π Estimation

π is a widely used constant that enthusiasts around the world challenged to get to the most precise value. In Machine Learning, especially Monte Carlo simulation, this exercise demonstrates the enormous power of random numbers.

The experiment design is straight-forward: the probability of randomly selected points that fall inside of a circle inscribed in a square is the area ratio bewtween the circle and the square.

- Draw a unit square with a inscribed circle, with its center at the origin of a coordinate;
- Square's area is 1;
- Since the circle radius is 1, its area, per formula πr^2, is just π;
- Use a quarter of the circle and the square for simplicity:
    - points inside of the circle follow $\sqrt{a^2 + b^2} <= 1$

Therefore, the probability of any points land inside of the circle = circle area / area outside of circle.

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

Solve for π,

$$\pi = 4p$$

Numpy random uniform number generator is used to create data points (a, b).

In [48]:

```
import numpy as np
counter_circle = 0.
ratio = 0.
iteration = 1000000
for a, b in zip(np.random.uniform(0,1,iteration),np.random.uniform(0,1,itera
    if np.sqrt(a**2 + b**2) <= 1.:
        counter_circle = counter_circle + 1
ratio = counter_circle/iteration
print(f"Estimated pi = {4.0*ratio:.3f} (pi ~= 3.142)")
```

```
Estimated pi = 3.142 (pi ~= 3.142)
```

Amazing! One would start believing "...order comes out of chaos..."

In [ ]: