

**Міністерство освіти і науки України  
Національний технічний університет України  
"Київський політехнічний інститут імені Ігоря Сікорського"  
Фізико-технічний інститут**

**«Хмарні технології»**

**Лабораторна робота №5**

**Виконав:  
студент гр. ФБ-92  
Курганський Л.С.**

**Київ – 2022**

## Мета роботи: Елементи машинного навчання у AWS Sagemaker

### Завдання:

1. Обрати датасет у репозиторії <https://archive.ics.uci.edu/ml/index.php> (варіанти датасету мають бути погоджені з викладачем та не перетинатися)
2. Вивчити його особливості
3. Вирішити задачу класифікації / кластеризації за допомогою можливостей AWS Sagemaker.
4. Результати оформити протоколом

### Хід виконання роботи:

Було обрано датасет [“Iris”](#). Мета цього датасету дати є класифікація квітки за її зовнішніми характеристиками.

Завантаження даних:

```
# Loading the data
columns = ['sepal_length', 'sepal_width', 'petal_length', 'petal_width', 'class']

df = pd.read_csv("iris.csv", header=None, names = columns)
df.head()
```

	sepal_length	sepal_width	petal_length	petal_width	class
0	5.1	3.5	1.4	0.2	Iris-setosa
1	4.9	3.0	1.4	0.2	Iris-setosa
2	4.7	3.2	1.3	0.2	Iris-setosa
3	4.6	3.1	1.5	0.2	Iris-setosa
4	5.0	3.6	1.4	0.2	Iris-setosa

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 150 entries, 0 to 149
Data columns (total 5 columns):
#   Column          Non-Null Count  Dtype
---  ---
0   sepal_length    150 non-null   float64
1   sepal_width     150 non-null   float64
2   petal_length    150 non-null   float64
3   petal_width     150 non-null   float64
4   class           150 non-null   object
dtypes: float64(4), object(1)
memory usage: 6.0+ KB
```

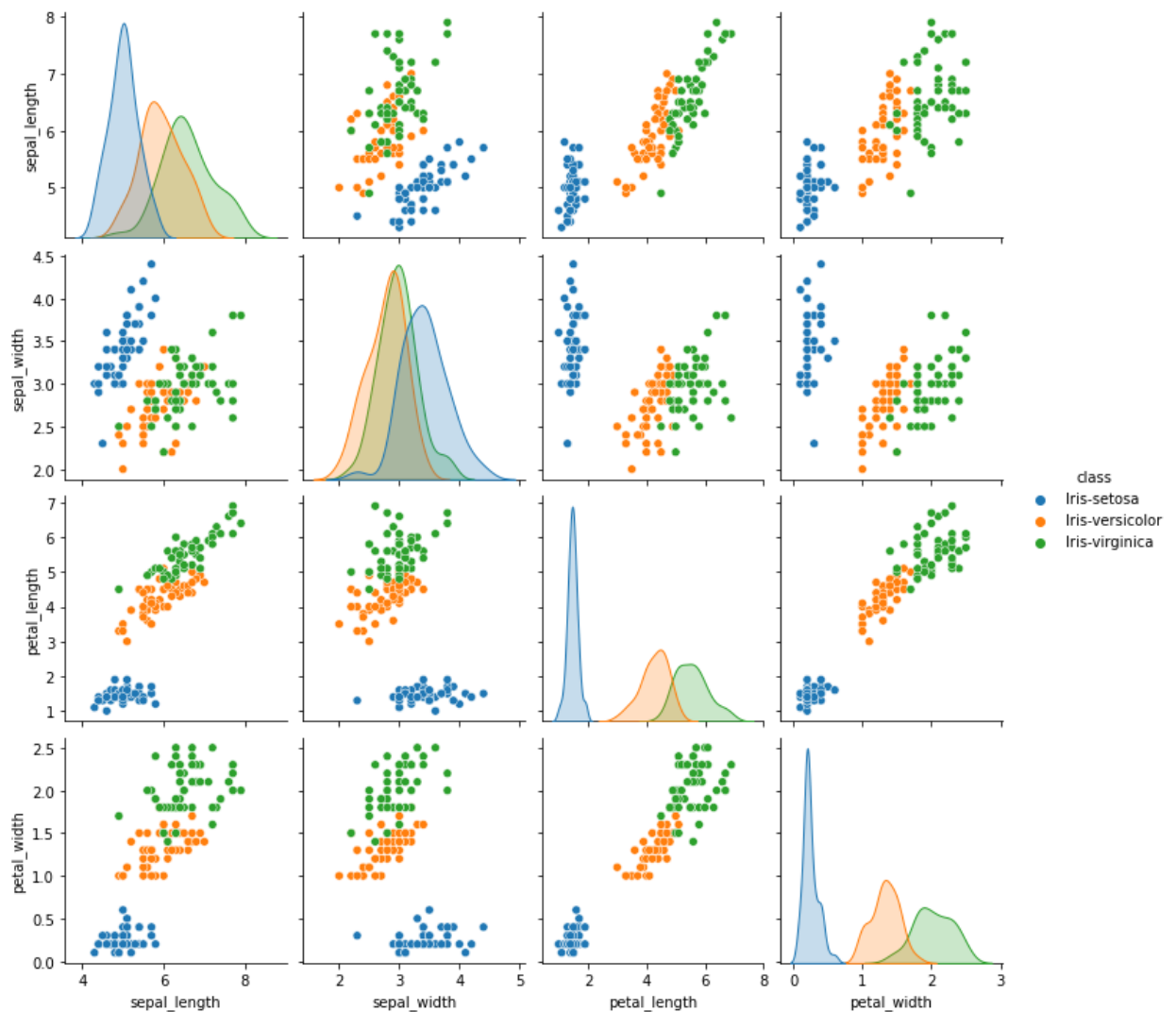
Отже, маємо 4 цифрових стовпця, які характеризують об'єкт та його клас.

Зробивши перевірку, зрозуміло що немає відсутніх даних.

```
# Checking of Nan  
df.isnull().values.any()  
  
False
```

Для отримання більшого розуміння можна провести візуалізацію залежності даних відповідно їх класу.

```
# visualization  
sns.pairplot(df[['sepal_length', 'sepal_width', 'petal_length', 'petal_width', 'class']], hue = 'class')
```



Завдяки графікам, зрозуміло що «petal\_length» і «petal\_width» мають гарну залежність, і тому підходять для виконання задачі, а «sepal\_width» та «sepal\_length» навпаки.

Знайшовши кореляцію можна вибирати дані для створення моделі.

```
# Correlation
df[['sepal_length', 'sepal_width', 'petal_length', 'petal_width']].corr()
```

	sepal_length	sepal_width	petal_length	petal_width
sepal_length	1.000000	-0.109369	0.871754	0.817954
sepal_width	-0.109369	1.000000	-0.420516	-0.356544
petal_length	0.871754	-0.420516	1.000000	0.962757
petal_width	0.817954	-0.356544	0.962757	1.000000

Далі розділяємо датасет для тренування і тестування:

```
# Splitting to train and test
x_train, x_test, y_train, y_test = train_test_split(
    df[['sepal_length', 'sepal_width', 'petal_length', 'petal_width']],
    df['class'], random_state = 5)
```

І обираємо модель для класифікації – k-NN (k-nearest neighbors).

```
# fit and test model
model = KNeighborsClassifier(n_neighbors=3)

model.fit(x_train, y_train)
y_pred = model.predict(x_test)
```

Спочатку були використані усі дані для побудови моделі і точність становила – 94.7 %

```
# check accuracy
print("Accuracy:", metrics.accuracy_score(y_test, y_pred))
```

Accuracy: 0.9473684210526315

Але, прибравши дані з слабкою кореляцією маємо такий же результат.




```
# Split without some data
x_train, x_test, y_train, y_test = train_test_split(
    df[['petal_length', 'petal_width']],
    df['class'], random_state = 5)
model = KNeighborsClassifier(n_neighbors=3)
model.fit(x_train, y_train)
y_pred = model.predict(x_test)
print("Accuracy:", metrics.accuracy_score(y_test, y_pred))
```

Accuracy: 0.9473684210526315


А отже, «sepal\_width» та «sepal\_length» не впливають на класифікацію.

## AWS Sagemaker


Спочатку створимо на бакеті з другої лабораторної папки: output, raw, train.

Name ▲	Type ▼
 output/	Folder
 raw/	Folder
 train/	Folder

Завантажимо датасет у папку raw.

Name ▲	Type ▼	Last modified
 iris.data	data	May 11, 2022, 18:15:21 (UTC+03:00)

І створюємо notebook instances.

Notebook instances				
<div><input type="text" value="Search notebook instances"/></div>				
	Name ▼	Instance	Creation time ▼	Status
<input type="radio"/>	lab5	ml.t2.medium	May 11, 2022 15:22 UTC	 Pending

Заходимо в jupyter, створюємо файл і розпочинаємо роботу з ним.

```
# setup variables to point to s3
filename = 'iris_recordio_train.data'
bucket = 'lkurgan'
data_dir = 'dataset'
dataset_name = 'iris.data'
raw_prefix = 'raw'
train_prefix = 'train'
output_prefix = 'output'
train_path = f'{train_prefix}/{filename}'
s3_train_data = f's3://{bucket}/{train_prefix}'
output_location = f's3://{bucket}/{output_prefix}'
```

```
%env DATA_DIR=$data_dir
%env S3_DATA_BUCKET_NAME = $bucket/$raw_prefix
%env DATASET_NAME = $dataset_name
%env TRAINING_PATH = $bucket/$train_prefix
```

```
env: DATA_DIR=dataset
env: S3_DATA_BUCKET_NAME=lkurgan/raw
env: DATASET_NAME=iris.data
env: TRAINING_PATH=lkurgan/train
```

Встановлення базових параметрів.

ФБ-92 Курганський Л.С.

Завантаження датасету.

```
!aws s3 cp s3://$S3_DATA_BUCKET_NAME/$DATASET_NAME ./$DATA_DIR/  
download: s3://lkurgan/raw/iris.data to $data_dir/iris.data
```

Зчитування даних.

```
# Loading the data  
columns = ['sepal_length', 'sepal_width', 'petal_length', 'petal_width', 'class']  
  
df = pd.read_csv(f"{data_dir}/iris.data", header=None, names = columns)  
df['class'].replace(['Iris-setosa', 'Iris-versicolor', 'Iris-virginica'], [1, 2, 3], inplace=True)  
df.head()
```


	sepal_length	sepal_width	petal_length	petal_width	class
0	5.1	3.5	1.4	0.2	1
1	4.9	3.0	1.4	0.2	1
2	4.7	3.2	1.3	0.2	1
3	4.6	3.1	1.5	0.2	1
4	5.0	3.6	1.4	0.2	1

Розподіл даних.

```
# Splitting to train and test  
x_train, x_test, y_train, y_test = train_test_split(  
    df[['sepal_length', 'sepal_width', 'petal_length', 'petal_width']],  
    df['class'], random_state = 5)
```

Перетворення даних в байти і завантаження на бакет.

```
buf = io.BytesIO()  
sagemaker.amazon.common.write_numpy_to_dense_tensor(buf, np.array(x_train).astype('float32'),  
np.array(y_train).astype('float32'))  
buf.seek(0)  
boto3.resource('s3').Bucket(bucket).Object(f'{train_path}').upload_fileobj(  
buf)
```

Name ▲	Type ▼	Last modified ▼
 iris_recordio_train.data	data	May 12, 2022, 01:02:46 (UTC+03:00)

## Створення моделі:

```
container = sagemaker.amazon.amazon_estimator.get_image_uri(boto3.Session().region_name, 'knn')
role = sagemaker.get_execution_role()
sess = sagemaker.Session()

# create knn estimator
knn = sagemaker.estimator.Estimator(container,
                                     role,
                                     train_instance_count=1,
                                     train_instance_type='ml.m5.4xlarge',
                                     output_path=output_location,
                                     sagemaker_session=sess)

# set hyperparameters
knn.set_hyperparameters(predictor_type='classifier',
                        feature_dim=4,
                        k=3,
                        sample_size=x_train.shape[0])

knn.fit({'train': s3_train_data}, job_name=f"iris-job-{int(time.time())}")
```

```
2022-05-12 16:12:05 Uploading - Uploading generated training model
2022-05-12 16:12:05 Completed - Training job completed
Training seconds: 182
Billable seconds: 182
```

### iris-job-1652371658

[Clone](#)[Create model package](#)[Stop](#)[Create model](#)

#### Job settings

Job name	SageMaker metrics time series
iris-job-1652371658	Disabled
ARN	Training time (seconds)
arn:aws:sagemaker:eu-west-3:631835917390:training-job/iris-job-1652371658	182
	Billable time (seconds)
	182
Status	Managed spot training savings
✔ Completed	0%
<a href="#">View history</a>	
Creation time	Tuning job source/parent
May 12, 2022 16:07 UTC	-
Last modified time	IAM role ARN
May 12, 2022 16:12 UTC	arn:aws:iam::631835917390:role/service-role/AmazonSageMaker-ExecutionRole-20220511T182196 <a href="#">↗</a>

## Загрузка моделі:

```
# deploy the model
knn_predictor = knn.deploy(initial_instance_count=1,
                           instance_type='ml.t2.medium',
                           endpoint_name="iris-endpoint")
knn_predictor.serializer = csv_serializer
knn_predictor.deserializer = json_deserializer
```

Тестування.

```
result = knn_predictor.predict(x_test.values)
```

Отримання значень.

```
result = [x['predicted_label'] for x in result["predictions"]]
```

Точність.

```
print("Accuracy:", metrics.accuracy_score(y_test, result))
```

Accuracy: 0.9473684210526315

Видалення ендпоінта.

```
sagemaker.Session().delete_endpoint(knn_predictor.endpoint)
```

The endpoint attribute has been renamed in sagemaker>=2.  
See: <https://sagemaker.readthedocs.io/en/stable/v2.html> for details.

## Перелік проблем:

Необхідні певні знання машинного навчання, для дослідження датасету.

У методі був вказаний не підходящий тип інстанса, проте «ml.m5.xlarge» підійшов.

**ResourceLimitExceeded:** An error occurred (ResourceLimitExceeded) when calling the CreateTrainingJob operation: The requested resource training-job/ml.m4.xlarge is not available in this region

## Висновок:

AWS надає хмарно-обчислювальну платформу «SageMaker», завдяки якій можна створювати, навчати і використовувати ML моделі на серверах сервісу.