МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМ. ІГОРЯ СІКОРСЬКОГО»

ФІЗИКО-ТЕХНІЧНИЙ ІНСТИТУТ

Кафедра фізико-технічних засобів захисту інформації

Лабораторна робота № 3

з дисципліни: «Автоматизація обробки ІзОД»

Керівник: Виконав:

Прогонов Дмитро Олександрович студент 5 курсу групи ФЕ-91мп

Захищено з оцінкою Павлусь Олександр Сергійович

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

дата, підпис

Київ – 2020 р.

**Вхідні дані**

Тестовий пакет – MIRFlickr-20k (https://press.liacs.nl/mirflickr/#sec\_download)

Вибірка зображень – 250 зображень;

Формування вибірки зображень – псевдовипадкове, з використанням генератора Мерсена (стартове значення співпадає з номером студента в загальному списку групи) за модулем кількості зображень в тестовому пакеті.

**Завдання**

1. Сформувати тестову вибірку зображень з вихідного пакета;
2. Для кожного каналу кольору кожного зображення з тестового пакета обчислити наступні характеристики:
   1. Математичне сподівання і дисперсію;
   2. Коефіцієнти асиметрії та ексцесу (нормалізований);
3. Використовуючи моделі SPAM і CC-PEV (<http://dde.binghamton.edu/download/feature_extractors/>), розрахувати вектора характеристик кожного каналу кольору кожного зображення;
4. Отримані параметри зображень упакувати в матрицю ознак (кожен рядок відповідає результатам для окремого каналу кольору тестового зображення, кожен стовпець - параметру зображення). Матриці ознак побудувати окремо для статистичних характеристик зображень (1-4 центральні моменти), а також моделей SPAM и CC-PEV;
5. Побудувати вектор міток класів зображень. Назва цільового класу для кожного студента визначається згідно з позицією студента у списку групи (див. Вкладений файл - наприклад, для першого студента в списку групи цільової клас "explore", для другого студента - мітка "sky", для третього студента - мітка "nikon" і т.д.);
6. псевдовипадковий чином розділити вихідний пакет зображень на 2 рівні частини (тестова і контрольна підвибірки). З використанням тестової підвибірки провести настройку наступних класифікаторів:
   1. Лінійна регресія;
   2. Робастна регресія;
   3. Логістична регресія;
   4. Метод опорных векторів (SVM);
7. Використовуючи налаштовані класифікатори з п. 6 провести обробки зображень з контрольного підпакету. Оцінити ймовірності правильної класифікації (0 і 1 класи), а також ймовірності помилок першого (помилкове спрацьовування) і другого (пропуск цілі) роду;
8. Повторити пп. 6-10 разів для отримання усередненої точності класифікації

**Хід роботи:**

1) Для кожного каналу кольору кожного зображення з тестового пакета обчислити наступні характеристики: Математичне сподівання і дисперсію; Коефіцієнти асиметрії та ексцесу (нормалізований);

RED = 0

GREEN = 1

BLUE = 2

*# Expected value & Variance*

*#RED*

sum\_val = sum(values[RED])

M\_red = 0

**for** index **in** range(len(values[RED])):

p = (values[RED][index] / sum\_val)

M\_red += p \* index

D\_red = 0

**for** index **in** range(len(values[RED])):

p = (values[RED][index] / sum\_val)

D\_red += p \* ((index - M\_red) \*\* 2)

print("Red: Expected value - **{0:.2f}**, Variance - **{1:.2f}**"\

.format(M\_red, D\_red))

*#GREEN*

sum\_val = sum(values[GREEN])

M\_green = 0

**for** index **in** range(len(values[GREEN])):

p = (values[GREEN][index] / sum\_val)

M\_green += p \* index

D\_green = 0

**for** index **in** range(len(values[GREEN])):

p = (values[GREEN][index] / sum\_val)

D\_green += p \* ((index - M\_green) \*\* 2)

print("Green: Expected value - **{0:.2f}**, Variance - **{1:.2f}**"\

.format(M\_green, D\_green))

*#BLUE*

sum\_val = sum(values[BLUE])

M\_blue = 0

**for** index **in** range(len(values[BLUE])):

p = (values[BLUE][index] / sum\_val)

M\_blue += p \* index

D\_blue = 0

**for** index **in** range(len(values[BLUE])):

p = (values[BLUE][index] / sum\_val)

D\_blue += p \* ((index - M\_blue) \*\* 2)

print("Blue: Expected value - **{0:.2f}**, Variance - **{1:.2f}**"\

.format(M\_blue, D\_blue))

Отримуємо:

Red: Expected value - 109.20, Variance - 6114.74

Green: Expected value - 100.81, Variance - 5584.68

Blue: Expected value - 94.57, Variance - 5972.40

sum\_val = sum(arr\_values)

ans = 0

**for** index **in** range(len(arr\_values)):

p = (arr\_values[index] / sum\_val)

ans += p \* ((index - M) \*\* power)

**return** ans

*#Asymmetry and excess*

Asym\_red = E\_operator(values[RED], M\_red, 3) / (D\_red \*\* (3 / 2))

Asym\_green = E\_operator(values[GREEN], M\_green, 3) / (D\_green \*\* (3 / 2))

Asym\_blue = E\_operator(values[BLUE], M\_blue, 3) / (D\_blue \*\* (3 / 2))

Отримуємо:

Red: Asymmetry - 0.243, Excess - 1.615

Green: Asymmetry - 0.389, Excess - 1.437

Blue: Asymmetry - 0.516, Excess - 1.921

2) Використовуючи моделі SPAM і CC-PEV розрахувати вектора характеристик для кожного зображення.

За допомогою моделі SPAM та СС-PEV було отримано набір ознак зображень із тестової вибірки. Дані були занесені у таблицю. Де кожному рядку відповідає зображення, для якого розраховувався набір ознак, а у кожному стовпчику значення ознак для одного зображення.

Для 8 варіанту заданий тег - “water”

Сформуємо вектори з мітками класів для 8 варіанту.

Y = Series()

**for** name **in** image\_names:

**with** open(os.path.join(DIRNAME\_TAGS, 'tags**%d**.txt' % (name,))) **as** file:

set\_tags = set(tag.strip() **for** tag **in** file.readlines())

*#print('tags%d.txt' % (name,), ' ',set\_tags)*

Y.set\_value(name, int(MAIN\_TAG **in** set\_tags))

Y = pd.DataFrame(Y, columns=['target'])

Пакети даних

data\_static = DataFrame()

num = 0

d = []

**for** image\_name **in** image\_names:

image = np.array(Image.open(os.path.join(DIRNAME\_IMAGES, 'im**%d**.jpg' % (image\_name))))

data = []

**for** name, num **in** COLOR.items():

a = image[:, :, num].ravel()

data.extend([np.mean(a), np.var(a), sp.stats.skew(a), sp.stats.kurtosis(a)])

d.append([image\_name, \*data])

data\_static = pd.DataFrame.from\_records(d)

data\_static = data\_static.rename(columns={0: 'name'}).set\_index('name')

3) Навчання моделі

**import** **itertools**

**from** **sklearn.linear\_model** **import** LinearRegression, LogisticRegression

**from** **sklearn.svm** **import** SVC

**from** **sklearn.model\_selection** **import** KFold, cross\_val\_score, train\_test\_split

**from** **sklearn.metrics** **import** accuracy\_score, recall\_score, confusion\_matrix

N = 10

**def** plot\_confusion\_matrix(cm, classes,

normalize=**True**,

title='Confusion matrix',

cmap=plt.cm.Blues):

*"""*

*This function prints and plots the confusion matrix.*

*Normalization can be applied by setting `normalize=True`.*

*"""*

plt.imshow(cm, interpolation='nearest', cmap=cmap)

plt.title(title)

plt.colorbar()

tick\_marks = np.arange(len(classes))

plt.xticks(tick\_marks, classes)

plt.yticks(tick\_marks, classes)

**if** normalize:

cm = cm.astype('float') / cm.sum(axis=1)[:, np.newaxis]

print("Normalized confusion matrix")

**else**:

print('Confusion matrix, without normalization')

print(cm)

thresh = cm.max() / 2.

**for** i, j **in** itertools.product(range(cm.shape[0]), range(cm.shape[1])):

plt.text(j, i, round(cm[i, j], 3),

horizontalalignment="center",

color="white" **if** cm[i, j] > thresh **else** "black")

plt.tight\_layout()

plt.ylabel('True label')

plt.xlabel('Predicted label')

font = {'size' : 15}

plt.rc('font', \*\*font)

З використанням тестової та контрольної вибірки налаштувати наступні класифікатори:

* Логістична регресія;
* Лінійна регресія;
* Метод опорних векторів (SVM);

Код програми:

**for** c **in** (LogisticRegression, LinearRegression, SVC):

cls = c()

cnf\_matrix = []

**for** i **in** range(N):

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.5)

cls.fit(X\_train, y\_train)

y\_predict = cls.predict(X\_test)

cnf\_matrix.append(confusion\_matrix(y\_test, y\_predict.astype(bool).astype(int)))

cnf\_matrix = np.mean(np.array(cnf\_matrix), axis=0)

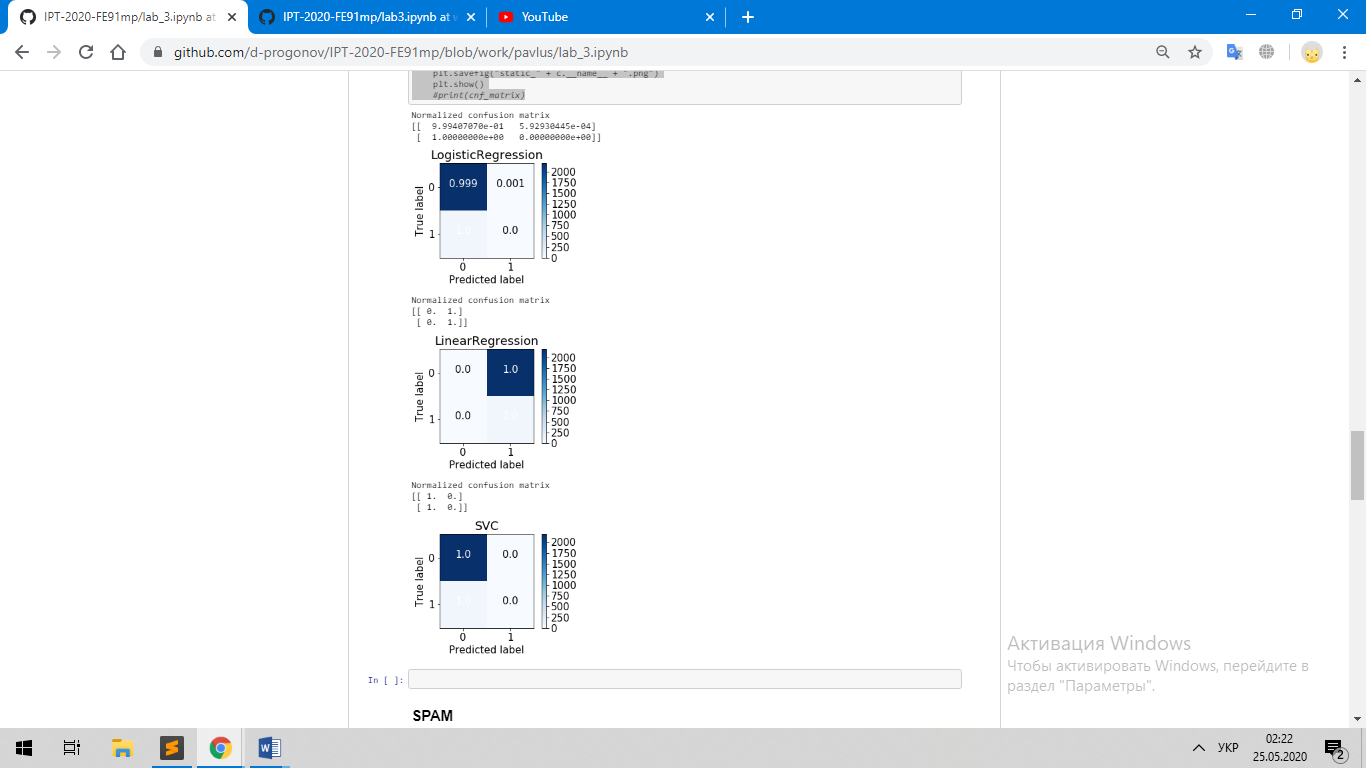
plt.figure(figsize=(4, 3))

plot\_confusion\_matrix(cnf\_matrix, classes=['0', '1'],

title=c.\_\_name\_\_)

plt.savefig("static\_" + c.\_\_name\_\_ + ".png")

plt.show()

Для налаштованих класифікаторів отримуємо такі результати:

**ВИСНОВКИ**

В даній лабораторній роботі було проаналізовано вибірку з 250 зображень датасету MIRFlickr-20k. Було знайдено що всі канали охоплюють увесь спектр значень. Було знайдено математичне очікування – 109 для чевоного каналу, 101 для зеленого і 95 для синього каналу відповідно і дисперсію 6114.74, 5584.68 та 5972.40.

У роботі були проаналізовані загальні методи класифікації об'єктів та систем. Було отримано матрицю ознак, за допомогою якої було налаштовано класифікатори.

Використовуючи мітку цільового класу «water» було сформовано дані для розрізняння що відповідають та не відповідають моделі. Також було натреновано моделі Лінійної та Логістичної Регресії для розрізняння. За даними таблиць результатів бачимо що він складав 1 та 0.999 відповідно для двох моделей.