Plotting the bar graph

In [22]:
```
plt.bar(['Logistic Regression','SVM','Random Forest','KNN','Naive Bayes'],acc,co
plt.ylabel('Accuracy Score')
plt.xlabel('Algortihms')
plt.show()

plt.bar(['Logistic Regression','SVM','Random Forest','KNN','Naive Bayes'],roc,co
plt.ylabel('ROC AUC')
plt.xlabel('Algortihms')
plt.show()


#Random forest has highest accuracy 95%+- and ROC_AUC curve 95%+-
#model can be improve more if we take same count of labels
#in our model 30% is diabetic and 70% non diabetic patient as mentioned in the b
```

```
ValueError                                Traceback (most recent call last)
~\AppData\Local\Temp/ipykernel_10160/3124584757.py in <module>
----> 1 plt.bar(['Logistic Regression','SVM','Random Forest','KNN','Naive Baye
s'],acc,color=['salmon','brown','red','yellow','orange'],label='Accuracy')
      2 plt.ylabel('Accuracy Score')
      3 plt.xlabel('Algortihms')
      4 plt.show()
      5

~\anaconda3\lib\site-packages\matplotlib\pyplot.py in bar(x, height, width, bot
tom, align, data, **kwargs)
   2649         x, height, width=0.8, bottom=None, *, align='center',
   2650         data=None, **kwargs):
-> 2651     return gca().bar(
   2652         x, height, width=width, bottom=bottom, align=align,
   2653         **({"data": data} if data is not None else {}), **kwargs)

~\anaconda3\lib\site-packages\matplotlib\__init__.py in inner(ax, data, *args,
```

```
~\anaconda3\lib\site-packages\matplotlib\__init__.py in inner(ax, data, *args,
**kwargs)
   1359     def inner(ax, *args, data=None, **kwargs):
   1360         if data is None:
-> 1361             return func(ax, *map(sanitize_sequence, args), **kwargs)
   1362
   1363         bound = new_sig.bind(ax, *args, **kwargs)

~\anaconda3\lib\site-packages\matplotlib\axes\_axes.py in bar(self, x, height,
width, bottom, align, **kwargs)
   2302                 yerr = self._convert_dx(yerr, y0, y, self.convert_yunit
s)
   2303
-> 2304         x, height, width, y, linewidth, hatch = np.broadcast_arrays(
   2305             # Make args iterable too.
   2306             np.atleast_1d(x), height, width, y, linewidth, hatch)
```

```
<__array_function__ internals> in broadcast_arrays(*args, **kwargs)

~\anaconda3\lib\site-packages\numpy\lib\stride_tricks.py in broadcast_arrays(su
bok, *args)
    536     args = [np.array(_m, copy=False, subok=subok) for _m in args]
    537
--> 538     shape = _broadcast_shape(*args)
    539
    540     if all(array.shape == shape for array in args):

~\anaconda3\lib\site-packages\numpy\lib\stride_tricks.py in _broadcast_shape(*a
rgs)
    418     # use the old-iterator because np.nditer does not handle size 0 arr
ays
    419     # consistently
--> 420     b = np.broadcast(*args[:32])
    421     # unfortunately, it cannot handle 32 or more arguments directly
```

```
ValueError: shape mismatch: objects cannot be broadcast to a single shape
```