Recurrent neural algorithms to classify Type Ia supernovae

Hirmans Tabaharizato (hirmans@aims.ac.za)
African Institute for Mathematical Sciences (AIMS)

Supervised by: Prof. Bruce Bassett
University of Cape Town, African Institute for Mathematical Sciences, South Africa
Co-supervised by: Dr. Erick Jonathan Almaraz Aviña
African Institute for Mathematical Sciences, South Africa

14 May 2020
Submitted in partial fulfillment of a structured masters degree at AIMS South Africa

Abstract

A large amount of astronomical research is devoted to shed light on the nature of dark energy and dark matter. In the upcoming years the Large Synoptic Survey Telescope (LSST) will start operations collecting a large amount of data from astronomical objects. In this essay we discuss the classification of type Ia supernovae using the Supernovae Photometric Classification Challenge data. The goal of this essay was to build a supervised learning algorithm to classify type Ia supernovae. We built a model to classify light curves using recurrent neural networks and long-short term memory cells. The SPCC catalog simulates the real observing conditions (sky noise, atmosphere transparency) for the Dark Energy Survey (DES) telescope. Before implementing the classifier, we split the data set into the photometric and the spectroscopic samples, which contain type Ia supernovae in different proportions. We trained the model using the spectroscopic sample to classify type Ia supernovae in the photometric sample. We got the accuracy, the confusion matrix, the receiver operating characteristic (ROC) curve, and the area under the curve (AUC). The classifier performed well in the spectroscopic sample but performed poorly in the photometric sample. Particularly, the amount of false positive was large. Further improvements are needed.

Keywords: Cosmology, supernovae, machine learning.

Declaration

I, the undersigned, hereby declare that the work contained in this research project is my original work, and that any work done by others or by myself previously has been acknowledged and referenced accordingly.

Hirmans Tabaharizato, 14 May 2020

Contents

1. Introduction

It has been a long time that humans want to answer the question of who we are? Where did we come from? The study of the universe perhaps a way to respond to those questions. At the end of the twentieth century, the discovery that the expansion of the Universe is driven by the cosmological constant Λ which contributes to Einstein's equation (Wikipedia, a). Where,

$$R_{\mu\nu} = \frac{1}{2}Rg_{\mu\nu} + \frac{8\pi G}{c^4}T_{\mu\nu} + \Lambda g_{\mu\nu} \tag{1.0.1}$$

With $g_{\mu\nu}$ the metrics which describe the structure of space-time and G gravitational constants (or universal constant) and $R\mu\nu$ and $T_{\mu\nu}$ are the Ricci tensor and Stress-energy tensor, respectively. The measure of this Λ is very powerful in the research area in astronomy. The study of type Ia supernovae is among the one of finding measurements of the cosmological constant. This project talks more about type Ia supernovae which consider as "standard candles" (Mazure and Basa, 2007) in terms of the measurements of the distance between two astronomical objects in general. More precisely, this focuses on the classification of type Ia supernova using recurrent neural networks.

So the strategies to accomplish this task are subdivided into three parts whose the first part represents the explanation type Ia supernova including the supernova classification and the collection techniques (photometric and spectroscopic observations) of data from astronomical objects including the supernova (death of a star) which represent the large explosion of a star.

In the second part, we will talk about the tools for classification that contain the techniques to measure the performance of the classifier (or algorithm) including the confusion matrix which contributes a lot of the classification problem. But before that, we talk about machine learning which is a subfield of artificial intelligence. And for the creation of a model, we use recurrent neural networks including long short term memory cells to extract the long dependencies in the data. And on the last of the second part, we will see the metrics for the classification to conclude how good a model for classifier including the area under the curve of receiver operating characteristic and precision-recall curve.

In the third part, we will discuss the application of tools for classification of type Ia supernova using the "Supernova Photometric Classification Challenge" (SPCC) data (www.hep.anl.gov/SNchallenge). We train the model using the spectroscopic sample and test it on the photometric sample. And in the last of the third part, we discuss the confusion matrix.

2. Type Ia supernovae

In this section we will talk about supernovae. Our discussion will focus on the type 1a (SNe1a). These objects are very useful to study the nature of dark energy, which causes the accelerating expansion of the universe at late times.

2.1 Supernova classification

In 1941 Rudolph Lao Bernard Minkowski noticed that supernovae are classified into two main types: the type I is subdivided into three subtypes (Type Ia, Type Ib, and Type Ic) and the type II is subdivided also into subcategories as follows: Type IIb, Type IIn, Type IIP, and Type IIL. The difference between type I and type II is that the first one contains no hydrogen line in its spectrum, while the second one has a hydrogen line in its spectrum. Table 2.1 shows the chemical elements present in each type.

Type of supernova	subtype	chemical elements (as line present in the spectrum)
I	Ia	No Hydrogen (H), Calcium, Silicon (Si), Sulfur
	Ib	No Hydrogen (H) , Helium (H)
	Ic	No Hydrogen (H), No Helium (He)
II	IIb	weak Hydrogen (H), Helium
	IIn	Hydrogen (H)
	IIP	Hydrogen (H)
	IIL	Hydrogen (H)

Table 2.1: Classification of supernovae

The type II and Ib/Ic phenomena are similar because both arise from the core-collapse of a massive star on the final stage of its evolution.

2.2 Type Ia supernova

The type Ia supernova (SNeIa) is a phenomenon produced by a thermonuclear explosion of a white dwarf whose mass is near to $1M_{\odot}$ (mass of the Sun, $M_{\odot} = 1.989 \times 10^{30} kg$) but with a diameter hundred times smaller the Sun (Schneider (2015)). This phenomenon occurs when the white dwarf mass in a binary system exceeds the Chandrasekhar limit by absorbing gas from another star.

Type Ia supernovae serve as a useful tool to probe the nature of dark energy, which in the standard model of cosmology (ΛCDM) is described by a cosmological constant (Λ). Although this model successfully encompasses the observational evidence available at the present time, there is no theoretical understanding of the physical mechanism that determines the value of the cosmological constant. The lack of understanding of the nature of dark energy is perhaps one of the most challenging questions of modern physics. This problem impels the quest for more observational evidence that might constrain further the properties of this mysterious physical agent.

In this respect, SNeIa serve as useful as "standard candles" to measure cosmological distances and probe the expansion of the universe.

Table 2.2: two examples Telescope which can observe astronomical objects

	HSC	PSF
i=22.5 objects	10 sec	1 hour
objects/FoV	$10\hat{}4-5$	2400

2.3 Observational strategies.

In this section we discuss the observational methods used to find supernovae in modern astronomical surveys.

2.3.1 Photometric observations.

Photometry is a technique used to measure the light and to count the number of photons per second coming from an astronomical source in the specific wavelengths or pass-bands known as filters.

From the measured flux the distance to the object is given by Avia (2018)

$$F = \frac{L}{4\pi d^2}, \tag{2.3.1}$$

where F represents the brightness (flux) of the object, L represents the luminosity, and d is the distance to the object.

2.3.2 Spectroscopic observations.

Spectroscopy is a technique used to study the spectrum of the light received from the astronomical objects being observed. This technique allows us to know the some properties of the objects such as their chemical composition, the temperature, the density, and relative motion (velocity). Spectroscopy is also used to analyze the interaction between matter and electromagnetic radiation emitted from an object.

In cosmology spectroscopy is also used to measure the distance from a source. However, spectroscopy is is very expensive in terms of telescope time. For example, the HSC (Hyper Supreme-Cam) is a gigantic digital still camera of 8.2 m which is being built by the National Astronomical Observatory of Japan in collaboration with international academic and industrial partners (see table 2.2).

Photometric and spectroscopic observations are useful tools to collect astronomical data. However, spectroscopic surveys are very expensive, so there are not enough resources to follow each one of the objects. This limitation has motivated the use of photometric information to classify astronomical objects. However, classification algorithms that are trained using spectroscopic data to classify photometric data perform poorly. This is because spectroscopic observations are not representative of the photometric sample. Spectroscopic observations prioritize following up objects at lower redshifts and larger brightness than in photometric surveys. For instance in the Dark Energy Survey (DES) they are only 20 percent of sample for spectroscopic classifications where up to 4000 SNe Ia light curves sample. Even in the Pan-STARRS Medium Deep Survey the size of the supernovae sample is also poor, amounting only to 5000 objects. However, only 10 percent out of this sample was spectroscopic followed up. Furthermore, the upcoming Large Synoptic Survey Telescope (LSST) is expected to collect almost $\sim 100,000$ SNe Ia, but only a small fraction will be spectroscopically confirmed. In view of this problem, it is necessary to develop new classification tools to extract the most available information from these surveys. In the next section, we discuss a very promising tool for addressing this problem.

3. Tools for classification.

We now discuss the tools that we'll use to implement the algorithms of classification. We use machine learning (ML), which is one of the subfields of Artificial Intelligence (AI). The impact of AI on modern society can be seen in many technological applications. In the context of astronomy, ML algorithms have been applied on many tasks including the description of complicated relationships in the data, clustering and detection of outliers, generation of simulated data, and classification and exploration of data sets. Machine learning has great potential in the astronomy community . For instance, supernova classification is crucial for obtaining cosmological constraints from type Ia supernovae in upcoming surveys such as LSST.

In machine learning there are three main problems to solve:

- Supervised learning (when one has labeled data),
- Unsupervised learning (for unlabeled data),
- Semi-supervised learning (some of the data are labeled).

Supervised learning problems subdivide into two groups: regression and classification problems. In our case, we work on a classification problem.

3.1 Machine learning for classification.

A supervised learning classification algorithm is a machine learning task to learn a function that maps a set of input features to class labels (the target variable). Let X be the input features which are represented as vectors of input space N_X and Y the target variable. Then, the supervised learning function can be written as:

$$Y = f(X), \tag{3.1.1}$$

where f is the mapping function from the input to the target. The goal is to approximate the mapping function so that when we have new input data, we can predict the corresponding class labels. This process is called a supervised learning algorithms.

There are different methods to learn this mapping, such as standard neural networks, convolutional neural networks, recurrent neural networks (RNN), and other machine learning algorithms. In all cases we need a training set to fit the parameters of the algorithm, and a test set to asses the performance of the model. To have a nice test set two conditions are needed:

- a large amount of data.
- the test set has to be representative of the training set.

In this work we use RNNs to classify SNe Ia.

3.2 Recurrent Neural Networks.

RNNs is one type of networks used to learn sequential data. At the end of the 1980's several researchers introduced simple partially recurrent neural networks to learn sequences of characters. Many other applications of RNNs include machine translation, sentiment classification, speech recognition, language model, etc. In our case RNNs are used to classify astronomical objects by studying the light curves. Different types of RNNs can be used to accomplish this task including the unidirectional recurrent neural networks (also called forward propagation), where the data is scanned from the left to the right, bidirectional recurrent networks (or forward and backward propagation), i.e, the data can be scanned in both directions. In any case, it has been shown that long short-term memory (LSTM) cells render better results than recurrent neural cells by capturing long-term dependencies or correlations in th data.

In mathematical terms, simple RNNs cells are defined as follows. Let $S^{<t>}$ be a new state at time t depending on the input $X^{<t>}$ and the previous state at time $t-1$, $S^{<t-1>}$:

$$S^{<t>} = F_w(S^{<t-1>}, X^{<t>}), \tag{3.2.1}$$

where,

- $X^{<t>}$ represents the input at time t with dimension T_X
- $S^{<t-1>}$ represents the state at time step $t-1$.
- F_w represents an activation function.

In neural networks the activation function is a function used to determine the target from the input features of the data (see figure 3.1). In general, this function is a non-linear function whose form depends on the task. There are several activation functions F_w including rectified linear units (ReLU), hyperbolic tangent, sigmoid, and softmax.

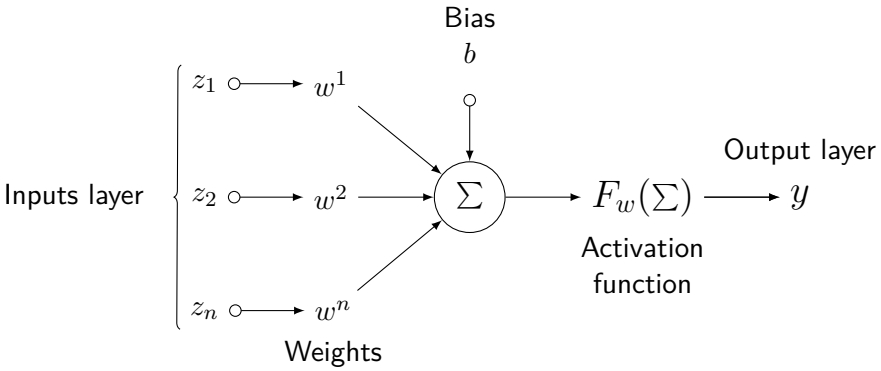

Figure 3.1: The architecture of neural networks from input layer to output layer (the index n indicate the number of input, $\sum = \sum_{i=1}^{n} w^i z_i + b$)

These functions are defined as follows:

Rectified linear unit.

The rectified linear unit is given by:

$$\text{ReLU}(z) = \begin{cases} 0 & \text{if } z < 0, \\ z & \text{if } z \geq 0. \end{cases}$$

The output is the identity function if the argument is positive and zero otherwise (see figure 3.2). This function is normally used in the hidden layers of the networks.

Hyperbolic tangent.

This function is given by:

$$\tanh(z) = \frac{\sinh(z)}{\cosh(z)}, \tag{3.2.2}$$

where $\sinh(z) = \frac{e^z - e^{-z}}{2}$ and $\cosh(z) = \frac{e^z + e^{-z}}{2}$ and the range of $\tanh$ is between -1 to 1.

Sigmoid function.

This activation function is referred to as logistic function or S-shape curve (see figure 3.2) function in some literature.

$$\sigma(z) = \frac{e^z}{1 + e^z} = \frac{1}{1 + e^{-z}}, \tag{3.2.3}$$

where z represents the weighted sum of the parameter of the layer and the bias. This function is normally used in binary classification problems in the output layer. When z goes to infinity $\sigma(z)$ tends to 1. On the other hand, σ goes to zero if z tends to minus infinity. So the goal of this function is to make a prediction in binary classification problems. In this case:

$$\text{prediction} = \begin{cases} \text{class } 0 \text{ if } \sigma(z) < 0.5, \\ \text{class } 1 \text{ if } \sigma(z) \geq 0.5 \end{cases}$$

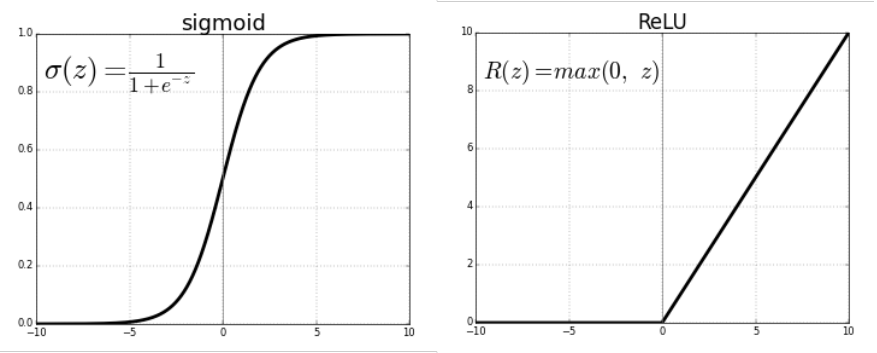

Figure 3.2: Sigmoid-$\sigma(z)$ and ReLU(z) activation functions diagram

Softmax function

Softmax function is also a type of logistic function that normalizes the input value into a vector values. This vector value computes the probability distribution sum to 1. This function is given by the following formula:

$$\text{Softmax}(x) = \frac{e^{z_i}}{\sum_{i=1}^{n} e^{z_i}}, \tag{3.2.4}$$

z_i indicate the elements of the input. This function is used in the multivariate model. The range of this function is between 0 and 1.

For example,

$$S^{<t>} = \tanh(W_{SS}S^{<t-1>} + W_{SX}X^{<t>} + b_S), \tag{3.2.5}$$

with the coefficients W_{SS}, W_{SX} are respectively the weights matrices at previous times steps and input, and the b_S represents the biases which have the same dimensions as $S^{<t>}$. To simplify the notation in equation 3.2.5, we can rewrite this expression as follows:

$$S^{<t>} = \tanh(W_S[S^{<t-1>},\ X^{<t>}] + b_S), \tag{3.2.6}$$

where

$$\begin{cases} W_S = [W_{SS} \vdots W_{SX}] \ \ (\star_1) \\ \\ [S^{<t-1>},\ X^{<t>}] = \begin{bmatrix} S^{<t-1>} \\ \cdots\cdots \\ X^{<t>} \end{bmatrix} \ \ (\star_2) \end{cases}$$

So the product of $(\star_1)$ and $(\star_2)$ gives back the term $W_{SS}S^{<t-1>} + W_{SX}X^{<t>}$ inside of the equation 3.2.5.

The expression 3.2.6 also ensures the connection between output and input at time step t. When the output (target) is given by the function G_w (sigmoid σ, softmax in general) that depends on new state at the time t $S^{<t>}$:

$$\begin{aligned} Y^{<t>} &= G_w(S^{<t>}) \\ &= \text{softmax}(W_Y S^{<t>} + b_Y) \end{aligned} \tag{3.2.7}$$

With $W_{YS} = W_Y$ indicate weight matrix of dimension $T_Y \times T_S$. As we see, $Y^{<t>}$ is a set of the output depended by the new state at time step t, $S^{<t>}$.

In consequence, both relations 3.2.6 and 3.2.7 represent the general definition of forward propagation of RNNs, i.e, RNNs scan only the data from the left (previous information) to right (current information) until the end of times series.

Thus, the goal of RNNs is to build a function mapping from the feature (input) to predict specific objects a target (output) such as a label. The figure 3.3 illustrate the structure of a single time stepe in a RNNs.

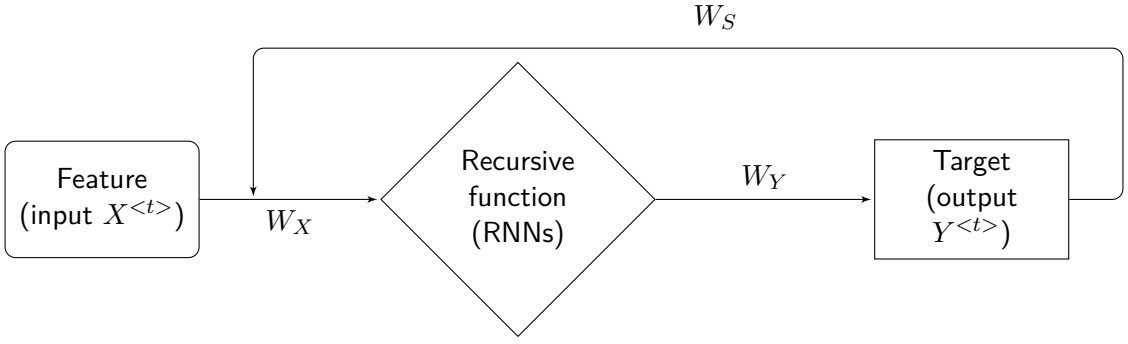

Figure 3.3: Elemental diagram of a RNNs cell

If we consider n times steps, we have:

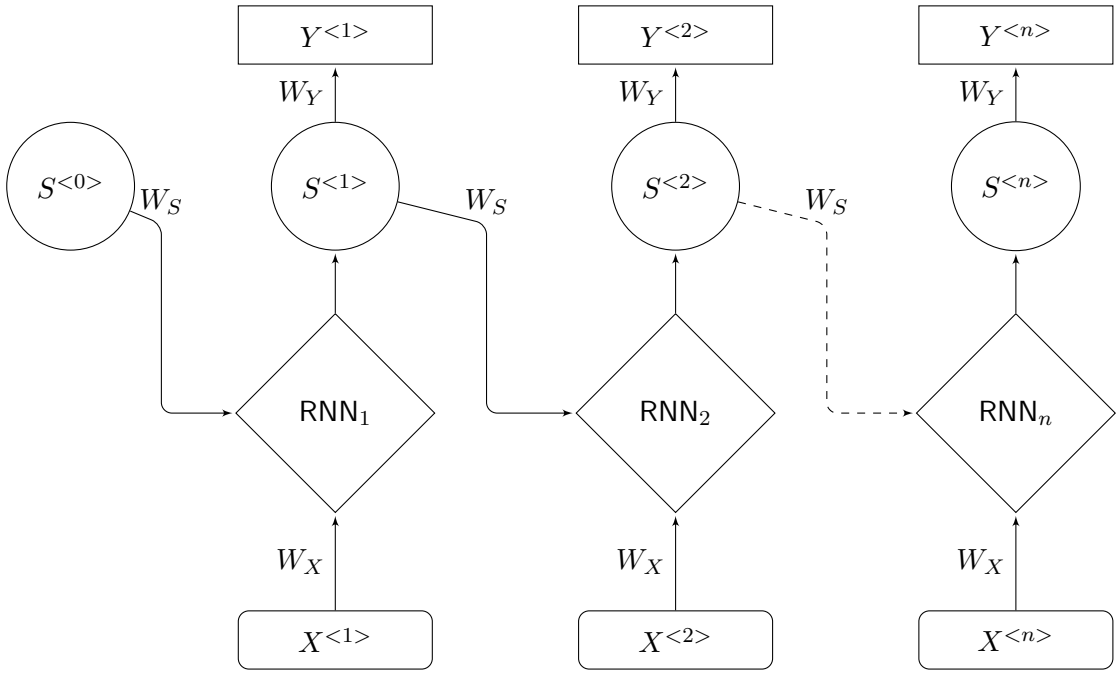

Figure 3.4: Recurrent neural network diagram which represents a prediction of a target $(Y^{<t>})_{1\leq t\leq n}$ activated by input space $(X^{<t>})_{1\leq t\leq n}$. With $\text{RNN}_1 = \tanh(S^{<0>}, X^{<1>})$, $\text{RNN}_2 = \tanh(S^{<1>}, X^{<2>})$, $\text{RNN}_n = \tanh(S^{<n-1>}, X^{<n>})$. In this case, the length of input sequence and output sequence are the same.

In a multi hidden layer architecture, the flow of the information is same, but the difference is that one has more than one hidden layer in every time step, where the first output becomes an input for the next step.

3.2.1 Long short-term memory.

The long short-term memory is type of cell used in RNNs architectures. This type of RNNs is more powerful than simple RNNs because LSTM cells are particularly used to extract long dependencies in the data. LSTM cells incorporate a forget gate, an update gate and an output gate. In general, they are sigmoid functions that depend on the hidden state $S^{<t-1>}$ at time step $t-1$ and input $X^{<t>}$ at time step. But the difference between them is that the weight matrices and their biases are not the same. If we denote $f^{<t>}$, $u^{<t>}$ and $o^{<t>}$ the forget gate , update gate and output gate, respectively, we have:

$$\begin{cases} f^{<t>} = \sigma(W_f[S^{<t-1>}, X^{<t>}] + b_f), \\ u^{<t>} = \sigma(W_u[S^{<t-1>}, X^{<t>}] + b_u), \\ o^{<t>} = \sigma(W_o[S^{<t-1>}, X^{<t>}] + b_o), \end{cases} \tag{3.2.8}$$

where

$$\begin{cases} W_k = [W_{kS} \vdots W_{kX}] \text{ with index } k \text{ may be } f, u, o \ (I) \\ [S^{<t-1>}, \ X^{<t>}] = \begin{bmatrix} S^{<t-1>} \\ \cdots\cdots \\ X^{<t>} \end{bmatrix} \ (II) \end{cases}$$

The new cell memory $\tilde{c}^{<t>}$ which is gieven by:

$$\tilde{c}^{<t>} = \tanh(W_c[S^{<t-1>}, X^{<t>}] + b_c), \tag{3.2.9}$$

with

$$W_c = [W_{cS} \vdots W_{cX}]$$

The equation 3.2.9 is used to coordinate the next input at time step t observed with the previous hidden layer at time step $t-1$.

The goal of these relations is to obtain the current cell memory at time t and the output as hidden layer at time step t through previous cell memory and hidden layer at time step $t-1$. The figure 3.1 illustrate the inner structure of LSTM cell. The current memory cell $(c^{<t>})$ combines the output of the forget and update cells. The action of this cell is divided into two options as follows:

1- to forget the previous memory.

2- to generate the new memory.

Where,

$$c^{<t>} = u^{<t>} \circ \tilde{c}^{<t>} + f^{<t>} \circ c^{<t-1>}, \tag{3.2.10}$$

where $\circ$ represents dot products. From the relation 3.2.10, we can compute the output (hidden layer) $S^{<t>}$ at time step t that is given by product between the output gate and hyperbolic tangent of current cell memory at time step t, where,

$$S^{<t>} = o^{<t>} \circ \tanh(c^{<t>}). \tag{3.2.11}$$

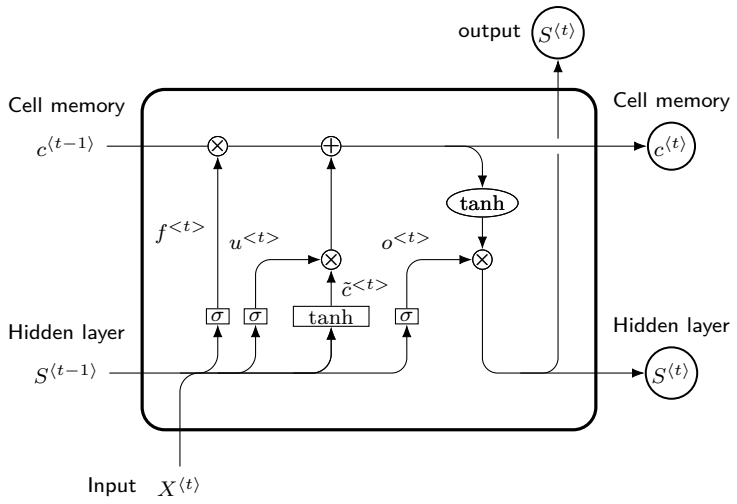

Figure 3.5: Representation of Long short term memory (LSTM)

3.3 Metrics for classification.

In this section we talk about the measure of performance of the algorithm. There are different types of metrics including precision-recall, accuracy, receiver operating characteristic (ROC), etc.

3.3.1 Confusion matrix.

Consider a problem of prediction of K classes for a test set of n samples. Let's denote C_j $(1 \leq j \leq K)$ the prediction class values and C'_i $(1 \leq i \leq K)$ actual class values correspond to the true class values. Both classes can be arranged in a confusion matrix C_M as shown in table 3.1.

C_M a matrix with $(c_{ij})_{1 \leq i,j \leq K}$ elements, where i and j indicate the columns and rows, respectively.

	C_1	C_2	$\cdots$	$\cdots$	C_K
C'_1	c_{11}	c_{21}	$\cdots$	$\cdots$	c_{K1}
C'_2	c_{12}	c_{22}	$\cdots$	$\cdots$	c_{K2}
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
$\vdots$	$\vdots$	$\vdots$	$\cdots$	$\ddots$	$\vdots$
C'_K	c_{1K}	c_{2K}	$\cdots$	$\cdots$	c_{KK}

Table 3.1: Confusion matrix for classification

Where c_{ij} terms represent number of objects in prediction class value (C_j) and number of objects in true class value (C_i). So, the elements present in the diagonal of C_M (where i=j) indicate the correct classified objects. The diagonal-off elements correspond to the number of missclasified objects.

From this matrix 3.1, we can measure the performance of the algorithm (i.e classifier) In binary classification problems, the confusion matrix is simplified to:

$$C_M = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \tag{3.3.1}$$

So the goal of this matrix is to indicate how good a model for predicting objects in a classification is. The presented elements values in each columns indicate us the numbers of prediction values and in each rows represent actual values. To make clear, the following table 3.2 can explain it. The elements of in diagonal (a_{11} and a_{22}) of matrix C indicate the correct of classification in a model of prediction and the other elements (a_{21} and a_{12}) the incorrect prediction also called type errors. In general, a_{11} refereed to true positive (TP) value of the thing being studied during prediction and a_{22} refereed as a false negative (FN) value of correct prediction (see table 3.2 to be clear Data Science and Machine Learning).

		Prediction value (PV)		
		Positive (1)	Negative(0)	Total of AV (each rows)
Actual values (AV)	Positive (1)	True Positive (TP)	False Negative (FN)	TP+FN
	Negative (0)	False Positive (FP)	True Negative (TN)	FP+TN
	Total of PV (each columns)	TP+FP	FN+TN	

Table 3.2: Confusion matrix of two class: Positive (1) and Negative (0)

The meaning of the four entries of matrix (TP, FP, TN, and FN) is the following:

1- **True positive (TP)** is the number of correct predictions for the positive class

2. **False Positive (FP)** indicates the number of incorrect predictions of the positive class.

3. **False Negative (FN)** indicates the number of incorrect predictions for the negative class . This is also another type of error because in our model we predict this number is negative but in the actual value, we have the number of positive (1) in our case.

4. **True negative (TN)** is the number of predictions for the negative class.

Precision is the total number of true positive divided by the sum of the number of a true positives and the number of false positives.

$$\text{Precision} = \frac{\text{TP}}{\text{TP+FP}}. \tag{3.3.2}$$

The **recall** also called sensitivity is:

$$\text{Recall} = \frac{\text{TP}}{\text{TP+FN}}. \tag{3.3.3}$$

The **accuracy** is the sum of the number of true positive and the number of true negative divided by the four outputs (TP+FP+FN+TN):

$$\text{Accuracy} = \frac{\text{TP+TN}}{\text{TP+FP+FN+TN}}. \tag{3.3.4}$$

In other over words, the accuracy is a ratio of the trace of confusion matrix and the sum of all elements of confusion matrix, where,

$$\text{Accuracy} = \frac{Tr(C_M)}{c}, \tag{3.3.5}$$

where

$$c = \text{TP} + \text{FP} + \text{TN} + \text{FN}.$$

In consequence, if the accuracy is high we can conclude that the performance of algorithm is good.

3.3.2 Receiver operating characteristic (ROC) curve.

To know the performance of machine learning algorithm (MLA) in a binary classification, we use ROC -curves. This metric is a graph given by the false positive rate (FPR) (in x-axis) versus true positive rate (TPR) (in the y-axis) through various number of classification thresholds

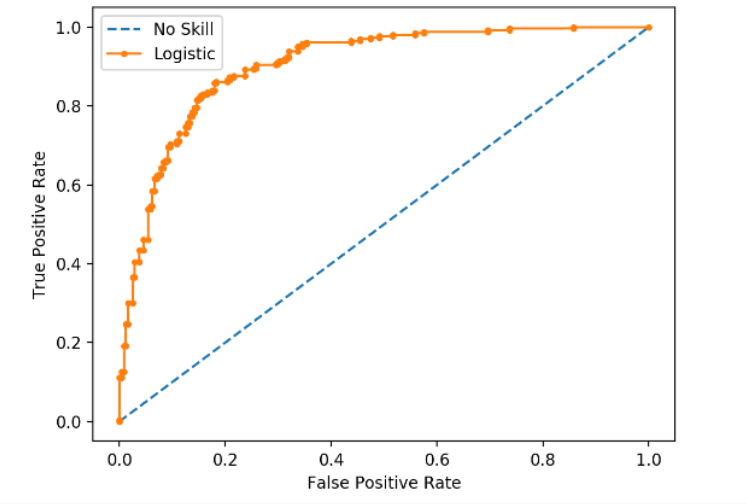

Figure 3.6: ROC Curve Plot for a No Skill Classifier and a Logistic Regression Model (Brownlee (2020))

Where,

$$\text{TPR} = f(\text{FPR}) \ (f : \text{function}) \tag{3.3.6}$$

TPR is the ratio between true positive (TP) and the sum of the true positive number and the false positive (FP) number.

$$\text{TPR} = \frac{\text{TP}}{\text{TP}+\text{FN}}. \tag{3.3.7}$$

TPR is also called the recall given by equation 3.3.3.

The FPR is given by division between false negative numbers and the sum of false positive and true negative numbers:

$$\text{FPR} = \frac{\text{FP}}{\text{FP}+\text{TN}}. \tag{3.3.8}$$

Where, FPR is also call called contamination.

So the area under the ROC curve is important to summarize if the the performance of model is good or bad.

3.3.3 Precision-recall curve (PRC).

By definition, Precision-recall curve is a plot of precision in the y-axis and recall in the x-axis.

The goal of this metric is to assure the performance of a classifier. In general, this metric is used for imbalanced data.

3.4 Log loss

Log loss also called logarithm loss is the most commonly technique to measure the error predicted by classification model. This function allows us to know how good a prediction model is. Log loss is very important in a classification problem. The goal of this loss function in neural network is to reduce the value into negligible value (i.e near to zero or 0). Hence, we have a good model if the log loss decrease to zero, whereas this loss increases as we have a bad model. In binary classification, the log loss is defined as follows:

$$Loss(y) = -\frac{1}{N}\sum_{i=1}^{N} y_i \log(p(y_i)) + (1 - y_i)\log(1 - p(y_i)), \tag{3.4.1}$$

where

- y is class label (0 for first class and 1 for second class).
- p(y) indicate the predicted probability of first class for all N points.
- $1 - p(y)$ indicate the predicted probability of second class for all N points.

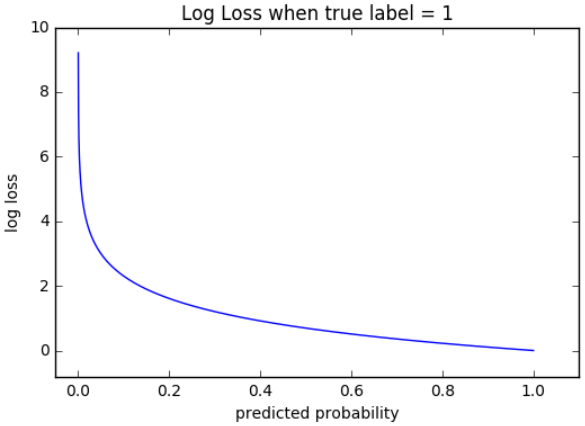

Figure 3.7: Binary cross entropy or log loss

That's all about the chapter 3, in the next chapter, we will see the application and result in classification astronomical objects more precisely classification of type Ia supernovae and non type Ia supernovae by using binary classification.

4. Application in Type Ia Supernovae.

We explain how to classify type Ia supernovae In photometric and spectroscopic surveys. In this chapter, we talk about the classification of Type Ia supernova from the Supernova Photometric Classification Challenge data (SPCC). We apply the above method to classify type Ia supernova. To accomplish this task, we catch first the description of SPCC catalog.

4.1 Description of the SPCC catalog.

In this subsection, we talk about the description of SPCC data using python language. First, we create a library as a dictionary for supernova type as shown in the table 4.1:

SN-type	Interger code
Ia	1
II (IIn, IIP, IIL)	2 (21, 22, 23)
Ibc (Ib, Ic)	3 (32, 33)
other	66
rejected	-1

Table 4.1: Integer code of Types of supernovae

And the dictionary of the passbands (griz filter) (see table 4.2)

Colors (filter)	code
green	g
red	r
blue	i
black	z

Table 4.2: Filter in SPCC catalog

After our implementation, we describe the distribution of the objects in the data set (see the table 4.3):

Type of astronomical objects	Number
Ia	879
II	0
IIn	44
IIP	218
IIL	2
Ibc	13
Ib	55
Ic	45
NO_SPEC_CONF	17065

Table 4.3: Number of astronomical objects per type

.

From table 4.3, we can see that the SPCC data contains almost 900 (more precisely 879) type Ia supernovae, while there are 17065 objects no spectroscopically confirmed. On the other hand, there are 1256 spectroscopically confirmed objects in this data. In any case, the true identity of the objects is praised into the file DES UNBLIND+HOSTZ.KEY, which contains of 19761 objects.

The following figure 4.1 represents the light curves of 4 random variable is chosen by randomly the four evolution of light curves through time pointed on the telescope to describe the nature of data set.

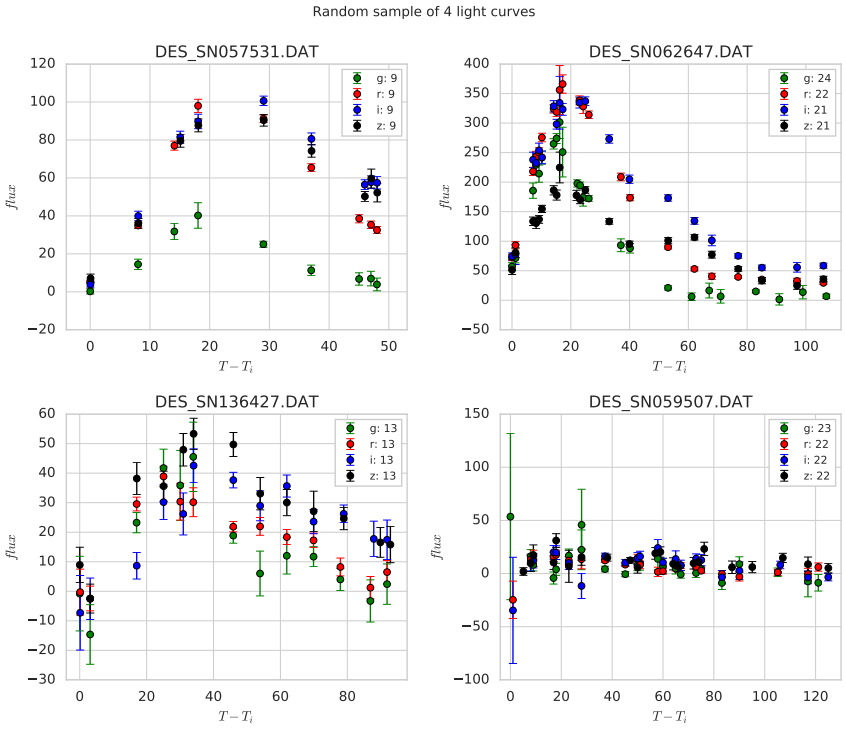

Figure 4.1: Information of random sample of four light curves from SPCC data

4.2 Distribution of SPCC data.

In this section, we talk about the distribution of data through the spectroscopic sample and photometric sample including the distribution of SNeIa or non SNeIa in both samples. There are 1145 object in te spectroscopic sample and 16383 objects in photometric sample. Where the size of the photometric sample. The distribution in redshift in fugure 4.2.

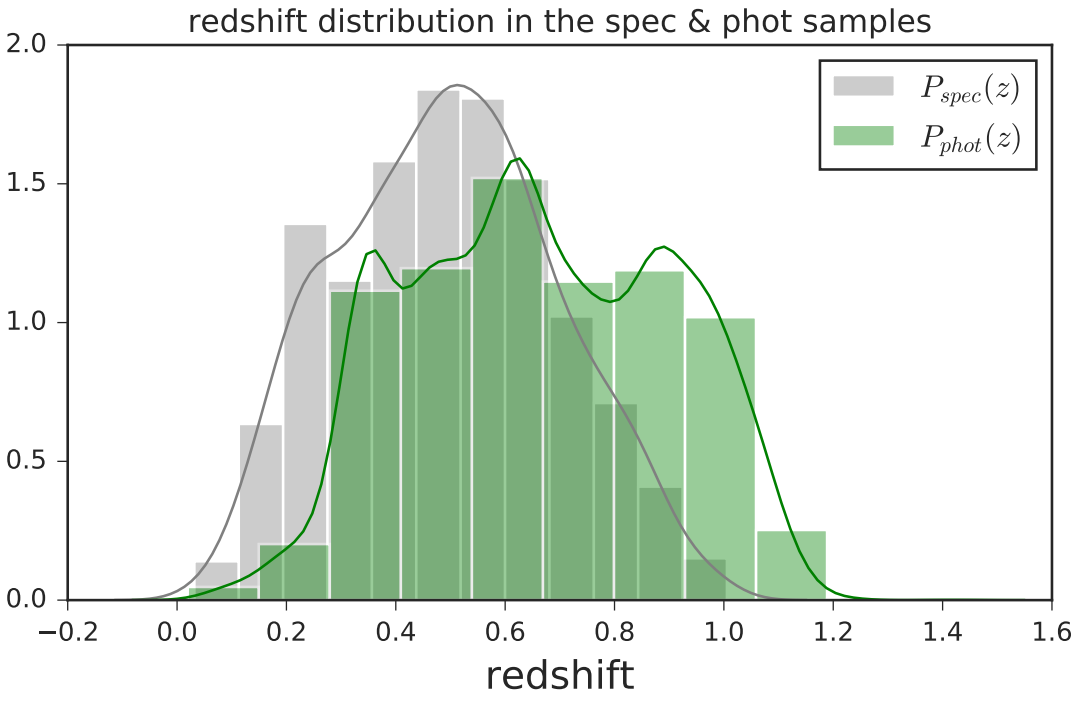

Figure 4.2: Distribution of redshift in the spectroscopic and photometric sample

From figure 4.2, we can describe that the photometric sample is shifted to larger redshifts.

4.2.1 Prior distribution.

We see the probability corresponding to the existence of Type Ia supernova in the spectroscopic sample is around 0.711, whereas in the photometric sample the probability is 0.289. That means that spectroscopic sample, there is a lot of the SNaIa (almost $70\,\%$ of data), whereas only $30\,\%$ of non-type Ia supernova objects and in the photometric sample, the non-type Ia supernovae (non-SNeIa) are dominant with almost $74\,\%$ of the sample, whereas 26 percent correspond to SNeIa.

Numerically, in the spectroscopic sample:

$$\begin{cases} P_{spec}(y=1) = 0.711 \\ P_{spec}(y=0) = 1 - P_{spec}(y=1) = 0.289 \end{cases} \tag{4.2.1}$$

and in photometric sample,

$$\begin{cases} P_{phot}(y=1) = 0.259 \\ P_{phot}(y=0) = 1 - P_{phot}(y=1) = 0.741 \end{cases} \tag{4.2.2}$$

Where $y = 1$ indicate SNeIa class,whereas $y = 0$ the non SNeIa class. Figure 4.3 shows the redshift distributions in each sample.

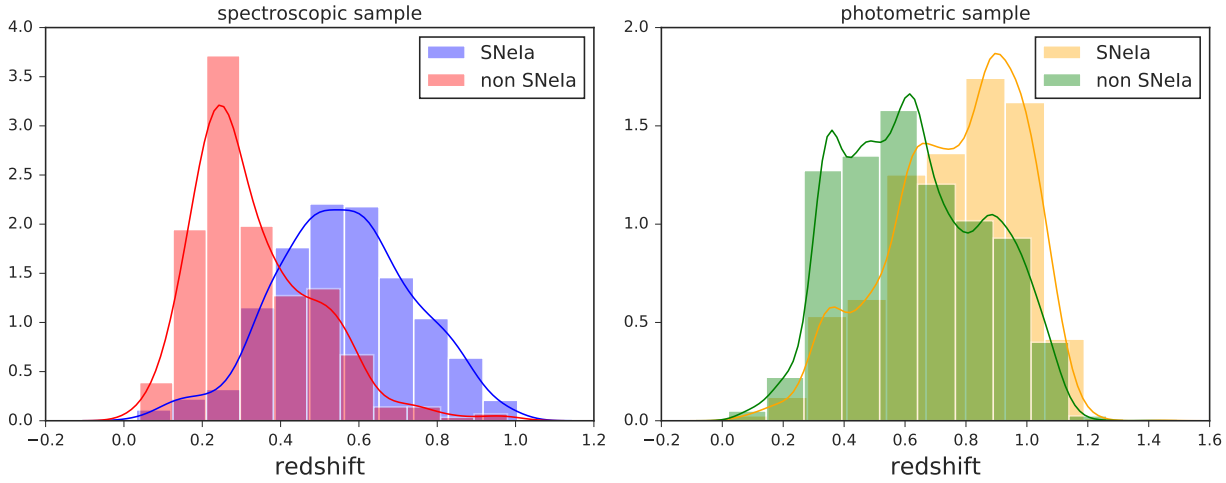

Figure 4.3: Redshift distribution SNela and non-SNela into the spectroscopic sample (in the left) and into the photometric sample (on the right)

As we see in the figure (4.3), in the both samples, SNela extend to a larger redshift.

4.2.2 Conditional distribution.

In this subsection, we present the redshift distribution for each type in bother sample see figure (4.4)

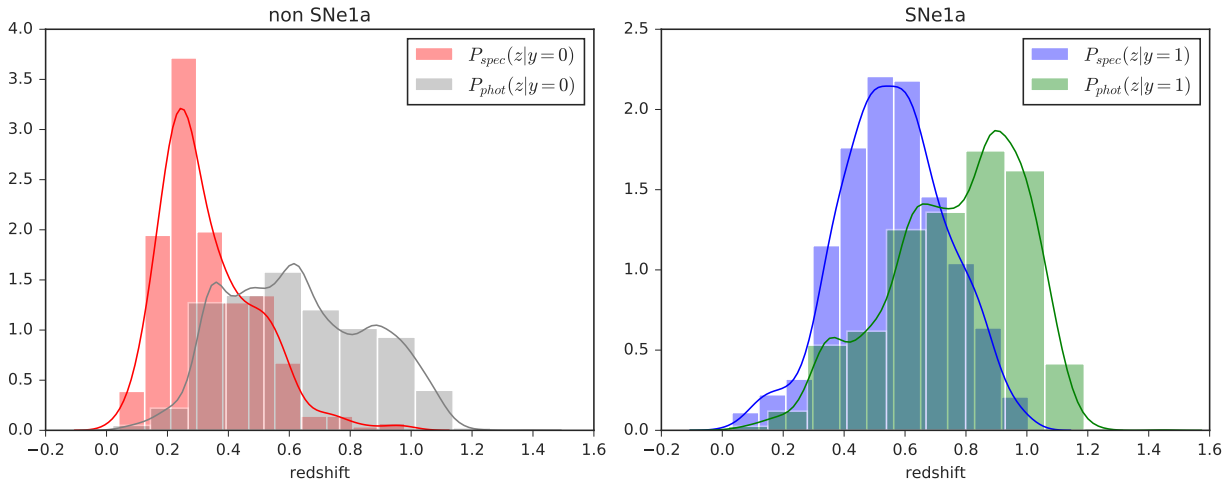

Figure 4.4: redshift distribution from photometric sample and spectroscopic sample for SNela and non SNela

The photometric sample extends to a larger redshift than the spectroscopic sample both for SNela and non-SNela.

4.2.3 Gaussian process interpolation.

In this subsection, we took randomly one sub-data from SPCC data (see figure (4.5)).

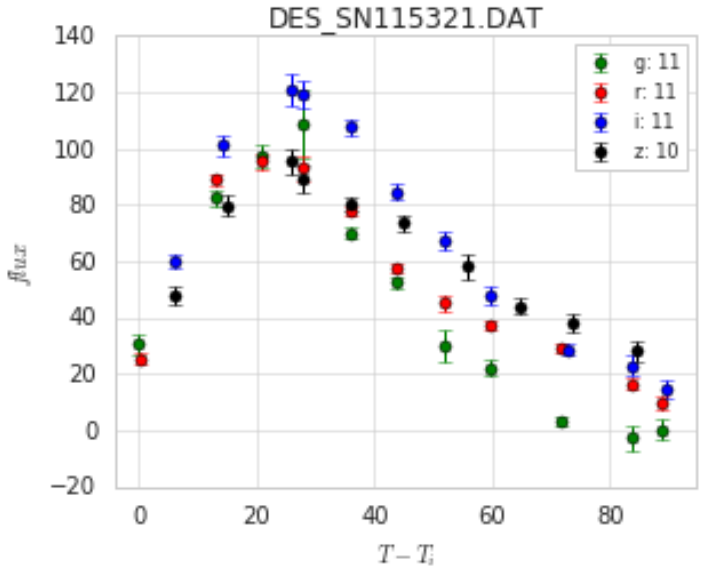

Figure 4.5: High light curve measurements through times $T - T_i$

This figure indicates the evolution through time of the griz filters . As we see the distribution of observational all points is pretty irregular in time. To make an interpolation on regular distribution, we use the Gaussian process interpolation. So the following figure 4.6 shows the result using the library sklearn.gaussian_process.

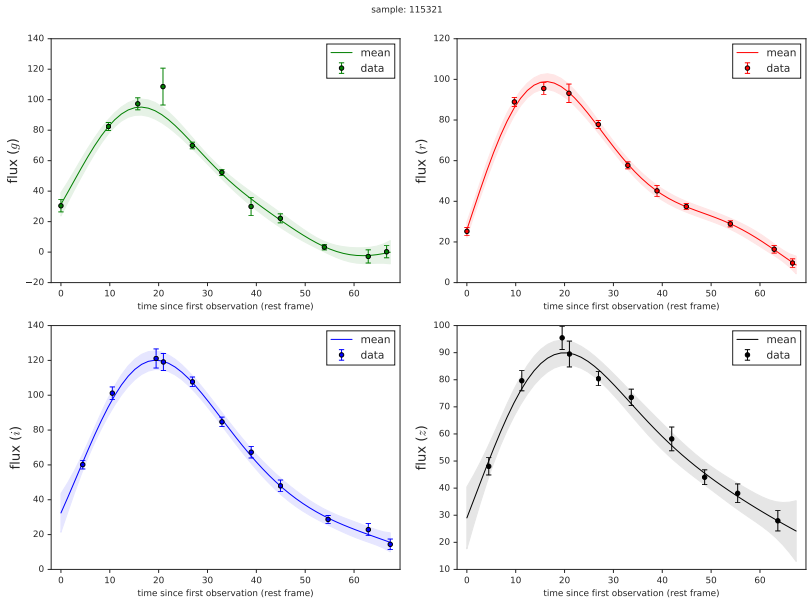

Figure 4.6: Random sample of one light curve

This result on 4.6 in the particular data sample DES_SN115321.DAT represents the evolution of flux from explosion through time since first observation.

The solid line is the mean bands represent the ine sign definition the the mean

4.3 Training and test SPCC data.

In this section, we talk about the classification of the type Ia supernovae and the non type Ia supernovae using RNNs (more precisely LSTM) to predict it the class. So in this case, we use binary classification with non SNeIa the negative class and SNeIa the positive class.

Here, they are 50 time steps. For of each time step the input has a nine-component vector containing the infrastructure at t and the interpolated flux and error for the four passbands. At the last step of the network, the classifier computes the probability prediction. When the probability of prediction is greater than 0.5, then the model predicts a SNeIa, when the probability of prediction is less than 0.5, then the model predicts a non-SNeIa . Therefore,

$$\begin{cases} \text{if } p > 0.5, \text{ the model predicts SNeIa,} \\ \text{and } p < 0.5, \text{ the model predicts non-SNeIa.} \end{cases}$$

So our data set After running our classifier 50 times we get the confusion matrix mean. :

$$CM = \begin{pmatrix} 6652.34 \pm 869.9347702 & 5485.66 \pm 869.9347702 \\ 217.34 \pm 112.56369042 & 4027.66 \pm 112.56369042 \end{pmatrix} \tag{4.3.1}$$

The errors from the relation (4.3.1) indicate the standard deviation of confusion matrix. And the area under curve of ROC curve is 0.837 ± 0.027.

4.3.1 Plot of Accuracy and Loss.

As we have discussed in previous chapter that the accuracy is a metric how well a classifier make predictions. In this case, we verify the performance of the model using as a training set the spectroscopic sample and as a test set the photometric sample. The plots 4.7 show the accuracy and loss function.

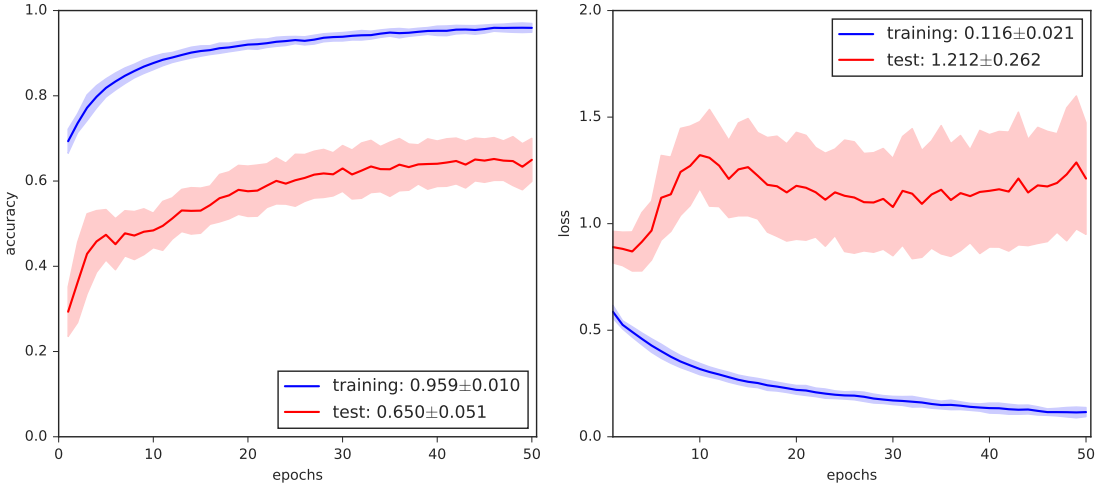

Figure 4.7: Accuracy and loss

As we see in the plot of the 4.7, the performance of training set is better than testing set. Because, for the training, when the number of epoch is increasing, the accuracy is also increasing and tend to 1.0 on the left panel. In this case we have 0.959 ± 0.010 for accuracy training. And on the right plot, we see also that when the number of epoch is increasing, the loss decreases for the training set (the blue curve). The performance of the training set in this case is good. Whereas for test set (the red curve), the performance is poor. That means, the performance of test set is poor because the spectroscopic sample (training) is different than the photometric sample (test set).

4.3.2 Plot ROC curve and Precision-Recall curve.

In plot belong (see figure 4.8), we show ROC curve and the precision-recall curve for the model. So before interpretation of its plot,remember, the performance of the classifier is better when the area under curve is close to 1 (i.e, if the true positive rate converge high when the false positive rate goes to 1).

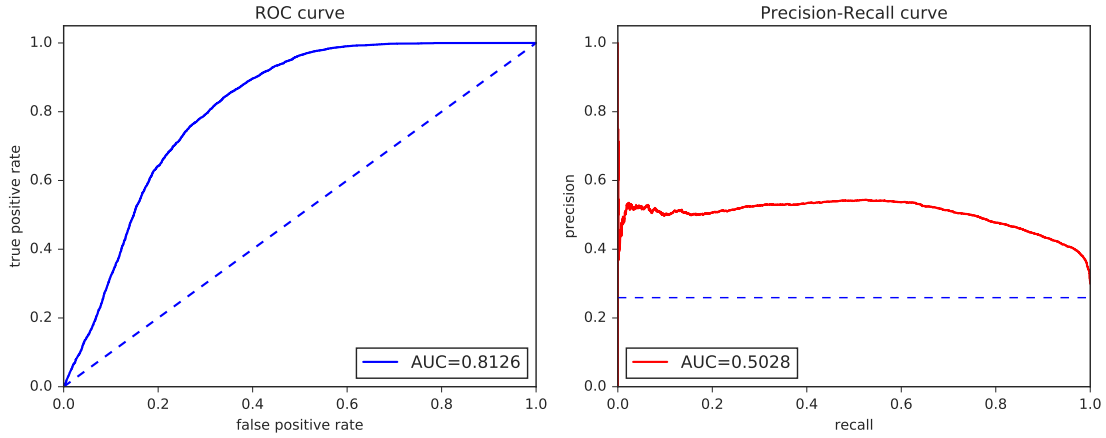

Figure 4.8: Roc curve and Precision-Recall curve

As we can see on the left plot, the AUC is equals to 0.8126. However, we need to improve it.

4.3.3 Plot of Confusion matrix.

In this subsection, we give the prediction values of type Ia supernova and non type Ia supernova. In this case, the four outputs in the table 3.2 (TP, FP, FN, TN) are represented as follows:

- TP: is the number of correctly classified SNe1a
- FP: is the number of non-SNe1a misclassified as SNe1a
- TN: is the number of correctly classified non-SNe1a
- FN: is the number of SNe1a misclassified as non-SNe1a

In consequence, in the figure 4.9 on the top right of the confusion matrix, we have the false positive values and this value is a large amount missclassifications (where FP = 7109 values). This means that there is a large fraction of non-type Ia supernovae that are missclassifed as SNeIa.

In order to improve our classifier, we need to develop some algorithms that can classify more accurately

objects of the photometric sample from the spectroscopic datasets. We leave this task for the future work.

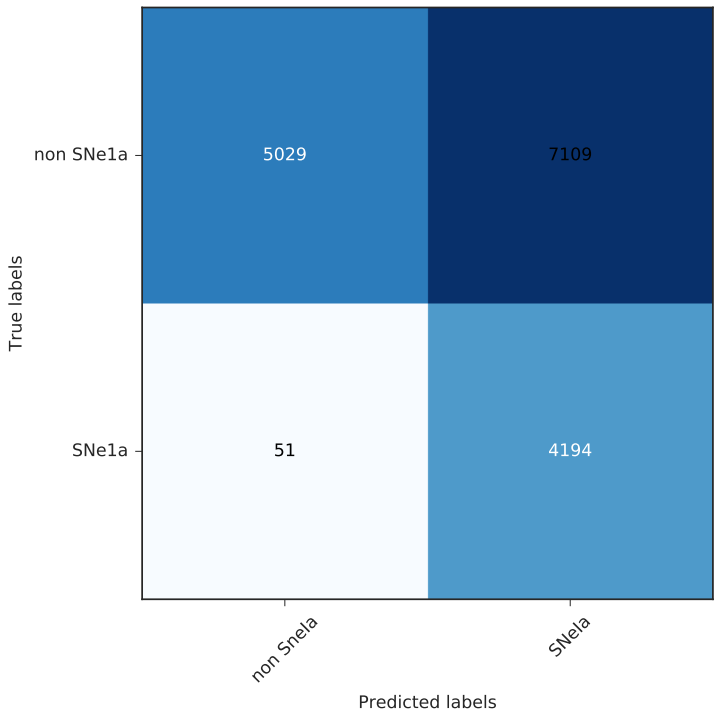

Figure 4.9: Confusion matrix for predicting SNIa and non-SNIa

5. Conclusions

Type Ia supernovae are used to study the nature of dark energy. The amount of spectroscopically confirmed type Ia supernovae is small. There won't be enough spectroscopical resources for identifying objects in future surveys. However, photometric techniques are less expensive than spectroscopy. In this essay, we discussed the problem of classifying type Ia supernovae using photometric and spectroscopic samples. Here we focused on the implementation of a recurrent neural network algorithm to classify these objects in the SPCC catalog. We trained the algorithm using the spectroscopic sample and tested it using the photometric sample. The classifier model performed well on the spectroscopic sample but on the photometric sample it performed poorly. We obtained a 96 $\%$ accuracy on the training data and 65 $\%$ on the test data. The loss function increased on the test set. We used the confusion matrix to know the performance of the classifier. The results in the confusion matrix indicate that there was a large amount of false positives. We need to improve the classifier to reduce the errors.

Acknowledgments

My sincere gratitude goes to the African Institute of Mathematics (AIMS) for the post graduate scholarship that enabled me to undertake this masters program. Generally, I will like to appreciate the AIMS family for making my experience in South Africa worthwhile. I gladly admit that it was the best of times, and there is no doubt that this will form part of my most cherished moments.

The completion of this essay was made possible due to the academic guidance of Prof. Bruce Bassett and Dr. Erick Jonathan Almaraz Avi$\tilde{n}a$. I sincerely appreciate your time, patience, willingness to teach and principles of prompt response that eased the burden of this essay phase.

To my lovely Mum, I am dedicating this essay to you, in memory of the beautiful time you spent teaching me invaluable life principles. And my four brothers I love you all.

Finally, I give glory to the Almighty God for his mercy, grace and strength in time of need.

References

A medium corporation (USA). Typesetting neural network diagrams with tex. , https://medium.com/momenton/typesetting-neural-network-diagrams-with-tex-4920b6b9fc19, Accessed May 2020.

E. J. A. Aviña. *Cosmological Implications observational constraints on Dark energy as a scalar field condensate whith and inverse power lqw potential*. Phd, Universidad Nacional Automa de Mexico, 2018.

D. Baron. Mahine learning in astronomy: a parctical overview. *Arxiv*, 2019.

K. Boone. Avocado: Photometric classification of astronomical transients with gaussian process augmentation. 2019.

J. Brownlee. *Probability for Machine Learning, Discover How to Harness Uncertainty With Python*. Unpublished manuscript, 2020.

T. Charnock and A. Moss. Deep recurrent neural networks for supernovae classification. *The American Astronomical Society*, 2019.

Data Science and Machine Learning. Confusion matrix. , https://manisha-sirsat.blogspot.com/2019/04/confusion-matrix.html, Accessed May 2020.

D. A. Graves. *Supervised Sequence labelling with Recurrent Neural Networks*. Springer-Verlag, 2007.

R. Kessler, B. Bassett, P. Belov, V. Bhatnagar, H. Campbel, A. Conley, J. A. Frieman, A. Glazov, S. González-Gaitán, R. Hlozek, S. Jha, S. Kuhlmann, M. Kunz, H. Lampeitl, A. Mahabal, J. Newling, R. C. Nichol, D. Parkinson, N. S. Philip, D. Poznanski, J. W. Richards, S. A. Rodney, M. Sako, D. P. Schneider, M. Smith, M. Stritzinger, and M. Varughese. Results from the supernova photometric classification challenge. *The Astronomical Society of the Pacific, USA*, 2010.

D. A. Mazure and D. S. Basa. *Exploding Superstars understanding Supernovae and Gama-Ray Bursts*. Editions Dunod, Paris, 2007.

A. Möller and T. de Boissière. Supernnova: an open-source framework for bayesian, neural network based supernova classification. *Monthly Notices of the Royal Astronomical Society*, 2019.

C. E. Nwankpa, W. Ijomah, A. Gachagan, and S. Marshall. Activation functions: Comparison of trends in practice and research for deep learning. *Arxiv*, 2018.

P. Schneider. *Extragalactic Astronomy and Cosmology*. Springer-Verlag Berlin Heidelberg 2006, 2015.

D. I. F. Suny. *Astrophysics is Easy!, An Introduction for the Amateur Astronomer*. Springer-Verlag, 2007.

Wikipedia. Cosmological constant. Wikipedia, the Free Encyclopedia, https://en.wikipedia.org/wiki/Cosmological_constant, Accessed may 2020a.

Wikipedia. Activation function. Wikipedia, the Free Encyclopedia, https://en.wikipedia.org/wiki/Activation_function, Accessed may 2020b.

Wikipedia. Kriging. Wikipedia, the Free Encyclopedia, https://en.wikipedia.org/wiki/Kriging, Accessed may 2020c.

Wikipedia. Subaru telescope. Wikipedia, the Free Encyclopedia, https://en.wikipedia.org/wiki/Subaru_Telescope, Accessed may 2020d.