)
= 0.92224

Ans 4.


Ans-4 Scores on the SAT.
with a mean = 530
and standard deviation of 110
P(X > 500) = ?
μ = 530
σ = 110
X = 500
Normal distribution
Z = (500 - 530) / 110
Z = -0.27273



To find the probability of z score in the STD table

$\Rightarrow 0.3936$

To get score above 500 ..

$P(X > 500) = P(Z > -0.272)$

$= 1 - 0.3936$

$= 0.6064$

∴ probability scoring above 500 on the SAT 60.64 %.

Ans 5.

Ans-5) In Edward's Theater chain how much money spend means of 4.11 dollar and standard deviation 1.37 dollar

What $P(X < 3 \text{ dollar}) = ?$

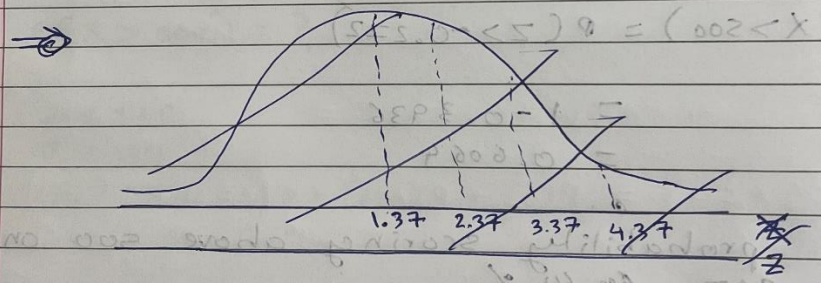
$\mu = 4.11$

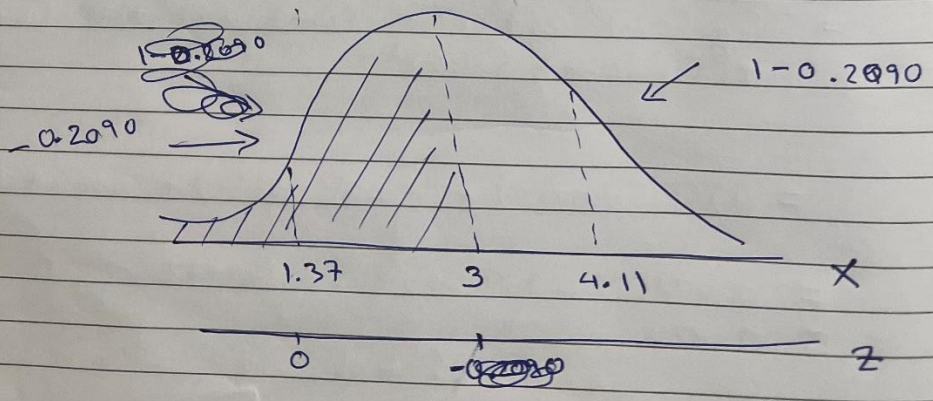
$\sigma = 1.37$ ∴ Normal distribution

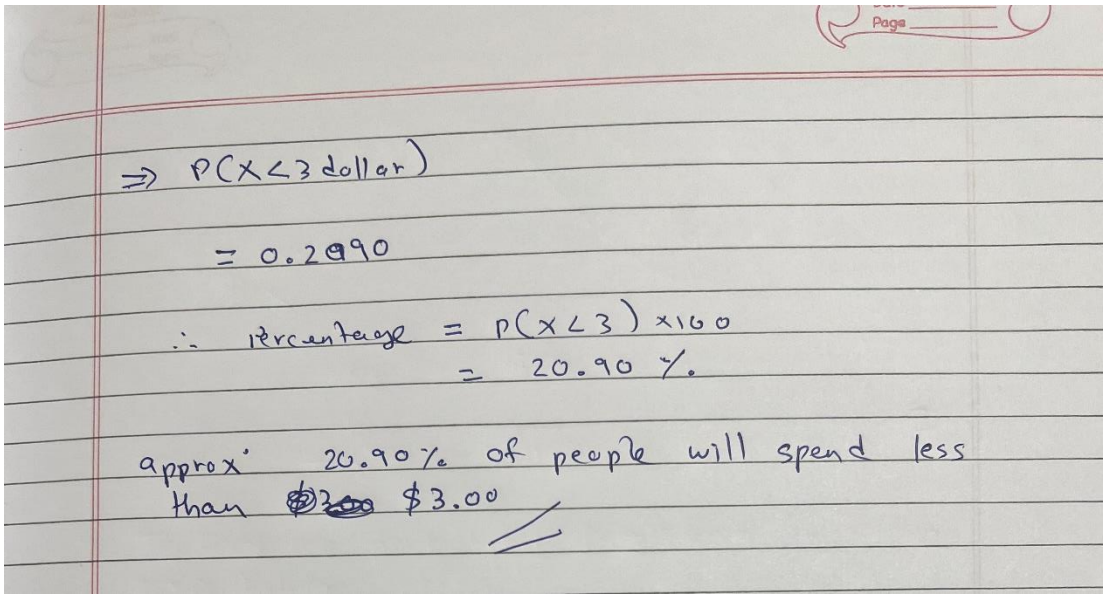
$x = 3$

$$z = \frac{(3 - 4.11)}{1.37}$$

$$z = -0.8102$$







Ans 6.

```python
import numpy as np

# Decision Tree  accuracy
dt_accuracy = np.array([93, 94, 89, 88, 78, 89, 76, 98])
dt_mean_accuracy = np.mean(dt_accuracy)

# Logistic Regression  accuracy
lr_accuracy = np.array([78, 90, 89, 76, 89])
lr_mean_accuracy = np.mean(lr_accuracy)

# Print the mean accuracy for each model
print("Mean Accuracy for Decision Tree:", dt_mean_accuracy)
print("Mean Accuracy for Logistic Regression:", lr_mean_accuracy)
```

✓ 0.0s Python

Mean Accuracy for Decision Tree: 88.125

Mean Accuracy for Logistic Regression: 84.4

1. Are the two populations paired or independent? Explain your answer.

   The two populations are independent, as they are the accuracy results of two different models obtained from separate sets of data, not related to each other.

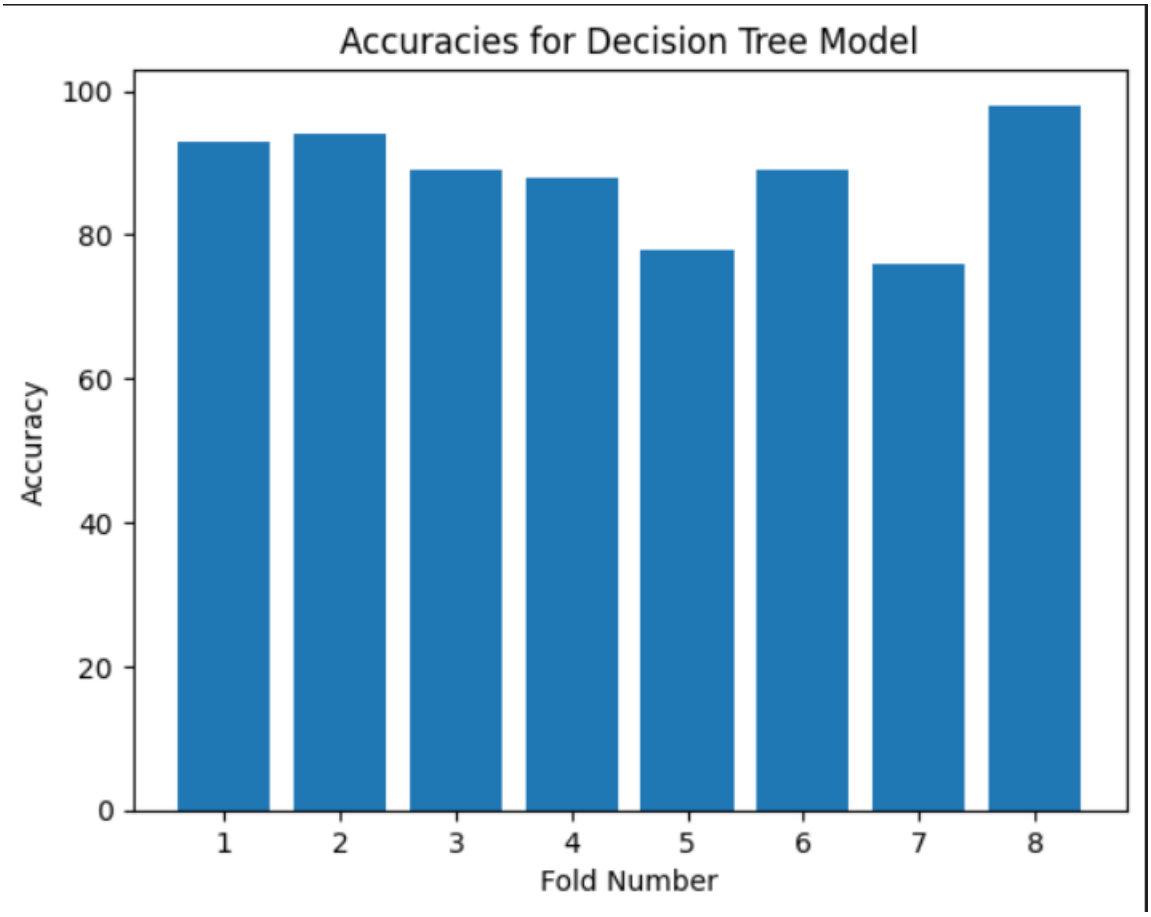
2. Graph the data as you see fit. Why did you choose the graph(s) that you did and what does it (do they) tell you?

```python
import matplotlib.pyplot as plt

accuracy_dTree = [93, 94, 89, 88, 78, 89, 76, 98]

plt.bar(range(1, 9), accuracy_dTree)
plt.title("Accuracies for Decision Tree Model")
plt.xlabel("Fold Number")
plt.ylabel("Accuracy")
plt.show()
```

✓ 0.1s Python

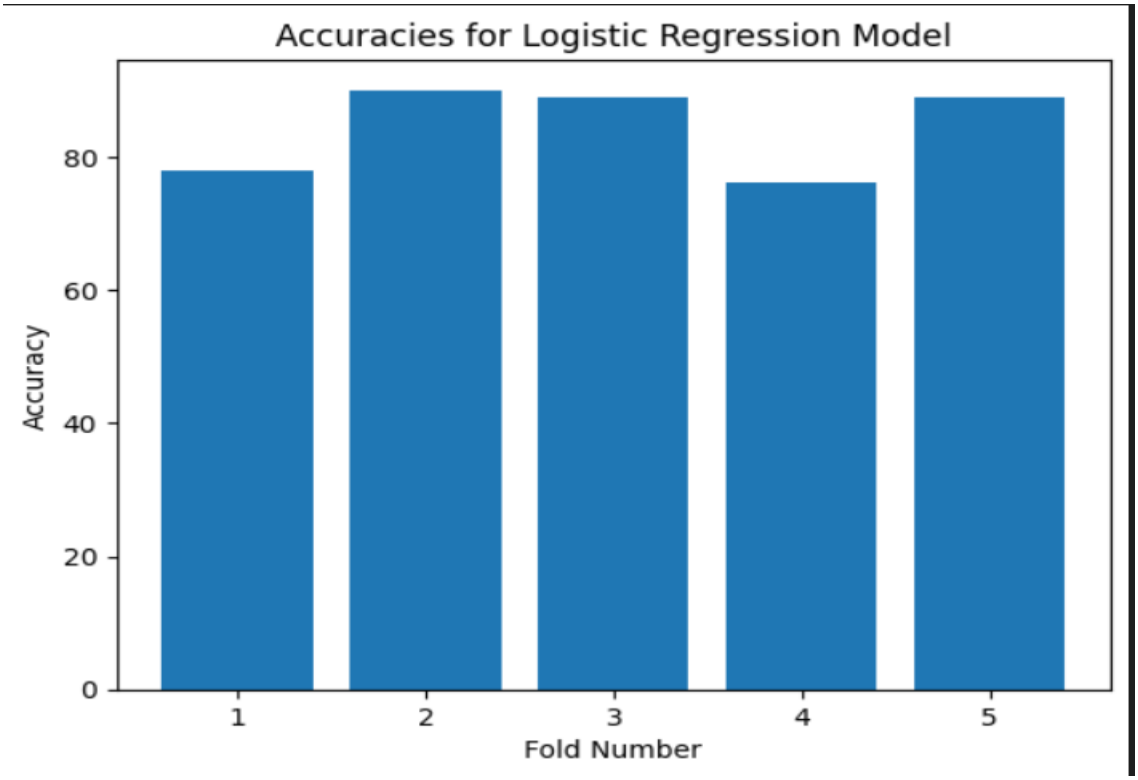


```python
import matplotlib.pyplot as plt

accuracy_linearR = [78, 90, 89, 76, 89]

plt.bar(range(1, 6), accuracy_linearR)
plt.title("Accuracies for Logistic Regression Model")
plt.xlabel("Fold Number")
plt.ylabel("Accuracy")
plt.show()
```

✓ 0.0s Python



Easy to understand comparison of the accuracy of the model across different folds. From the bar graph, we can see the accuracy of the model on each fold and get a general idea of how well the model performed across all the folds.

3. Choose a test appropriate for the hypothesis above, and justify your choice based on your answers to parts (a) and (b). Then perform the test by computing a p-value, and making a reject or not reject decision. Do use python or any programming language for this, and show your work. Finally, state your conclusion in the context of the problem.

Before performing this model we have to calculate certain value such as accuracy of Decision Tree and Linear Regression then we have to calculate the mean of this model to compare the accuracy after that we have to find the standard deviation for both the model.

Calculate the degree of freedom for the null hypothesis using the formula len(accuracy_dTree) + len(accuracy_linearR) - 2.

Means length of the accuracy of Decision tree plus length of linear regression minus whole value.

```
import scipy.stats as stats
```
✓ 0.0s Python

```
# calculate means
mean_dTree = np.mean(accuracy_dTree)
mean_linearR = np.mean(accuracy_linearR)
```
✓ 0.0s Python

```
# calculate standard deviation
std_dTree = np.std(accuracy_dTree, ddof=1)
std_linearR = np.std(accuracy_linearR, ddof=1)
```
✓ 0.0s Python

```
# calculate degree the null hypothesis
degree_hy = len(accuracy_dTree) + len(accuracy_linearR) - 2
```
✓ 0.0s Python

The t-statistic measures the difference between the means of the two accuracy scores in standard error units.

```python
# calculate the t-statistic
t = (mean_dTree - mean_linearR) / np.sqrt(std_dTree**2/len(accuracy_dTree) + std_linearR**2/len(accuracy_linearR))
```

✓ 0.0s Python

```python
# calculate the p-value
p = stats.t.sf(np.abs(t), degree_hy) * 2
```

✓ 0.0s Python

```python
print("t-statistic: ", t)
print("p-value: ", p)
```

✓ 0.0s Python

After check, If the p-value is less than a 0.05, then the result can be considered the null hypothesis can be rejected, indicating that there is evidence that the means of the two accuracy scores are different or else we can't reject the null hypothesis.

t-statistic: 0.9163203472079554

p-value: 0.3791497502435084

The p-value is 0.379, which is greater than 0.05 that means we fail to reject the null hypothesis that the mean accuracy of decision tree model and logistic regression model is the same. We cannot conclude that the mean accuracy of the two models is different.