****

Faculty of Computers and Artificial Intelligence

Computer Science Department

2021/2022

**CS 395 Selected Topics in CS-1**

**Research Project**

Report Submitted for Fulfillment of the Requirements and ILO’s for Selected Topics in CS-1 course for Fall 2021

Team No. 24

|  |  |  |  |
| --- | --- | --- | --- |
|  | ID | Name | Grade |
|  | 201900344 | Salma ahmed essa hassan |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Delivered to:

**Dr. Wessam El-Behaidy**

**Eng. Islam Gamal**

**Eng. Muhammed Kamal**

I. NUMERICAL DATASET

1. Project Introduction

* 1. **Dataset Name**

Bank marketing dataset (prediction term deposit subscription)

* 1. **Number of classes and their labels**

(Specify number of classes and their labels.)

* 1. **Dataset Samples Numbers**

(The total number of samples in dataset)

**Number of rows : 1136**

* 1. **Training, Validation and Testing**

(The number of samples used in training, validation and testing.)

* **Training: 9487**
* **Testing: 252**
* **Validation: 1423**

1. Implementation Details
   * 1. **Extracted Features**

(How many features were extracted, their names, the dimension of resulted features)

|  |  |  |
| --- | --- | --- |
| Feature | Feature\_Type | Description |
| age | numeric | age of a person |
| job | Categorical,nominal | type of job ('admin.','blue-collar','entrepreneur','housemaid','management','retired','self-employed','services','student','technician','unemployed','unknown') |
| marital | categorical,nominal | marital status ('divorced','married','single','unknown'; note: 'divorced' means divorced or widowed) |
| education | categorical,nominal | ('basic.4y','basic.6y','basic.9y','high.school','illiterate','professional.course','university.degree','unknown') |
| default | categorical,nominal | has credit in default? ('no','yes','unknown') |
| housing | categorical,nominal | has housing loan? ('no','yes','unknown') |
| loan | categorical,nominal | has personal loan? ('no','yes','unknown') |
| month | categorical,ordinal | last contact month of year ('jan', 'feb', 'mar', …, 'nov', 'dec') |
| day*of*week | categorical,ordinal | last contact day of the week ('mon','tue','wed','thu','fri') |
| duration | numeric | last contact duration, in seconds . Important note: this attribute highly affects the output target (e.g., if duration=0 then y='no') |
| campaign | numeric | number of contacts performed during this campaign and for this client (includes last contact) |
| pdays | numeric | number of days that passed by after the client was last contacted from a previous campaign (999 means client was not previously contacted) |
| previous | numeric | number of contacts performed before this campaign and for this client |

**Extract Feature:**

|  |  |  |
| --- | --- | --- |
| Feature | Feature\_Type | Description |
| deposit | binary | Has the client subscribe a term deposit?  (yes, no) |

* + 1. **Cross-validation**

(Is cross-validation is used in any of implemented models? If yes, specify the number of fold and ratio of training/validation)

* + 1. **Artificial Neural Network (ANN)**
* **Hyper-parameters**

(Specify all the hyper-parameters (initial learning rate, optimizer, regularization, batch size, no. of epochs…) with their specified value in implementation)

* + 1. **Support Vector Machine** **(SVM)**
* **Hyper-parameters**

(Specify all the hyper-parameters (optimizer, regularization, …) with their specified value in implementation)

**From sklearn import SVM**

Support Vector Machine (SVM) is a supervised machine learning algorithm that can be employed for both classification and regression purposes as support vector classification (SVC) and support vector regression (SVR).

The main idea is to identify the optimal separating hyperplane which maximizes the margin of the training data. Let us understand this objective term by term

**Parameters are as follows:**

C *float, default=1.0*

Regularization parameter. The strength of the regularization is inversely proportional to C. Must be strictly positive. The penalty is a squared l2 penalty.

kernel*{‘linear’, ‘poly’, ‘rbf’, ‘sigmoid’, ‘precomputed’} or callable, default=’rbf’*

Specifies the kernel type to be used in the algorithm. If none is given, ‘rbf’ will be used. If a callable is given it is used to pre-compute the kernel matrix from data matrices; that matrix should be an array of shape (n\_samples, n\_samples).

degree*int, default=3*

Degree of the polynomial kernel function (‘poly’). Ignored by all other kernels.

gamma*{‘scale’, ‘auto’} or float, default=’scale’*

Kernel coefficient for ‘rbf’, ‘poly’ and ‘sigmoid’.

coef0*float, default=0.0*

Independent term in kernel function. It is only significant in ‘poly’ and ‘sigmoid’.

tol*float, default=1e-3*

Tolerance for stopping criterion.

cache\_size*float, default=200*

Specify the size of the kernel cache (in MB).

max\_iter*int, default=-1*

Hard limit on iterations within solver, or -1 for no limit.

max\_iter*int, default=-1*

Hard limit on iterations within solver, or -1 for no limit.

**Methods:**

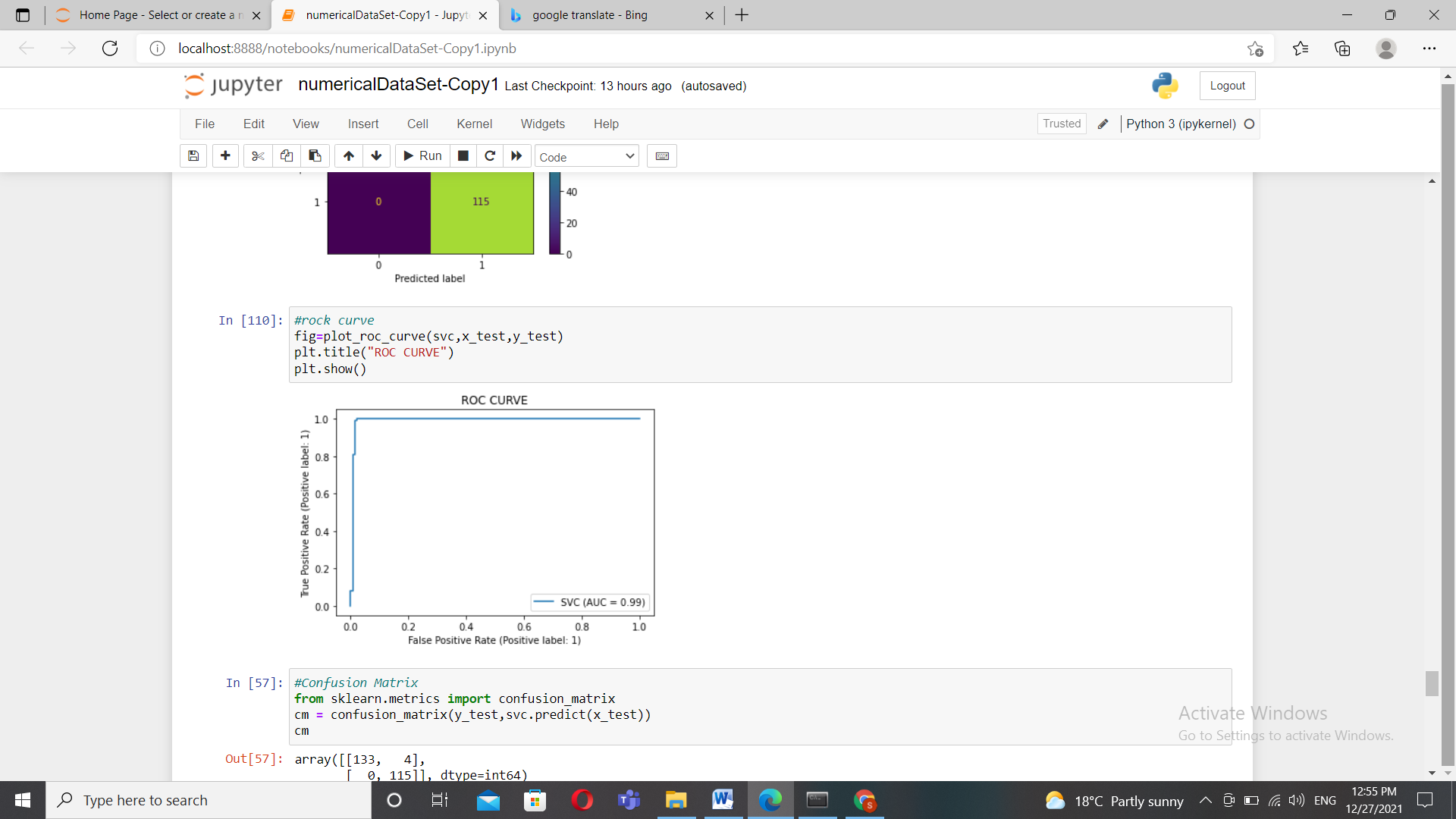
|  |  |
| --- | --- |
| [**decision\_function**](https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html#sklearn.svm.SVC.decision_function)(X) | Evaluate the decision function for the samples in X. |
| [**fit**](https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html#sklearn.svm.SVC.fit)(X, y[, sample\_weight]) | Fit the SVM model according to the given training data. |
| [**get\_params**](https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html#sklearn.svm.SVC.get_params)([deep]) | Get parameters for this estimator. |
| [**predict**](https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html#sklearn.svm.SVC.predict)(X) | Perform classification on samples in X. |
| [**predict\_log\_proba**](https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html#sklearn.svm.SVC.predict_log_proba)(X) | Compute log probabilities of possible outcomes for samples in X. |
| [**predict\_proba**](https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html#sklearn.svm.SVC.predict_proba)(X) | Compute probabilities of possible outcomes for samples in X. |
| [**score**](https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html#sklearn.svm.SVC.score)(X, y[, sample\_weight]) | Return the mean accuracy on the given test data and labels. |
| [**set\_params**](https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html#sklearn.svm.SVC.set_params)(\*\*params) | Set the parameters of this estimator. |

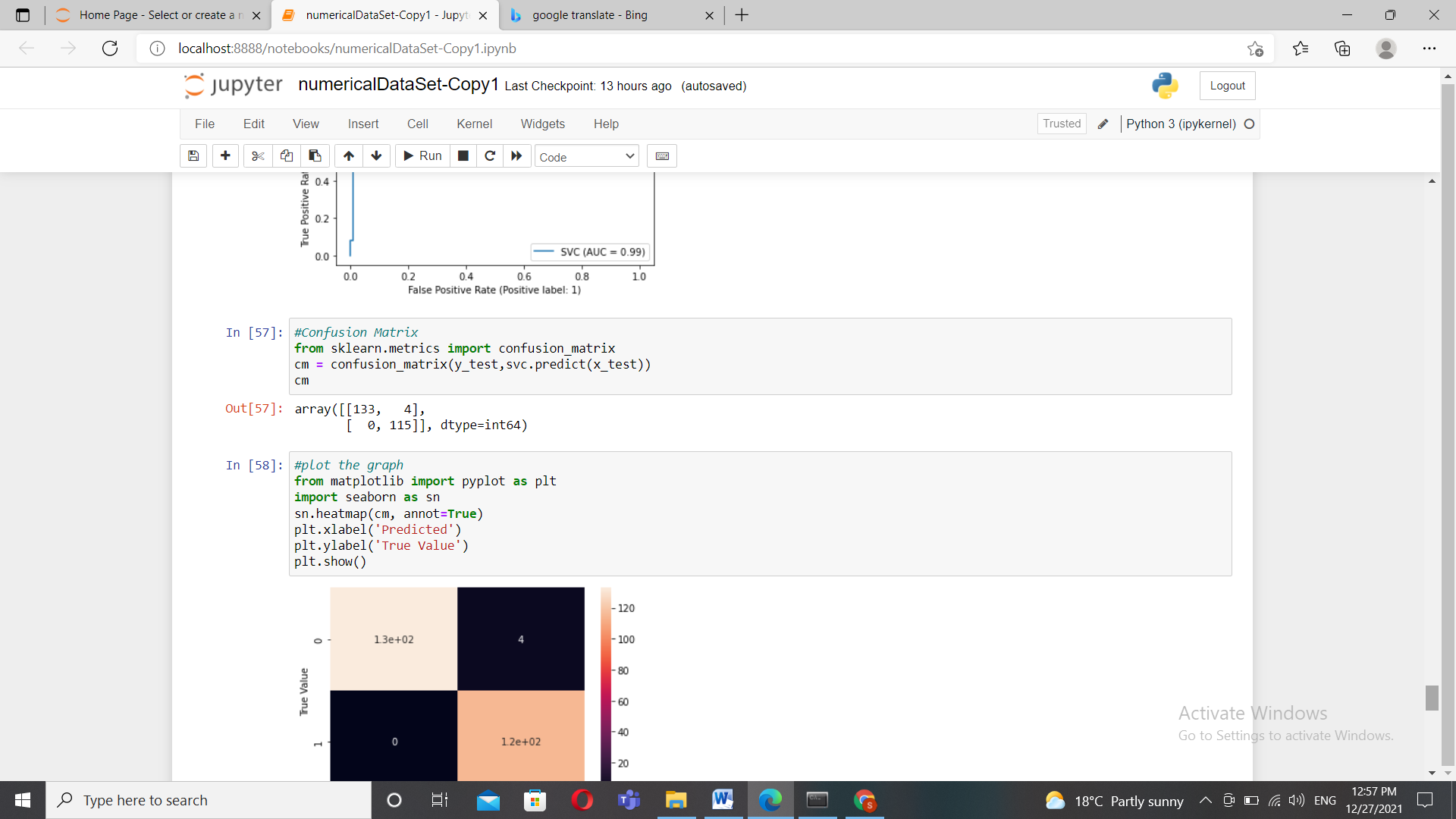
**We used kernel and gamma as a hyper parameters in our module.**

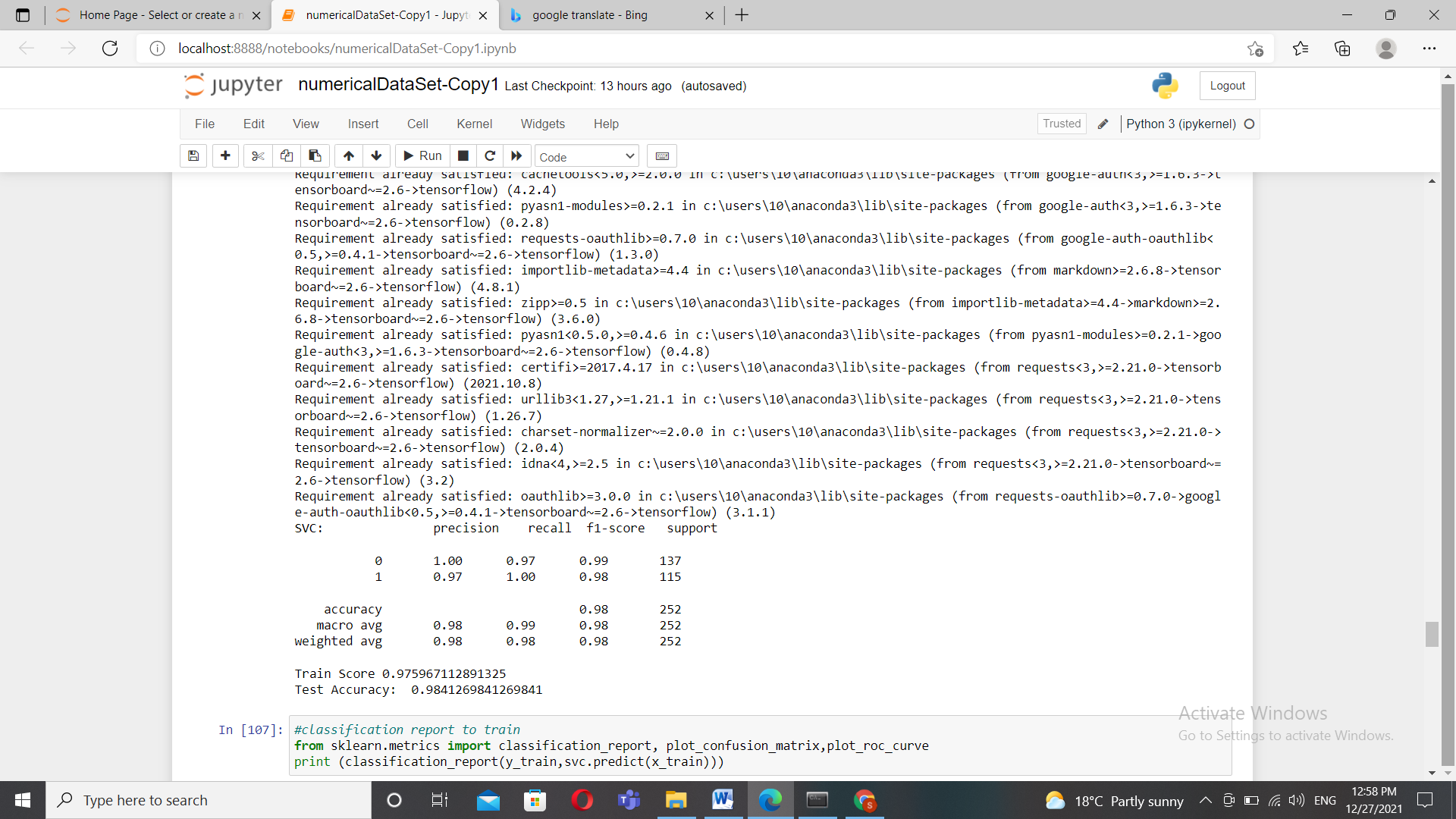
1. Models Results

**For each model you should show all these results for your model on testing data** (loss curve, accuracy, confusion matrix, ROC curve)

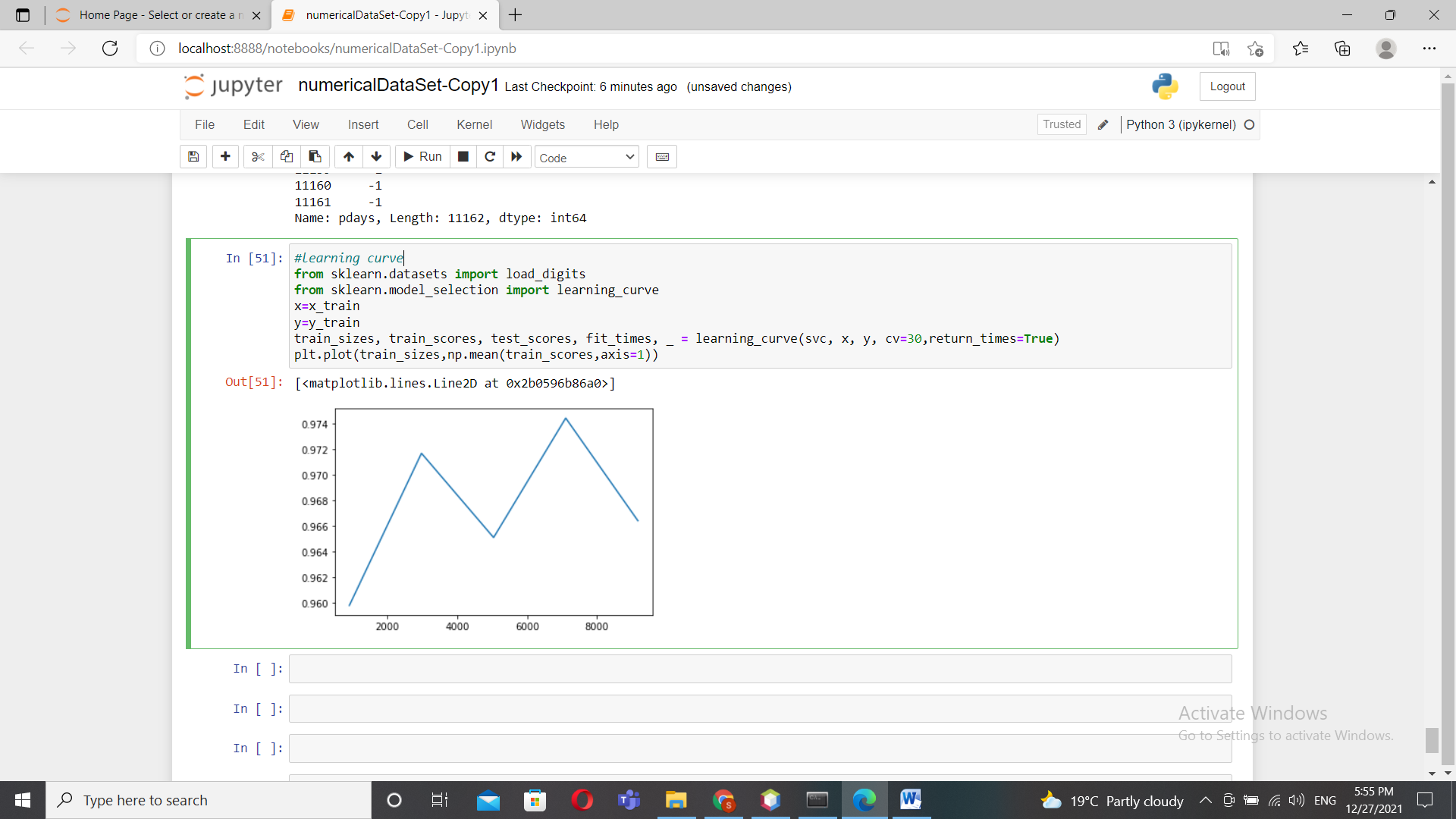
* 1. **ANN Results**

 **b.SVM Results**





Accuracy



II. IMAGE DATASET

1. Project Introduction

* 1. **Dataset Name**

Cats vs Dogs

* 1. **Number of classes and their labels**

2 classes, cats and dogs

* 1. **Dataset Images Numbers and size**

12,501\*2 image, with total size of 874 M

* 1. **Training, Validation and Testing**

25,002

2. Implementation Details

* + 1. **Extracted Features**

(How many features were extracted, their names, the dimension of resulted features)

* + 1. **Cross-validation**

(Is cross-validation is used in any of implemented models? If yes, specify the number of fold and ratio of training/validation)

* + 1. **Artificial Neural Network (ANN)**
* **Hyper-parameters**

(Specify all the hyper-parameters (initial learning rate, optimizer, regularization, batch size, no. of epochs…) with their specified value in implementation)

Epochs : 100

Batch size: 128

optimizer=tf.keras.optimizers.Adam(),

* + 1. **Support Vector Machine** **(SVM)**
* **Hyper-parameters**

(Specify all the hyper-parameters (optimizer, regularization, …) with their specified value in implementation)

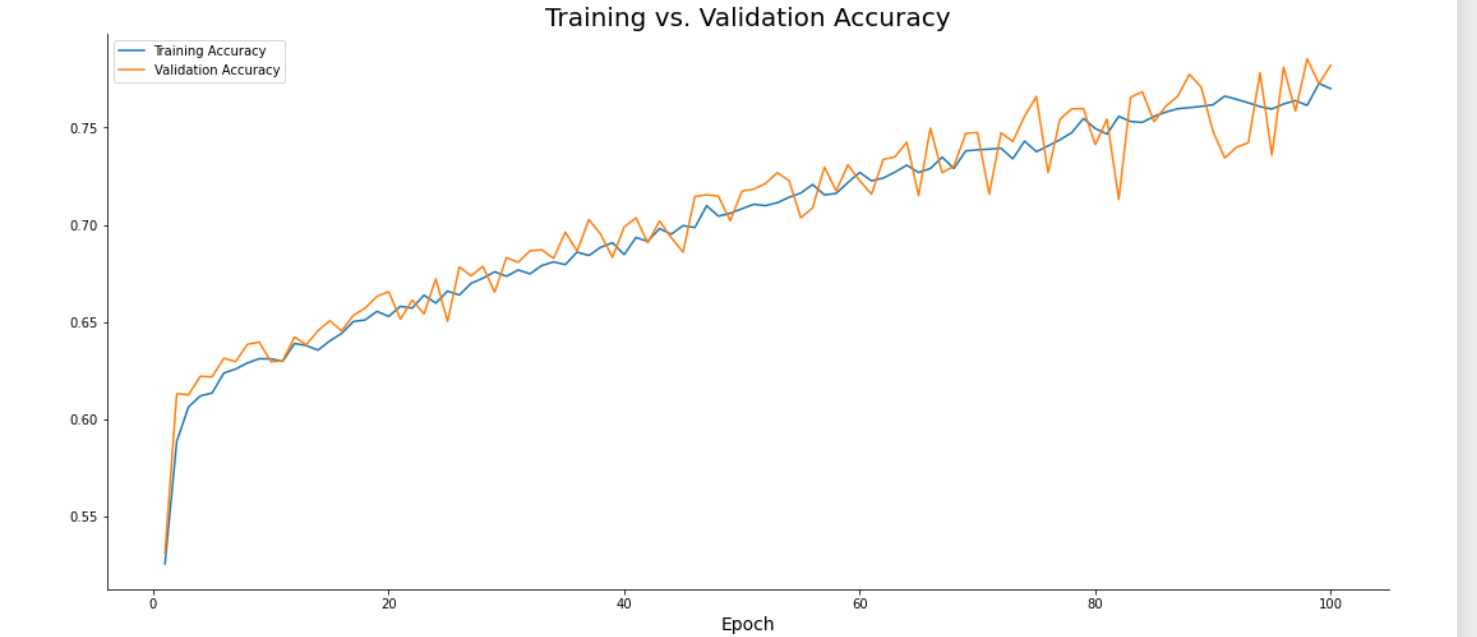
SVC(C=100, gamma='auto')

3. Models Results

**For each model you should show all these results for your model on testing data** (loss curve, accuracy, confusion matrix, ROC curve)

* 1. **ANN Results**

****

****

* 1. **SVM Results**