

**Лабораторное занятие 9.** Рассмотреть четкие расчеты по методу бэггинга и ансамблей. Написание программы.

**9-зертханалық сабақ.** Бэггинг және ансамблдер әдісіне нақты есеп қарастыру.  
Бағдарламасын жазу.

**Код построения доверительного интервала с помощью бэггинга:**

```
import numpy as np
```

```
def get_bootstrap_samples(data, n_samples):
```

```
    # функция для генерации подвыборок с помощью бутстрэпа
```

```
    indices = np.random.randint(0, len(data), (n_samples, len(data)))
```

```
    samples = data[indices]
```

```
    return samples
```

```
def stat_intervals(stat, alpha):
```

```
    # функция для интервальной оценки
```

```
    boundaries = np.percentile(stat, [100 * alpha / 2., 100 * (1 - alpha / 2.)])
```

```
    return boundaries
```

```
# сохранение в отдельные numpy массивы данных по лояльным и уже  
# бывшим клиентам
```

```
loyal_calls = telecom_data[telecom_data['Churn'] == False]['Customer service  
calls'].values
```

```
churn_calls = telecom_data[telecom_data['Churn'] == True]['Customer service  
calls'].values
```

```
# ставим seed для воспроизводимости результатов
```

```
np.random.seed(0)
```

```
# генерируем выборки с помощью бутстрэпа и сразу считаем по каждой из  
# них среднее
```

```
loyal_mean_scores = [np.mean(sample)
```

```
                     for sample in get_bootstrap_samples(loyal_calls, 1000)]
```

```
churn_mean_scores = [np.mean(sample)
```

```
for sample in get_bootstrap_samples(churn_calls, 1000)]
```

```
# выводим интервальную оценку среднего
```

```
print("Service calls from loyal: mean interval", stat_intervals(loyal_mean_scores,  
0.05))
```

```
print("Service calls from churn: mean interval",  
stat_intervals(churn_mean_scores, 0.05))
```