**“MAJOR PROJECT TITLE”**

**Major Project Report**

*Submitted in Partial Fulfillment of the Requirements for the Degree of*

**BACHELOR OF TECHNOLOGY**

**IN**

**ELECTRICAL ENGINEERING**

By

Name of Student:

(Roll No.)



**Department of Electrical Engineering**

**Institute of Technology**

**NIRMA UNIVERSITY**

**Ahmedabad 382 481**

**May 2018**

**CERTIFICATE**

This is to certify that the Major Project Report entitled

“\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_”submitted by Mr./Ms. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_) towards the partial fulfillment of the requirements for the award of degree in Bachelor of Technology in the field of Electrical Engineering of Nirma University is the record of work carried out by him/her under our supervision and guidance. The work submitted has in our opinion reached a level required for being accepted for examination. The results embodied in this major project work to the best of our knowledge have not been submitted to any other University or Institution for award of any degree or diploma.

**Date:**

**Industry - Guide** **Institute - Guide**

|  |  |
| --- | --- |
| Name of Guide | Name of Guide |
| Designation | Designation |
| Address | Address |

|  |  |
| --- | --- |
| **Head of Department** | **Director** |
| Department of Electrical Engineering | Institute of Technology |
| Institute of Technology | Nirma University |
| Nirma University | Ahmedabad |
| Ahmedabad |  |

|  |  |
| --- | --- |
|  | CONTENTS |
| **Acknowledgement** | **(i)** | |
| **Abstract** | **(ii)** | |
| **List of Figures/Captions**  Output of Linear Regression  Output of Polynomial Regression  Output of Sentiment Analysis | **(iii)** | |
| **List of Tables**  Data for Bird Species Classification  Dataset for Polynomial Regression | **(iv)** | |
|  | |
| **Contents** | **(iv)** | |
| **Chapter 1: Basics of Machine Learning**  **1.1 Terminologies** | **4** | |
| **1.2 Linear and Polynomial regression using Python** | **5** | |
| **Chapter 2: Sentimental Analysis of Tweeter** |  |
| **2.1 Code** |  |
| **Chapter 3: Different Packages of Python** |  |
| **3.1 Numpy**  **3.2 Pandas**  **3.3 RE (regular expression)**  **3.4 Tweepy**  **3.5 Textblob** |  |
| **Chapter 4: MxNet** |  |
| **4.1 jkl** |  |

**Chapter 1: Basics of Machine Learning**

Machine learning is turning data into information. It uses statistics. With so much of the economic activity dependent on information, you can’t afford to be lost in the data. Machine learning will help you get through all the data and extract some information.

**1.1 Key terminology**

Example of building a bird classification system

We chose to measure weight, wingspan, whether it has webbed feet, and the color of its back.

The Four things we’ve measured are called features also called as attributes. Each of the rows in table 1.1 is an instance made up of features.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Weight(g) | Wingspan(cm) | Webbed Feet | Back Color | Species |
| 1 | 1000.1 | 125.0 | No | Brown | Buteo jamaicensis |
| 2 | 3000.7 | 200.0 | No | Gray | Sagittarius serpentarius |
| 3 | 3300.0 | 220.3 | No | Gray | Sagittarius serpent Arius |
| 4 | 4100.0 | 136.0 | Yes | Black | Gavia immer |
| 5 | 3.0 | 11.0 | No | Green | Calothorax Lucifer |
| 6 | 570.0 | 75.0 | No | Black | Campephilus principalis |

Table 2: Data for Bird Species Classification

How do we then decide if a bird at our feeder is an Ivory-billed Woodpecker or something else? This task is called classification, and there are many machine learning algorithms that are good at classification.

To train the algorithm we feed it quality data known as a training set. A training set is the set of training examples. We’ll use to train our machine learning algorithms.

In table 1.1 our training set has six training examples. Each training example has four features and one target variable; this is depicted in figure 1.2. The target variable is what we’ll be trying to predict with our machine learning algorithms.

In classification the target variable takes on a nominal value, and in the task of regression its value could be continuous. The machine learns by finding some relationship between the features and the target variable.

To test machine learning algorithms what’s usually done is to have a training set of data and a separate dataset, called a test set. The test set is fed to the program. The target variable for each example from the test set isn’t given to the program, and the program decides which class each example should belong to. The target variable or class that the training example belongs to is then compared to the predicted value, and we can get a sense for how accurate the algorithm is. Classification and regression are examples of supervised learning. This set of problems is known as supervised because we’re telling the algorithm what to predict.

The opposite of supervised learning is a set of tasks known as unsupervised learning. In unsupervised learning, there’s no label or target value given for the data.

**1.2 Linear and Polynomial regression using Python**

**Linear regression**

Regression was invented by the cousin of Charles Darwin, Francis Galton. Galton did his first regression in 1877 to estimate the size of pea seeds based on the size of their parents’ seeds.

Our goal when using regression is to predict a numeric target value. Linear regression means you can add up the inputs multiplied by some constants to get the output.

There’s another type of regression called nonlinear regression in which this isn’t true the output may be a function of the inputs multiplied together.

General approach to regression

1. Collect: Any method.

2. Prepare: We’ll need numeric values for regression. Nominal values should be mapped to binary values.

3. Analyze: It’s helpful to visualized 2D plots. Also, we can visualize the regression weights if we apply shrinkage methods.

4. Train: Find the regression weights.

5. Test: We can measure the R2, or correlation of the predicted value and data, to measure the success of our models.

6. Use: With regression, we can forecast a numeric value for a number of inputs. This is an improvement over classification because we’re predicting a continuous value rather than a discrete category.

Our input data is in the matrix X, and our regression weights in the vector w. For a given piece of data X1 our predicted value is given by. We have the xs and ys, but how can we find the ws? One way is to find the ws that minimize the error. We define error as the difference between predicted y and the actual y. Using just the error will allow positive and negative values to cancel out, so we use the squared error.

**Code:**

import numpy  
import matplotlib.pyplot as plot  
import pandas  
from sklearn.model\_selection import train\_test\_split  
from sklearn.linear\_model import LinearRegression

# Import the dataset  
dataset = pandas.read\_csv('salaryData.csv')  
x = dataset.iloc[:, :-1].values  
y = dataset.iloc[:, 1].values  
  
# Split the dataset into the training set and test set  
# We're splitting the data in 1/3, so out of 30 rows, 20 rows will go into the training set,  
# and 10 rows will go into the testing set.  
xTrain, xTest, yTrain, yTest = train\_test\_split(x, y, test\_size = 1/3, random\_state = 0)

linearRegressor = LinearRegression()

yPrediction = linearRegressor.predict(xTest)

plot.scatter(xTrain, yTrain, color = 'red')  
plot.plot(xTrain, linearRegressor.predict(xTrain), color = 'blue')  
plot.title('Salary vs Experience (Training set)')  
plot.xlabel('Years of Experience')  
plot.ylabel('Salary')  
plot.show()

plot.scatter(xTest, yTest, color = 'red')  
plot.plot(xTrain, linearRegressor.predict(xTrain), color = 'blue')  
plot.title('Salary vs Experience (Test set)')  
plot.xlabel('Years of Experience')  
plot.ylabel('Salary')  
plot.show()



Figure 1: Output of Linear Regression

Explanation:

Libraries used

* matplotlib.pyplot- plot a variety of graphs and charts
* numpy - NumPy stands for Numerical Python.it is a library consisting of multidimensional array objects and a collection of routines for processing those arrays.
* Pandas- Pandas is an open-source Python Library providing high-performance data manipulation and analysis tool
  + read\_csv()- to import our dataset using pandas
* sklearn- These libraries make lives easier by just exposing APIs to us so that we can call those APIs and get our results.

1. We separate the features and the dependent variable using x = dataset.iloc[:, :-1].values  
   y = dataset.iloc[:, 1].values
2. we can split our data into training and test sets using train\_test\_split method
3. we create our object using linearRegressor = LinearRegression()
4. we make the model learn by feeding it with training data linearRegressor.fit(xTrain, yTrain)
5. to test your model by feeding it with test data yPrediction = linearRegressor.predict(xTest)

**Polynomial regression**

Model for a single predictor, *X*, is:

Y=β0+β1X+β2X2+…+βhX^h+ϵ  
where h is called the **degree** of the polynomial.

Although this model allows for a nonlinear relationship between Y and X, polynomial regression is still considered linear regression since it is linear in the regression coefficients, β1,β2,...,βhβ1,β2,...,βh

Code:

import matplotlib.pyplot as plt

import numpy as np

from sklearn.linear\_model import Ridge

from sklearn.preprocessing import PolynomialFeatures

from sklearn.pipeline import make\_pipeline

y\_train=[[45000],[50000],[60000],[80000],[110000],[150000],[200000],[300000],[500000],[1000000]]

x\_plot = np.linspace(0,len(y\_train), len(y\_train))#returns evenly spaced numbers array between start=0, stop=28, number needed=26

plt.scatter(x\_plot, y\_train, color='navy', s=30, marker='o', label="training points")

colors = ['teal', 'yellowgreen', 'gold', 'red','green','violet','grey']

x\_plot = x\_plot.reshape(-1,1)

for count, degree in enumerate([1, 2, 3]):

model = make\_pipeline(PolynomialFeatures(degree), Ridge())

model.fit(x\_plot, y\_train)

y\_plot = model.predict(x\_plot)

plt.plot(x\_plot, y\_plot, color=colors[count], linewidth=2,

label="degree %d" % degree)

plt.legend(loc='lower right')

plt.show()

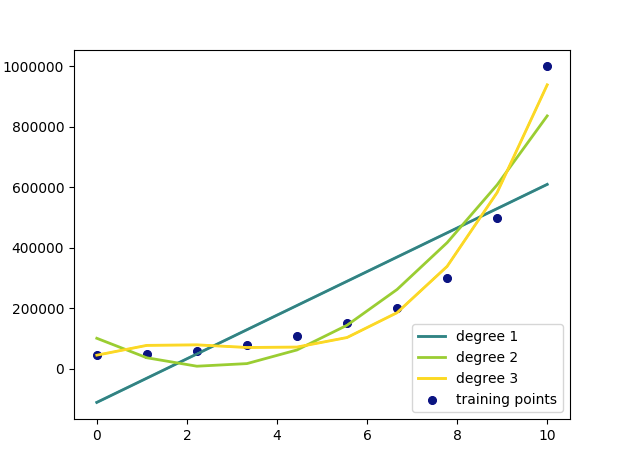


Figure 2: Output of Polynomial Regression



Table 2: Dataset for Polynomial Regression

**Chapter 2: Sentimental Analysis of Tweeter**

Sentiment Analysis is the process of ‘computationally’ determining whether a piece of writing is positive, negative or neutral. It’s also known as **opinion mining**, deriving the opinion or attitude of a speaker.

In marketing field companies use it to develop their strategies, to understand customers’ feelings towards products or brand, how people respond to their campaigns or product launches and why consumers don’t buy some products.

In political field, it is used to keep track of political view, to detect consistency and inconsistency between statements and actions at the government level. It can be used to predict election results as well.

Sentiment analysis also is used to monitor and analyse social phenomena, for the spotting of potentially dangerous situations and determining the general mood of the blogosphere

**2.1 Code:**

import re

import tweepy

from tweepy import OAuthHandler

from textblob import TextBlob

class TwitterClient(object):

'''

Generic Twitter Class for sentiment analysis.

'''

def \_\_init\_\_(self):

'''

Class constructor or initialization method.

'''

# keys and tokens from the Twitter Dev Console

consumer\_key = 'XXXXXXXXXXXXXXXXXXXXXXXX'

consumer\_secret = 'XXXXXXXXXXXXXXXXXXXXXXXXXXXX'

access\_token = 'XXXXXXXXXXXXXXXXXXXXXXXXXXXX'

access\_token\_secret = 'XXXXXXXXXXXXXXXXXXXXXXXXX'

# attempt authentication

try:

# create OAuthHandler object

self.auth = OAuthHandler(consumer\_key, consumer\_secret)

# set access token and secret

self.auth.set\_access\_token(access\_token, access\_token\_secret)

# create tweepy API object to fetch tweets

self.api = tweepy.API(self.auth)

except:

print("Error: Authentication Failed")

def clean\_tweet(self, tweet):

'''

Utility function to clean tweet text by removing links, special characters

using simple regex statements.

'''

return ' '.join(re.sub("(@[A-Za-z0-9]+)|([^0-9A-Za-z \t])

|(\w+:\/\/\S+)", " ", tweet).split())

def get\_tweet\_sentiment(self, tweet):

'''

Utility function to classify sentiment of passed tweet

using textblob's sentiment method

'''

# create TextBlob object of passed tweet text

analysis = TextBlob(self.clean\_tweet(tweet))

# set sentiment

if analysis.sentiment.polarity > 0:

return 'positive'

elif analysis.sentiment.polarity == 0:

return 'neutral'

else:

return 'negative'

def get\_tweets(self, query, count = 10):

'''

Main function to fetch tweets and parse them.

'''

# empty list to store parsed tweets

tweets = []

try:

# call twitter api to fetch tweets

fetched\_tweets = self.api.search(q = query, count = count)

# parsing tweets one by one

for tweet in fetched\_tweets:

# empty dictionary to store required params of a tweet

parsed\_tweet = {}

# saving text of tweet

parsed\_tweet['text'] = tweet.text

# saving sentiment of tweet

parsed\_tweet['sentiment'] = self.get\_tweet\_sentiment(tweet.text)

# appending parsed tweet to tweets list

if tweet.retweet\_count > 0:

# if tweet has retweets, ensure that it is appended only once

if parsed\_tweet not in tweets:

tweets.append(parsed\_tweet)

else:

tweets.append(parsed\_tweet)

# return parsed tweets

return tweets

except tweepy.TweepError as e:

# print error (if any)

print("Error : " + str(e))

def main():

# creating object of TwitterClient Class

api = TwitterClient()

# calling function to get tweets

tweets = api.get\_tweets(query = 'Donald Trump', count = 200)

# picking positive tweets from tweets

ptweets = [tweet for tweet in tweets if tweet['sentiment'] == 'positive']

# percentage of positive tweets

print("Positive tweets percentage: {} %".format(100\*len(ptweets)/len(tweets)))

# picking negative tweets from tweets

ntweets = [tweet for tweet in tweets if tweet['sentiment'] == 'negative']

# percentage of negative tweets

print("Negative tweets percentage: {} %".format(100\*len(ntweets)/len(tweets)))

# percentage of neutral tweets

print("Neutral tweets percentage: {} % \

".format(100\*len(tweets - ntweets - ptweets)/len(tweets)))

# printing first 5 positive tweets

print("\n\nPositive tweets:")

for tweet in ptweets[:10]:

print(tweet['text'])

# printing first 5 negative tweets

print("\n\nNegative tweets:")

for tweet in ntweets[:10]:

print(tweet['text'])

if \_\_name\_\_ == "\_\_main\_\_":

# calling main function

main()

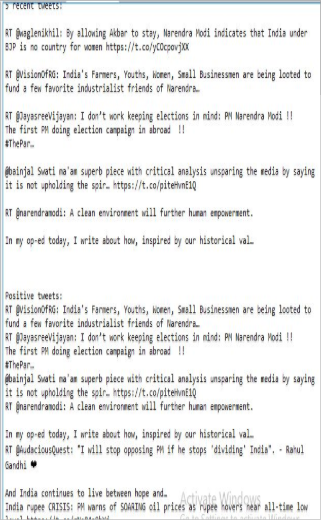


Figure 2: output of Sentiment Analysis

   We follow these 3 major steps in our program:

* Authorize twitter API client.
* Make a GET request to Twitter API to fetch tweets for a particular query.
* Parse the tweets. Classify each tweet as positive, negative or neutral.

**Chapter 3: Different Packages of Python**

**3.1 Numpy**

* Numerical Python
* Consists of multidimensional arrays
* Also has routines that process the data
* All operations like mathematical, logical, Fourier transforms, linear algebra
* Together with Matplotlib and Scipy it provides all functions of matlab
* np.array is used to create an array object
* The most important object defined in NumPy is an N-dimensional array type called **ndarray.**  It describes the collection of items of the same type. Items in the collection can be accessed using a zero-based index.
* Every item in an ndarray takes the same size of block in the memory. Each element in ndarray is an object of data-type object (called **dtype**)
* The basic ndarray is created using an array function in NumPy - numpy.array

**3.2 Pandas**

* Pandas is library used for data analysis and modelling
* DataFrame is a way to represent and work with tabular data
* Data can be read in DataFrame from various formats likecsv,Microsoft Excel,SQL database and fast HDF5 format
* The advantage of using pandas with Numpy is that it allows us to have columns of different data type
* Using iloc function we can retrieve subsets of dataset
* We can add labels to columns and rows using loc function
* Fast and efficient DataFrame object with default and customized indexing.
* Tools for loading data into in-memory data objects from different file formats.
* Data alignment and integrated handling of missing data.
* Reshaping and pivoting of date sets.
* Label-based slicing, indexing and subsetting of large data sets.
* Columns from a data structure can be deleted or inserted.
* Group by data for aggregation and transformations.
* High performance merging and joining of data.
* Time Series functionality.

**3.3 RE (Regular Expression)**

* It’s a sequence of characters used to find and replace pattern in a string
* Module re in Python is used to search for string, break into a substring, replace string re.match(),re.search(),re.findall(),re.split(),re.sub(),re.compile() are some methods used
* re.match is used to match pattern if it occurs at the start of string .re.search does not restrict to start condition.
* re.split is used to split string by some pattern
* re.sub is used to replace a word in a string with a another word.
* You can easily tackle many basic patterns in Python using the ordinary characters. Ordinary characters are the simplest regular expressions. They match themselves exactly and do not have a special meaning in their regular expression syntax. Examples are 'A', 'a', 'X', '5'.
* The match() function returns a match object if the text matches the pattern. Otherwise it returns None
* Special characters are characters which do not match themselves as seen but actually have a special meaning when used in a regular expression
  + . - A period. Matches any single character except newline character. re.search(r'Co.k.e', 'Cookie').group()
  + \w - Lowercase w. Matches any single letter, digit or underscore. re.search(r'Co\wk\we', 'Cookie').group()
* **Repetitions**
  + + - Checks for one or more characters to its left.

re.search(r'Co+kie', 'Cooookie').group() gives 'Cooookie'

* + \* - Checks for zero or more characters to its left

re.search(r'Ca\*o\*kie', 'Caokie').group() gives 'Caokie'

* Parts of a regular expression pattern bounded by parenthesis() are called groups. The parenthesis does not change what the expression matches, but rather forms groups within the matched sequence.

**3.4 Tweepy**

* Tweepy supports accessing Twitter via Basic Authentication and the newer method, OAuth. Twitter has stopped accepting Basic Authentication so OAuth is now the only way to use the Twitter API.
* Using OAuth has these advantages:
  + Tweets can be customized to have a string which identifies the app which was used.
  + It doesn’t reveal user password, making it more secure.
  + It's easier to manage the permissions, for example a set of tokens and keys can be generated that only allows reading from the timelines, so in case someone obtains those credentials, he/she won’t be able to write or send direct messages, minimizing the risk.
  + The application doesn't reply on a password, so even if the user changes it, the application will still work
* Tweepy provides access to the well documented Twitter API.
* With tweepy, it's possible to get any object and use any method that the official Twitter API offers.
* One of the main usage cases of tweepy is monitoring for tweets and doing actions when some event happens. Key component of that is the StreamListener object, which monitors tweets in real time and catches them.

**3.5 Textblob**

* TextBlob is a python library and offers a simple API to access its methods and perform basic NLP tasks.
* Tokenization refers to dividing text or a sentence into a sequence of tokens, which roughly correspond to “words”. This is one of the basic tasks of NLP. To do this using TextBlob, follow the two steps:
  + Create a **textblob** object and pass a string with it.
  + Call **functions** of textblob in order to do a specific task.
* Noun Phrase Extraction- Noun Phrase extraction is particularly important when you want to analyze the “who” in a sentence.
* Part-of-speech Tagging- Part-of-speech tagging or grammatical tagging is a method to mark words present in a text on the basis of its definition and context. In simple words, it tells whether a word is a noun, or an adjective, or a verb, etc.
* Words Inflection and Lemmatization- Inflection is a process of word formation in which characters are added to the base form of a word to express grammatical meanings. Word inflection in TextBlob is very simple, i.e., the words we tokenized from a textblob can be easily changed into singular or plural
* TextBlob library also offers an in-build object known as *Word.*
* N-grams- A combination of multiple words together are called N-Grams. N grams (N > 1) are generally more informative as compared to words, and can be used as features for language modelling.  N-grams can be easily accessed in TextBlob using the ***ngrams*** function, which returns a tuple of n successive words.
* Sentiment analysis is basically the process of determining the attitude or the emotion of the writer, i.e., whether it is positive or negative or neutral.
* The sentiment function of textblob returns two properties, polarity, and subjectivity.
  + Polarity is float which lies in the range of [-1,1] where 1 means positive statement and -1 means a negative statement. Subjective sentences generally refer to personal opinion, emotion or judgment whereas objective refers to factual information. Subjectivity is also a float which lies in the range of [0,1].

# GluonCV: a Deep Learning Toolkit for Computer Vision

GluonCV provides implementations of state-of-the-art (SOTA) deep learning algorithms in computer vision. It aims to help engineers, researchers, and students quickly prototype products, validate new ideas and learn computer vision.

GluonCV features:

1. training scripts that reproduce SOTA results reported in latest papers,
2. a large set of pre-trained models,
3. carefully designed APIs and easy to understand implementations,
4. community support.

**Code:**

**import** **mxnet** **as** **mx**

**import** **gluoncv**

*# you can change it to your image filename*

filename = 'classification-demo.png'

*# you may modify it to switch to another model. The name is case-insensitive*

model\_name = 'ResNet50\_v1d'

*# download and load the pre-trained model*

net = gluoncv.model\_zoo.get\_model(model\_name, pretrained=True)

*# load image*

img = mx.image.imread(filename)

*# apply default data preprocessing*

transformed\_img = gluoncv.data.transforms.presets.imagenet.transform\_eval(img)

*# run forward pass to obtain the predicted score for each class*

pred = net(transformed\_img)

*# map predicted values to probability by softmax*

prob = mx.nd.softmax(pred)[0].asnumpy()

*# find the 5 class indices with the highest score*

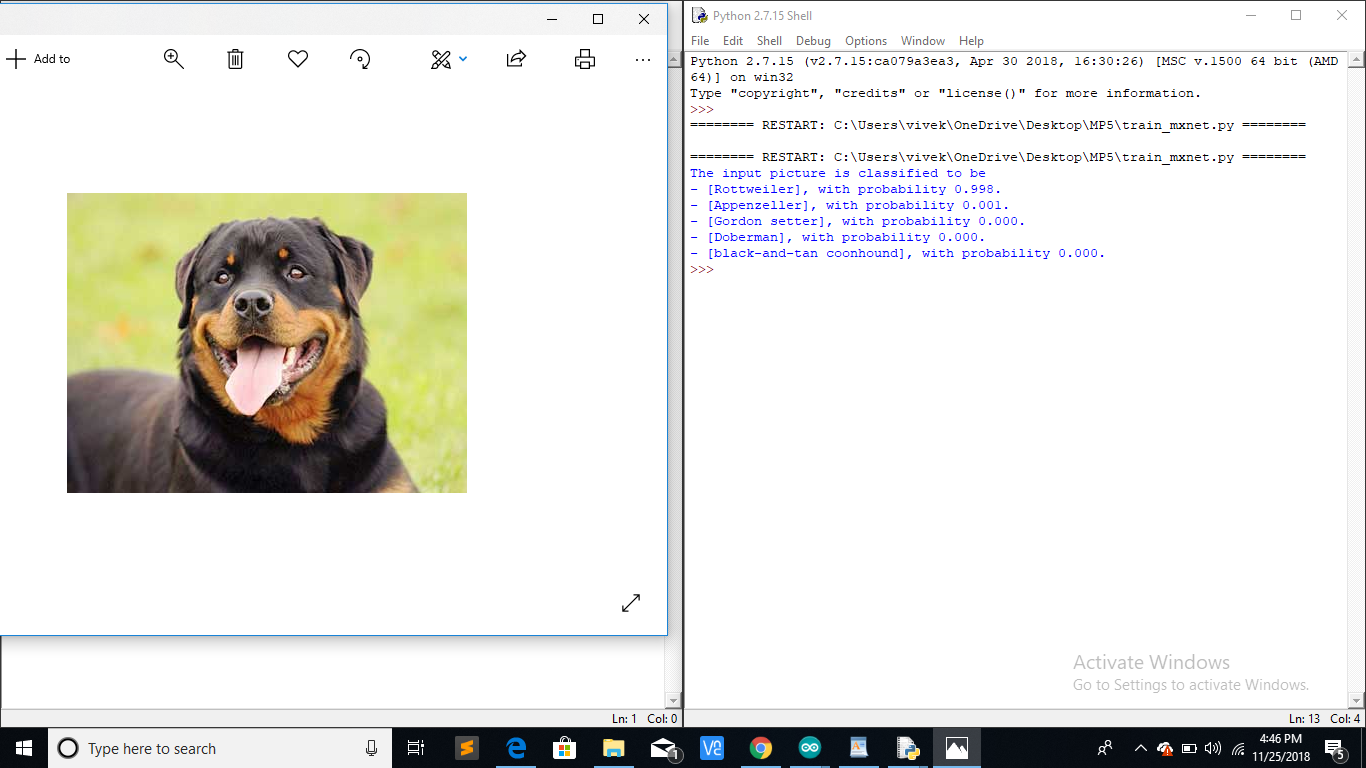
ind = mx.nd.topk(pred, k=5)[0].astype('int').asnumpy().tolist()

*# print the class name and predicted probability*

**print**('The input picture is classified to be')

**for** i **in** range(5):

**print**('- [*%s*], with probability *%.3f*.'%(net.classes[ind[i]], prob[ind[i]]))

****

**TO WHOMSOEVER IT MAY CONCERN**

This is to certify that Name of student (Roll No.), a student of B.Tech. in Programme Name from Institute of Technology, Nirma University worked in Name of Organization/Industry as a project trainee during dates/period of dissertation. During this period he/she was found regular and had done his/her project on “Title of Major Project”, under my supervision.

He/she has worked with utmost dedication and high level of engineering and analytical competence.

We wish him all the best for his/her future endeavors.

Date: (Signature)

Name of the External Guide

Affiliation of the External Guide

(Signature)

Name of the Faculty Guide

**Annexure - VI**

**Undertaking for Originality of the Work**

I, \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_, Roll No.\_\_\_\_\_\_\_\_\_\_ , give undertaking that the Major Project entitled

“\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_” submitted by me, towards the partial fulfillment of the requirements for the degree of Bachelor of Technology in **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** of Nirma University, Ahmedabad, is the original work carried out by me and I give assurance that no attempt of plagiarism has been made. I understand that in the event of any similarity found subsequently with any other published work or any project report elsewhere; it will result in severe disciplinary action.

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

Signature of Student

Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Place: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Endorsed by:

(Signature of External Guide)

(Signature of Internal Guide)

**Annexure - VII**

**Nirma University**

**Institute of Technology**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Department B.Tech. in \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

B. Tech. Major Project

PROJECT JOINING REPORT (PJR)

1) NAME OF STUDENT: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

2) ROLL NO: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. DATE OF JOINING: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
2. NAME OF COMPANY \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

/ORGANIZATION \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

5) PROJECT SITE ADDRESS: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

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

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

1. PHONE \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
2. FAX \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
3. EMAIL ADDRESS: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
4. WEB SITE ADDRESS: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
5. REPORTING TO \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(Name of In-charge \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

& Designation)

1. EMAIL ADDRESS: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
2. WORKING HOURS \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
3. LUNCH BREAK \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
4. ANY OTHER REMARKS:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
5. SIGN OF STUDENT: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
6. DATE: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

|  |  |
| --- | --- |
| EC. ENGG. DEPARTMENT USE |  |
| DATE :\_\_\_\_\_\_\_\_SIGN :\_\_\_\_\_\_\_\_\_ | SIGN OF Engineer In-charge/ Project |
|  | (with Date & Seal of Company) |
|  |  |