DATA-PORTFOLIO

Ruperto P. Bonet

ruperto.bonet@gmail.com
Telf:+44(79) 0169-3987

PhD in Computational Mechanics

(CIMEC-UNL, ARGENTINA-1998)

4.94/5 Bachelor Degree in Mathematics-Statistics

(Havana University, Cuba, 1986)

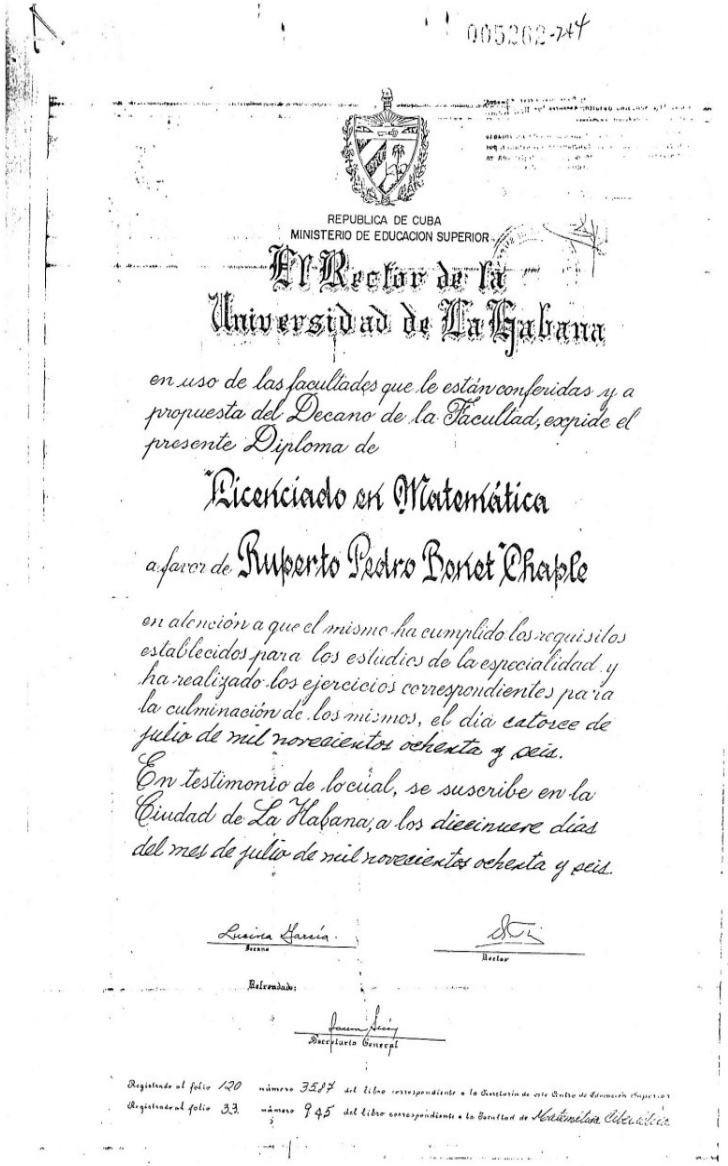
OUTLINE

Personal Description	2
Academic Degrees	3
Research Experiences Fellowship Programs	4
Research Experiences Research Stages	5
Research Experiences: Simulation Projects of engineering problems	6
Publishings	10
Experiences in Statistics	11
AI and Big Data	15
Cloud and Computing	16
Java-Hadoop	17
Spark	18
Experiences as a Big Data Analytics Tutor	19
Machine learning: Classification Training	23
Machine learning: Binary Classification	29
Lumped Sample Classification Prediction Method	34
Machine learning Algorithms	41
Applied AI and Deep Learning Serialization Keras	48
Tourism Enhancers Index by neighborhoods in London Project	53
Applied AI and Image Processing	60
Applied AI and Natural Language Processing	70

PERSONAL DESCRIPTION

- Enthusiastic and successful communicator teacher at all educational levels.
- Passionate scientific researcher in Applied Mathematics focused on Numerical Methods, Computational Mechanics, Simulation and Mathematical Modelling of real-world problems.
- Used to learning new technologies by self and willing to work in different environments and share as well its knowledge.
- A friendly person that participates in very active networking through Facebook, Linkedin, and other social media.
- A person with a cultural wealth acquired living and working around the world.

ACADEMIC DEGREES



005262-74

REPUBLICA DE CUBA
MINISTERIO DE EDUCACION SUPERIOR

El Rector de la
Universidad de La Habana

en uso de las facultades que le están conferidas y a propuesta del Decano de la Facultad, expide el presente Diploma de

Licenciado en Matemática

a favor de Ruperto Pedro Bonet Chaple

en atención a que el mismo ha cumplido los requisitos establecidos para los estudios de la especialidad y ha realizado los ejercicios correspondientes para la culminación de los mismos, el día catorce de julio de mil novecientos ochenta y seis.

En testimonio de lo cual, se suscribe en la Ciudad de La Habana, a los diecinueve días del mes de julio de mil novecientos ochenta y seis.

Luisa García
Decano

Rector

Refrendado:

Secretaria General

Registrado al folio 120 número 3597 del libro correspondiente a la Secretaría de este Centro de Educación Superior
Registrado al folio 33 número 945 del libro correspondiente a la Facultad de Matemática Elemental

REPÚBLICA ARGENTINA

Universidad Nacional del Litoral

Facultad de Ingeniería y Ciencias Hídricas

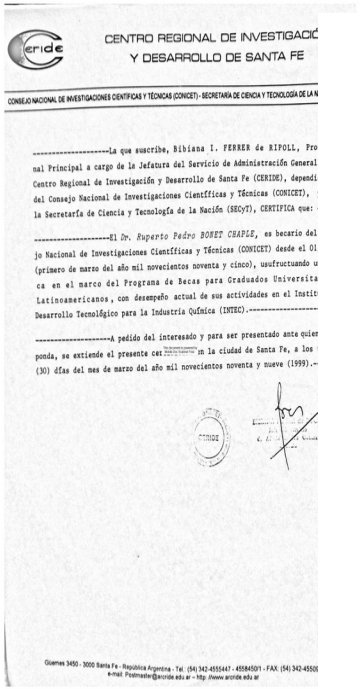
Por cuanto: el Licenciado en Matemática Ruperto Pedro Bonet Chaple, Pasaporte B-0273184, nacido en La Habana, República de Cuba, el 27 de marzo de 1957, ha aprobado la carrera de Doctorado en Ingeniería Mención: Mecánica Computacional, el 4 de diciembre de 1998.

Por tanto: de conformidad con lo que dispone el Estatuto vigente, se le otorga el presente diploma de Doctor en Ingeniería Mención: Mecánica Computacional.

Santa Fe, 17 de junio de 1999.

Registrado Universidad Nº 49.762 2ª Serie

RESEARCH EXPERIENCES
FELLOWSHIP PROGRAMS



CENTRO REGIONAL DE INVESTIGACIÓ
Y DESARROLLO DE SANTA FE

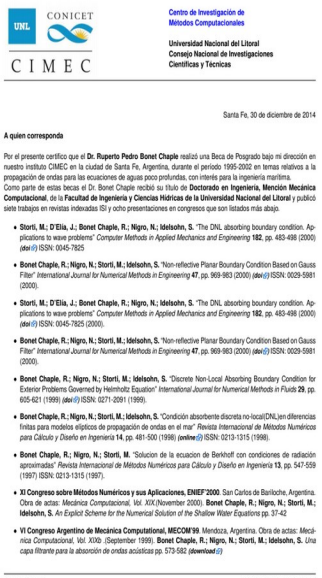
CONSEJO NACIONAL DE INVESTIGACIONES CIENTÍFICAS Y TÉCNICAS (CONICET) - SECRETARÍA DE CIENCIA Y TECNOLOGÍA DE LA N

----------La que suscribe, Bibiana I. FERRER de RIPOLL, Pro
nal Principal a cargo de la Jefatura del Servicio de Administración General
Centro Regional de Investigación y Desarrollo de Santa Fe (CERIDE), dependi
del Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET),
la Secretaría de Ciencia y Tecnología de la Nación (SECyT), CERTIFICA que:

----------El *Dr. Ruperto Pedro BONET CHAPLE*, es becario del
jo Nacional de Investigaciones Científicas y Técnicas (CONICET) desde el 01
(primero de marzo del año mil novecientos noventa y cinco), usufructuando u
ca en el marco del Programa de Becas para Graduados Universita
Latinoamericanos, con desempeño actual de sus actividades en el Instit
Desarrollo Tecnológico para la Industria Química (INTEC).----------

----------A pedido del interesado y para ser presentado ante quien
ponda, se extiende el presente cer in la ciudad de Santa Fe, a los
(30) días del mes de marzo del año mil novecientos noventa y nueve (1999).-

Güemes 3450 - 3000 Santa Fe - República Argentina - Tel. (54) 342-4555447 - 455845071 - FAX: (54) 342-4550
e-mail: Postmaster@ceride.edu.ar – http: //www.ceride.edu.ar



UNL CONICET
CIMEC

Centro de Investigación de
Métodos Computacionales

Universidad Nacional del Litoral
Consejo Nacional de Investigaciones
Científicas y Técnicas

Santa Fe, 30 de diciembre de 2014

A quién corresponda

Por el presente certifico que el **Dr. Ruperto Pedro Bonet Chaple** realizó una Beca de Posgrado bajo mi dirección en nuestro instituto CIMEC en la ciudad de Santa Fe, Argentina, durante el período 1995-2002 en temas relativos a la propagación de ondas para las ecuaciones de aguas poco profundas, con interés para la ingeniería marítima.
Como parte de estas becas el Dr. Bonet Chaple recibió su título de **Doctorado en Ingeniería, Mención Mecánica Computacional**, de la **Facultad de Ingeniería y Ciencias Hídricas de la Universidad Nacional del Litoral** y publicó siete trabajos en revistas indexadas ISI y ocho presentaciones en congresos que son listados más abajo.

- **Storti, M.; D'Elia, J.; Bonet Chaple, R.; Nigro, N.; Idelsohn, S.** "The DNL absorbing boundary condition. Applications to wave problems" *Computer Methods in Applied Mechanics and Engineering* **182**, pp. 483-498 (2000) (*doi*) ISSN: 0045-7825
- **Bonet Chaple, R.; Nigro, N.; Storti, M.; Idelsohn, S.** "Non-reflective Planar Boundary Condition Based on Gauss Filter" *International Journal for Numerical Methods in Engineering* **47**, pp. 969-983 (2000) (*doi*) ISSN: 0029-5981 (2000).
- **Storti, M.; D'Elia, J.; Bonet Chaple, R.; Nigro, N.; Idelsohn, S.** "The DNL absorbing boundary condition. Applications to wave problems" *Computer Methods in Applied Mechanics and Engineering* **182**, pp. 483-498 (2000) (*doi*) ISSN: 0045-7825 (2000).
- **Bonet Chaple, R.; Nigro, N.; Storti, M.; Idelsohn, S.** "Non-reflective Planar Boundary Condition Based on Gauss Filter" *International Journal for Numerical Methods in Engineering* **47**, pp. 969-983 (2000) (*doi*) ISSN: 0029-5981 (2000).
- **Bonet Chaple, R.; Nigro, N.; Storti, M.; Idelsohn, S.** "Discrete Non-Local Absorbing Boundary Condition for Exterior Problems Governed by Helmholtz Equation" *International Journal for Numerical Methods in Fluids* **29**, pp. 605-621 (1999) (*doi*) ISSN: 0271-2091 (1999).
- **Bonet Chaple, R.; Nigro, N.; Storti, M.; Idelsohn, S.** "Condición absorbente discreta no-local(DNL)en diferencias finitas para modelos elípticos de propagación de ondas en el mar" *Revista Internacional de Métodos Numéricos para Cálculo y Diseño en Ingeniería* **14**, pp. 481-500 (1998) (*online*) ISSN: 0213-1315 (1998).
- **Bonet Chaple, R.; Nigro, N.; Storti, M.** "Solucion de la ecuacion de Berkhoff con condiciones de radiación aproximadas" *Revista Internacional de Métodos Numéricos para Cálculo y Diseño en Ingeniería* **13**, pp. 547-559 (1997) ISSN: 0213-1315 (1997).
- **XI Congreso sobre Métodos Numéricos y sus Aplicaciones, ENIEF'2000**. San Carlos de Bariloche, Argentina. Obra de actas: *Mecánica Computacional, Vol. XIX* (November 2000). **Bonet Chaple, R.; Nigro, N.; Storti, M.; Idelsohn, S.** *An Explicit Scheme for the Numerical Solution of the Shallow Water Equations* pp. 37-42
- **VI Congreso Argentino de Mecánica Computational, MECOM'99**. Mendoza, Argentina. Obra de actas: *Mecánica Computacional, Vol. XIXb.* (September 1999). **Bonet Chaple, R.; Nigro, N.; Storti, M.; Idelsohn, S.** *Una capa filtrante para la absorción de ondas acústicas* pp. 573-582 (*download*)

Dr. Mario A. Storti
CIMEC (UNL/CONICET)
Predio CONICET-Santa Fe
Colect. Ruta Nac. 168 Km 472, Paraje El Pozo
3000 Santa Fe, Argentina

Tel: +54-342- 451 1594 (ext. 1015)
Fax: +54-342- 451 1169
<mario.storti@gmail.com>
www.cimec.santafe-conicet.gov.ar/storti

Instituto de Desarrollo Tecnológico
para la Industria Química
Güemes 3450
3000 SANTA FE (ARGENTINA)
TEL. 54 (342) 4550153/4550154/4550157
FAX: 54 (342) 4550944 NO. 54 (342) 4553961
e-mail@intec.unl.edu.ar

UNIVERSIDAD NACIONAL DEL LITORAL — CONSEJO NACIONAL DE INVESTIGACIONES CIENTIFICAS Y TECNICAS

Santa Fe, 10 de abril de 2000

Sra. Costanza Riveros
Jefa del Dpto. Becas Internas del CONICET
Rivadavia 1917
1033 – CAPITAL FEDERAL

Comunico a Ud. que a partir del 1º de abril de 2000, el Lic. Ruperto P. BONET CHAPLE, Pasaporte Nro. 80273184, ha iniciado en el INTEC las tareas inherentes a la Beca Posdoctoral Interna Tipo B Orientada, que le fuera otorgada por el CONICET, bajo la Dirección y Co-dirección de los Dres. Mario STORTI y Sergio R. IDELSOHN, respectivamente.

Motiva esta comunicación, que a la fecha el Lic. Bonet Chaple no ha recibido del CONICET la documentación correspondiente para hacer formal toma de posesión de dicha Beca.

Sirva la presente de fehaciente certificación.

Sin otro particular, la saluda atentamente.

Dr. ALBERTO E. CARDONO
Director Técnico
Instituto de Desarrollo Tecnológico
para la Industria Química
INTEC

Nota: Esta comunicación fue enviada por Fax con fecha 06/04/00 cuya fotocopia se adjunta.

c.c. Dpto. de Personal e Impuestos.

CIMNE

Prof. Ruperto Pedro Bonet
Centro Internacional de Métodos
Computacionales en Ingeniería (CIMEC)
INTEC
Güemes, 3450
3000 Santa Fe
ARGENTINA

Fax: +54/342/455.66.73

Barcelona, 7 de Marzo 2001

Estimado Prof. Bonet,

Me complace confirmarle nuestra invitación a que visite CIMNE del 16 de Abril al 15 de Junio de 2001.

Durante ese período se podrá profundizar en las posibilidades del cálculo finitesimal en problemas de flujo superficial y otros. Posteriormente, usted podría implementar los algoritmos correspondientes y de esta manera avanzaríamos en una línea de colaboración que esperamos sea fructífera.

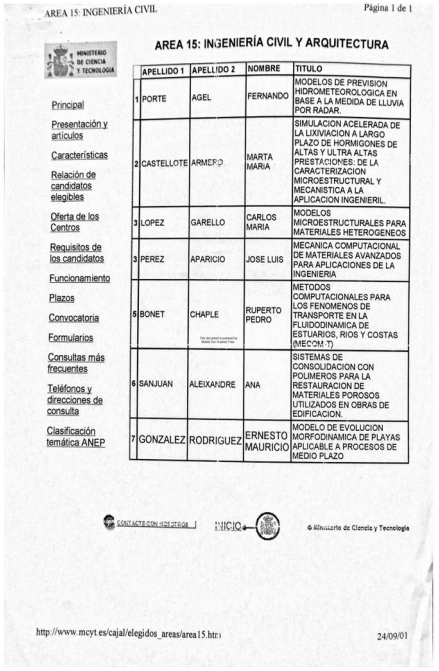
Esperando poder verle en Barcelona el próximo mes, reciba un cordial saludo

CENTRO INTERNACIONAL DE MÉTODOS NUMÉRICOS EN INGENIERÍA
EUGENIO OÑATE
Director

e-mail: onate@cimne.upc.es

EO/mc

Centro Internacional de Métodos Numéricos en Ingeniería
Edificio C-1, Campus Norte UPC, Gran Capitán, s/n, 08034 Barcelona, España – Teléfono: 34-93 203 72 16 – Fax 34-93 401 65 17 – e-mail: cimne@cimne.upc.es



AREA 15: INGENIERÍA CIVIL — Página 1 de 1

MINISTERIO DE CIENCIA Y TECNOLOGÍA

AREA 15: INGENIERÍA CIVIL Y ARQUITECTURA

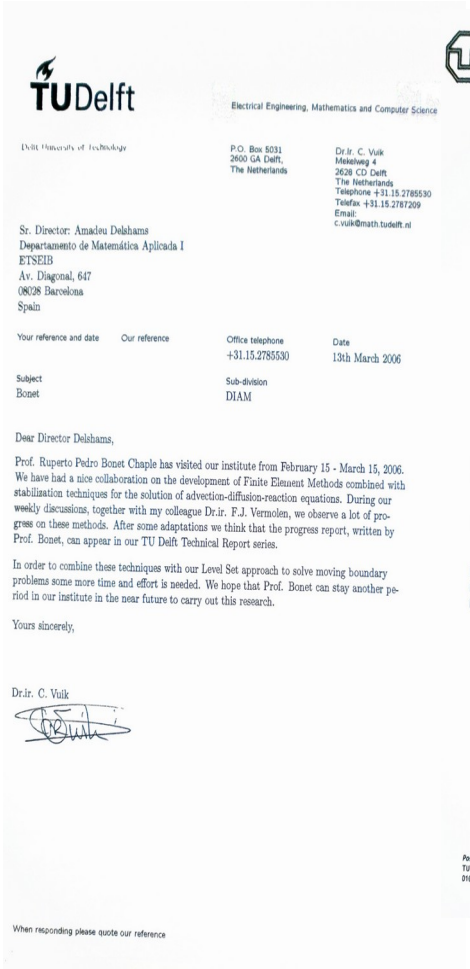
Principal
Presentación y artículos
Características
Relación de candidatos elegibles
Oferta de los Centros
Requisitos de los candidatos
Funcionamiento
Plazos
Convocatoria
Formularios
Consultas más frecuentes
Teléfonos y direcciones de consulta
Clasificación temática ANEP

	APELLIDO 1	APELLIDO 2	NOMBRE	TITULO
1	PORTE	AGEL	FERNANDO	MODELOS DE PREVISION HIDROMETEOROLOGICA EN BASE A LA MEDIDA DE LLUVIA POR RADAR.
2	CASTELLOTE	ARMERO	MARTA MARIA	SIMULACION ACELERADA DE LA LIXIVIACION A LARGO PLAZO DE HORMIGONES DE ALTAS Y ULTRA ALTAS PRESTACIONES: DE LA CARACTERIZACION MICROESTRUCTURAL Y MECANISTICA A LA APLICACION INGENIERIL.
3	LOPEZ	GARELLO	CARLOS MARIA	MODELOS MICROESTRUCTURALES PARA MATERIALES HETEROGENEOS
3	PEREZ	APARICIO	JOSE LUIS	MECANICA COMPUTACIONAL DE MATERIALES AVANZADOS PARA APLICACIONES DE LA INGENIERIA
5	BONET	CHAPLE	RUPERTO PEDRO	METODOS COMPUTACIONALES PARA LOS FENOMENOS DE TRANSPORTE EN LA FLUIDODINAMICA DE ESTUARIOS, RIOS Y COSTAS (MECOM_T)
6	SANJUAN	ALEIXANDRE	ANA	SISTEMAS DE CONSOLIDACION CON POLIMEROS PARA LA RESTAURACION DE MATERIALES POROSOS UTILIZADOS EN OBRAS DE EDIFICACION.
7	GONZALEZ	RODRIGUEZ	ERNESTO MAURICIO	MODELO DE EVOLUCION MORFODINAMICA DE PLAYAS APLICABLE A PROCESOS DE MEDIO PLAZO

CONTACTE CON NOSOTROS | INICIO | © Ministerio de Ciencia y Tecnología

http://www.mcyt.es/cajal/elegidos_areas/area15.htm 24/09/01

RESEARCH EXPERIENCES
RESEARCH STAGES



TU Delft

Delft University of Technology

Electrical Engineering, Mathematics and Computer Science

P.O. Box 5031
2600 GA Delft,
The Netherlands

Dr.ir. C. Vuik
Mekelweg 4
2628 CD Delft
The Netherlands
Telephone +31.15.2785530
Telefax +31.15.2787209
Email:
c.vuik@math.tudelft.nl

Sr. Director: Amadeu Delshams
Departamento de Matemática Aplicada I
ETSEIB
Av. Diagonal, 647
08028 Barcelona
Spain

Your reference and date	Our reference	Office telephone	Date
		+31.15.2785530	13th March 2006

Subject: Bonet

Sub-division: DIAM

Dear Director Delshams,

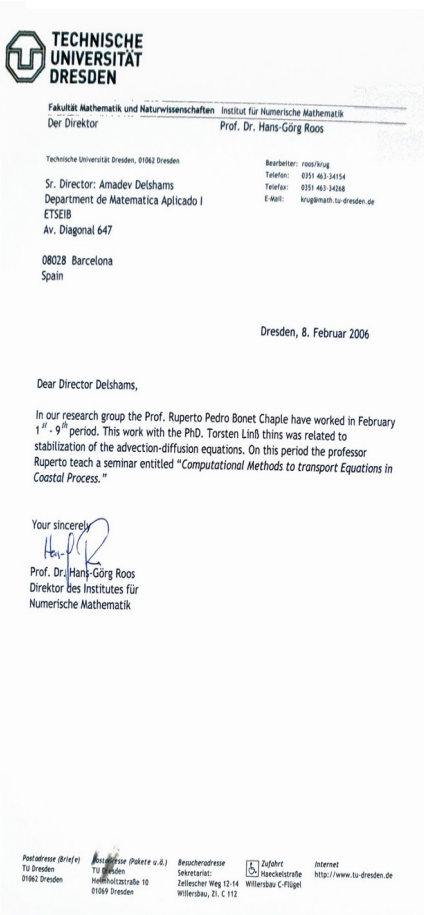
Prof. Ruperto Pedro Bonet Chaple has visited our institute from February 15 - March 15, 2006. We have had a nice collaboration on the development of Finite Element Methods combined with stabilization techniques for the solution of advection-diffusion-reaction equations. During our weekly discussions, together with my colleague Dr.ir. F.J. Vermolen, we observe a lot of progress on these methods. After some adaptations we think that the progress report, written by Prof. Bonet, can appear in our TU Delft Technical Report series.

In order to combine these techniques with our Level Set approach to solve moving boundary problems some more time and effort is needed. We hope that Prof. Bonet can stay another period in our institute in the near future to carry out this research.

Yours sincerely,

Dr.ir. C. Vuik

When responding please quote our reference



TECHNISCHE UNIVERSITÄT DRESDEN

Fakultät Mathematik und Naturwissenschaften Institut für Numerische Mathematik
Der Direktor Prof. Dr. Hans-Görg Roos

Technische Universität Dresden, 01062 Dresden

Bearbeiter: roos/krug
Telefon: 0351 463-34154
Telefax: 0351 463-34268
E-Mail: krug@math.tu-dresden.de

Sr. Director: Amadev Delshams
Department de Matematica Aplicado I
ETSEIB
Av. Diagonal 647

08028 Barcelona
Spain

Dresden, 8. Februar 2006

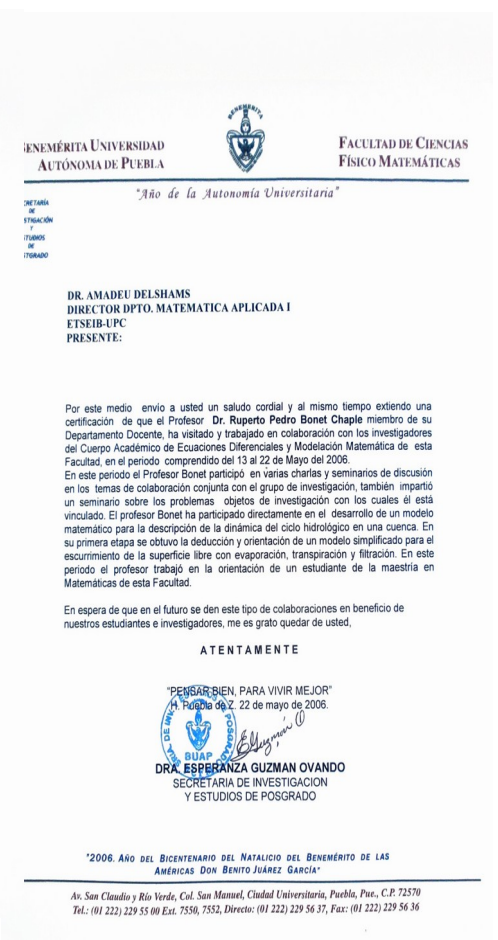
Dear Director Delshams,

In our research group the Prof. Ruperto Pedro Bonet Chaple have worked in February 1st - 9th period. This work with the PhD. Torsten Linß thins was related to stabilization of the advection-diffusion equations. On this period the professor Ruperto teach a seminar entitled *"Computational Methods to transport Equations in Coastal Process."*

Your sincerely,

Prof. Dr. Hans-Görg Roos
Direktor des Institutes für
Numerische Mathematik

Postadresse (Briefe) TU Dresden 01062 Dresden | Hausadresse (Pakete u.ä.) TU Dresden Hettnerstraße 10 01069 Dresden | Besucheradresse Sekretariat: Zellescher Weg 12-14 Willersbau, Zi. C 112 | Zufahrt Haeckelstraße Willersbau C-Flügel | Internet http://www.tu-dresden.de



BENEMÉRITA UNIVERSIDAD AUTÓNOMA DE PUEBLA

FACULTAD DE CIENCIAS FÍSICO MATEMÁTICAS

"Año de la Autonomía Universitaria"

SECRETARÍA DE INVESTIGACIÓN Y ESTUDIOS DE POSGRADO

DR. AMADEU DELSHAMS
DIRECTOR DPTO. MATEMATICA APLICADA I
ETSEIB-UPC
PRESENTE:

Por este medio envio a usted un saludo cordial y al mismo tiempo extiendo una certificación de que el Profesor **Dr. Ruperto Pedro Bonet Chaple** miembro de su Departamento Docente, ha visitado y trabajado en colaboración con los investigadores del Cuerpo Académico de Ecuaciones Diferenciales y Modelación Matemática de esta Facultad, en el periodo comprendido del 13 al 22 de Mayo del 2006.
En este periodo el Profesor Bonet participó en varias charlas y seminarios de discusión en los temas de colaboración conjunta con el grupo de investigación, también impartió un seminario sobre los problemas objetos de investigación con los cuales él está vinculado. El profesor Bonet ha participado directamente en el desarrollo de un modelo matemático para la descripción de la dinámica del ciclo hidrológico en una cuenca. En su primera etapa se obtuvo la deducción y orientación de un modelo simplificado para el escurrimiento de la superficie libre con evaporación, transpiración y filtración. En este periodo el profesor trabajó en la orientación de un estudiante de la maestría en Matemáticas de esta Facultad.

En espera de que en el futuro se den este tipo de colaboraciones en beneficio de nuestros estudiantes e investigadores, me es grato quedar de usted,

ATENTAMENTE

"PENSAR BIEN, PARA VIVIR MEJOR"
H. Puebla de Z. 22 de mayo de 2006.

DRA. ESPERANZA GUZMAN OVANDO
SECRETARIA DE INVESTIGACION
Y ESTUDIOS DE POSGRADO

"2006. Año del Bicentenario del Natalicio del Benemérito de las Américas Don Benito Juárez García"

Av. San Claudio y Río Verde, Col. San Manuel, Ciudad Universitaria, Puebla, Pue., C.P. 72570
Tel.: (01 222) 229 55 00 Ext. 7550, 7552, Directo: (01 222) 229 56 37, Fax: (01 222) 229 56 36

Departament de Matemàtica Aplicada I
UNIVERSITAT POLITÈCNICA DE CATALUNYA

Escola Tècnica Superior d'Enginyeria Industrial de Barcelona
Diagonal, 647
E-08028 Barcelona
Tel. 93 401 65 49
Fax 93 401 17 13

Yo, Yuri Fedorov Kuzmin, el miembro del equipo de investigacion del proyecto MTM2009-12672,

Certifico, que recibi el investigador Ruperto Pedro Bonet Chaple en el departamento de Matematica Aplicada I de la UPC, durante el periodo 3 de febrero -- 19 de diciembre de 2014, y que hemos realizado juntos con el un trabajo sobre temas del proyecto mencionado (estudio de singularidades complejas de soluciones de algunos sistemas integrables y descripcion de las variedades invariantes complejas).

UPC Departament de Matemàtica Aplicada I

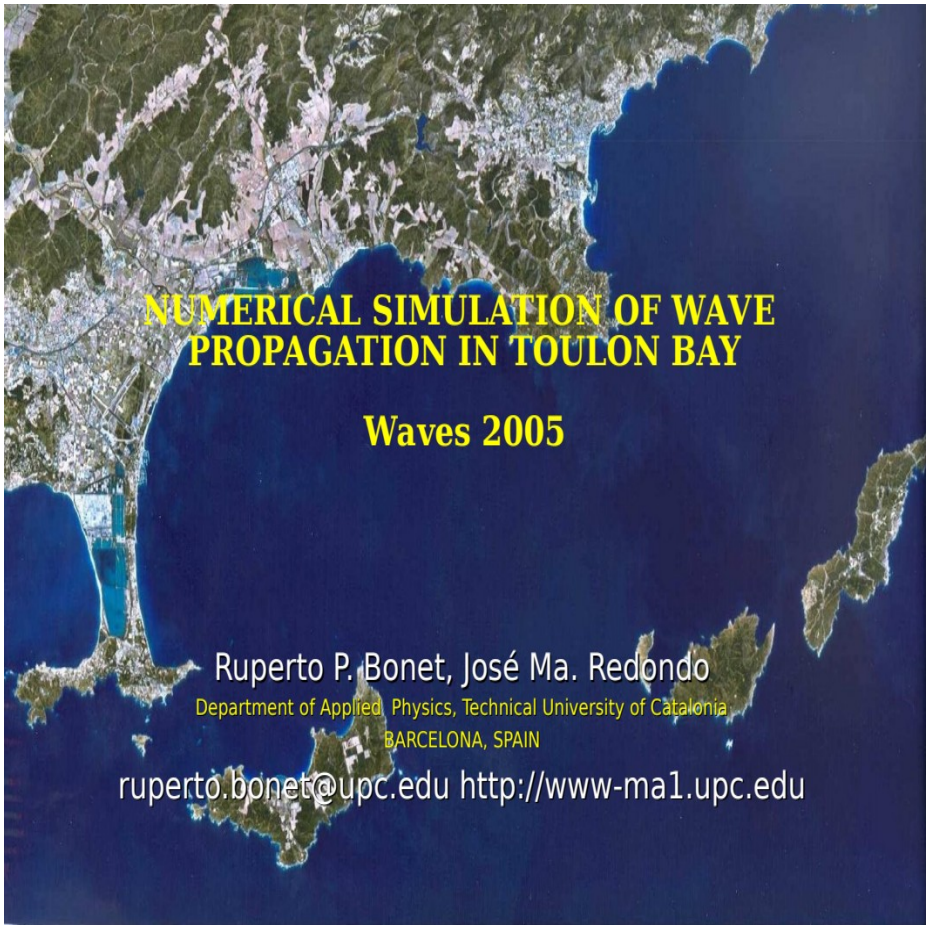
Barcelona, 30 de abril de 2015

Yuri Fedorov Kuzmin

23/01/2017

RESEARCH EXPERIENCES

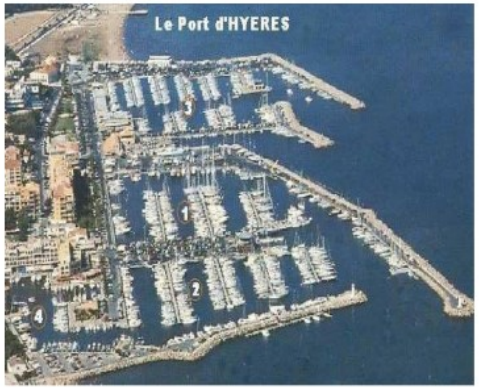
SIMULATION PROJECTS OF ENGINEERING PROBLEMS



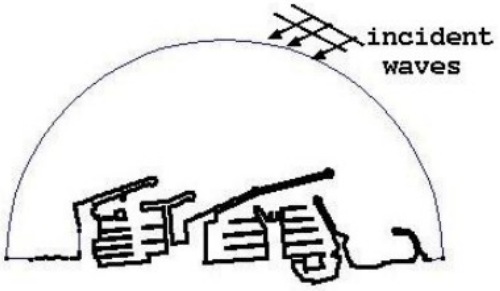
RESEARCH EXPERIENCES

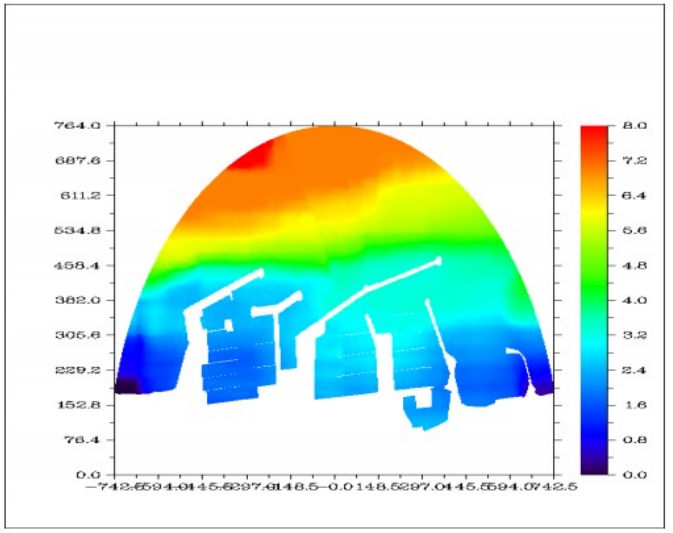
SIMULATION PROJECTS OF ENGINEERING PROBLEMS

Waves in Hyeres Harbor



Period : 6s
Incidence angle : South-East
Fully Reflecting at coastline

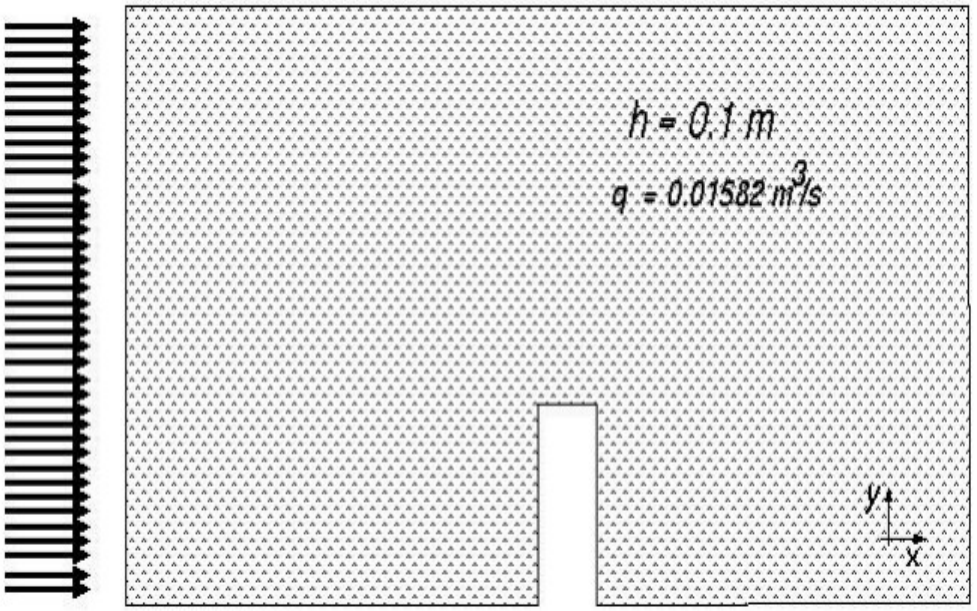




Skills: Data entry, data preparation, C++, MATLAB, modelling, FEM

SPECIFIC PHYSICAL PROBLEMS

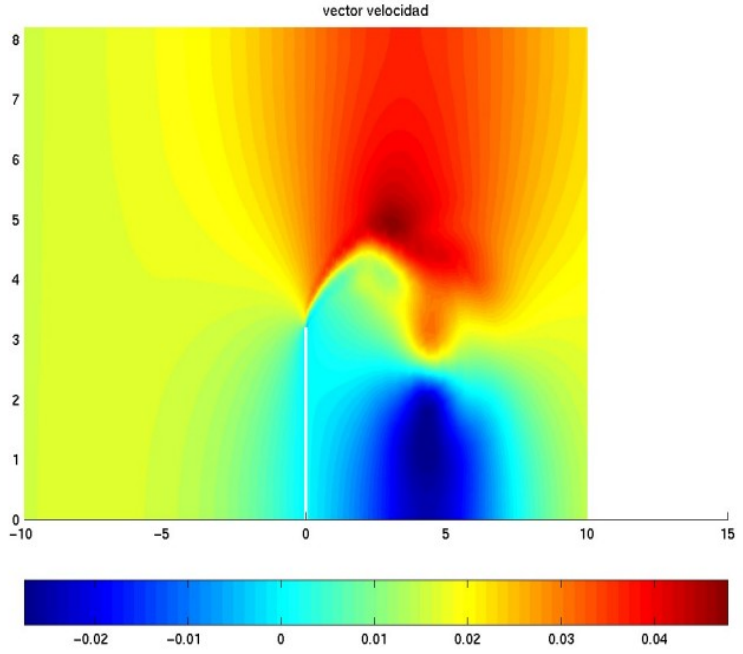
INVISCID FLOW IN OPEN CHANNELS WITH LATERAL CONTRACTION



RESEARCH EXPERIENCES

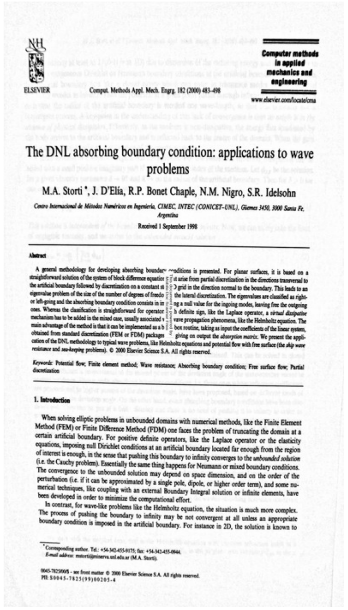
SIMULATION PROJECTS OF ENGINEERING PROBLEMS

VELOCITY FLOW MAP



Skills: Data entry, data preparation, C++, MPI, MATLAB, modelling, FEM

PUBLISHINGS



ELSEVIER

Comput. Methods Appl. Mech. Engrg. 182 (2000) 483–498

Computer methods in applied mechanics and engineering

www.elsevier.com/locate/cma

The DNL absorbing boundary condition: applications to wave problems

M.A. Storti *, J. D'Elía, R.P. Bonet Chaple, N.M. Nigro, S.R. Idelsohn

Centro Internacional de Métodos Numéricos en Ingeniería, CIMEC, INTEC (CONICET–UNL), Güemes 3450, 3000 Santa Fe, Argentina

Received 1 September 1998

Abstract

A general methodology for developing absorbing boundary conditions is presented. For planar surfaces, it is based on a straightforward solution of the system of block difference equations that arise from partial discretization in the directions transversal to the artificial boundary followed by discretization on a constant step grid in the direction normal to the boundary. This leads to an eigenvalue problem of the size of the number of degrees of freedom in the lateral discretization. The eigenvalues are classified as right- or left-going and the absorbing boundary condition consists in using a null value for the ingoing modes, leaving free the outgoing ones. Whereas the classification is straightforward for operators with definite sign, like the Laplace operator, a *virtual dissipative* mechanism has to be added in the mixed case, usually associated with wave propagation phenomena, like the Helmholtz equation. The main advantage of the method is that it can be implemented as a black box routine, taking as input the coefficients of the linear system, obtained from standard discretization (FDM or FDM) packages and giving on output the *absorption matrix*. We present the application of the DNL methodology to typical wave problems, like Helmholtz equations and potential flow with free surface (the ship wave resistance and sea-keeping problems). © 2000 Elsevier Science S.A. All rights reserved.

Keywords: Potential flow; Finite element method; Wave resistance; Absorbing boundary condition; Free surface flow; Partial discretization

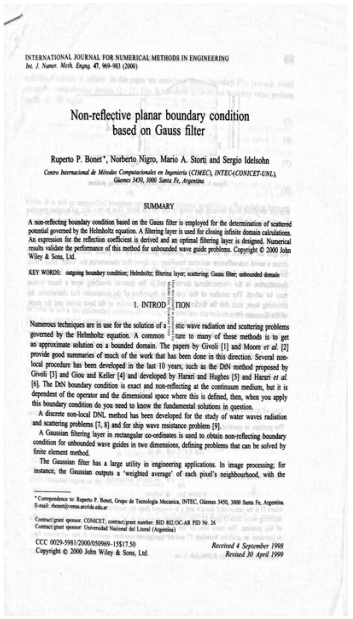
1. Introduction

When solving elliptic problems in unbounded domains with numerical methods, like the Finite Element Method (FEM) or Finite Difference Method (FDM) one faces the problem of truncating the domain at a certain artificial boundary. For positive definite operators, like the Laplace operator or the elasticity equations, imposing null Dirichlet conditions at an artificial boundary located far enough from the region of interest is enough, in the sense that pushing this boundary to infinity converges to the *unbounded solution* (i.e. the Cauchy problem). Essentially the same thing happens for Neumann or mixed boundary conditions. The convergence to the unbounded solution may depend on space dimension, and on the order of the perturbation (i.e. if it can be approximated by a single pole, dipole, or higher order term), and some numerical techniques, like coupling with an external Boundary Integral solution or infinite elements, have been developed in order to minimize the computational effort.

In contrast, for wave-like problems like the Helmholtz equation, the situation is much more complex. The process of pushing the boundary to infinity may be not convergent at all unless an appropriate boundary condition is imposed in the artificial boundary. For instance in 2D, the solution is known to

* Corresponding author. Tel.: +54-342-4551075; fax: +54-342-455-0944.
E-mail address: mstorti@intec.unl.edu.ar (M.A. Storti).

0045-7825/99/$ – see front matter © 2000 Elsevier Science S.A. All rights reserved.
PII: S0045-7825(99)00265-4



INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN ENGINEERING
Int. J. Numer. Meth. Engng. 47, 969–983 (2000)

Non-reflective planar boundary condition based on Gauss filter

Ruperto P. Bonet *, Norberto Nigro, Mario A. Storti and Sergio Idelsohn

Centro Internacional de Métodos Computacionales en Ingeniería (CIMEC), INTEC-(CONICET-UNL), Güemes 3450, 3000 Santa Fe, Argentina

SUMMARY

A non-reflecting boundary condition based on the Gauss filter is employed for the determination of scattered potential governed by the Helmholtz equation. A filtering layer is used for closing infinite domain calculations. An expression for the reflection coefficient is derived and an optimal filtering layer is designed. Numerical results validate the performance of this method for unbounded wave guide problems. Copyright © 2000 John Wiley & Sons, Ltd.

KEY WORDS: outgoing boundary condition; Helmholtz; filtering layer; scattering; Gauss filter; unbounded domain

1. INTRODUCTION

Numerous techniques are in use for the solution of scattering and radiation wave problems governed by the Helmholtz equation. A common feature to many of these methods is to get an approximate solution on a bounded domain. The papers by Givoli [1] and Moore *et al.* [2] provide good summaries of much of the work that has been done in this direction. Several non-local procedure has been developed in the last 10 years, such as the DtN method proposed by Givoli [3] and Grou and Keller [4] and developed by Harari and Hughes [5] and Harari *et al.* [6]. The DtN boundary condition is exact and non-reflecting at the continuum medium, but it is dependent of the operator and the dimensional space where this is defined, then, when you apply this boundary condition do you need to know the fundamental solutions in question.

A discrete non-local DNL method has been developed for the study of water waves radiation and scattering problems [7, 8] and for ship wave resistance problem [9].

A Gaussian filtering layer in rectangular co-ordinates is used to obtain non-reflecting boundary condition for unbounded wave guides in two dimensions, defining problems that can be solved by finite element method.

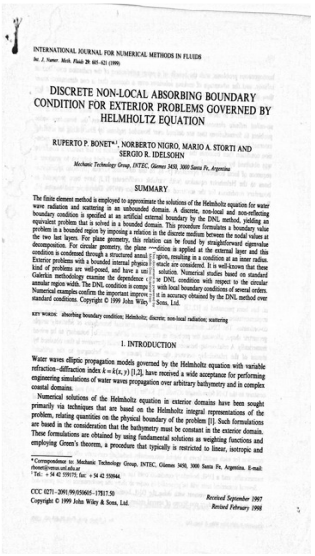
The Gaussian filter has a large utility in engineering applications. In image processing, for instance, the Gaussian outputs a 'weighted average' of each pixel's neighbourhood, with the

* Correspondence to: Ruperto P. Bonet, Grupo de Tecnología Mecánica, INTEC, Güemes 3450, 3000 Santa Fe, Argentina.
E-mail: rbonet@cimec.arcride.edu.ar

Contract/grant sponsor: CONICET; contract/grant number: BID 802/OC-AR PID Nr. 26
Contract/grant sponsor: Universidad Nacional del Litoral (Argentina)

CCC 0029-5981/2000/050969–15$17.50

Received 4 September 1998
Revised 30 April 1999



INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
Int. J. Numer. Meth. Fluids 28: 601–621 (1998)

DISCRETE NON-LOCAL ABSORBING BOUNDARY CONDITION FOR EXTERIOR PROBLEMS GOVERNED BY HELMHOLTZ EQUATION

RUPERTO P. BONET*,†, NORBERTO NIGRO, MARIO A. STORTI AND SERGIO R. IDELSOHN

Mechanic Technology Group, INTEC, Güemes 3450, 3000 Santa Fe, Argentina

SUMMARY

The finite element method is employed to approximate the solutions of the Helmholtz equation for water wave radiation and scattering in an unbounded domain. A discrete, non-local and non-reflecting boundary condition is specified at an artificial external boundary by the DNL method, yielding an equivalent problem that is solved in a bounded domain. This procedure formulates a boundary value problem in a bounded region by imposing a relation in the discrete medium between the nodal values at the two last layers. For plane geometry, this relation can be found by straightforward eigenvalue decomposition. For circular geometry, the plane wave solution is applied at the external layer and this condition is condensed through a structured annular region, resulting in a condition at an inner radius. Exterior problems with a bounded internal physics and unbounded exterior domains are considered. It is well-known that these kind of problems are well-posed, and have a unique solution. Numerical studies based on standard Galerkin methodology examine the dependence of the DNL condition with respect to the circular annular region width. The DNL condition is compared with local boundary conditions of several orders. Numerical examples confirm the important improvement in accuracy obtained by the DNL method over standard conditions. Copyright © 1999 John Wiley & Sons, Ltd.

KEY WORDS: absorbing boundary condition; Helmholtz; discrete; non-local radiation; scattering

1. INTRODUCTION

Water waves elliptic propagation models governed by the Helmholtz equation with variable refraction–diffraction index $k = k(x, y)$ [1,2], have received a wide acceptance for performing engineering simulations of water waves propagation over arbitrary bathymetry and in complex coastal domains.

Numerical solutions of the Helmholtz equation in exterior domains have been sought primarily via techniques that are based on the Helmholtz integral representations of the problem, relating quantities on the physical boundary of the problem [1]. Such formulations are based in the consideration that the bathymetry must be constant in the exterior domain. These formulations are obtained by using fundamental solutions as weighting functions and employing Green's theorem, a procedure that typically is restricted to linear, isotropic and

* Correspondence to: Mechanic Technology Group, INTEC, Güemes 3450, 3000 Santa Fe, Argentina. E-mail: rbonet@cimec.unl.edu.ar
† Tel.: + 54 42 559573; fax: + 54 42 550944.

CCC 0271–2091/99/050605–17$17.50

Revised September 1997
Revised February 1998

Numerical solutions of a Simplified Model for the interaction of growth factors concentration and Capillary Sprout Growth during Tumor Angiogenesis

Ruperto P. Bonet Chaple *

Department of Mathematics
National University of San Luis,
Ejército de los Andes 950 Bloque 2
5700 San Luis, Argentina

Abstract

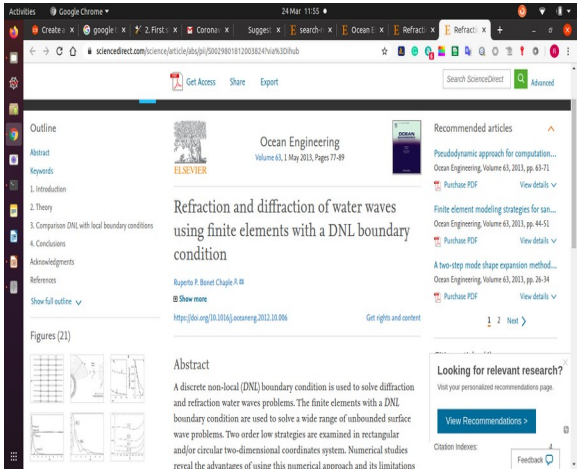
We present the first numerical tests relative to a 1D-mathematical model which describes the interaction between the growth factors and development of capillary sprouts during tumor angiogenesis stage, based on the presented model in H.M.Byrne and M.A.J.Chaplain. Also, we examine numerically the behaviour of the growth factor concentration induced by tumor growth, based on the model equation proposed by F.Vermolen. These numerical results has been obtained using an explicit method on each equation and in each time step the coupling terms have been updated by means of an iterative process, where the growth factor concentration and tip density gradients must be to evaluated numerically. In this case, these terms have been computed by central differences at the center of each mesh element.

1 Growth Factor concentration mathematical model

In this section the model based on the ideas of [3.Vermolen] is presented. Firstly, the model for the regeneration, decay and transport of the growth factor is given, and subsequently the tumor growth process as a result of the

* Corresponding author. Department of Mathematics
National University of San Luis, Ejército de los Andes 950 Bloque 2 (5700)
San Luis, Argentina. Tel : +54(02652)42-67-59; Fax : +54(02652)44-40-14
E-mail address : rupertobonet@modelplusmat.com
Web page : http://www.modelplusmat.com

Preprint submitted to Applied Mathematical Modelling *14 October 2013*



Activities Google Chrome 24 Mar 11:55

sciencedirect.com/science/article/abs/pii/S0029801812003624?via%3Dihub

Get Access Share Export Search ScienceDirect Advanced

Outline
Abstract
Keywords
1. Introduction
2. Theory
3. Comparison DNL with local boundary conditions
4. Conclusions
Acknowledgments
References
Show full outline
Figures (21)

ELSEVIER

Ocean Engineering
Volume 63, 1 May 2013, Pages 77-89

Refraction and diffraction of water waves using finite elements with a DNL boundary condition

Ruperto P. Bonet Chaple, R.
Show more
https://doi.org/10.1016/j.oceaneng.2012.10.006 Get rights and content

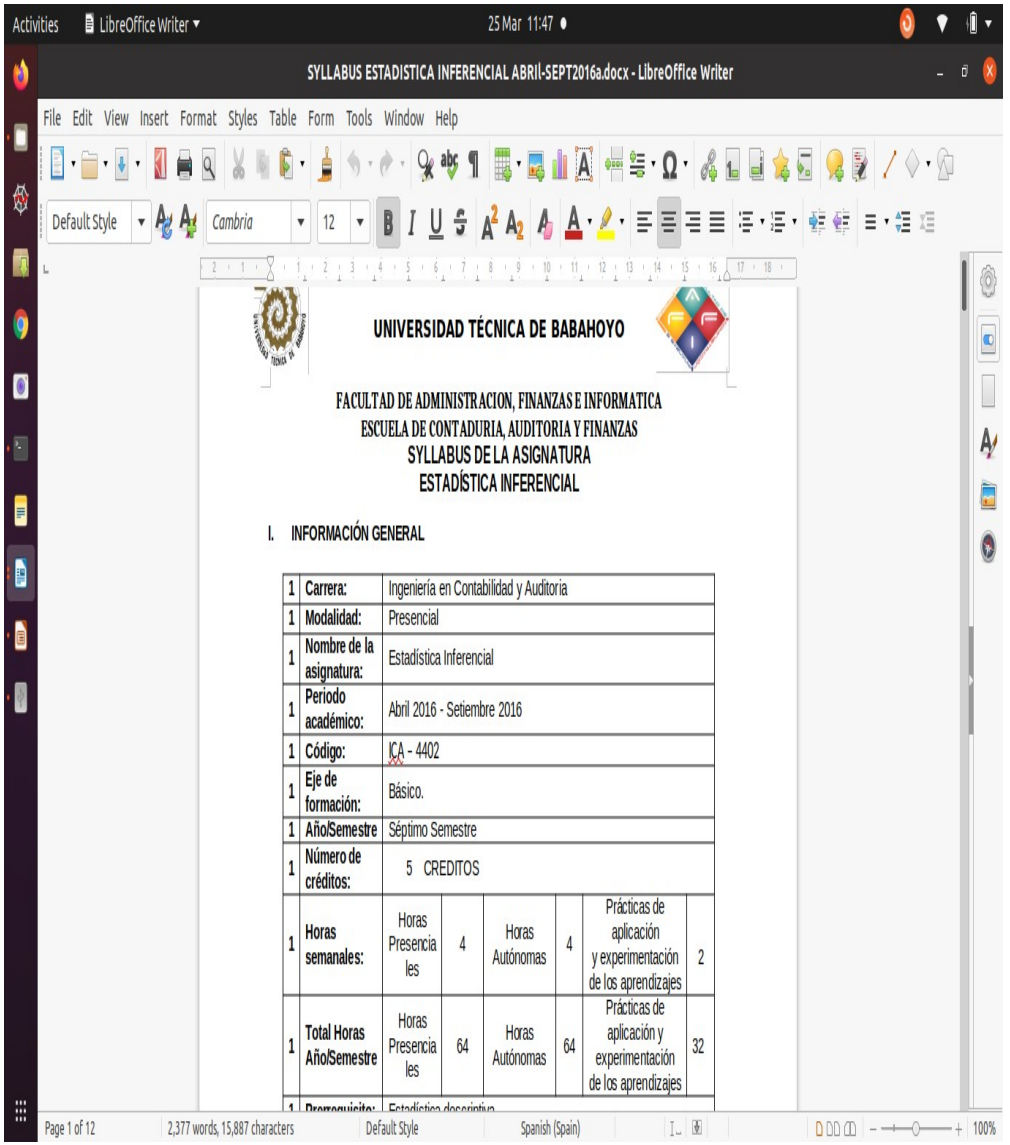
Abstract

A discrete non-local (DNL) boundary condition is used to solve diffraction and refraction water waves problems. The finite elements with a *DNL* boundary condition are used to solve a wide range of unbounded surface wave problems. Two order low strategies are examined in rectangular and/or circular two-dimensional coordinates system. Numerical studies reveal the advantages of usine this numerical approach and its limitations

Recommended articles
Pseudodynamic approach for computation...
Ocean Engineering, Volume 63, 2013, pp. 63-71
Purchase PDF View details
Finite element modeling strategies for san...
Ocean Engineering, Volume 63, 2013, pp. 44-51
Purchase PDF View details
A two-step mode shape expansion method...
Ocean Engineering, Volume 63, 2013, pp. 26-34
Purchase PDF View details
1 2 Next

Looking for relevant research?
Visit your personalized recommendations page.
View Recommendations >
Citation Indexes:
Feedback

EXPERIENCES in STATISTICS



Activities LibreOffice Writer 25 Mar 11:47

SYLLABUS ESTADISTICA INFERENCIAL ABRIL-SEPT2016a.docx - LibreOffice Writer

File Edit View Insert Format Styles Table Form Tools Window Help

Default Style Cambria 12

UNIVERSIDAD TÉCNICA DE BABAHOYO

FACULTAD DE ADMINISTRACION, FINANZAS E INFORMATICA
ESCUELA DE CONTADURIA, AUDITORIA Y FINANZAS
SYLLABUS DE LA ASIGNATURA
ESTADÍSTICA INFERENCIAL

I. INFORMACIÓN GENERAL

1	Carrera:	Ingeniería en Contabilidad y Auditoria					
1	Modalidad:	Presencial					
1	Nombre de la asignatura:	Estadística Inferencial					
1	Periodo académico:	Abril 2016 - Setiembre 2016					
1	Código:	ICA - 4402					
1	Eje de formación:	Básico.					
1	Año/Semestre	Séptimo Semestre					
1	Número de créditos:	5 CREDITOS					
1	Horas semanales:	Horas Presenciales	4	Horas Autónomas	4	Prácticas de aplicación y experimentación de los aprendizajes	2
1	Total Horas Año/Semestre	Horas Presenciales	64	Horas Autónomas	64	Prácticas de aplicación y experimentación de los aprendizajes	32
1	Prerrequisito:	Estadística descriptiva					

Page 1 of 12 2,377 words, 15,887 characters Default Style Spanish (Spain) 100%

EXPERIENCES in STATISTICS

UNIVERSIDAD TÉCNICA DE BABAHOYO
FACULTAD DE ADMINISTRACION FINANZAS E INFORMATICAS
ESCUELA DE CONTABILIDAD Y AUDITORIA

SYLLABUS ESTADÍSTICA DESCRIPTIVA

I. INFORMACIÓN GENERAL

1.1.	**Carrera:**	CONTADURIA, AUDITORIA Y FINANZAS						
1.2.	**Modalidad:**	PRESENCIAL						
1.3.	**Nombre de la asignatura:**	ESTADÍSTICA DESCRIPTIVA						
1.4.	**Periodo académico:**	SEPTIEMBRE 2015-DICIEMBRE 2015						
1.5.	**Código:**	X						
1.6.	**Eje de formación:**	PROFESIONAL						
1.7.	**Semestre:**	MATUTINO						
1.8.	**Número de créditos:**	X						
1.9.	**Horas semanales:**	**Horas Presenciales**	3	**Horas Autónomas**		**Prácticas de aplicación y experimentación de los aprendizajes**		
1.10.	**Total Horas Semestre: 160**	**Horas Presenciales**	64	**Horas Autónomas**	64	**Prácticas de aplicación y experimentación de los aprendizajes**	32	
1.11.	**Prerrequisito:**	ALGEBRA						
1.12.	**Correquisito:**	ESTADISTICA INFERENCIAL						
1.13.	**Duración periodo lectivo:**	19 semanas	Fecha inicio	Septiembre 28/09/2015		Fecha culminación	26/11/ 2015	
1.14.	**Profesores responsables:**	Ec. Martha Acosta Roby, MCA Dra. Enrique Díaz Chong. Msc. Ing. Héctor Crespo Caicedo PhD. Lic.Ruperto Pedro Bonet Chaple		E-mail: mgacosta@utb.edu.ec hcrespoc@utb.edu.ec rbonet@utb.edu.ec				

EXPERIENCES in STATISTICS

UNIVERSIDAD TÉCNICA DE BABAHOYO
VICERRECTORADO DE INVESTIGACIÓN Y POSTGRADO
INSTITUTO DE INVESTIGACIÓN Y DESARROLLO

Certifica a:

Ing. Ruperto Bonett

Vicerrectorado de Investigación y Postgrado

Por haber participado como Capacitador en el PROGRAMA DE FORTALECIMIENTO DE CAPACIDADES DOCENTES EN INVESTIGACIÓN, en los Módulos de Concepción y Escritura de Artículos Científicos, MATLAB, e-learning/m-learning/b-learning, Matemática y Tecnología, Metodología de la Investigación, Diseño de Experimentos y Estadísticas en la Investigación durante los días los días 21 y 28 de octubre, 4 y 11 de noviembre y 2 de diciembre de 2015, en la Universidad Técnica de Babahoyo.

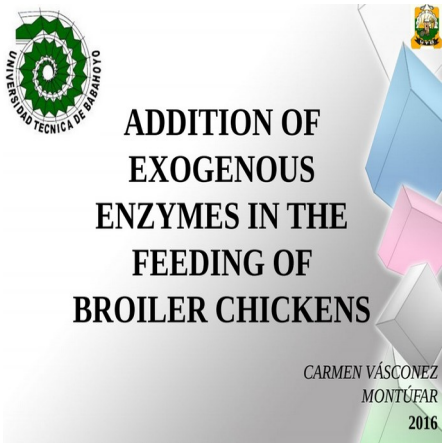
Babahoyo, 2 de diciembre de 2015

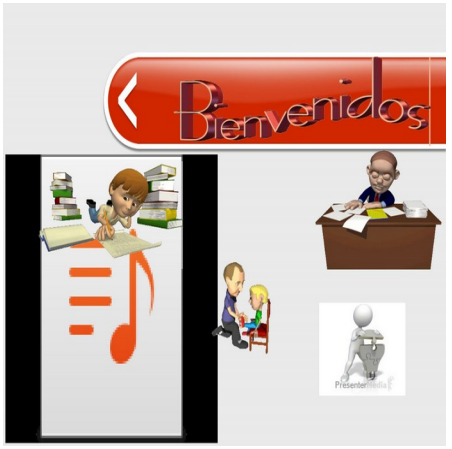
Dr . Rafael Falconí Montalván, MSc.
RECTOR

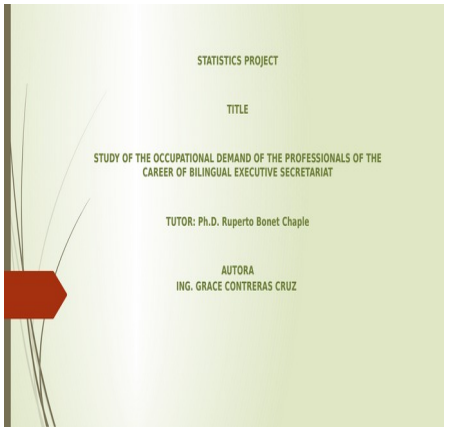
Lic. Adelita Pinto Yerovi, MSc.
VICERRECTORA DE INVESTIGACIÓN Y POSTGRADO

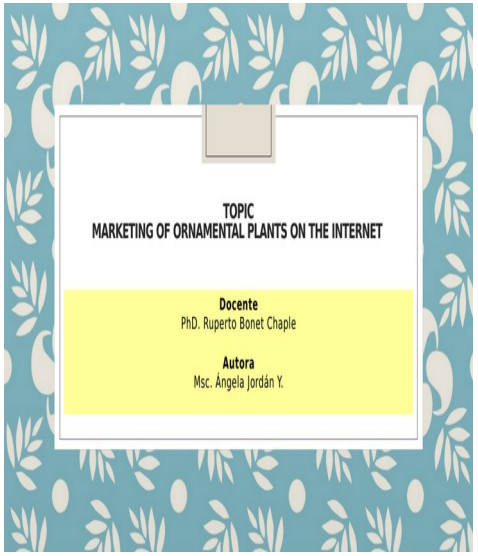
EXPERIENCES in STATISTICS

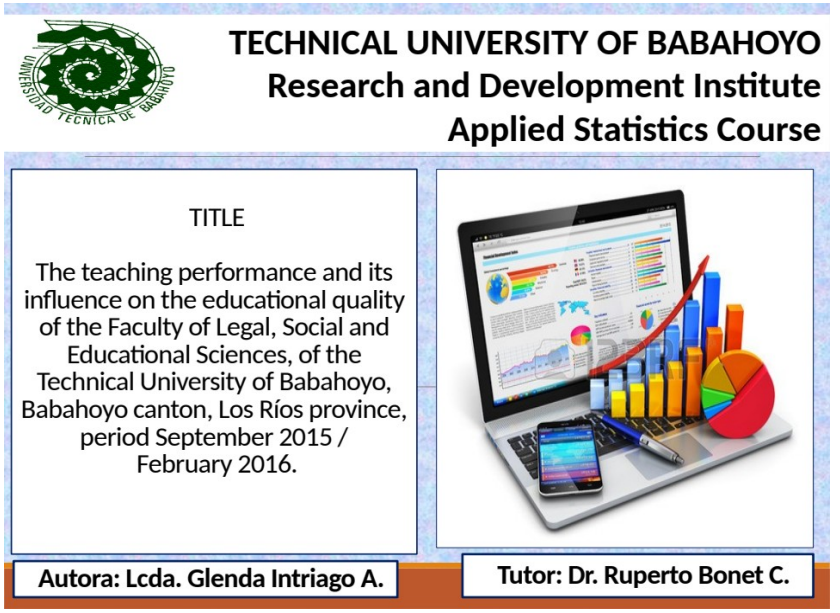
(ADVISERED PROJECTS)







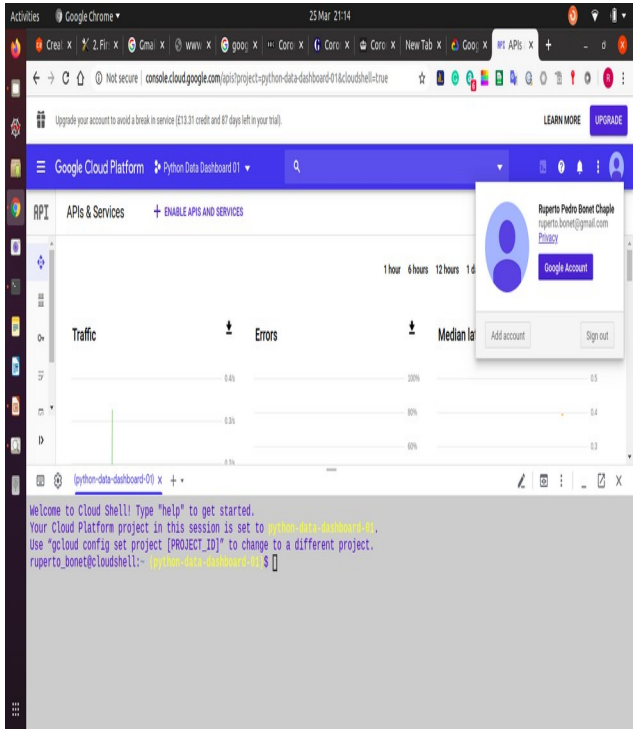


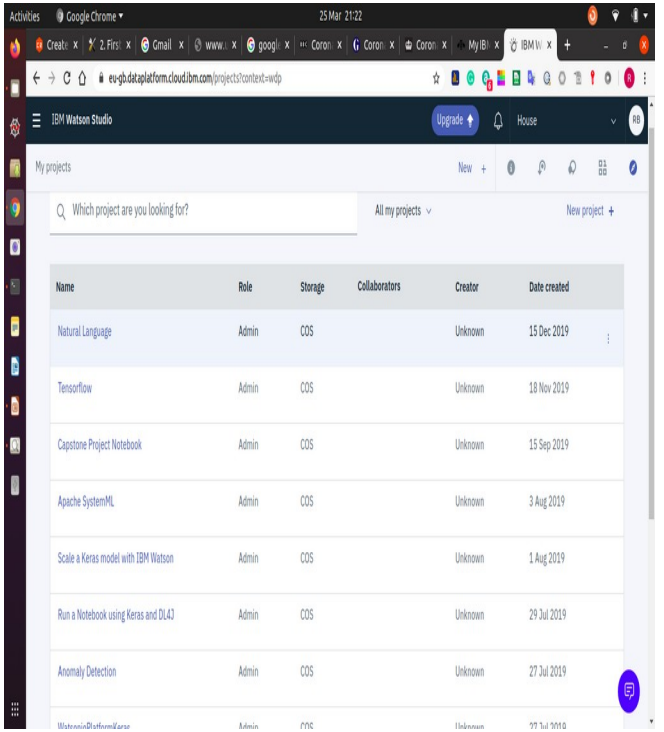


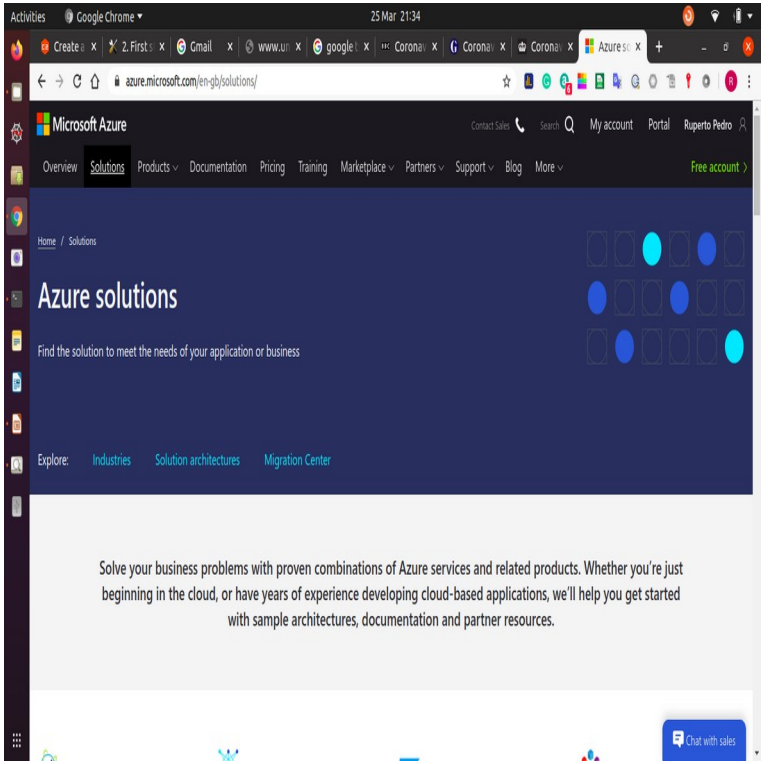
AI and BIG DATA

COURSERA CERTIFICATIONS

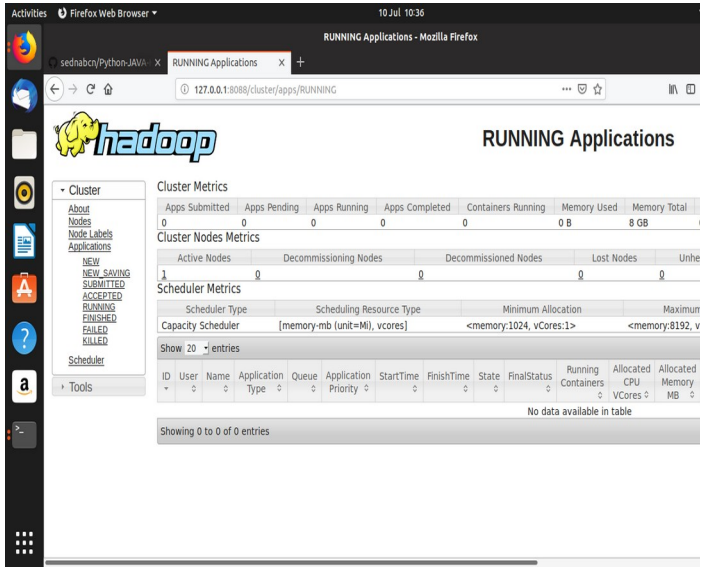
CLOUD COMPUTING







JAVA-HADOOP

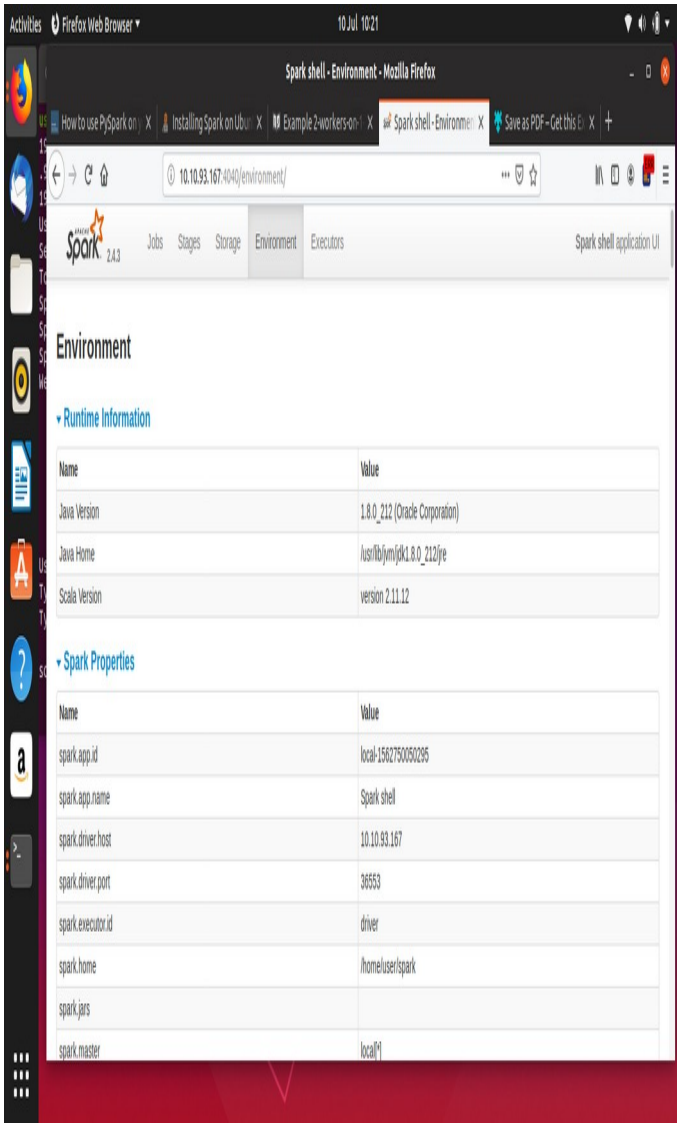


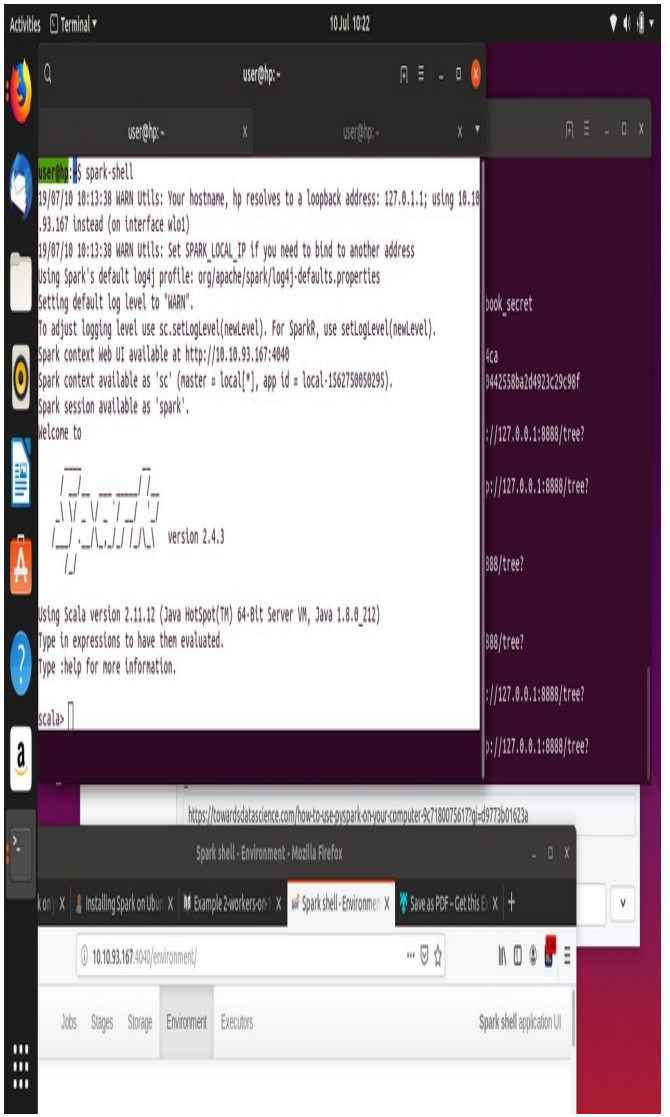

```
The Project Gutenberg EBook of Ulysses, by James Joyce

This eBook is for the use of anyone anywhere at no cost and with
almost no restrictions whatsoever. You may copy it, give it away or
re-use it under the terms of the Project Gutenberg License included
with this eBook or online at www.gutenberg.org


Title: Ulysses

Author: James Joyce

Release Date: August 1, 2008 [EBook #4300]
Last Updated: October 30, 2018

Language: English

Character set encoding: UTF-8

*** START OF THIS PROJECT GUTENBERG EBOOK ULYSSES ***



Produced by Col Choat, and David Widger.


cover


Ulysses


by James Joyce


Contents


 — I —

 [ 1 ]
 [ 2 ]
 [ 3 ]

 — II —

 [ 4 ]
 [ 5 ]
 [ 6 ]
 [ 7 ]
 [ 8 ]
 [ 9 ]
 [ 10 ]
 [ 11 ]
 [ 12 ]
 [ 13 ]
 [ 14 ]
 [ 15 ]

 — III —

 [ 16 ]
```

```
'(Lo)cra"   1
'1490 1
'1498,"     1
'35"  1
'40," 1
'A    2
'AS-IS".    1
'A_   1
'Absoluti   1
'Aesopi"    1
'Alack!     1
'Alack!"    1
'Alla 1
'Allegorical        1
'Alpha      1
'Alpha,"    1
'Alpine-glow"       1
'An   2
'And  3
'Antoni     1
'At   1
'BOILING"   2
'B_   1
'Batesian"  1
'Bathers    1
'Beta 2
'Beta"      1
'Big  1
'Bononiae   1
'But  1
'By   1
'Cave-men"  2
'Clean      1
'Come 1
'Cromagnards"       1
'Crookes    2
'DARWIN'S   2
'Dagoes,"   1
'Daily      1
'Day" 2
'De   1
'Death      1
'Defects".  1
'Defects,"  2
'Description        1
'Disposizione       2
'Doctrinal  1
'E    1
'Egli 1
'El   2
'Elements". 1
'Every      2
'Fifty-foot 1
'First      2
'Florentie  1
'For  2
'Gamma      1
'Gamma"     1
'Georgics"  1
'Go   1
'Guido      2
"Here 1
"Historia   1
"How  1
```

SPARK


Activities
Firefox Web Browser
10 Jul 10:20
Spark shell - Environment - Mozilla Firefox
10.10.93.167:4040/environment/
spark.master
local[*]
spark.repl.class.outputDir
/tmp/spark-07271a2f-35f4-48bd-9c88-408223fca7d0/repl-7bad3ce6-3883-4484-b05c-e64f587fa8da
spark.repl.class.uri
spark://10.10.93.167:36553/classes
spark.scheduler.mode
FIFO
spark.sql.catalogImplementation
hive
spark.submit.deployMode
client
spark.ui.showConsoleProgress
true
System Properties
Name
Value
SPARK_SUBMIT
true
awt.toolkit
sun.awt.X11.XToolkit
file.encoding
UTF-8
file.encoding.pkg
sun.io
file.separator
/
java.awt.graphicsenv
sun.awt.X11GraphicsEnvironment
java.awt.printerjob
sun.print.PSPrinterJob
java.class.version
52.0
java.endorsed.dirs
/usr/lib/jvm/jdk1.8.0_212/jre/lib/endorsed


Activities
Firefox Web Browser
10 Jul 10:21
Spark shell - Environment - Mozilla Firefox
10.10.93.167:4040/environment/
Jobs
Stages
Storage
Environment
Executors
Spark shell application UI
Environment
Runtime Information
Name
Value
Java Version
1.8.0_212 (Oracle Corporation)
Java Home
/usr/lib/jvm/jdk1.8.0_212/jre
Scala Version
version 2.11.12
Spark Properties
Name
Value
spark.app.id
local-1562750650295
spark.app.name
Spark shell
spark.driver.host
10.10.93.167
spark.driver.port
36553
spark.executor.id
driver
spark.home
/home/user/spark
spark.jars
spark.master
local[*]


Activities
Terminal
10 Jul 10:22
user@hp: ~
user@hp:~$ spark-shell
19/07/10 10:13:38 WARN Utils: Your hostname, hp resolves to a loopback address: 127.0.1.1; using 10.10.93.167 instead (on interface wlo1)
19/07/10 10:13:38 WARN Utils: Set SPARK_LOCAL_IP if you need to bind to another address
Using Spark's default log4j profile: org/apache/spark/log4j-defaults.properties
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
Spark context Web UI available at http://10.10.93.167:4040
Spark context available as 'sc' (master = local[*], app id = local-1562750650295).
Spark session available as 'spark'.
Welcome to
version 2.4.3
Using Scala version 2.11.12 (Java HotSpot(TM) 64-Bit Server VM, Java 1.8.0_212)
Type in expressions to have them evaluated.
Type :help for more information.
scala>
Spark shell - Environment - Mozilla Firefox
10.10.93.167:4040/environment/
Jobs
Stages
Storage
Environment
Executors
Spark shell application UI

EXPERIENCES AS A

BIG DATA ANALYTICS TUTOR

01/04/2020 Copy_of_pset1_q2.ipynb - Colaboratory

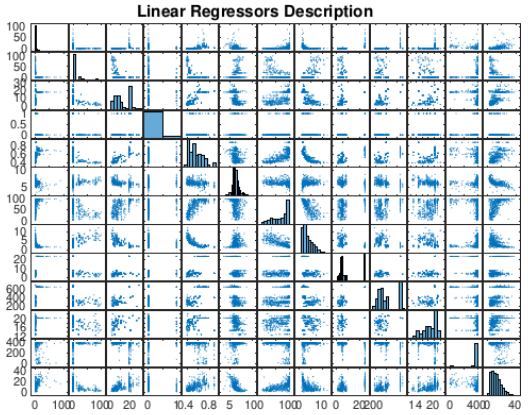
```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Sun Jan 19 19:35:36 2020

@author: sedna
"""

import pandas as pd
import matplotlib
import matplotlib.pyplot as plt
import statsmodels.api as sm
import numpy as np
import scipy.linalg as scla

matplotlib.rcParams['axes.labelsize'] = 14
matplotlib.rcParams['xtick.labelsize'] = 10
matplotlib.rcParams['ytick.labelsize'] = 12
matplotlib.rcParams['text.color'] = 'k'
#from google.colab import files
#uploaded=files.upload()
file='/content/intradayreturns.csv'
# Data
data =pd.read_csv(file, header=0)
data=data[sorted(data.columns)]

# headers
Date_time=data.columns
stocks=data.index
print(data.columns)
print(data.index)
#Sum by diary balance
print("Diary Balance","\n")
print(data.sum())

# Preparing data
data=data.replace(to_replace='infs',value=0,regex=True)
data=data.fillna(data.mean())
# Checking the dataset
"""
for ii in range(len(data.columns)):
    for jj in range(len(data.index)):
       if np.isnan(data.loc[data.index[jj],data.columns[ii]]) or not np.isfinite(
         print ("ii= %d, jj= %d".format(ii,jj))
"""
notnull_data=pd.notnull(data)
print("Nonull data in the intrayreturns dataset")
print(notnull_data)

stocks_hist=data.transpose()
z=stocks_hist[data.index[3]]
z.plot(figsize=(15,10))
plt.ylabel('Input/Output)')
plt.xlabel('Datetime Order')
```

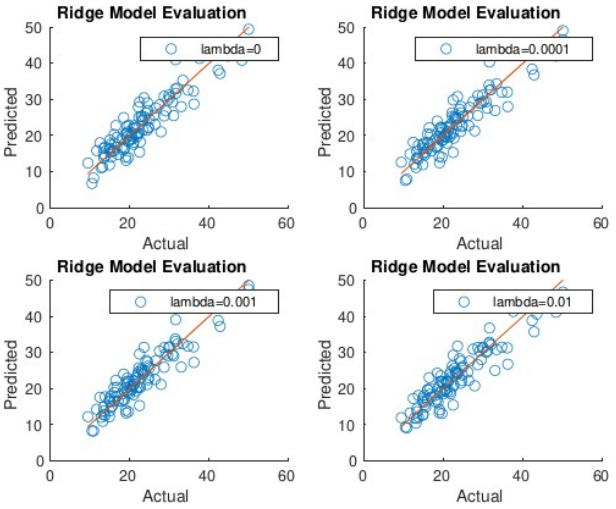
https://colab.research.google.com/drive/1OP2xh6XRY-sJ6C2Cq7YdjUWNeIsU_YG3#printMode=true 1/

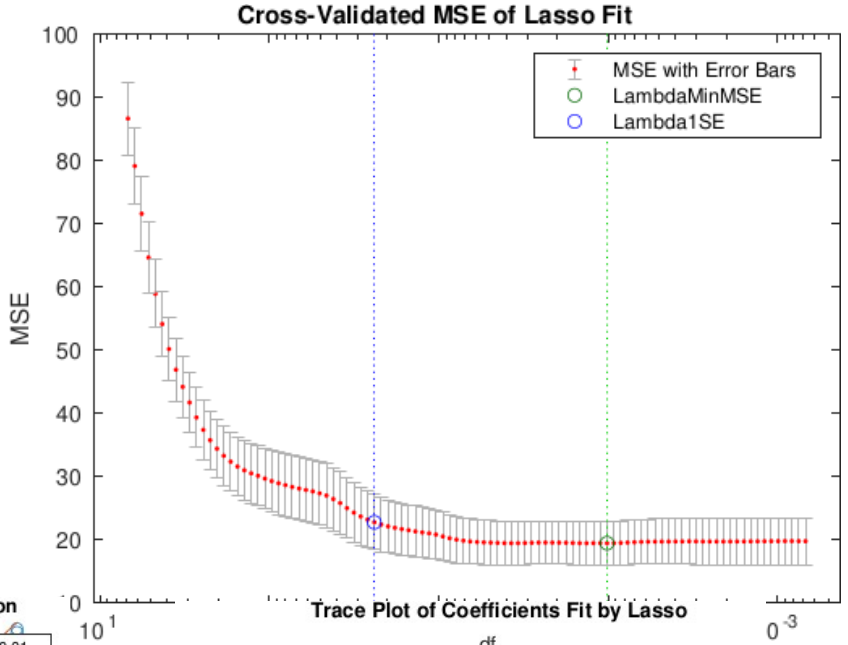
```
clear all;
clf;
%Boston housing data from: http://lib.stat.cmu.edu/datasets/boston
%originally from a paper by Harrison and Rubinfield (1978)

%  Variables in order:
%  CRIM     per capita crime rate by town
%  ZN       proportion of residential land zoned for lots over 25,000 sq.ft.
%  INDUS    proportion of non-retail business acres per town
%  CHAS     Charles River dummy variable (= 1 if tract bounds river; 0 otherwise)
%  NOX      nitric oxides concentration (parts per 10 million)
%  RM       average number of rooms per dwelling
%  AGE      proportion of owner-occupied units built prior to 1940
%  DIS      weighted distances to five Boston employment centres
%  RAD      index of accessibility to radial highways
%  TAX      full-value property-tax rate per $10,000
%  PTRATIO  pupil-teacher ratio by town
%  B        1000(Bk - 0.63)^2 where Bk is the proportion of blacks by town
%  LSTAT    % lower status of the population
%  MEDV     Median value of owner-occupied homes in $1000's

%read in the data, and define the 14 originally observed variables:


data = dlmread('housing.csv', ',', 0,0);


vCRIM = data(:,1);
vZN = data(:,2);
vINDUS = data(:,3);
vCHAS = data(:,4);
vNOX = data(:,5);
vRM = data(:,6);
vAGE = data(:,7);
vDIS = data(:,8);
vRAD = data(:,9);
vTAX = data(:,10);
vPTRATIO = data(:,11);
vB = data(:,12);
vLSTAT = data(:,13);
vMEDV = data(:,14);

%the outcome variable is Median value of owner-occupied homes in $1000's:

y=vMEDV;      %outcome variable
n=length(y);  %number of observations

%define all the regressors:

X0=[vCRIM,vCRIM.^2,vCRIM.^3,...       %cubic in crim (per capita crime rate)
   vZN>0,...                          %dummy for zn > 0
   vZN,vZN.^2,vZN.^3,...              %cubic in zn (proportion residential land zoned for lots over 25,000 ft2)
   vINDUS,vINDUS.^2,vINDUS.^3,...     %cubic in indus (proportion of non-retail business acres)
   vCHAS,...                          %Charles River dummy
   vNOX,vNOX.^2,vNOX.^3,...           %cubic in nox (nitrogen oxide concentration)
   vRM,vRM.^2,vRM.^3,...              %cubic in rm (average rooms per dwelling)
   vAGE,vAGE.^2,vAGE.^3,...           %cubic in age (proportion of owner-occupied units build pre 1940)
   vDIS,vDIS.^2,vDIS.^3,...           %cubic in dis (weighted mean of distances to 5 employment centers)
   vRAD,vRAD.^2,vRAD.^3,...           %cubic in rad (index of highway accessibility)
   vTAX,vTAX.^2,vTAX.^3,...           %cubic in tax (property tax rate per $10,000)
   vPTRATIO,vPTRATIO.^2,vPTRATIO.^3,...   %cubic in ptratio (pupil-teacher ratio by town)
   vB,vB.^2,vB.^3,...                 %cubic in black (1000*(proportion black - .63)2)
   vLSTAT,vLSTAT.^2,vLSTAT.^3,...     %cubic in lstat (percent lower SES)
   ];

p=size(X0,2);  %total number of regressors (excluding the intercept here)
```

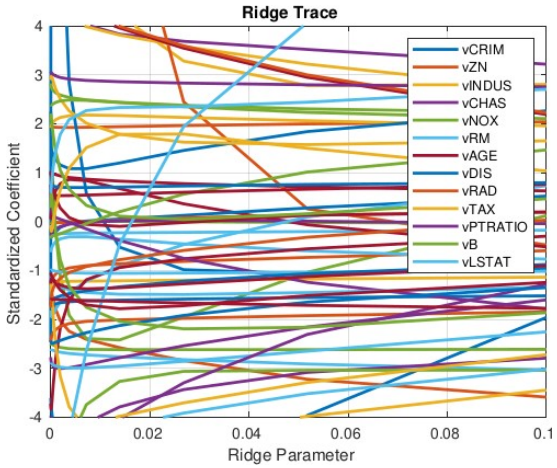
EXPERIENCES AS A BIG DATA ANALYTICS TUTOR

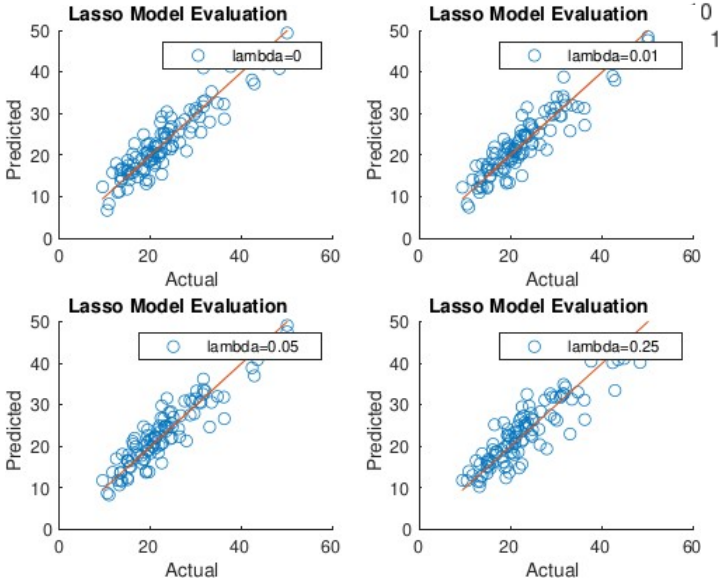
Boston Housing Dataset

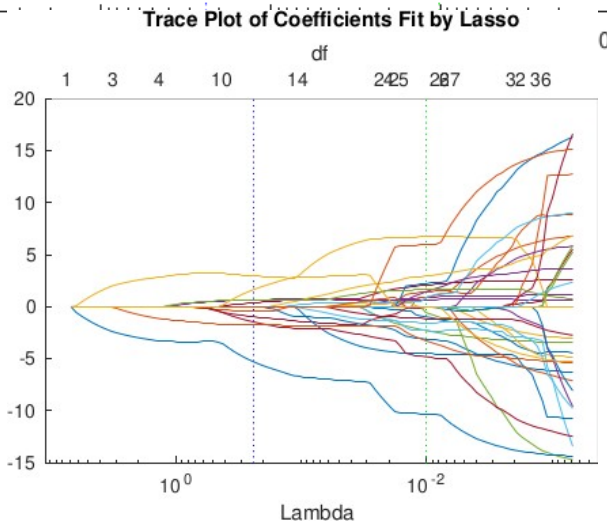








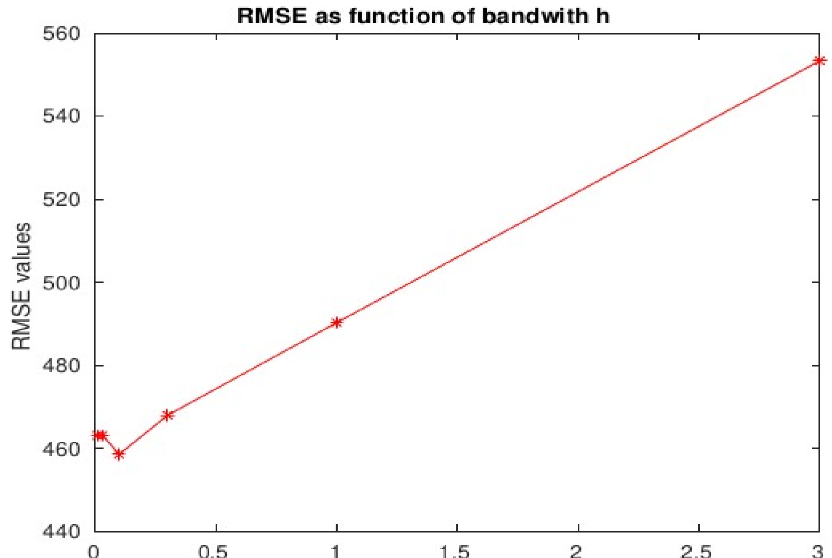


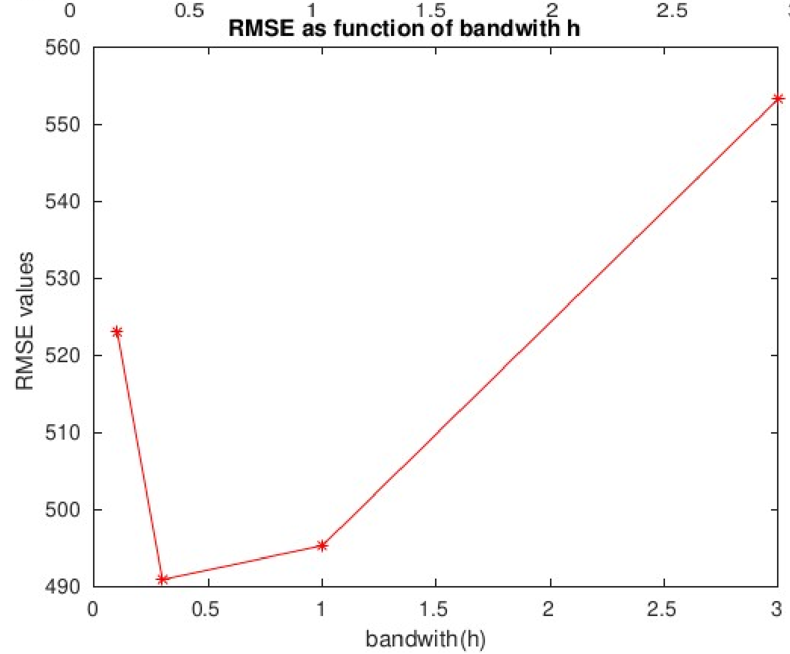


EXPERIENCES AS A BIG DATA ANALYTICS TUTOR

Smoking Effect on Birth Outcomes

```
clear all;

%DATA FROM: J. Abrevaya, "Estimating the Effect of Smoking on Birth Outcomes Using a
%Matched Panel Data Approach," Journal of Applied Econometrics, Vol. 21,
%No. 4, 2006, pp. 489-519.
%
%DATA DOWNLOADED FROM: http://qed.econ.queensu.ca/jae/2006-v21.4/abrevaya/
%
%THE DATA HAVE ALREADY BEEN RANDOMLY PERMUTED

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%READ DATA FROM FILE:
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

data = dlmread('birpanel.csv', ',', 0,0);
name={'momid3','idx','stateres','dmage','dmeduc','mplbir','nlbnl','gestat','dbirwt','cigar','smoke','male','year','married','hsgrad','somecoll','collgrad','agesq','black','adeqcode2','adeqcode3','novisit','pretri2','pret

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%DROP OBSERVATIONS FOR WHICH VARIABLE "cigar" IS UNKNOWN
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

ind=find(data(:,10)~=99);
data=data(ind,:);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%DEFINE RELEVANT VARIABLES FOR OUR EXERCISE:
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

dmage=data(:,4);    %age of mother (in years)
dmeduc=data(:,5);   %education of mother (in years)
gestat=data(:,8);   %length of gestation (in weeks)
dbirwt=data(:,9);   %birthweight (in grams)
cigar=data(:,10);   %number of cigarettes smoked per day (99=unknown)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%DEFINE OUTCOME VARIABLE = birthweight (in grams)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

y=dbirwt;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%DEFINE COVARIATES:
%   age of mother, education of mother,
%   length of gestation, number of cigarettes smoked per day
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

x=[dmage,dmeduc,gestat,cigar];

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Standardize all the covariates to have std = 1
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

for k=1:4
 x(:,k) = x(:,k) / std( x(:,k) );
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Decompose dataset into training set and validation set:
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

n=length(y);  %total sample size

yT=y(1:(n-100));
xT=x(1:(n-100),:);
nT=length(yT);      %sample size training set

yV=y((n-100+1):n);
xV=x((n-100+1):n,:);
nV=length(yV);          %sample size validation set

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%QUESTION2a)%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

h_vector=[0.01,0.03,0.1,0.3,1,3];
[ypred_vector,rmse_vector]=kernel_regression_vector(h_vector,nT,nV,xT,yT,xV,yV);

figure(1)
plot(h_vector,rmse_vector,'-*r')
xlabel('bandwith(h)')
ylabel('RMSE values')
title('RMSE as function of bandwith h')
disp (h_vector)
disp(rmse_vector)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%QUESTION2b)%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
disp('RMSE_OLS')
beta=(xT'*xT)^(-1)*xT'*yT;
xT_av=mean(xT,1);
yt_av=mean(yT);
beta_0=yt_av - dot(beta,xT_av);
y_OLS=beta_0 + xT*beta;
y_prediction=beta_0 +xV*beta;
RMSE_OLS =root_mean_square_error(y_prediction,yV);
disp(RMSE_OLS)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%QUESTION2c)%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
nrT=10000;
xrT=xT(1:nrT,:);
yrT=yT(1:nrT);
[ypred_vector,rmse_vector]=kernel_regression_vector(h_vector,nrT,nV,xrT,yrT,xV,yV);
figure(2)
plot(h_vector,rmse_vector,'-*r')
xlabel('bandwith(h)')
ylabel('RMSE values')
```





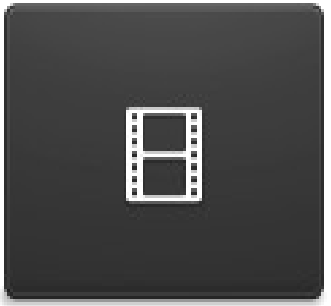
MACHINE LEARNING
SiMLeng-Statsmodels package

```
#!/usr/bin/env python3

from simulation_statsmodels import Simleng_strategies as Simleng

Simleng(0,-1).strategies()
```

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Fri Sep 28 14:46:37 2018

@author: sedna
"""
import pandas as pd
import numpy as np
from collections import OrderedDict
import matplotlib.pyplot as plt
from matplotlib import colors as mcolors
from biokit.viz import corrplot
import seaborn as sns
from scipy import stats
import statsmodels.api as sm
import statsmodels.discrete.discrete_model as smd
from statsmodels.multivariate.pca import PCA as smPCA
from sklearn.decomposition import PCA as skPCA
from sklearn import metrics
from statsmodels.stats.outliers_influence import variance_inflation_factor
from metrics_classifier_statsmodels import MetricBinaryClassifier as ac
from tools import Tools
from sklearn.discriminant_analysis import LinearDiscriminantAnalysis as LDA
from sklearn.discriminant_analysis import QuadraticDiscriminantAnalysis as QDA

colors = [ic for ic in mcolors.BASE_COLORS.values() if ic !=(1,1,1)]

class Data_base:

      def data_generation():
          """LOADING THE PIMA DATA SET"""
          data_train={}
          data_test={}
          pima_tr = pd.read_csv('pima.tr.csv', index_col =0)
          pima_te = pd.read_csv ('pima.te.csv', index_col =0)
          # Training aata
          df=pima_tr

          #df.dropna(how="all", lace=True) # drops the empty line at file-end
          columns=df.columns
          columns_train=columns[:7]
          columns_test=columns[-1]
          X_train=df[columns_train]
          Y_train=df[columns_test]
          # Testing data
          de=pima_te
          X_test=de[columns_train]
          Y_test=de[columns_test]
          data_train['train']=[columns_train,X_train,Y_train,df]
          data_test['test']=[columns_test,X_test,Y_test,de]
          return data_train,data_test

      def data_generation_binary_classification(**kwargs):
          """Generation of dummy variables to Binary Classification Task"""

          _,X_train,Y_train,_=Tools.data_extract_dict_to_list('train',**kwargs)
          _,X_test,Y_test,_=Tools.data_extract_dict_to_list('test',**kwargs)

          data_dummy_train={}
          data_dummy_test={}
          # Dummy variables to categorical variable
          V_train=Tools.data_dummy_binary_classification(Y_train,"No",0,1)
          V_test=Tools.data_dummy_binary_classification(Y_test,"No",0,1)
```

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Mon Sep 17 10:05:57 2018

@author: sedna
"""
#!/usr/bin/env python3.6
import numpy as np


class MetricBinaryClassifier():
    """Confusion matrix in statsmodels"""
    def __init__(self):
        self._paramaters=None

    def count(self,x):
          n=0
          for i in x:
              if i==True:
                  n+=1
          return n

    def metrics_binary(truth,predicted,misclassif):

        if len (truth) != len (predicted ):
            raise Exception ( " Wrong sizes ... " )
        total = len (truth )
        if total == 0:
            return 0

        #truth=np.array(truth).reshape(total,1)
        #predicted=np.array(predicted).reshape(total,1)
        # checking empty set putting array.size()>0

        f=lambda x,y: x == y==True
        t=lambda x,y: x==y
        h=lambda x:x==0
        g=lambda x:x==1

        #TRUTH :NEGATIVE
        atn=list(map(h,truth))
        #TRUTH :POSITIVE
        atp=list(map(g,truth))
        # TEST:FALSE
        apn=list(map(h,predicted))
        # TEST:TRUE
        app=list(map(g,predicted))
        # P[X=1] TRUTH POSITIVE
        pt1=atp.count(True)/total
        # P[X=0] TRUTH NEGATIVE
        pt0=atn.count(True)/total
        # P[X*=1] TEST TRUE
        pp1=app.count(True)/total
        # P[X*=0] TEST FALSE
        pp0=apn.count(True)/total

        BIAS=pp1-pt1

        fp_index=[]
        fn_index=[]
        #TN+TP
        a=list(map(t,truth,predicted))
        TNTP=a.count(True)
```

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Sat Sep 29 12:03:20 2018

@author: sedna
"""

class Tools:

    def data_unpack_kwargs(**kwargs):
        # data is a list of keys and values
        # kind='list' to list
        #kind='dict' to dict
        keyss=kwargs.keys()
        data={}
        for name in keyss:
            data[name]=kwargs[name]
        return data
    def data_list_unpack_dict(dict):
            data=[]
            for name in dict.keys():
                data_list=dict[name]
                if len(data_list)==1:
                    data_list=list(dict[name])[0]
                data.extend(data_list)
            return data
    def data_extract_dict_to_list(kind,**kwargs):
        data=Tools.data_unpack_kwargs(**kwargs)
        data_dict={}
        for k,name in enumerate(kwargs.keys()):
            if name==kind:
                data_dict[kind]=data[kind]
                data_list=Tools.data_list_unpack_dict(data_dict)
                return data_list
    def data_list_to_matrix(x,shape):
        import numpy as np
        Lx=len(x)
        Ls=len(shape)
        if Ls==2: [Nx,Ny]=shape[:]
        if Ls==3: [Nx,Ny,Nz]=shape[:]
        if Ls==2:
            matrix=np.empty((Nx,Ny))
            for ii in range(Lx):
                matrix[:,ii]=x[ii]
        assert Nx,Ny==matrix.shape
        return matrix
    def data_list_to_transform_to_integer(x):
        """get a new type for the members of the list """
        xx=[]
        for ii in x:
            xx.append(int(ii))
        return xx
    def add_data_list_of_several_lists(nn,x):
        LEN=0
        for ii in range(nn):
            LEN+=len(x[ii])
        return LEN,print(LEN)
    def add_data(x):
        import numpy as np
        n,m=x.shape
        x=np.array(x)
        sum_by_rows=[x[ii,:].sum() for ii in range(n)]
        return np.asarray(sum_by_rows).astype(float)
```



SiMLeng PROJECT

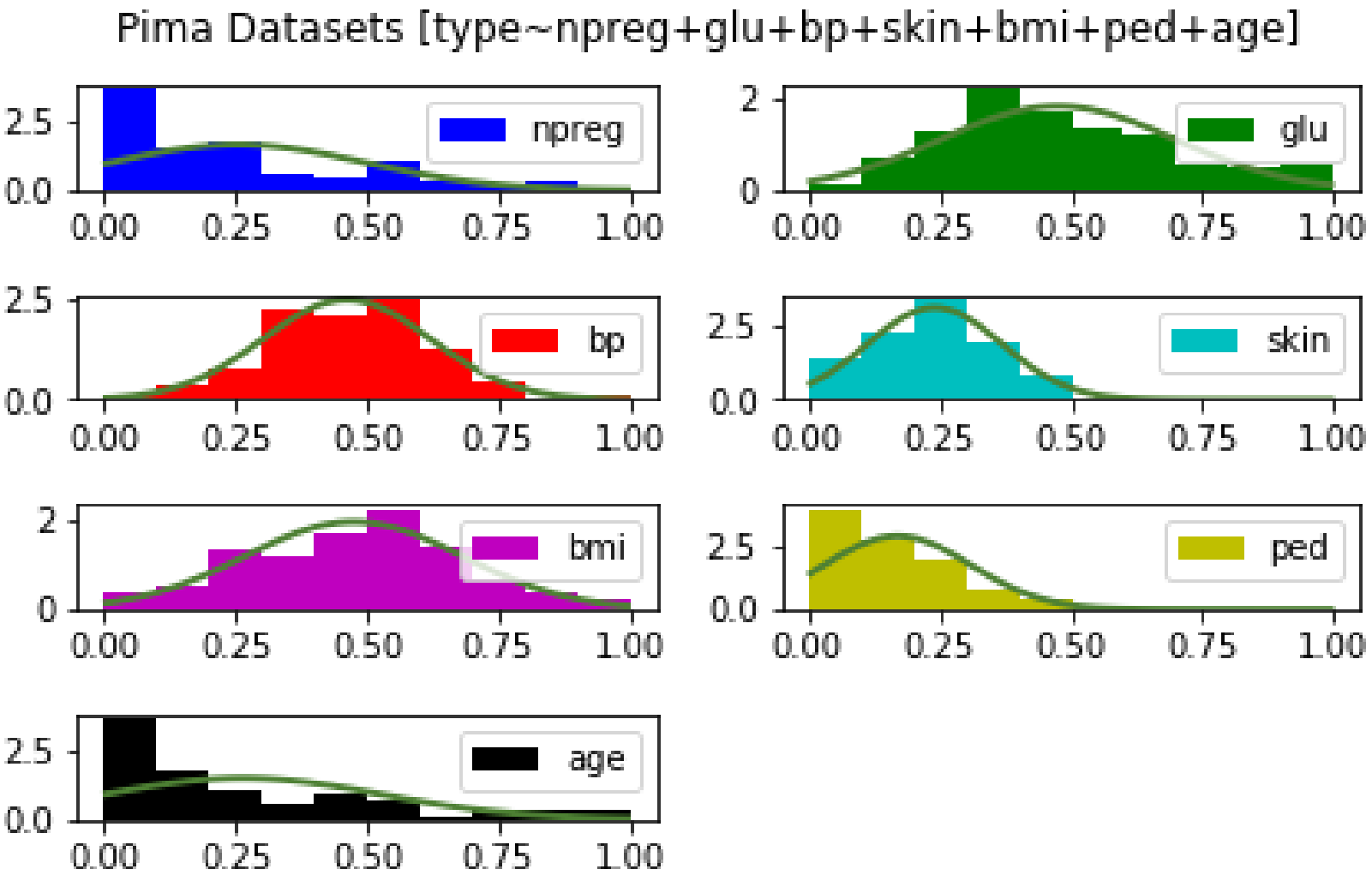
PIMA INDIAS DATASET

CLASSIFICATION TRAINING

RUPERTO P. BONET

rpbonetch@gmail.com

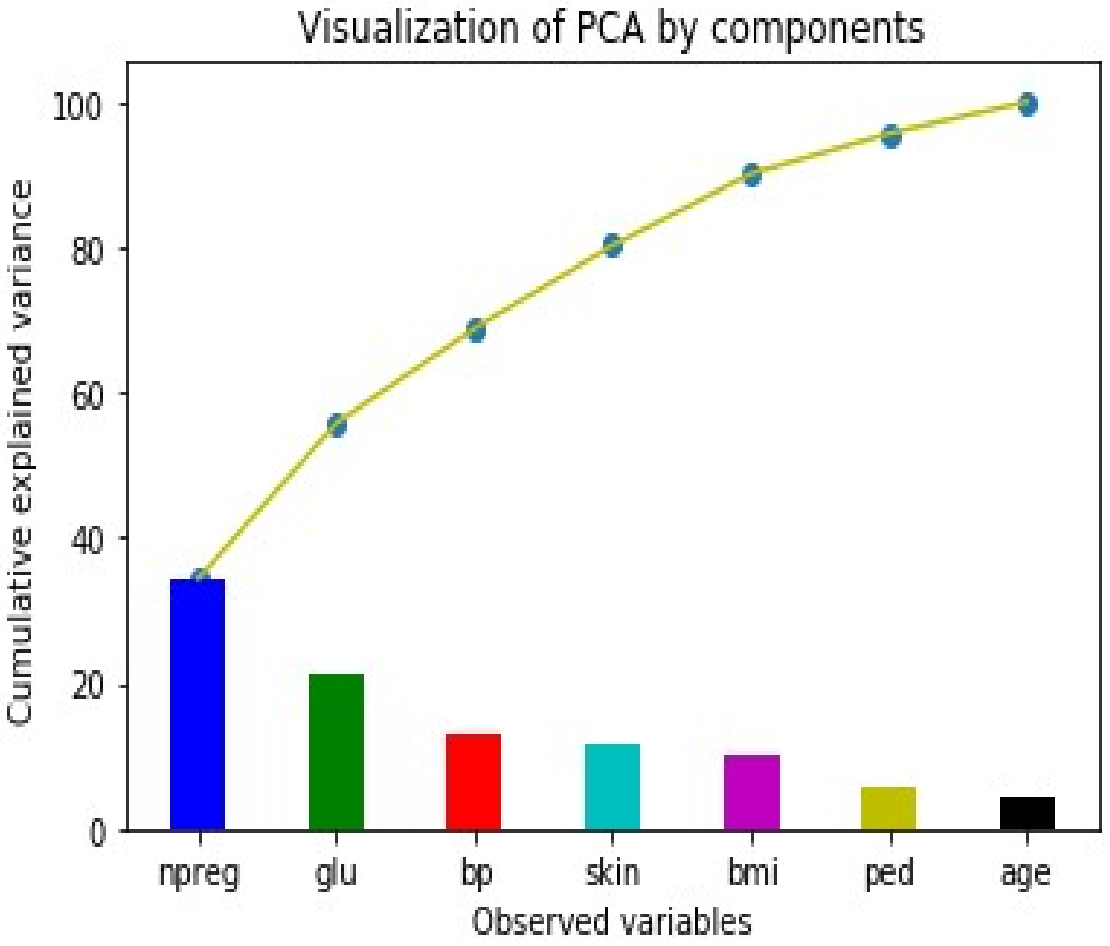
Predictors Histogram-Normal pdf



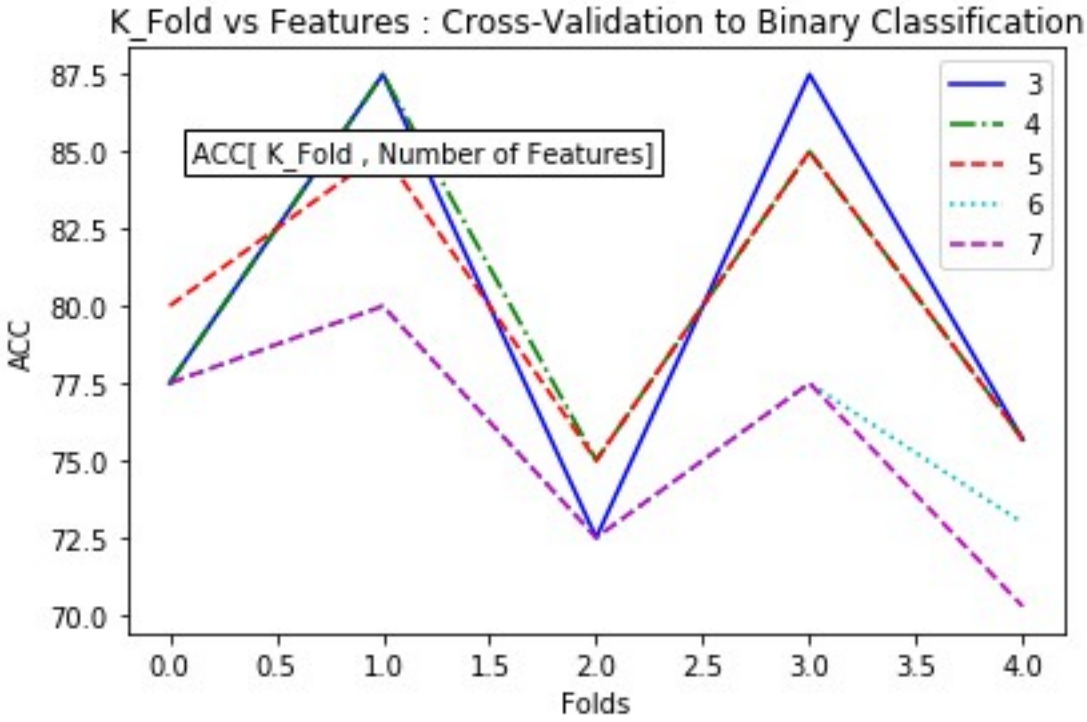
FEATURES-SELECTION
PCA (ncomp=7)

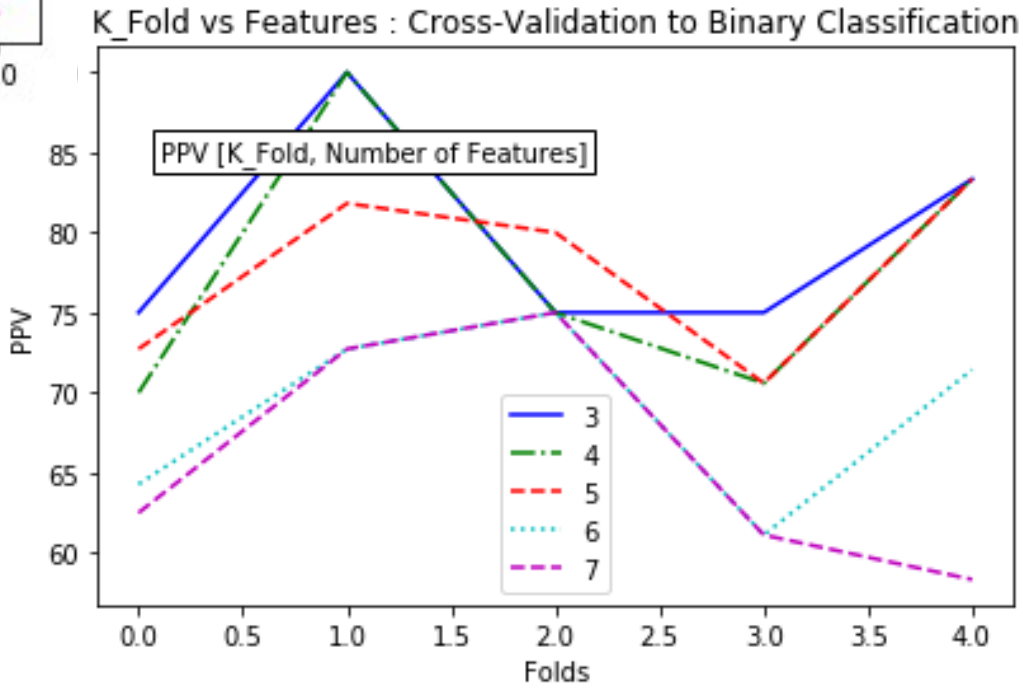
Eigenvalues in descending order

	obs_var	Eig.Abs	cum_var_exp
0	npreg	481.852	34.418
1	glu	299.289	55.796
2	bp	182.374	68.823
3	skin	159.994	80.251
4	bmi	138.027	90.110
5	ped	77.964	95.679
6	age	60.500	100.000

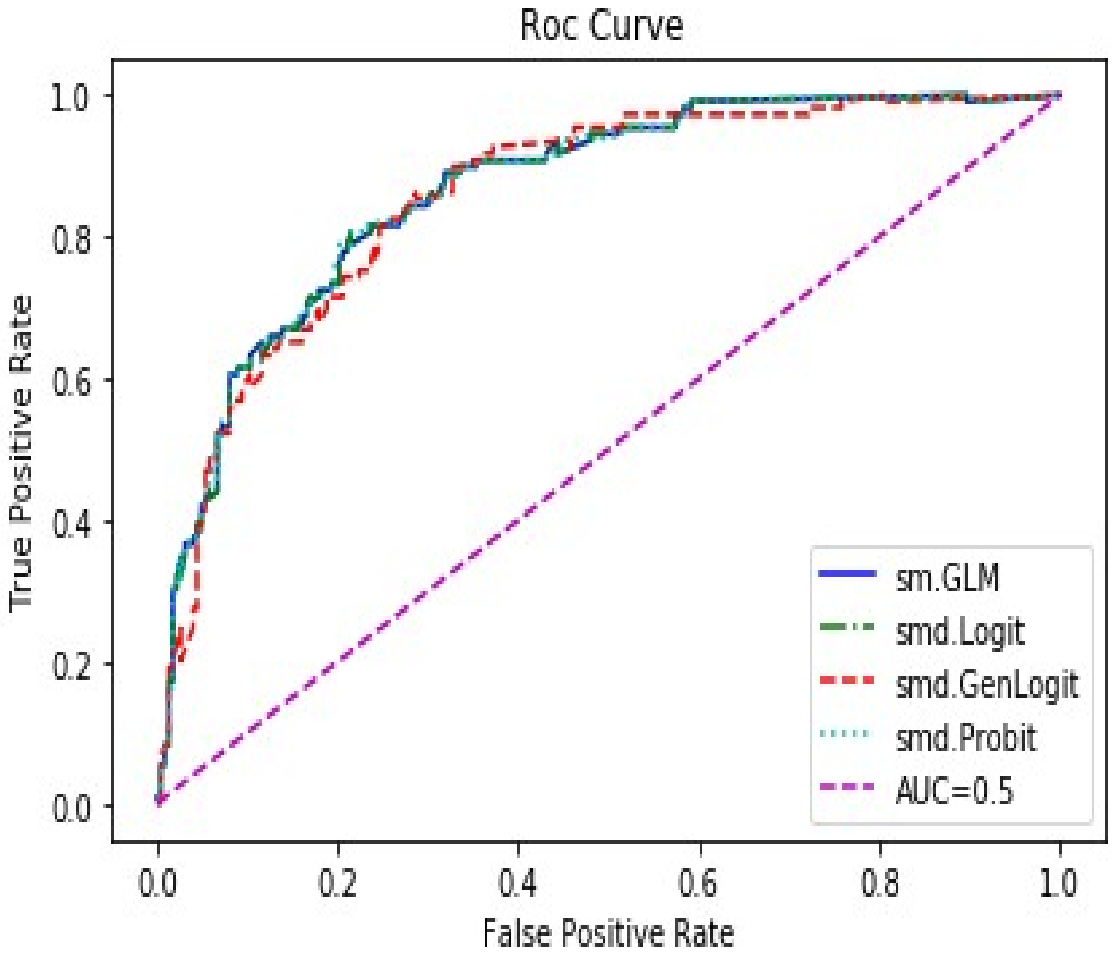


CROSS VALIDATION K_FOLD=5





BINARY CLASSIFICATION RESULTS



PRELIMINARY CONCLUSIONS

- Glu and ped are most relevant predictors.
- Genlogistic function with c lower than 0.5 has strong influence on the detection of TRUE NEGATIVE, and by consequence, it improves the prediction performance.
- The best prediction with 3 predictors is 81.93%, which has been obtained using GenLogistic function (with c=0.2) on the features subset: npreg, bmi and ped.
- The best prediction with 4 predictors is 80.12%.
- This prediction on the features subset: npreg, glu, bmi and ped has been obtained applying two different strategies:
  -QDA to the original variables, and
  -Logit to the transformed variables by PCA with four components
- This result is consistent for the study and this features subset represents 63.22% from training data information.

SiMLeng PROJECT

Genlogistic vs Logistic

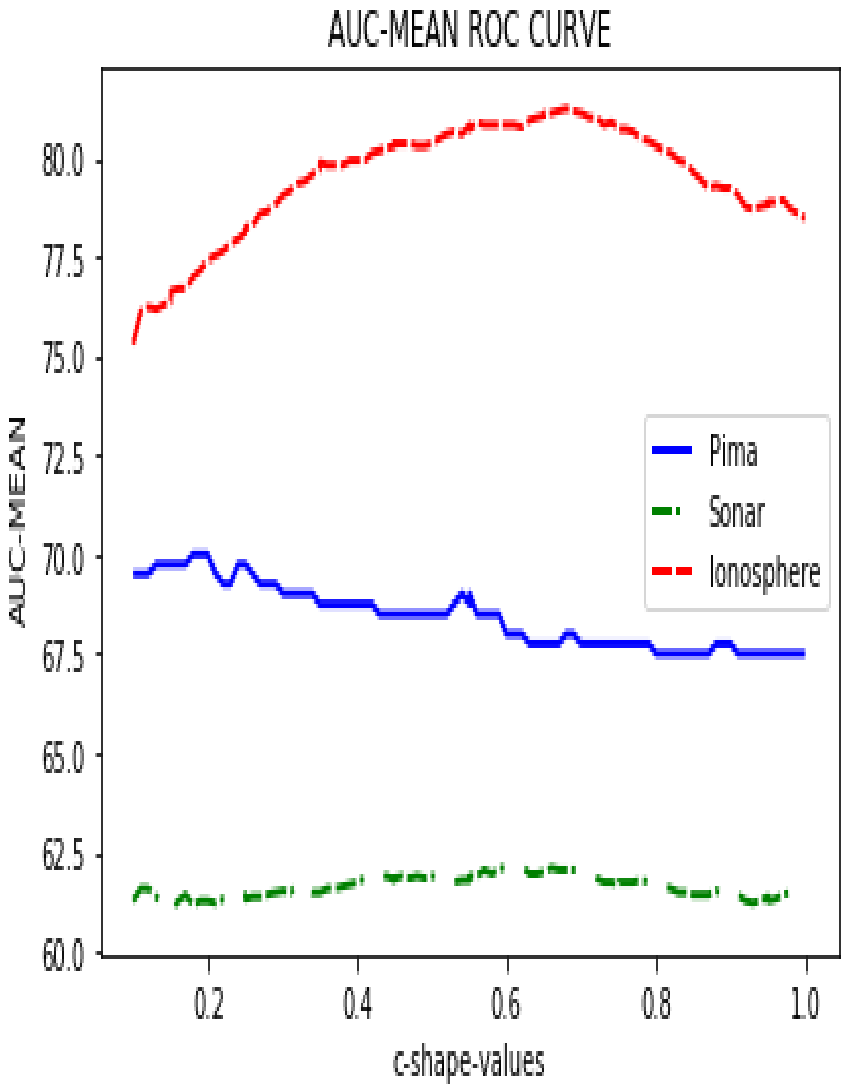
NOTE ON

BINARY CLASSIFICATION

RUPERTO P. BONET

rpbonetch@gmail.com

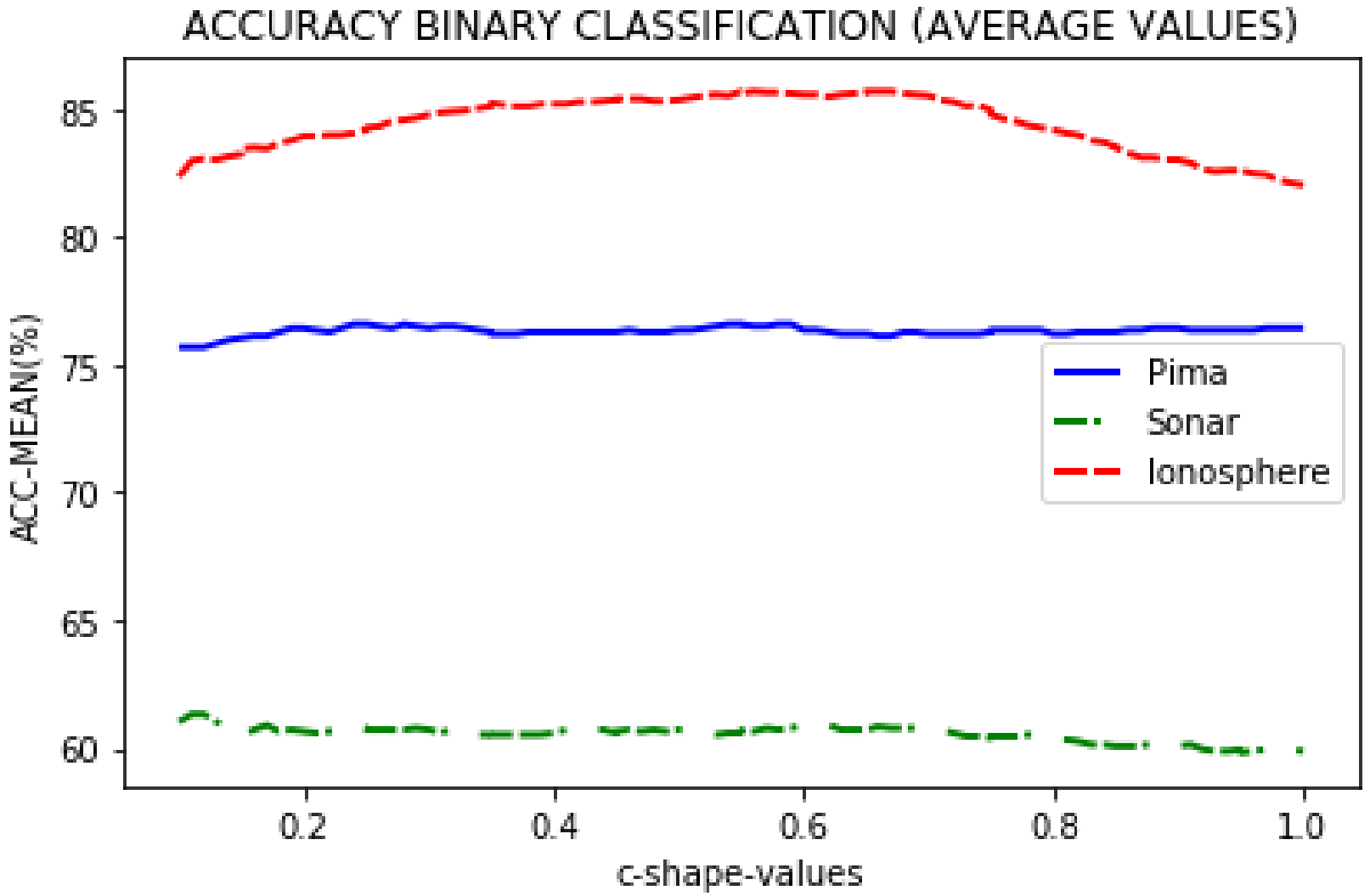
COMPARISON BETWEEN DATASETS
AUC MEAN(%)



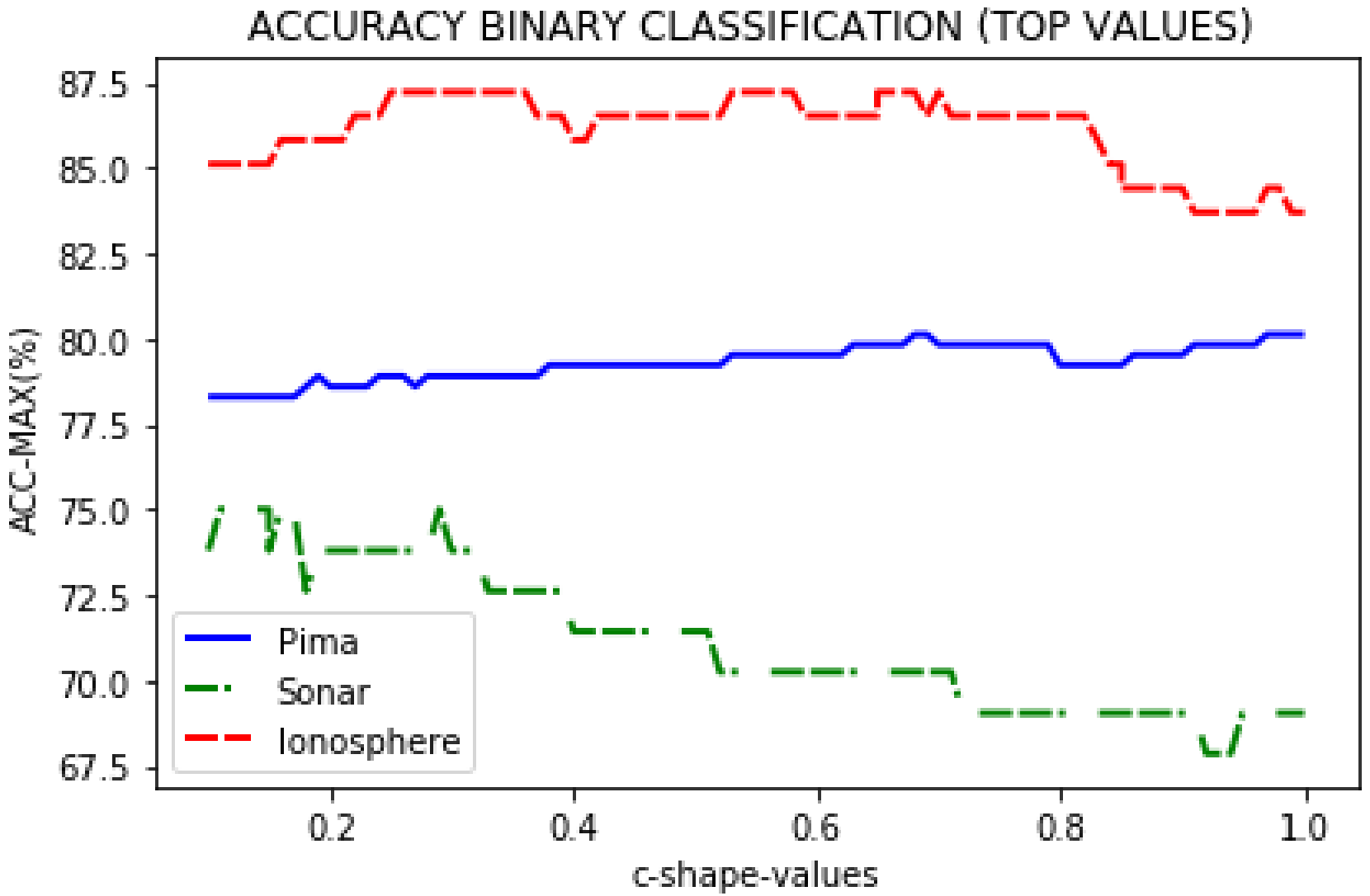

```python
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Sun Nov  4 22:17:43 2018
These classes are added to statsmodels package

@author: sedna
"""
import numpy as np
from statsmodels import base
from statsmodels.discrete.discrete_model import BinaryModel

class GenLogit(BinaryModel):
    __doc__ = """
    Binary choice genlogit model

%(params)s
    %(extra_params)s

    Attributes
    ----------
    endog : array
        A reference to the endogenous response variable
    exog : array
        A reference to the exogenous design.
    """ % {'params' : base._model_params_doc,
           'extra_params' : base._missing_param_doc}


    def get_c_value(self):
        #self.c= float(input("Shape genlogistic: "))
        #self.c=0.35
        self.c=0.2
        return self.c


    def cdf(self, X):
        """
        The genlogistic cumulative distribution function

        Parameters
        ----------
        X : array-like
            `X` is the linear predictor of the logit model.  See notes.
        c :  `c` is as a shape parameter c>0 [c=1 becomes to logistic case]
        Returns
        -------
         1/(1 + exp(-x))^c

        Notes
        -----
        In the genlogit model,

        .. math:: l\Lambda\left(x^{\prime}\beta,c\right)=\text{Prob}\left(Y=1|x\right)=\frac{1}{\left(1+e^{-x^{\prime}\beta}\right)^{c}}
        """

        c = self.get_c_value()
        X = np.asarray(X)
        return 1/(1+np.exp(-X))**c

    def pdf(self, X):
        """
        The logistic probability density function


        Parameters
        ----------
        X : array-like
            `X` is the linear predictor of the logit model.  See notes.
        c :  `c` as a shape parameter
        Returns
        -------
        pdf : ndarray
            The value of the Logit probability mass function, PMF, for each
```

```python
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Fri Nov  9 16:41:04 2018

@author: sedna
"""

import numpy as np
import pandas as pd
import statsmodels.api as sm
import statsmodels.discrete.discrete_model as smd
from statsmodels.multivariate.pca import PCA as smPCA
from metrics_classifier import MetricClassifier
from sklearn import metrics
from sklearn.model_selection import train_test_split
from collections import OrderedDict
import matplotlib.pyplot as plt
from tools import Tools
from matplotlib import colors as mcolors
# colors for grahics with matplolib and plotly

testsize=0.4
treshold=0.90
NDataSets=3

Datasets=['Pima','Sonar','Ionosphere']

ac=MetricClassifier()

def show_best_pca_predictors(x,ncomp,treshold):
        try:
            eigenvalues=smPCA(x,ncomp).eigenvals
        except:
            eigenvalues=smPCA(x,ncomp,method='eig').eigenvals

        order=[i for i,vals in sorted(enumerate(np.abs(eigenvalues)), \
                                      key=lambda x:x[1],reverse=True)]
        eigenvalues_sorted=[eigenvalues[order[ii]] for ii in range(ncomp)]
        features_sorted=[x.columns[order[ii]] for i in range(ncomp)]

        tot = sum(eigenvalues_sorted)
        best_predictors=[]
        cum_var_exp=0.0
        for i,i in enumerate(eigenvalues_sorted[:]):
            var_exp=(i/tot)
            cum_var_exp+=var_exp
            if cum_var_exp < treshold:
                best_predictors.append(features_sorted[ii])
        return best_predictors

def plot_matrix_matrix(X,Y,Title,xlabel,ylabel,Labels,\
                       Linestyle,kind,scale,grid,text,boxstyle,mode):
    """Daw matrix vs matrix."""
    """
    kind of graphic
    0-plot
    1-scatter
    2-stem
    scale of plot
    'equal'
    'Log'
    'semilogy'
    'semilogx'
    'Loglog'
    mode of graphic
    0-full
    1-split
    """

    if (len(Y.columns)>7 and mode==0):
        colors = [kc for kc in mcolors.CSS4_COLORS.values() if kc !=(1,1,1)]
        keys   = [kc  for kc in mcolors.CSS4_COLORS.keys() \
        if mcolors.CSS4_COLORS[kc]!=(1,1,1)]
```

COMPARISON BETWEEN DATASETS
ACC MEAN(%)



COMPARISON BETWEEN DATASETS
ACC TOP VALUES(%)



PRELIMINARY CONCLUSIONS

Results seem to indicate that Genlogistic function could reduce the false positive/ negative rate more than Logistic predictions. The following table shows the above observations:

Datasets	Pima Indias			Sonar			Ionosphere		
TOP VALUES(%)	AUC	ACC	FPFN	AUC	ACC	FPFN	AUC	ACC	FPFN
Logistic (c=1)	73.0	80.12	19.88	70.0	69.05	30.95	81.0	83.69	16.31
Genlogistic(0<c<1)	77.0	80.12	19.88	75.0	75.00	25.00	83.0	87.23	12.77

SiMLeng PROJECT

Lumped Sample Collocation Prediction Method

RUPERTO P. BONET

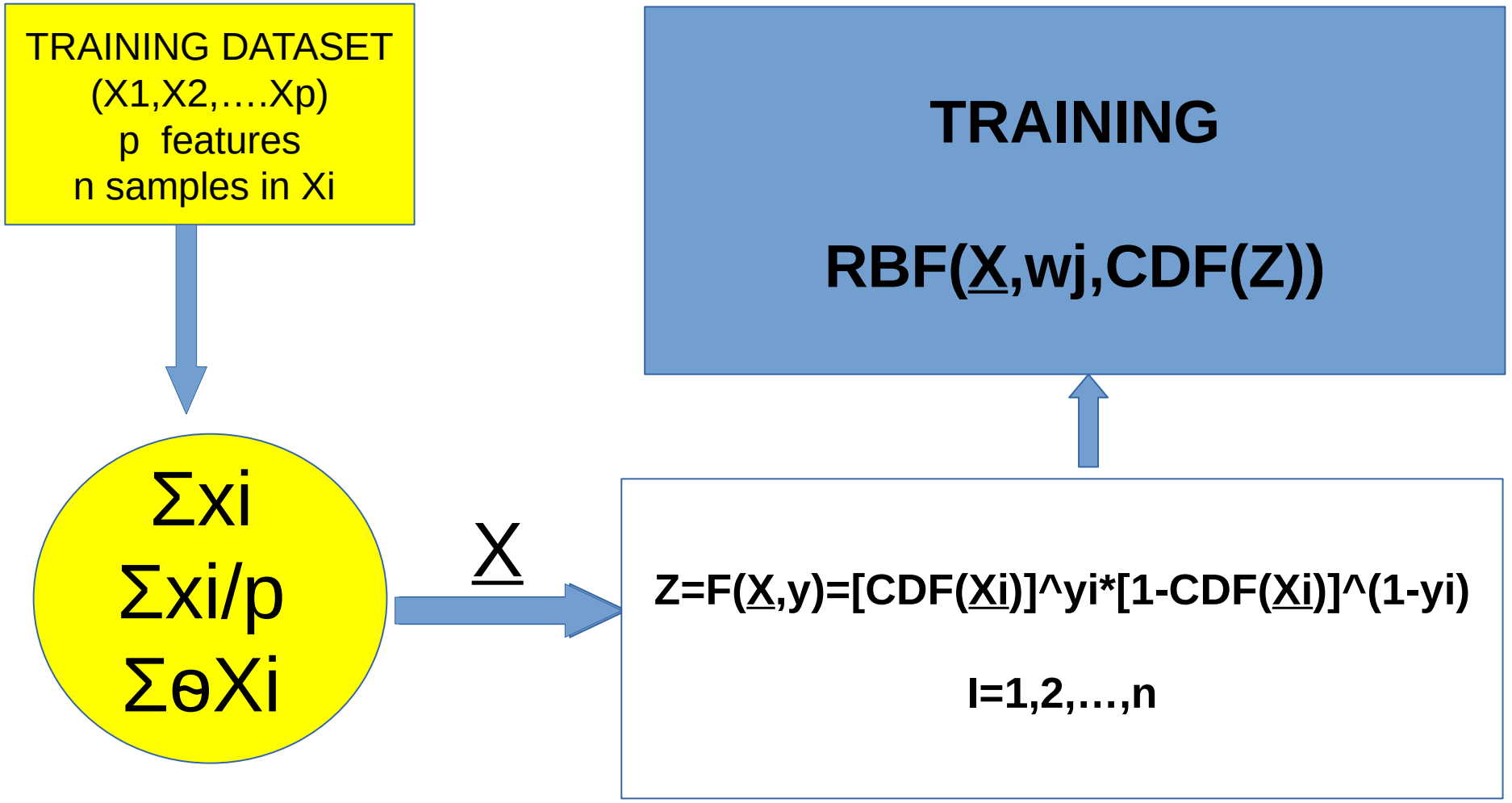
rpbonetch@gmail.com

INTRODUCTION

A numerical method based on PCA features selection and lumping transformations of samples in combination with One-Dimensional Radial Basis Functions Collocation method is implemented.

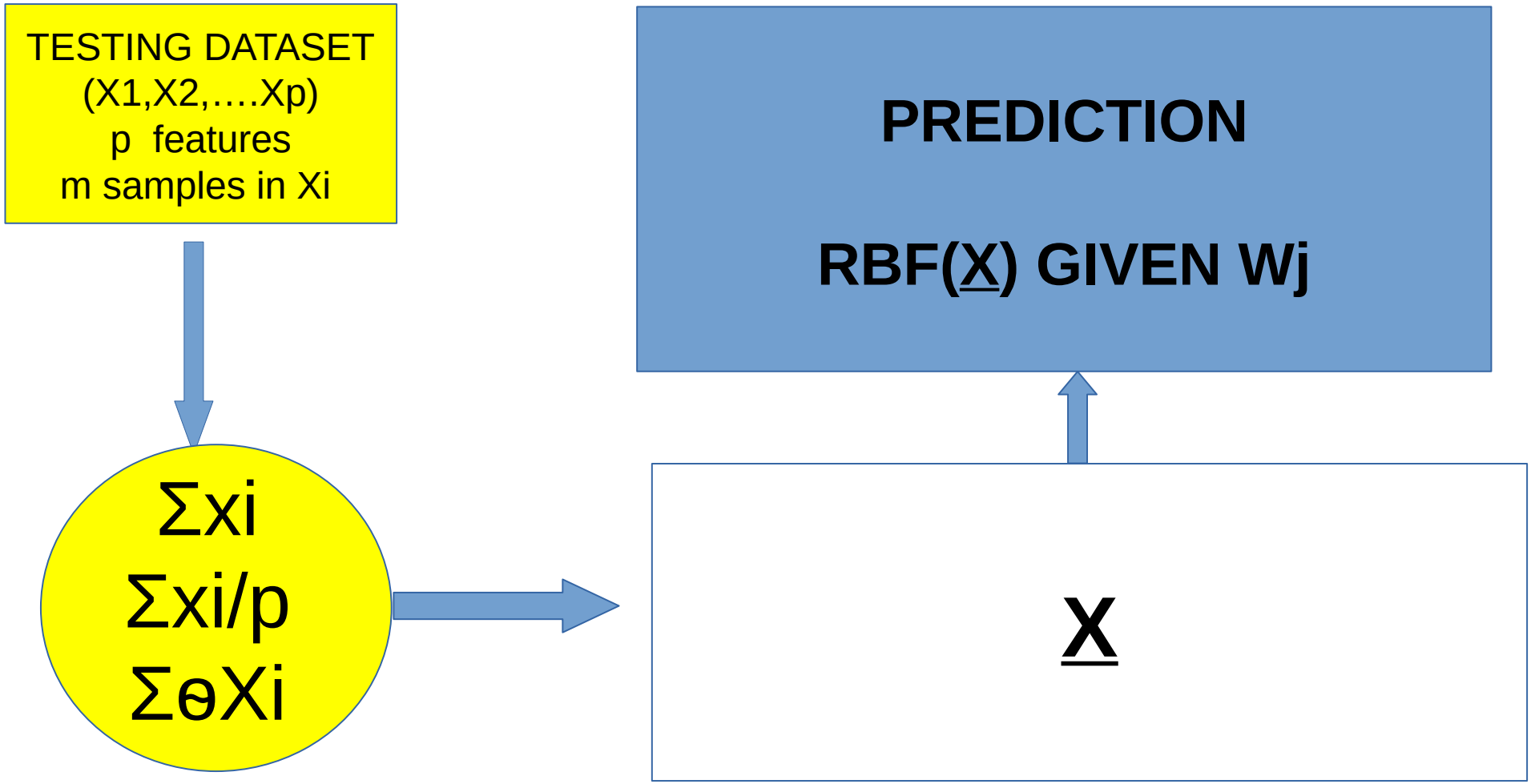
This experiment has been designed using Pima Indias, Sonar and Ionosphere datasets, which are related to making prediction and binary classification.

FEATURES-SAMPLE LUMPING COLLOCATION PREDICTION METHOD



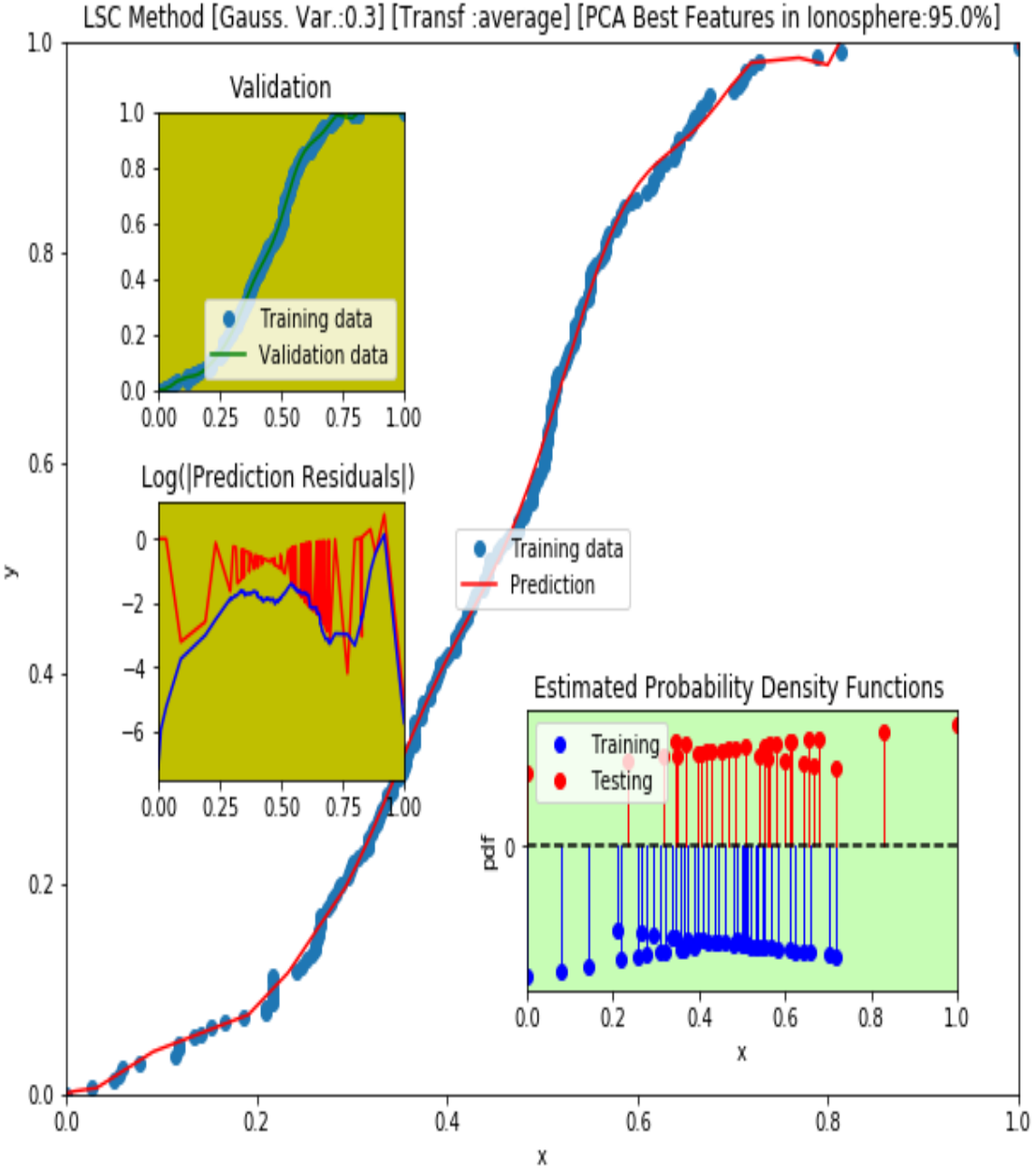
TRANSFORMATIONS

*FEATURES-SAMPLE LUMPING COL*LOCATION PREDICTION METHOD



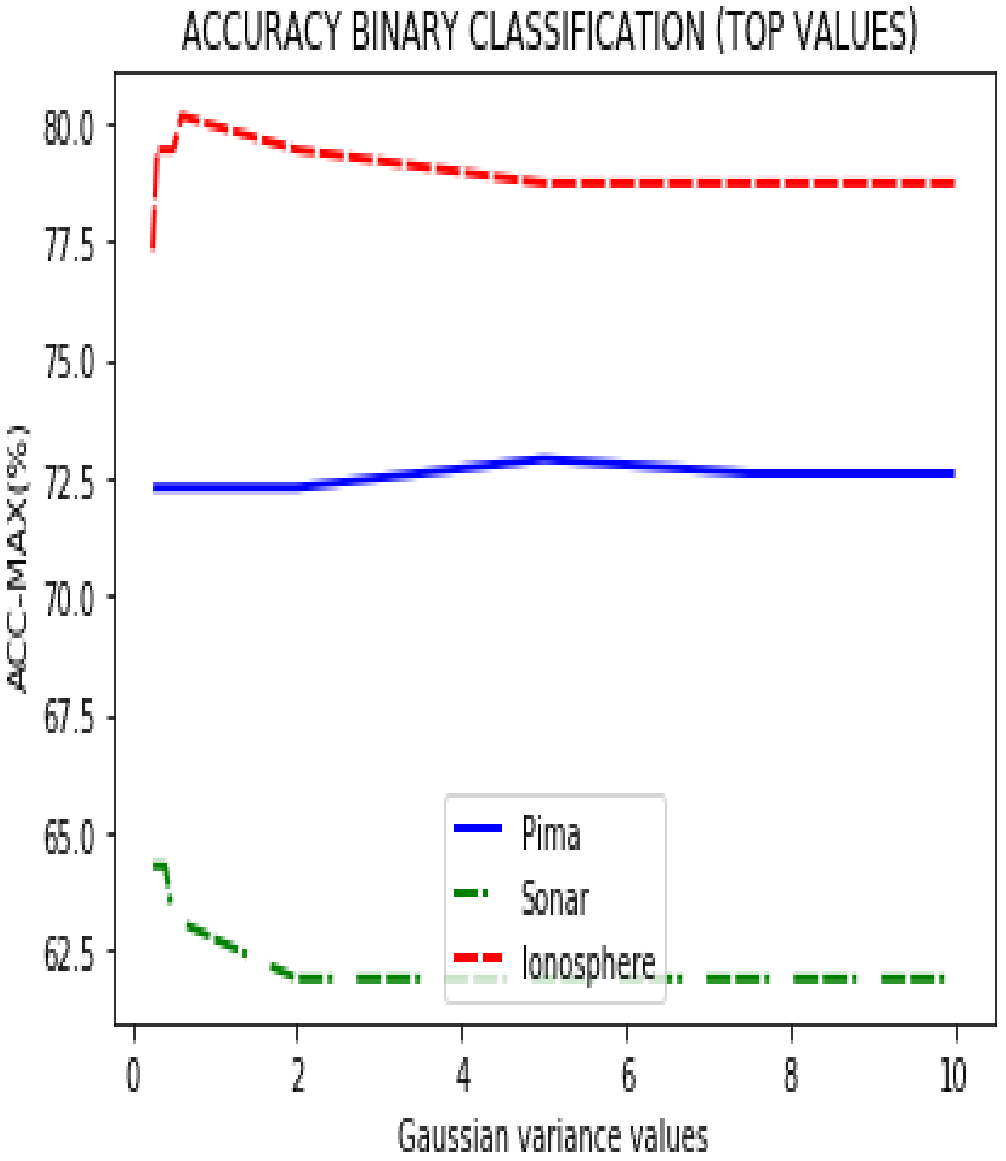
TRANSFORMATIONS

LSC METHOD [RBF+ Const Poly]




```python
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Sat Dec  1 19:18:27 2018

@author: sedna
"""
from IPython import get_ipython

get_ipython().magic('reset -sf')

import numpy as np
import pandas as pd
import statsmodels.api as sm
import statsmodels.discrete.discrete_model as smd
from statsmodels.stats.outliers_influence import variance_inflation_factor
from statsmodels.multivariate.pca import PCA as smPCA
from metrics_classifier import MetricClassifier
from sklearn import metrics
from sklearn.model_selection import train_test_split
from collections import OrderedDict,defaultdict
import matplotlib.pyplot as plt
from tools import Tools
from matplotlib import colors as mcolors
# colors for grahics with matplolib and plotly
from smt.surrogate_models import rbf


testsize=0.4
#treshold=0.95
NDataSets=3

features_treshold=[95.0]
#features_treshold=[90.0]
samples_transform=['sum','average','optim']
#samples_transform=['sum']

Datasets=['Pima','Sonar','Ionosphere']
Datasets=['Ionosphere']

ac=MetricClassifier()

def show_best_pca_predictors(x,ncomp,treshold):
        try:
            eigenvalues=smPCA(x,ncomp).eigenvals
        except:
            eigenvalues=smPCA(x,ncomp,method='eig').eigenvals

        order=[ii for ii,vals in sorted(enumerate(np.abs(eigenvalues)), \
                                        key=lambda x:x[1],reverse=True)]
        eigenvalues_PCA_sorted=[eigenvalues[order[ii]] for ii in range(ncomp)]
        features_PCA_sorted=[x.columns[order[ii]] for ii in range(ncomp)]

        key_colinearity= variance_influence_factors(x,features_PCA_sorted,0.95)

        features_sorted=[]
        eigenvalues_sorted=[]
        for feat,eig in zip(features_PCA_sorted,eigenvalues_PCA_sorted):
                    if feat in key_colinearity[0]:
                        if feat==key_colinearity[0][0]:
                            features_sorted.append(feat)
                            eigenvalues_sorted.append(eig)
                    else:
                        features_sorted.append(feat)

                        eigenvalues_sorted.append(eig)


        tot = sum(eigenvalues_sorted)
        best_predictors=[]
        cum_var_exp=0.0
        for ii,i in enumerate(eigenvalues_sorted[:]):
            var_exp=(i/tot)
            cum_var_exp+=var_exp
            if cum_var_exp < treshold:
                    best_predictors.append(features_sorted[ii])
        return best_predictors

def variance_influence_factors(x,predictors,vif_treshold):
```

LSC METHOD [RBF +CONST POLY] COMPARISON BETWEEN DATASETS




```python
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Fri Nov  9 16:41:04 2018

@author: sedna
"""


import numpy as np
import pandas as pd
import statsmodels.api as sm
import statsmodels.discrete.discrete_model as smd
from statsmodels.multivariate.pca import PCA as smPCA
from metrics_classifier import MetricClassifier
from sklearn import metrics
from sklearn.model_selection import train_test_split
from collections import OrderedDict
import matplotlib.pyplot as plt
from tools import Tools
from matplotlib import colors as mcolors
# colors for grahics with matplolib and plotly

testsize=0.4
treshold=0.90
NDataSets=3

Datasets=['Pima','Sonar','Ionosphere']

ac=MetricClassifier()

def show_best_pca_predictors(x,ncomp,treshold):
        try:
            eigenvalues=smPCA(x,ncomp).eigenvals
        except:
            eigenvalues=smPCA(x,ncomp,method='eig').eigenvals

        order=[ii for ii,vals in sorted(enumerate(np.abs(eigenvalues)), \
                                key=lambda x:x[1],reverse=True)]
        eigenvalues_sorted=[eigenvalues[order[ii]] for ii in range(ncomp)]
        features_sorted=[x.columns[order[ii]] for ii in range(ncomp)]

        tot = sum(eigenvalues_sorted)
        best_predictors=[]
        cum_var_exp=0.0
        for ii,i in enumerate(eigenvalues_sorted[:]):
            var_exp=(i/tot)
            cum_var_exp+=var_exp
            if cum_var_exp < treshold:
                best_predictors.append(features_sorted[ii])
        return best_predictors

def plot_matrix_matrix(X,Y,Title,xlabel,ylabel,Labels,\
                       Linestyle,kind,scale,grid,text,boxstyle,mode):
    """Draw matrix vs matrix."""
    """
    kind of graphic
    0-plot
    1-scatter
    2-stem
    scale of plot
    'equal'
    'Log'
    'semilogy'
    'semilogx'
    'Loglog'
    mode of graphic
    0-full
    1-split
    """

    if (len(Y.columns)>7 and mode==0):
        colors = [ic for ic in mcolors.CSS4_COLORS.values() if ic !=(1,1,1)]
        keys   = [kc  for kc in mcolors.CSS4_COLORS.keys() \
        if mcolors.CSS4_COLORS[kc]!=(1,1,1)]
```

PRELIMINARY CONCLUSIONS

The Lumped Sample Collocation Prediction Method is a global method which depends of Cumulative Distribution Function [cdf] used to compute the learning curve. Results seem to indicate that it has the same precision as linear methods. Best results were obtained adding features/sample columns and selecting the training points/labels as collocation points. Radial Basis Functions are employed to get the learning parameters.

https://github.com/sednabcn/SiMLeng/tree/BinaryClassification-LSCollocation

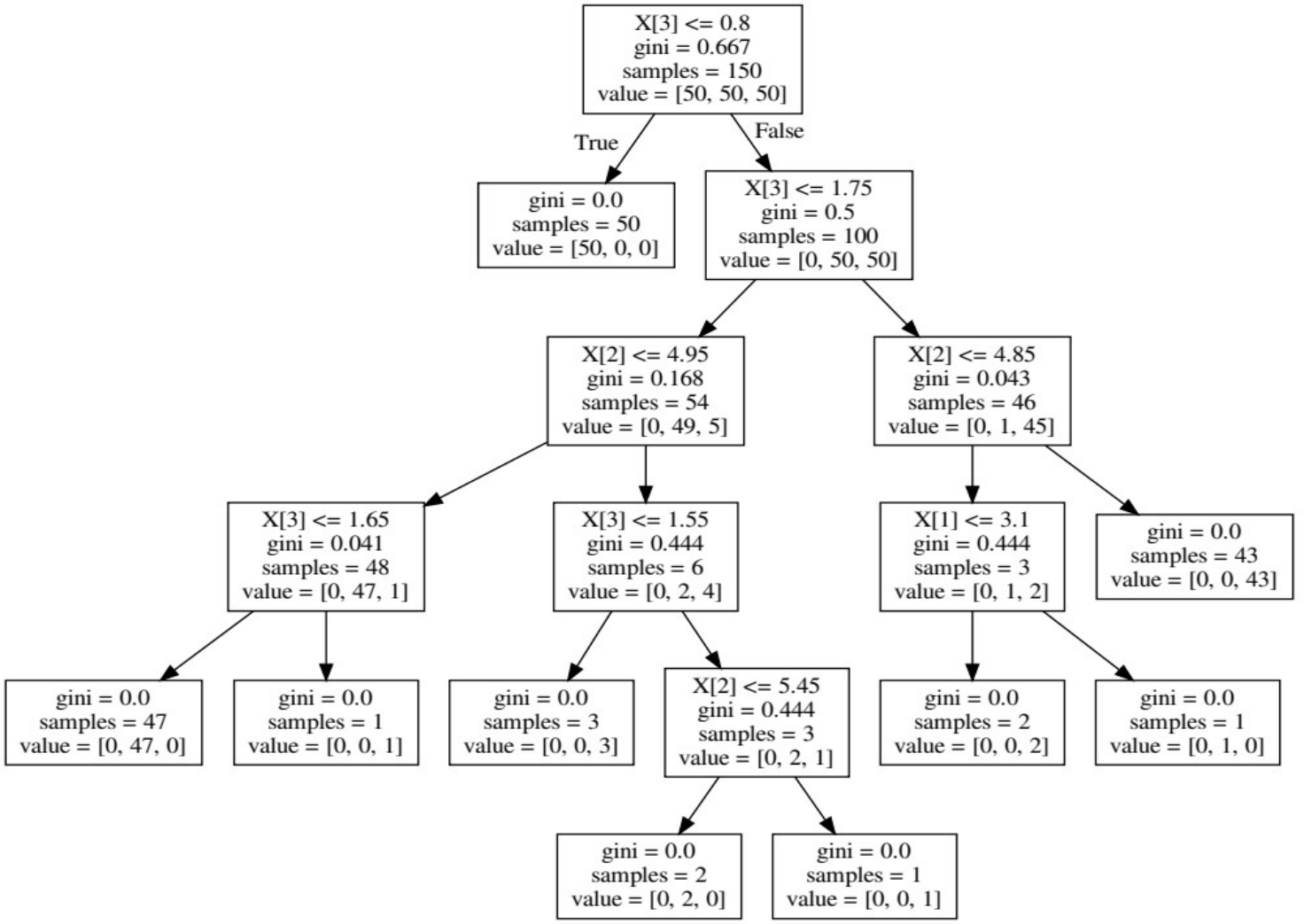
MACHINE LEARNING
SiMLeng-Scikit-learn package
(in progress)

```python
#!/usr/bin/env python3

from simulation_scikit_learn import Simleng_strategies as Simleng


Simleng(0,-1).strategies()
```

```python
#!/usr/bin/env python3
# -*- coding: utf-8 -*-

import pandas as pd
import numpy as np
import sklearn
import time
import nincrease
import seaborn as sns

from functools import wrap
from collections import OrderedDict
import matplotlib.pyplot as plt
from matplotlib import colors as mcolors
from biokit.viz import corrplot

from scipy import stats,linalg
from scipy.linalg import svd as sksvd
from scipy.stats import multivariate_normal
from mpl_toolkits.mplot3d import Axes3D

from tools import Tools


from sklearn import metrics
from sklearn import datasets
from sklearn import preprocessing
from sklearn import manifold
from sklearn import semi_supervised

from sklearn.preprocessing import Binarizer,
    FunctionTransformer, Imputer, KernelCenterer,
  LabelBinarizer, LabelEncoder, MultiLabelBinarizer,
  MinMaxScaler, MaxAbsScaler, QuantileTransformer,
  Normalizer, OneHotEncoder, RobustScaler,
  StandardScaler, add_dummy_feature,
  PolynomialFeatures, binarize, normalize,
  scale, robust_scale,
  maxabs_scale, minmax_scale, label_binarize, quantile_transform


from nincrease import fit_on_increasing_size,plot_r2_snr
from sklearn.manifold import MDS as skMDS, smacof as sksmacof,Isomap as skIsomap,TSNE as skTSNE,LocallyLinearEmbeddig as skLocalLEmbed,spectral_embedding as skSpectEmbed

from sklearn.model_selection import cross_val_score
from sklearn.model_selection import StratifiedKFold
from sklearn.decomposition import PCA as skPCA
from sklearn.grid_search import GridSearchCV
from sklearn.model_selection import KFold


from sklearn.feature_selection import SelectKBest as skSB
from sklearn.feature_selection import f_classif as skf_c
from sklearn.feature_selection import f_regression as skf_r
from sklearn.feature_selection import VarianceThreshold as skVT
from sklearn.feature_selection import RFE as skRFE
from sklearn.feature_selection import RFECV as skRFECV
from sklearn.feature_selection import mutual_info_classif as skmiC
from sklearn.feature_selection import mutual_info_regression as skmiR
from sklearn.feature_selection import SelectFromModel as skFM
from sklearn.grid_search import GridSearchCV as skGridSCV
from sklearn.learning_curve import learning_curve as sklearncur,validate_curve as skvalcur

from sklearn.pipeline import make_pipeline

from sklearn.pipeline import Pipeline



from sklearn import linear_model as sklm
from sklearn.linear_model import ARDRegression,BayesianRidge,ElasticNet,ElasticNetCV, Hinge,Huber,HuberRegressor,Lars, LarsCV, Lasso, LassoCV, LassoLars, LassoLarsCV, LassoLarsIC,Line
Perceptron,RANSACRegressor,RandomizedLasso,RandomizedLogisticRegression,Ridge,RidgeCV,\
RidgeClassifier,RidgeClassifierCV,SGDClassifier,SGDRegressor,SquaredLoss,TheilSenRegressor

from sklearn.linear_model import  base, bayes,cd_fast,coordinate_descent,\
 enet_path,huber, lars_path,lasso_path,lasso_stability_path,least_angle,logistic_regression_path,\
 logistic, omp,orthogonal_mp,orthogonal_mp_gram,passive_aggressive,perceptron,\
 randomized_l1, ransac,ridge,ridge_regression,sag,sag_fast, sgd_fast,stochastic_gradient,\
 theil_sen

from sklearn.random_projection import BaseRandomProjection, GaussianRandomProjection, SparseRandomProjection
from sklearn.semi_supervised import LabelPropagation, LabelSpreading

from sklearn import svm
from sklearn.svm import import SVC, NuSVC, SVR, NuSVR, OneClassSVM, LinearSVC, \
    LinearSVR
from sklearn.svm import SVC as skSVC
from sklearn.svm import SVR as skSVR
from sklearn.isotonic import IsotonicRegression as skIR
from sklearn.tree import DecisionTreeClassifier, DecisionTreeRegressor,\
      ExtraTreeClassifier, ExtraTreeRegressor, export_graphviz

from sklearn.gaussian_process import GaussianProcessClassifier as skGPC
from sklearn.gaussian_process import GaussianProcessRegressor as skGPR
from sklearn.gaussian_process import GaussianProcess as skGP
from sklearn.gaussian_process.correlation_model import absolute_exponential,squared_exponential,generalized_exponential,pure_nugget,cubic

from sklearn.gaussian_process.kernels import RBF as skRBF,Matern as skMatern
from sklearn.gaussian_process.kernels import RationalQuadratic as skRQ,ExpSineSquared as skESS, DotProduct as skDP, PairwiseKernel as skPK

from sklearn.kernel_approximation import RBFSampler as skRBFSampler,SkewedChi2Sampler as skSChi2Sampler,AdditiveChi2Sampler as skAdChi2Sampler,Nystroem as skNystroem
from sklearn.kernel_ridge import KernelRidge as skKRidge

from sklearn.discriminant_analysis import LinearDiscriminantAnalysis as skLDA
```

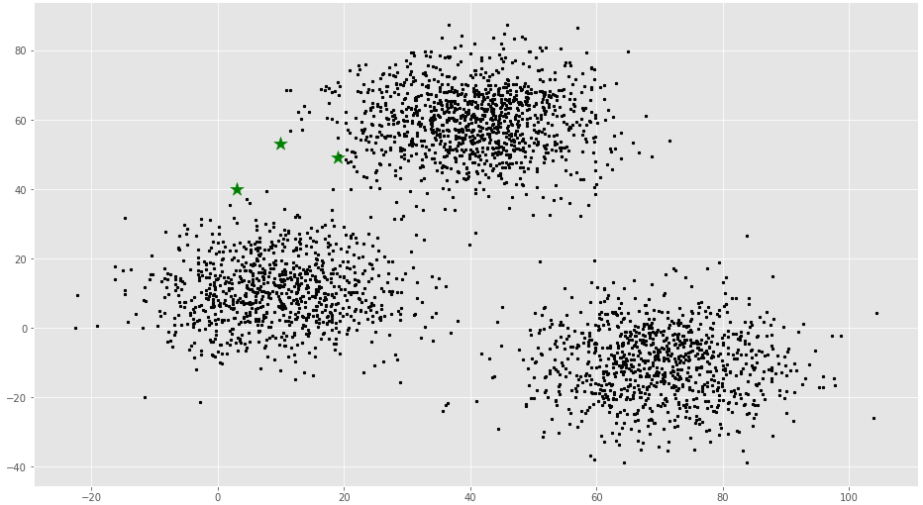
MACHINE LEARNING
Scikit-learn package
DECISION TREE
Banknote Authentication Dataset / Iris Dataset

```
Python 3.7.3 (default, Apr  3 2019, 05:39:12)
Type "copyright", "credits" or "license" for more information.

IPython 7.5.0 -- An enhanced Interactive Python.

Restarting kernel...

runfile('/home/sedna/PYTHON/PYML/DECISION-TREE/Decision-Tree-01.py', wdir='/home/sedna/PYTHO
Dataset Lenght:  625
Dataset Shape:  (625, 5)
Dataset:       0  1  2  3  4
0  B  1  1  1  1
1  R  1  1  1  2
2  R  1  1  1  3
3  R  1  1  1  4
4  R  1  1  1  5
Results Using Gini Index:
Predicted values:
['R' 'L' 'R' 'R' 'L' 'R' 'L' 'L' 'L' 'R' 'L' 'L' 'L' 'R' 'L' 'R' 'L'
 'L' 'R' 'L' 'R' 'L' 'L' 'R' 'L' 'L' 'L' 'R' 'L' 'L' 'L' 'R' 'L' 'L' 'L'
 'L' 'R' 'L' 'L' 'R' 'L' 'L' 'R' 'R' 'L' 'L' 'R' 'L' 'L' 'R' 'R' 'L' 'R'
 'R' 'L' 'R' 'R' 'L' 'L' 'R' 'R' 'L' 'L' 'L' 'L' 'L' 'R' 'R' 'L' 'L' 'R'
 'R' 'L' 'R' 'L' 'L' 'R' 'R' 'L' 'R' 'L' 'L' 'L' 'L' 'L' 'R' 'R' 'L' 'R' 'L'
 'R' 'R' 'L' 'L' 'L' 'R' 'R' 'L' 'L' 'L' 'R' 'L' 'R' 'R' 'R' 'R' 'R' 'R'
 'R' 'L' 'R' 'L' 'R' 'R' 'L' 'R' 'R' 'R' 'R' 'R' 'L' 'R' 'L' 'L' 'L' 'L'
 'L' 'L' 'L' 'R' 'R' 'R' 'R' 'L' 'R' 'R' 'L' 'L' 'R' 'L' 'R' 'L' 'R' 'R'
 'L' 'L' 'R' 'L' 'L' 'L' 'R' 'L' 'R' 'R' 'R' 'L' 'R' 'R' 'R' 'R' 'R' 'R'
 'L' 'L' 'R' 'R' 'R' 'R' 'L' 'R' 'R' 'L' 'R' 'L' 'L' 'L' 'L' 'R' 'R'
 'L' 'R' 'R' 'L' 'L' 'R' 'R' 'R']
Confusion Matrix:  [[ 0  6  7]
 [ 0 67 18]
 [ 0 19 71]]
Accuracy :  73.40425531914893
Report :                precision    recall  f1-score   support

           B       0.00      0.00      0.00        13
           L       0.73      0.79      0.76        85
           R       0.74      0.79      0.76        90

In [1]: racy                                0.73       188
   macro avg       0.49      0.53      0.51       188
weighted avg       0.68      0.73      0.71       188

Results Using Entropy:
Predicted values:
['R' 'L' 'R' 'L' 'R' 'L' 'R' 'L' 'R' 'R' 'R' 'R' 'L' 'L' 'R' 'L' 'R' 'L'
 'L' 'R' 'L' 'R' 'L' 'L' 'R' 'L' 'R' 'R' 'L' 'L' 'R' 'L' 'R' 'L' 'L' 'L'
 'L' 'L' 'R' 'L' 'R' 'L' 'R' 'R' 'L' 'L' 'R' 'L' 'L' 'R' 'L' 'L' 'L' 'L'
 'R' 'L' 'R' 'L' 'R' 'R' 'R' 'L' 'R' 'L' 'L' 'L' 'L' 'R' 'R' 'L' 'L' 'R'
 'R' 'L' 'R' 'L' 'L' 'R' 'R' 'L' 'L' 'L' 'R' 'L' 'R' 'R' 'R' 'R' 'R' 'R'
 'R' 'L' 'R' 'L' 'R' 'L' 'R' 'L' 'R' 'R' 'L' 'R' 'R' 'R' 'R' 'R' 'L' 'L'
 'L' 'L' 'L' 'R' 'R' 'R' 'R' 'L' 'R' 'R' 'L' 'L' 'R' 'L' 'R' 'L' 'R' 'R'
 'L' 'R' 'R' 'L' 'L' 'R' 'R' 'R' 'R' 'R' 'L' 'R' 'R' 'R' 'R' 'R' 'R' 'R'
 'R' 'L' 'R' 'L' 'R' 'L' 'R' 'L' 'R' 'L' 'R' 'L' 'L' 'L' 'L' 'L' 'R'
 'R' 'R' 'L' 'L' 'L' 'R' 'R' 'R']
Confusion Matrix:  [[ 0  6  7]
 [ 0 63 22]
 [ 0 20 70]]
Accuracy :  70.74468085106383
Report :                precision    recall  f1-score   support

           B       0.00      0.00      0.00        13
           L       0.71      0.74      0.72        85
           R       0.71      0.78      0.74        90

    accuracy                           0.71       188
   macro avg       0.47      0.51      0.49       188
weighted avg       0.66      0.71      0.68       188

/usr/local/lib/python3.7/dist-packages/scikit_learn-0.21.2-py3.7-linux-x86_64.egg/sklearn/metrics/classificatio
no predicted samples.
  'precision', 'predicted', average, warn_for)
/usr/local/lib/python3.7/dist-packages/scikit_learn-0.21.2-py3.7-linux-x86_64.egg/sklearn/metrics/classificatio
no predicted samples.
  'precision', 'predicted', average, warn_for)

In [2]: runfile('/home/sedna/PYTHON/PYML/DECISION-TREE/Decision-Tree-02.py', wdir='/home/sedna/P
(9, 3)
    2.771244718  1.784783929  0
0    1.728571     1.169761  0
1    3.678320     2.812814  0
2    3.961043     2.619950  0
3    2.999209     2.209014  0
4    7.497546     3.162954  1
[[ 1.72857131  1.16976141  0.        ]
 [ 3.67831985  2.81281357  0.        ]
 [ 3.96104336  2.61995032  0.        ]
 [ 2.99920892  2.20901421  0.        ]
 [ 7.49754587  3.16295355  1.        ]
 [ 9.00220326  3.33904719  1.        ]
 [ 7.44454233  0.47668338  1.        ]
 [10.12493903  3.23455098  1.        ]
 [ 6.64228735  3.31998376  1.        ]]
Split: [X1 < 6.642]
[X1 < 6.642]
  [X1 < 1.729]
    [0.0]
    [0.0]
  [X1 < 7.498]
    [X1 < 7.445]
      [1.0]
      [1.0]
    [X1 < 7.498]
      [1.0]
      [1.0]
Expected=0, Got=0
Expected=0, Got=0
Expected=0, Got=0
Expected=0, Got=0
Expected=1, Got=1
Expected=1, Got=1
Expected=1, Got=1
Expected=1, Got=1

In [3]: runfile('/home/sedna/PYTHON/PYML/DECISION-TREE/Decision-Tree-03.py', wdir='/home/sedna/PY
        0       1       2       3  4
0  3.62160  8.6661 -2.8073 -0.44699  0
1  4.54590  8.1674 -2.4586 -1.46210  0
2  3.86600 -2.6383  1.9242  0.10645  0
3  3.45660  9.5228 -4.0112 -3.59440  0
4  0.32924 -4.4552  4.5718 -0.98880  0
**************************
1372
Scores: [96.35036496350365, 97.08029197080292, 97.44525547445255, 98.17518248175182, 97.44525547
Mean Accuracy: 97.299%

In [4]:
```



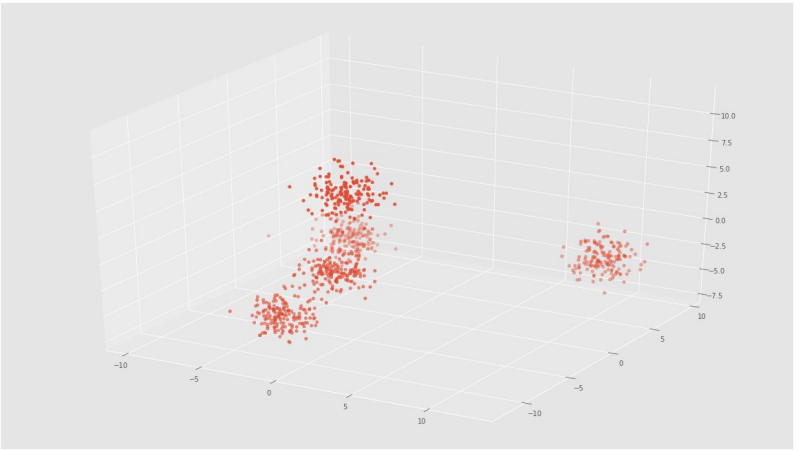
MACHINE LEARNING
Scikit-learn package
kMeans
An Artificial Dataset

```
Python 3.7.3 (default, Apr  3 2019, 05:39:12)
Type "copyright", "credits" or "license" for more information.

IPython 7.5.0 -- An enhanced Interactive Python.
runfile('/home/sedna/PYTHON/PYML/KMeans/kMeans1.py', wdir='/home/sedna/PYTHON/PYML/KMeans')
(3000, 3)
[[ 3. 40.]
 [19. 49.]
 [10. 53.]]

In [2]: runfile('/home/sedna/PYTHON/PYML/KMeans/kMeans1.py', wdir='/home/sedna/PYTHON/PYML/KMeans')
(3000, 3)
[[ 31.8263096   60.79510103]
 [  9.36565207  10.70099662]
 [ 78.51362742 -10.10562131]
 [ 61.67050215 -10.07078529]
 [ 48.05401289  58.04687427]]

In [3]:
```



k=3

k=4



MACHINE LEARNING
Scikit-learn package
k-Nearest-Neighbor
Iris Dataset

ACCURACY

Train Set	Test Set	k=3	k=5	k=10
66%	34%	60.487804878048 7%	62.926829268292686%	65.365853658536 6%
80%	20%	62.325581395348 8%	62.325581395348 4%	68.837209302325 59%

MACHINE LEARNING
Scikit-learn package
Naive-Bayes

ACCURACY

Train Set	Test Set	Pima-Indians Dataset	Iris Dataset	Ionosphere Dataset
67%	33%	74.015748031496%	98.0%	80.172413793103%
80%	20%	74.675324675324%	90.0%	88.732394366197%

MACHINE LEARNING
Scikit-learn package
Random-Forest

- **Petrol-consumption / Breast-cancer-wisconsin Dataset**

```
Python 3.7.3 (default, Apr  3 2019, 05:39:12)
Type "copyright", "credits" or "license" for more information.

IPython 7.5.0 -- An enhanced Interactive Python.
runfile('/home/sedna/PYTHON/PYML/RANDOM-FOREST/forest-00.py', wdir='/home/sedna/PYTHON/PYML/RANDOM-FOREST')
   Petrol_tax  Average_income  ...  Population_Driver_licence(%)  Petrol_Consumption
0         9.0           3571  ...                         0.525                 541
1         9.0           4092  ...                         0.572                 524
2         9.0           3865  ...                         0.580                 561
3         7.5           4870  ...                         0.529                 414
4         8.0           4399  ...                         0.544                 410

[5 rows x 5 columns]
Mean Absolute Error: 51.76500000000001
Mean Squared Error: 4216.166749999999
Root Mean Squared Error: 64.93201637097064

In [2]: runfile('/home/sedna/PYTHON/PYML/RANDOM-FOREST/forest-0classif.py', wdir='/home/sedna/PYTHON/PYML/RANDOM-FOREST')
Trees: 1
Scores: [56.09756097560976, 63.41463414634146, 60.97560975609756, 58.536585365853654, 73.17073170731707]
Mean Accuracy: 62.439%
Trees: 5
Scores: [70.73170731707317, 58.536585365853654, 85.36585365853658, 75.60975609756098, 63.41463414634146]
Mean Accuracy: 70.732%
Trees: 10
Scores: [75.60975609756098, 80.48780487804879, 92.68292682926829, 73.17073170731707, 70.73170731707317]
Mean Accuracy: 78.537%

In [3]: runfile('/home/sedna/PYTHON/PYML/RANDOM-FOREST/forest-001.py', wdir='/home/sedna/PYTHON/PYML/RANDOM-FOREST')
       CodeNumber  ClumpThickness  ...     Mitoses  CancerType
count  6.980000e+02      698.000000  ...  698.000000  698.000000
mean   1.071807e+06        4.416905  ...    1.590258    2.690544
std    6.175323e+05        2.817673  ...    1.716162    0.951596
min    6.163400e+04        1.000000  ...    1.000000    2.000000
25%    8.702582e+05        2.000000  ...    1.000000    2.000000
50%    1.171710e+06        4.000000  ...    1.000000    2.000000
75%    1.238354e+06        6.000000  ...    1.000000    4.000000
max    1.345435e+07       10.000000  ...   10.000000    4.000000

[8 rows x 10 columns]
Train_x Shape ::  (477, 9)
Train_y Shape ::  (477,)
Test_x Shape ::  (205, 9)
Test_y Shape ::  (205,)
Trained model ::  RandomForestClassifier(bootstrap=True, class_weight=None, criterion='gini',
            max_depth=None, max_features='auto', max_leaf_nodes=None,
            min_impurity_decrease=0.0, min_impurity_split=None,
            min_samples_leaf=1, min_samples_split=2,
            min_weight_fraction_leaf=0.0, n_estimators=10,
            n_jobs=None, oob_score=False, random_state=None,
            verbose=0, warm_start=False)
Actual outcome :: 2 and Predicted outcome :: 2
Actual outcome :: 2 and Predicted outcome :: 2
Actual outcome :: 2 and Predicted outcome :: 2
Actual outcome :: 4 and Predicted outcome :: 4
Actual outcome :: 4 and Predicted outcome :: 4
Train Accuracy ::  1.0
Test Accuracy  ::  0.9609756097560975
Confusion matrix  [[126   5]
 [  3  71]]
/usr/local/lib/python3.7/dist-packages/scikit_learn-0.21.2-py3.7-linux-x86_64.egg/sklearn/ensemble/forest.py:245: FutureWarning: The default value of n_estimators will change from 10 in version 0.20 to 100 in 0.22.
  "10 in version 0.20 to 100 in 0.22.", FutureWarning)

In [4]:
```

Petrol-consumption Dataset
TREES: 10
MEAN ACCURACY: 78.537%

Breast-cancer-wisconsin Dataset
TREES: 10
MEAN ACCURACY:96.09756%

MACHINE LEARNING
Scikit-learn package

SVM LINEAR CLASSIFICATION (TEST SIZE=30%)

METRICS	Breath-cancer dataset	Iris dataset	Wine dataset	Digits dataset (class =10)
ACCURACY	98.2456140%	93.3333333%	98.1481481%	96.6666666%
PRECISION	98.2469120%	93.4320987%	98.2322323%	96.6742587%
RECALL	98.2456140%	93.3333333%	98.1481481%	96.6666666%

SVM LINEAR PENALIZED ‘l1’ CLASSIFICATION (TEST SIZE=30%)

METRICS	Breath-cancer dataset	Iris dataset	Wine dataset	Digits dataset (class =10)
ACCURACY	98.8304093%	93.3333333%	100.0%	96.1111111%
PRECISION	98.8516746%	93.4558823%	100.0%	96.1411426%
RECALL	98.8304093%	93.3333333%	100.0%	96.1111111%

Applied AI and Deep Learning

Build and Serialisation Keras Model

Classification using Reuters Dataset

PARAMETERS	
X_train_rows	8982
X_test_rows	2246
max_words	1000
num_classes	46
test_split	0.2
validation_split	0.1
Epochs	3
batch_szie	32
Optimizer	Adam

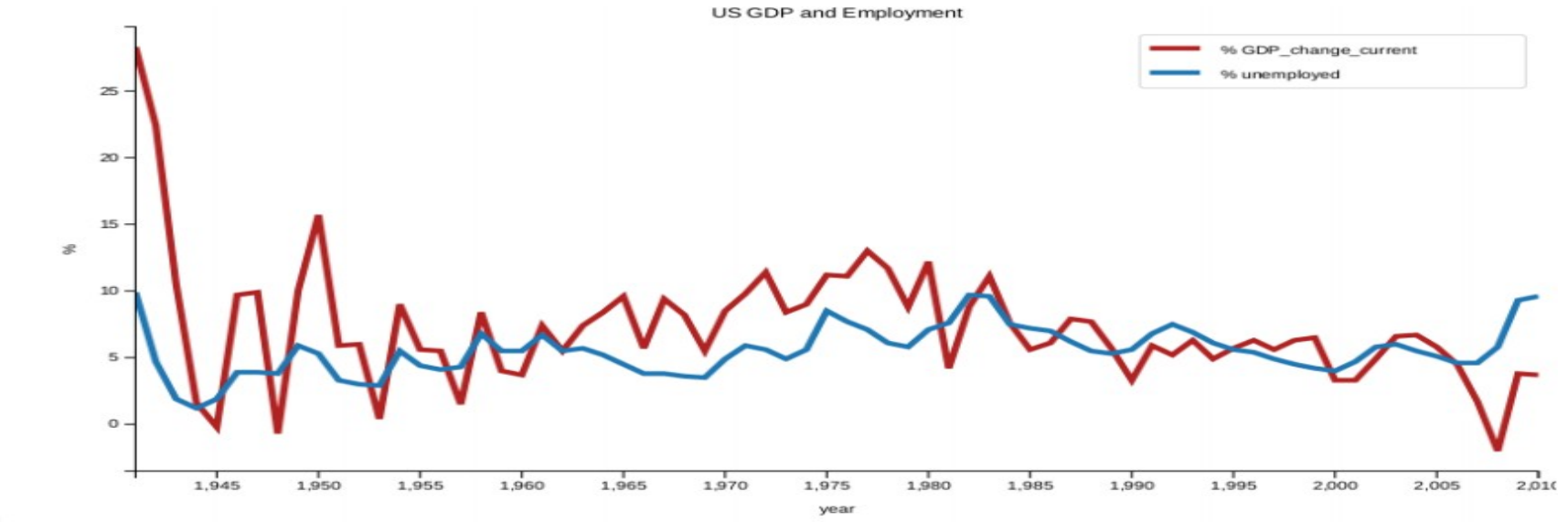
```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense (Dense)                (None, 512)               512512
_________________________________________________________________
dropout (Dropout)            (None, 512)               0
_________________________________________________________________
dense_1 (Dense)              (None, 46)                23598
=================================================================
Total params: 536,110
Trainable params: 536,110
Non-trainable params: 0
```

HISTORY

Epoch	val_loss	val_acc	loss	acc
0	0.969766	0.787542	1.440816	0.680193
1	0.851318	0.810901	0.783447	0.820859

Mpld3 Project
Analyzing US Economic Data and Building a Dashboard

Gross domestic product (GDP) is a measure of the market value of all the final goods and services produced in a period, often annually. GDP is an indicator of how well the economy is doing. A drop in GDP indicates the economy is producing less; similarly an increase in GDP suggests the economy is performing better. GDP is considered the "world's most powerful statistical indicator of national development and progress".[1] GDP can be determined in three ways, all of which should, in principle, give the same result. They are the production (or output or value added) approach, the income approach, or the speculated expenditure approach. Literature 1.-Lepenies, Philipp (2016). The Power of a Single Number: A Political History of GDP. New York: Columbia University Press. ISBN 978-0-231-17510-4.



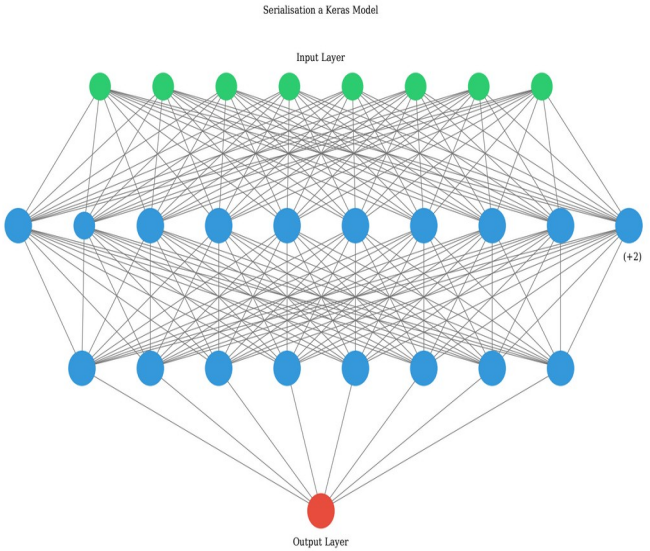
Applied AI and Deep Learning

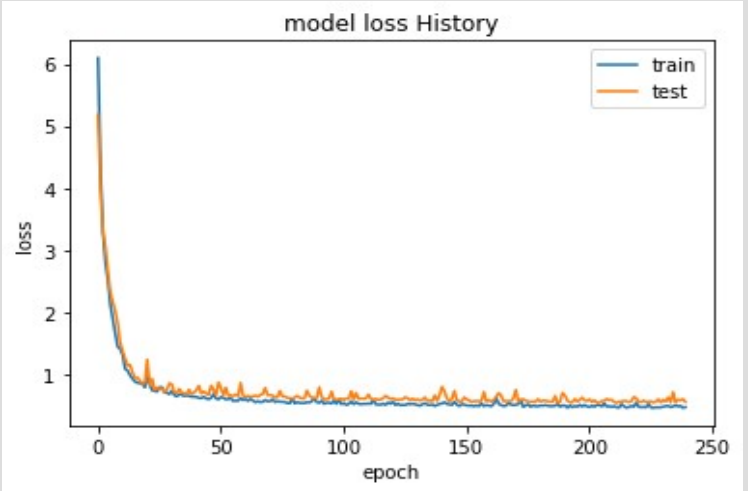
Build and Serialisation Keras Model

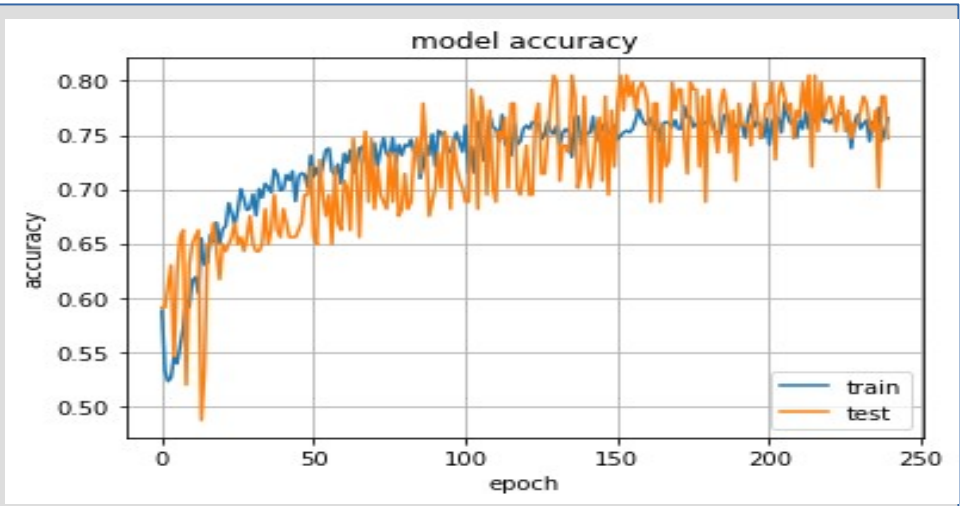
Classification using Pimas-Indians-Diabetes Dataset

PARAMETERS	
X_rows	768
features_size	8
num_classes	1
validation_split	0.2
Epochs	250
batch_szie	32
Optimizer	Adamax

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense_1 (Dense)              (None, 12)                108
_________________________________________________________________
dense_2 (Dense)              (None, 8)                 104
_________________________________________________________________
dense_3 (Dense)              (None, 1)                 9
=================================================================
Total params: 221
Trainable params: 221
Non-trainable params: 0
```







Anomaly Detection case using IBM Watson Study platform

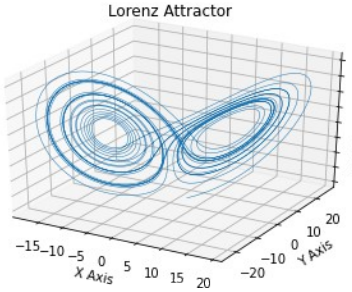
Taking data from https://raw.githubusercontent.com/romeokienzler/ developerWorks/master/lorenzattractor/

We have implemented re-use the unsupervised anomaly detection algorithm but turn it into a simpler feed forward neural network for supervised classification. For this purpose, we have trained the neural network from healthy and broken samples and at later stage hook it up to a message queue for real-time anomaly detection.

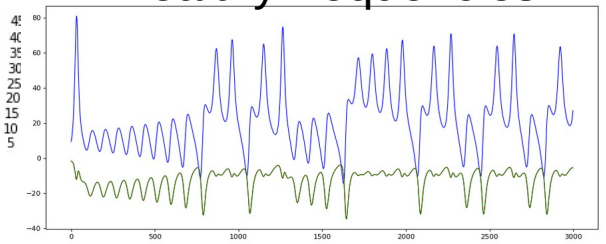
Model: "sequential"

Layer (type) Output Shape Param

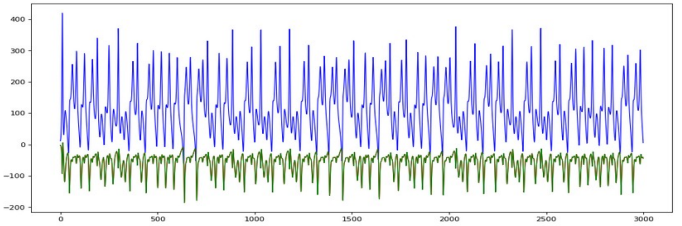
dense_1 (Dense) (None, 3000) 9003000

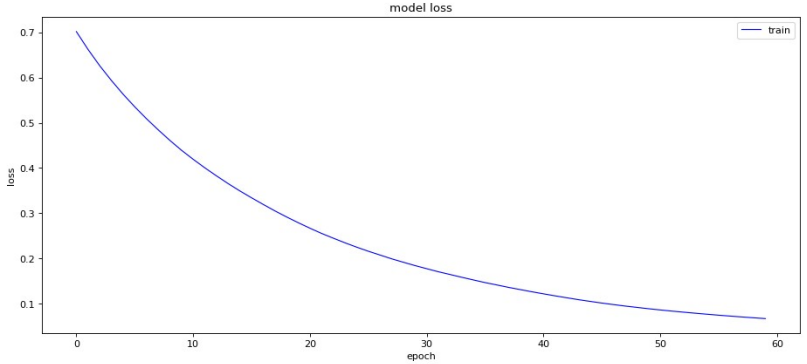


Healthy frequencies



Broken frequencies





Healthy frequencies Score

0.92995715

0.9455121

Broken frequencies Score

0.07092416

0.05365459

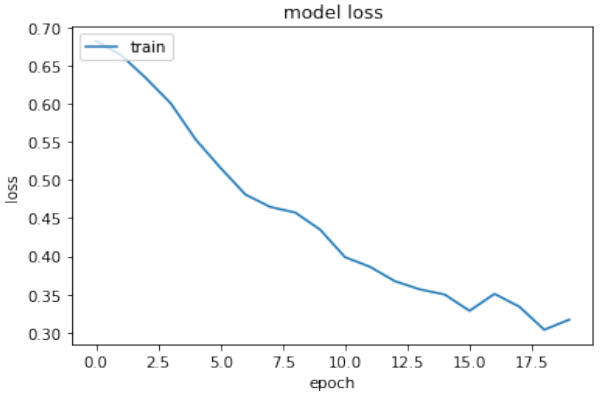
A KERAS MODEL RUNNING IN DL4J ON SPARK

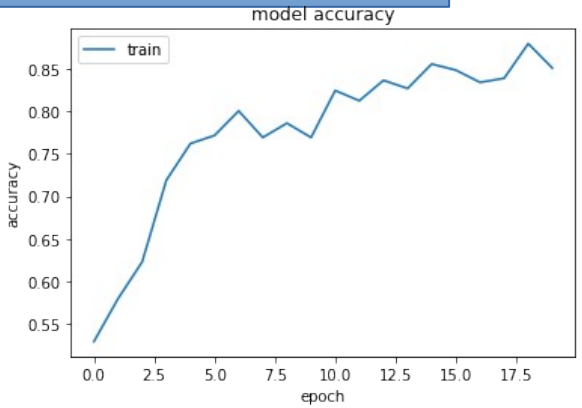
We have implemented a NN for Sonar Dataset. The file "sonar.mines" contains 111 patterns obtained by bouncing sonar signals off a metal cylinder at various angles and under various conditions. The file "sonar.rocks" contains 97 patterns obtained from rocks under similar conditions. The transmitted sonar signal is a frequency-modulated chirp, rising in frequency. The data set contains signals obtained from a variety of different aspect angles, spanning 90 degrees for the cylinder and 180 degrees for the rock.Each pattern is a set of 60 numbers in the range 0.0 to 1.0. Each number represents the energy within a particular frequency band, integrated over a certain period of time. The integration aperture for higher frequencies occur later in time, since these frequencies are transmitted later during the chirp. The label associated with each record contains the letter "R" if the object is a rock and "M" if it is a mine (metal cylinder). The numbers in the labels are in increasing order of aspect angle, but they do not encode the angle directly.

Model: "sequential"Layer (type) Output Shape Param

dense (Dense)	(None, 60)	3660
dense_1 (Dense)	(None, 30)	1830
dense_2 (Dense)	(None, 2)	62

Total params: 5,552 Trainable params: 5,552






```
!$SPARK_HOME/bin/spark-submit\
--class skymind.dsx.KerasImportCSVSparkRunner \
--files sonar.csv,my_modelx.h5 \
 --master $MASTER \
 dl4j-snapshot.jar \
 -batchSizePerWorker 15 \
 -indexLabel 60 \
 -train false \
 -numClasses 2 \
 -modelFileName my_modelx.h5 \
-dataFileName sonar.csv
```

SCORES

# of classes:	2
Accuracy:	0.8894
Precision:	0.8894
Recall:	0.8912
F1 Score:	0.8856

SiMLeng PROJECT

Tourism enhancers index by neighborhoods in London

RUPERTO P. BONET

ruperto.bonet@gmail.com

DESCRIPTION OF THE PROJECT

London is one of the most popular tourist destination in the whole world, join to Paris and Bangkok. Each and every year London city attracts millions of visitors from other countries. During 2019 London received 19.09 millions of tourists according to the Mastercard's annual Global Cities Index publishing.

The tourism industry is one of the engines of the economy growth in UK. London first works int the Tourism Sector Deal proposal. This proposal defends and enhances the Lodon's role as a gateway , open and welcoming destination for both business and leisure visitors
[https://www.londonfirst.co.uk/what-we-do/economy-and-tax/tourism]

In this project, motivated by the challenge of understanding London urban environments, and based on the opportunities created by geoenabled social data, we address the problem of comparing boroughs and neighborhoods in London city .The problem we study has applications to recommending locations in London city as tourist enhancers. Imagine an entrepreneur decides by looking for areas with tourist profile to open a new establishment related to the tourist target like as in the neighbourhoods tradillionaly most visited, for example, where should I open a basement to rent bycicles?: the methods developed in this paper allow to match each neighborhood in a borough with the most similar neighborhood in an other borough, or any other borough that someone wishes to compare.

Tourist-enhancers-index is a mean value index of turist activities by neighbourhood. This activities include the following categories: arts_entertainment, building, education, event, food, nightlife, parks_outdoors, shops and travel. Data are collected of several sources including Foursquare api.

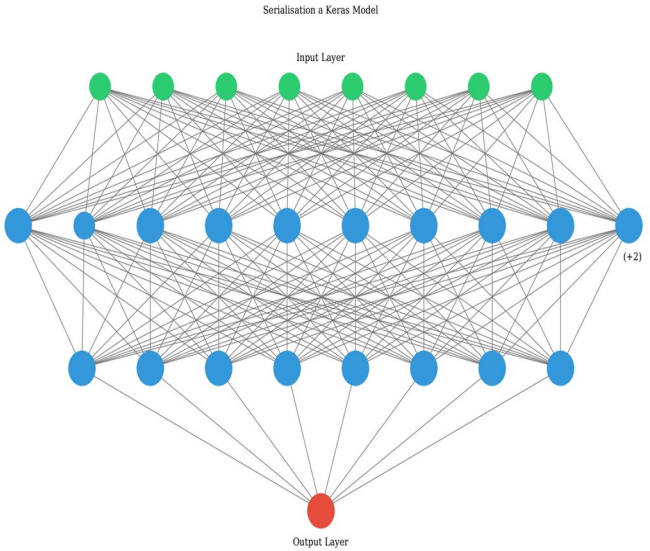
Applied AI and Deep Learning

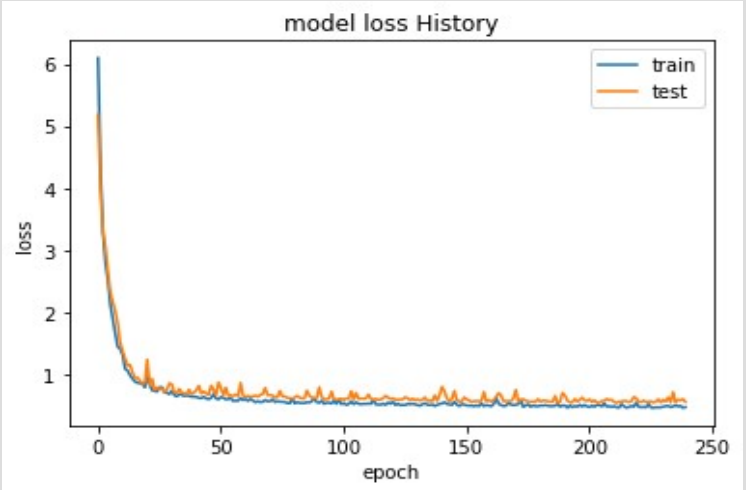
Build and Serialisation Keras Model

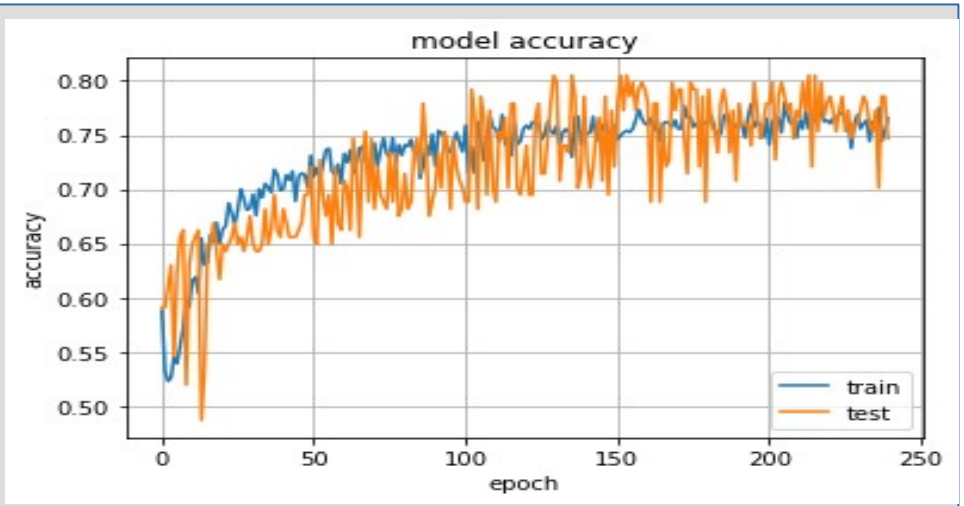
Classification using Pimas-Indians-Diabetes Dataset

PARAMETERS	
X_rows	768
features_size	8
num_classes	1
validation_split	0.2
Epochs	250
batch_szie	32
Optimizer	Adamax

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense_1 (Dense)              (None, 12)                108
_________________________________________________________________
dense_2 (Dense)              (None, 8)                 104
_________________________________________________________________
dense_3 (Dense)              (None, 1)                 9
=================================================================
Total params: 221
Trainable params: 221
Non-trainable params: 0
```

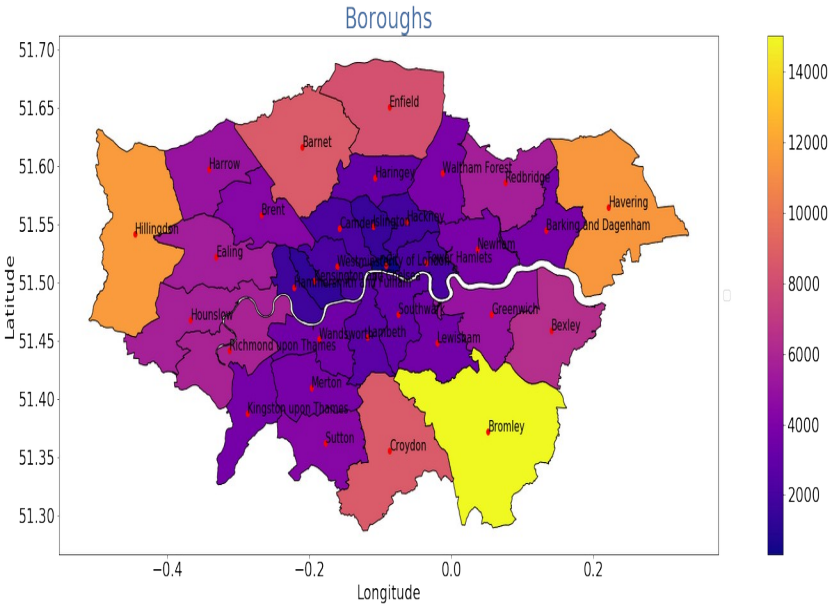


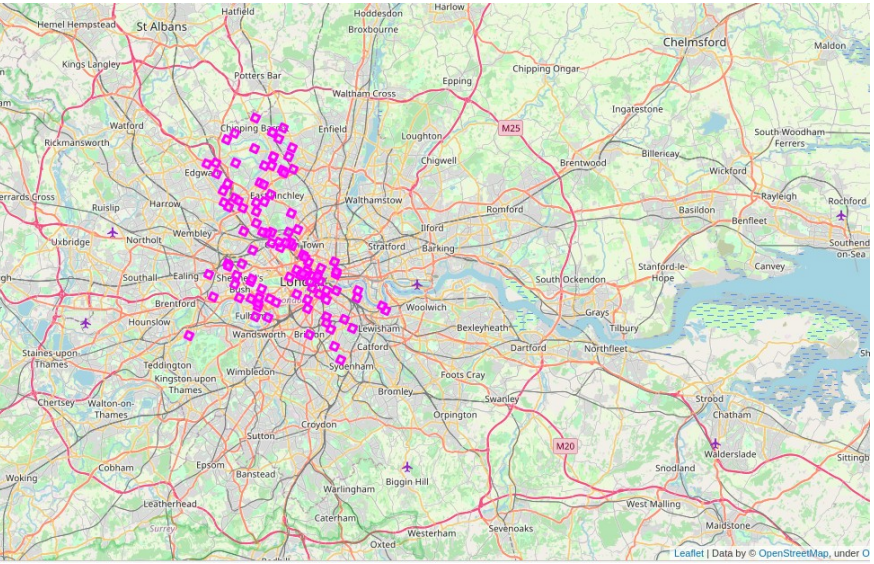




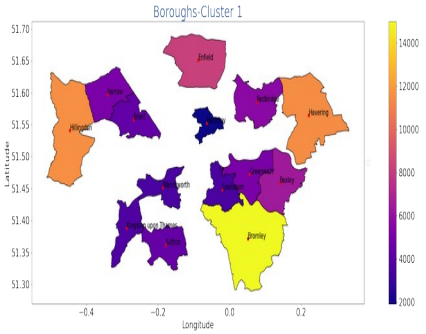
TOURIST-ENHANCERS INDEX BY NEIGBOURHOODS IN LONDON

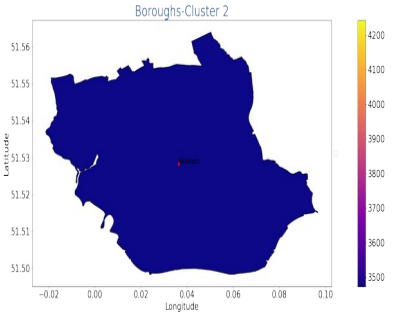
MOST VISITED PLACES IN LONDON

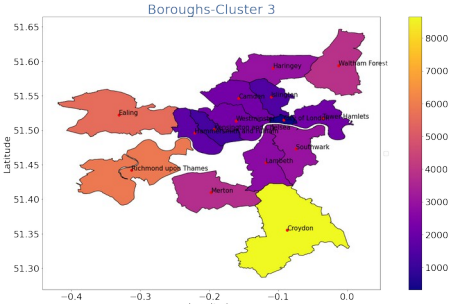


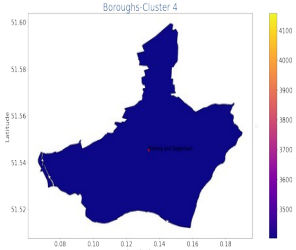


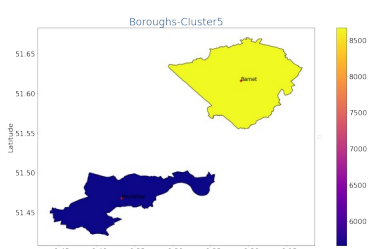
KMEANS CLUSTERING (K=5)











TOURIST-ENHANCERS INDEX BY NEIGBOURHOODS IN LONDON

RECOMENDATION ENGINE

GOOD CLASSIFIER (FAVOURITE-NON FAVOURITE)

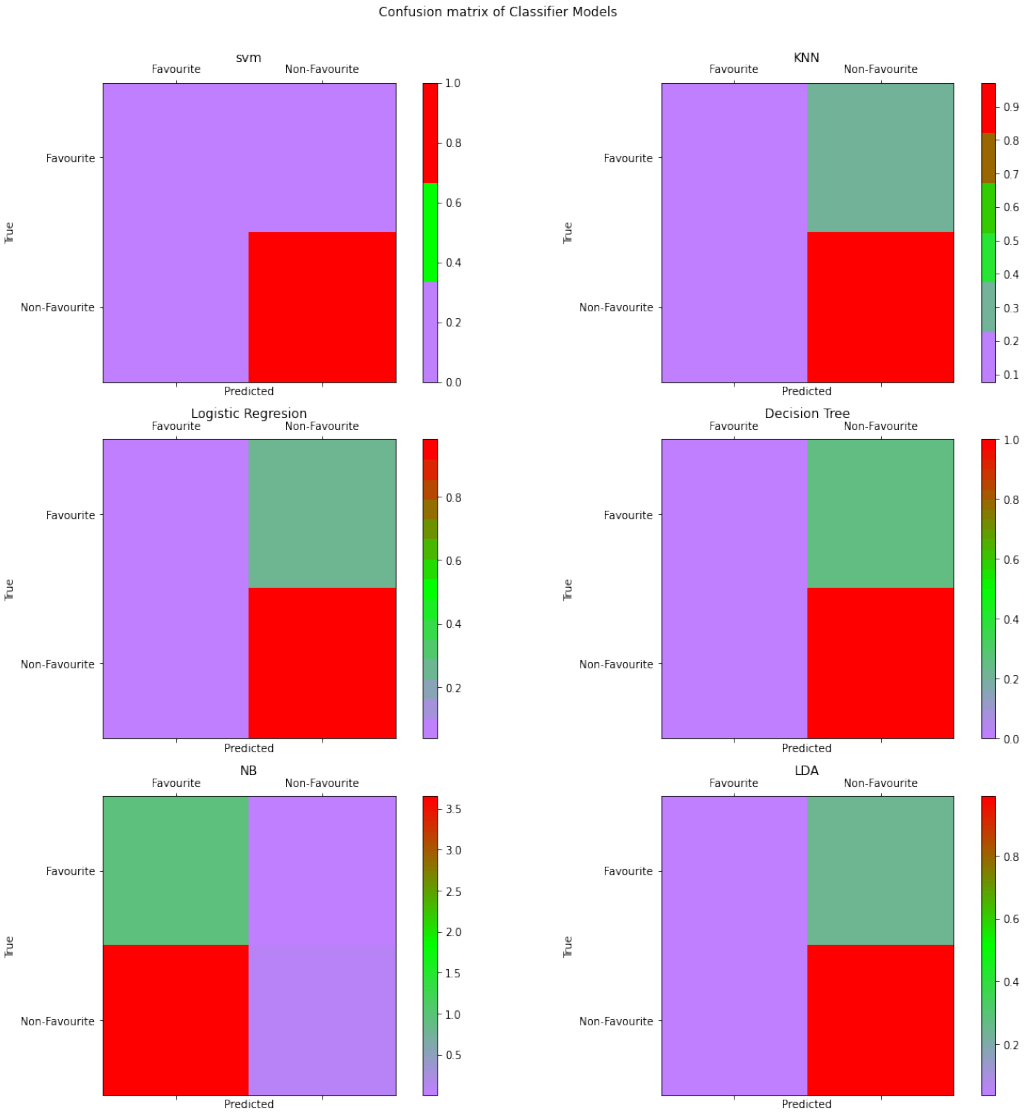
PARAMETERS

X_rows	515
features_size	9
num_classes	1
validation_split	0.25

SUPERVISED CLASSIFIERS ACCURACY

svm: 0.7984496124031008
KNN: 0.7906976744186046
Logistic Regresion:0.7906976744186046
Decision Tree: 0.7984496124031008
NB: 0.2558139534883721
LDA: 0.7984496124031008

CONFUSION MATRIX



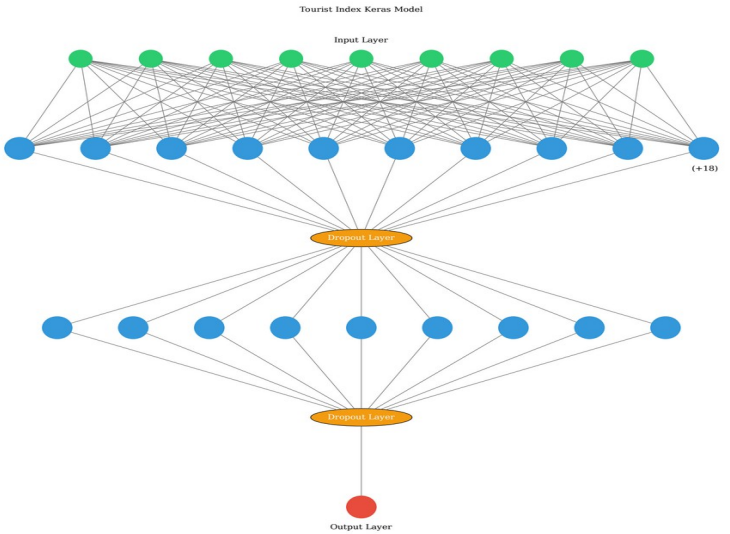
TOURIST-ENHANCERS INDEX BY NEIGBOURHOODS IN LONDON

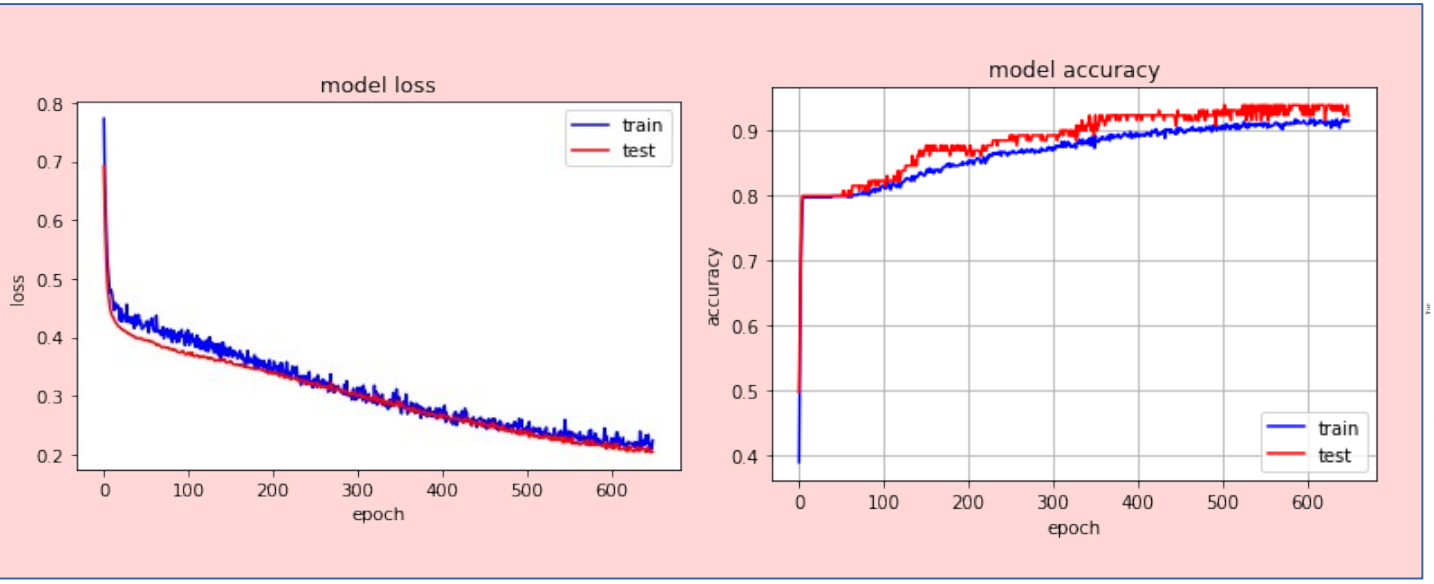
RECOMENDATION ENGINE

GOOD CLASSIFIER (FAVOURITE-NON FAVOURITE)

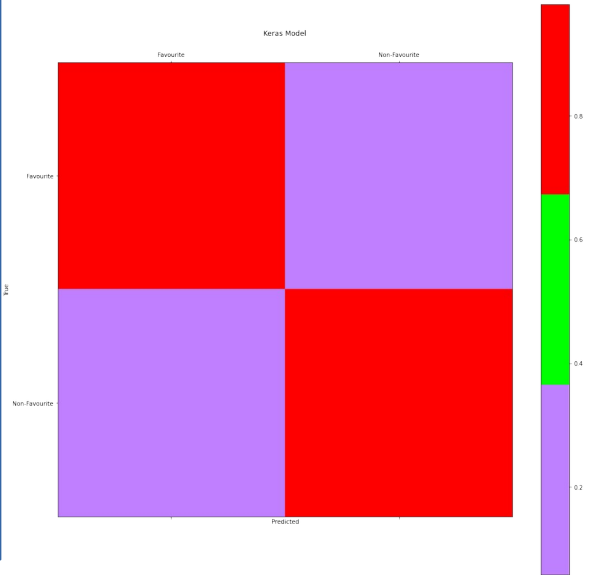
PARAMETERS	
X_rows	515
features_size	9
num_classes	1
validation_split	0.25
Epochs	250
batch_szie	50
Optimizer	Adam

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense_6 (Dense)              (None, 28)                280
_________________________________________________________________
dropout_2 (Dropout)          (None, 28)                0
_________________________________________________________________
dense_7 (Dense)              (None, 9)                 261
_________________________________________________________________
dropout_3 (Dropout)          (None, 9)                 0
_________________________________________________________________
dense_8 (Dense)              (None, 1)                 10
=================================================================
Total params: 551
Trainable params: 551
Non-trainable params: 0
```

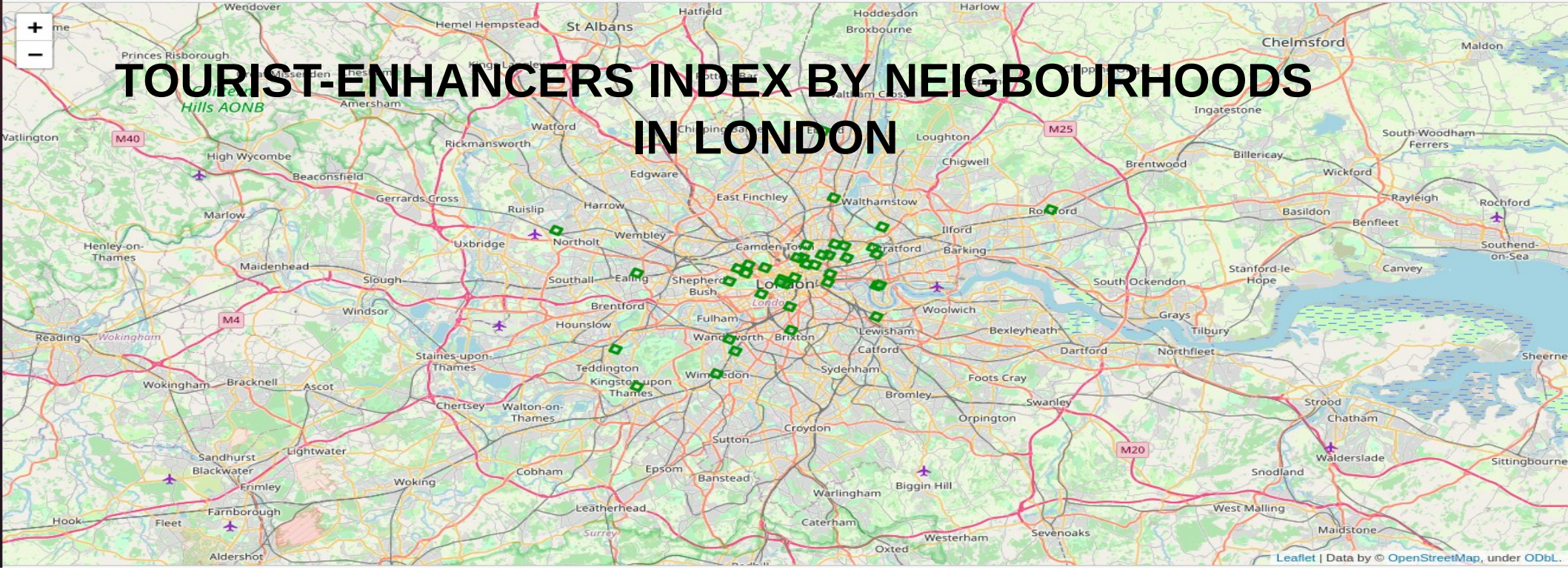


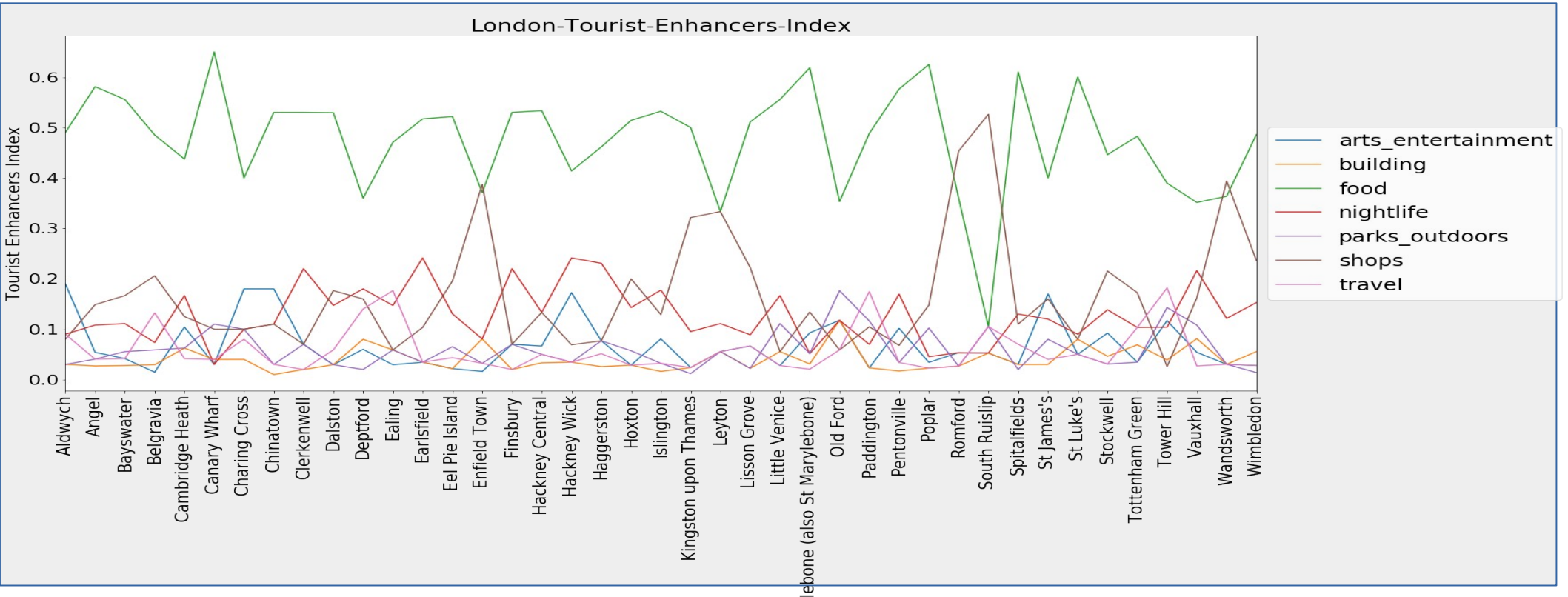


CONFUSION MATRIX



TOURIST-ENHANCERS INDEX BY NEIGBOURHOODS IN LONDON

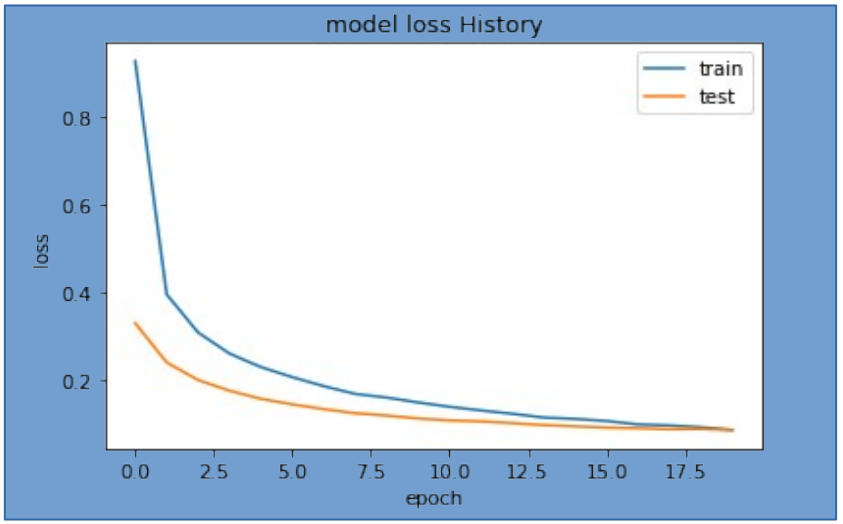


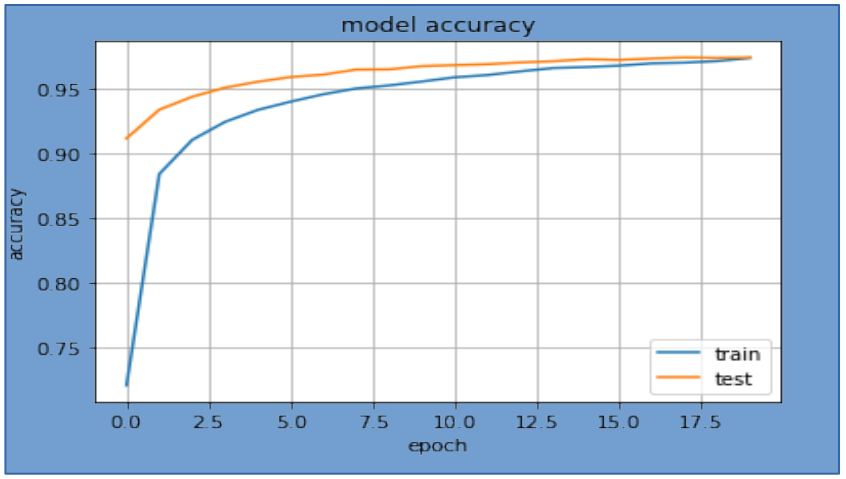


Applied AI and IMAGE PROCESSING

KERAS : MNIST HANDWRITTEN DIGIT RECOGNITION

PARAMETERS	
train samples	60000
test samples	10000
classes	10
hidden layers	128
dropout	0.25
learning rate	0.0002
batch_size	128
Validation split	0.2



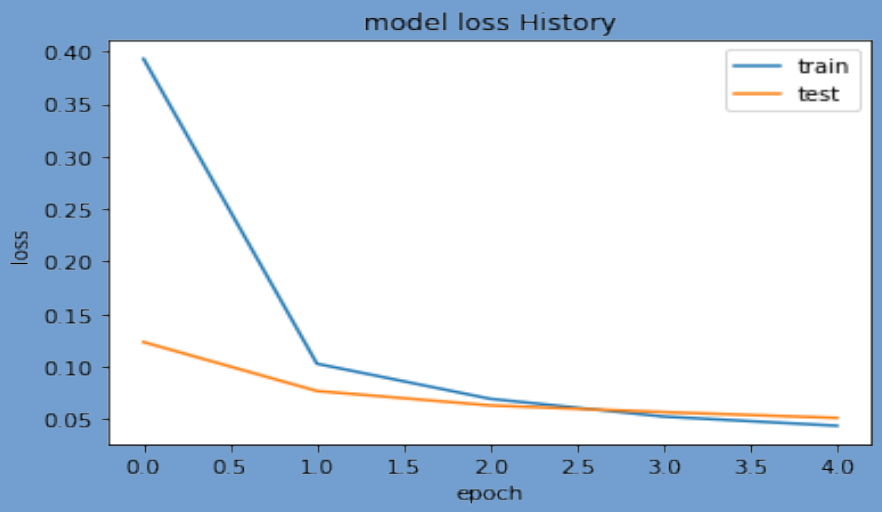



```
Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense_1 (Dense)              (None, 128)               100480
_________________________________________________________________
dropout_1 (Dropout)          (None, 128)               0
_________________________________________________________________
dense_2 (Dense)              (None, 128)               16512
_________________________________________________________________
dropout_2 (Dropout)          (None, 128)               0
_________________________________________________________________
dense_3 (Dense)              (None, 10)                1290
=================================================================
Total params: 118,282
Trainable params: 118,282
Non-trainable params: 0
```

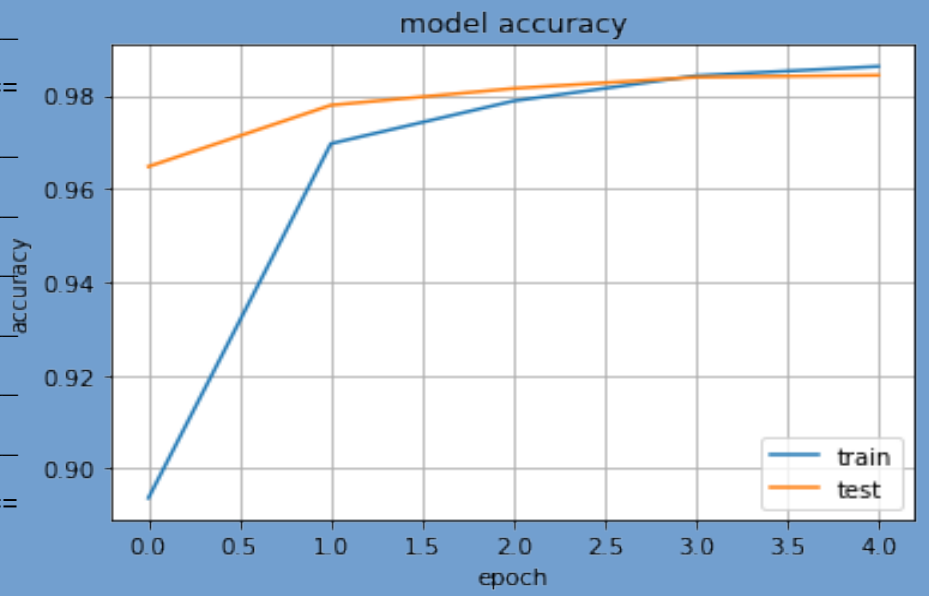
```
10000/10000 [====] - 0s 30us/step
Test score: 8.1528057133406 4
Test accuracy: 97.51999974250793
```

Applied AI and IMAGE PROCESSING
CONV2:LENET MNIST HANDWRITTEN DIGIT RECOGNITION

PARAMETERS	
train samples	60000
test samples	10000
classes	10
hidden layers	128
dropout	0.25
learning rate	0.0002
batch_size	128
Validation split	0.2

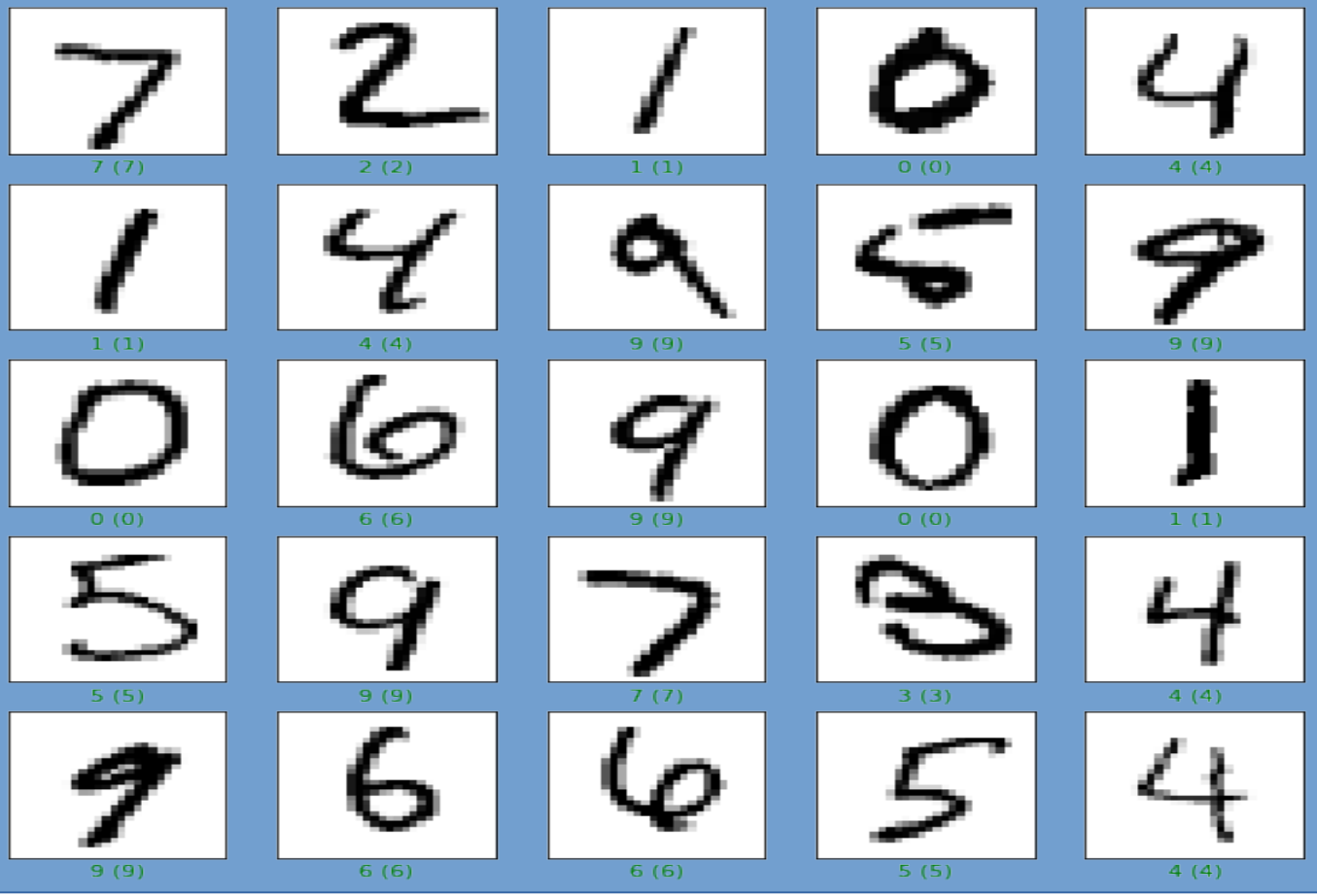



```
Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_1 (Conv2D)            (None, 20, 28, 28)        520
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 20, 14, 14)        0
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 50, 14, 14)        25050
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 50, 7, 7)          0
_________________________________________________________________
flatten_1 (Flatten)          (None, 2450)              0
_________________________________________________________________
dense_1 (Dense)              (None, 500)               1225500
_________________________________________________________________
dense_2 (Dense)              (None, 10)                5010
=================================================================
Total params: 1,256,080
Trainable params: 1,256,080
Non-trainable params: 0
```



Applied AI and IMAGE PROCESSING

CONV2:LENET MNIST HANDWRITTEN DIGIT RECOGNITION



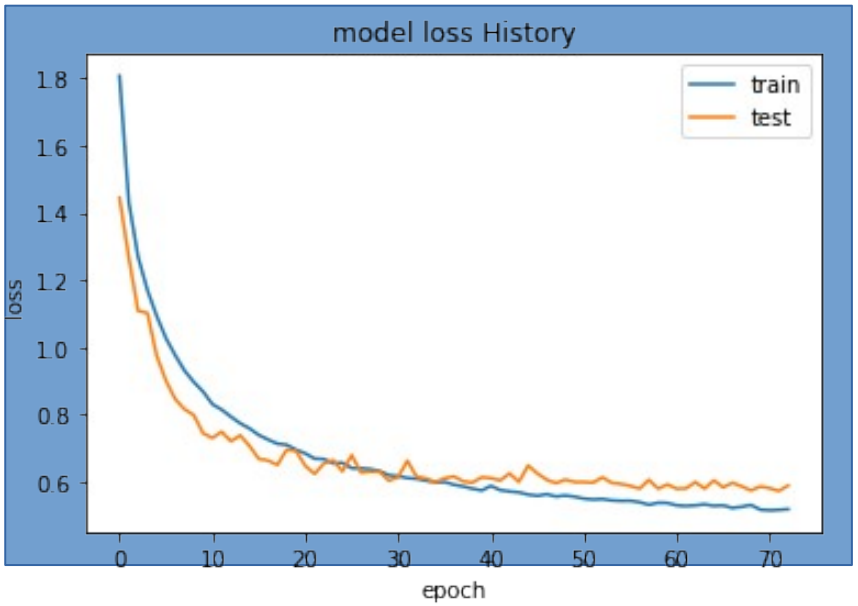

```
10000/10000 [==============================] - 8s 766us/step
        Test score: 3.788638093601912
        Test accuracy: 98.79000186920166
```

Applied AI and IMAGE PROCESSING

CONVNET : CIFAR10 CLASSIFICATION OF IMAGES

PARAMETERS

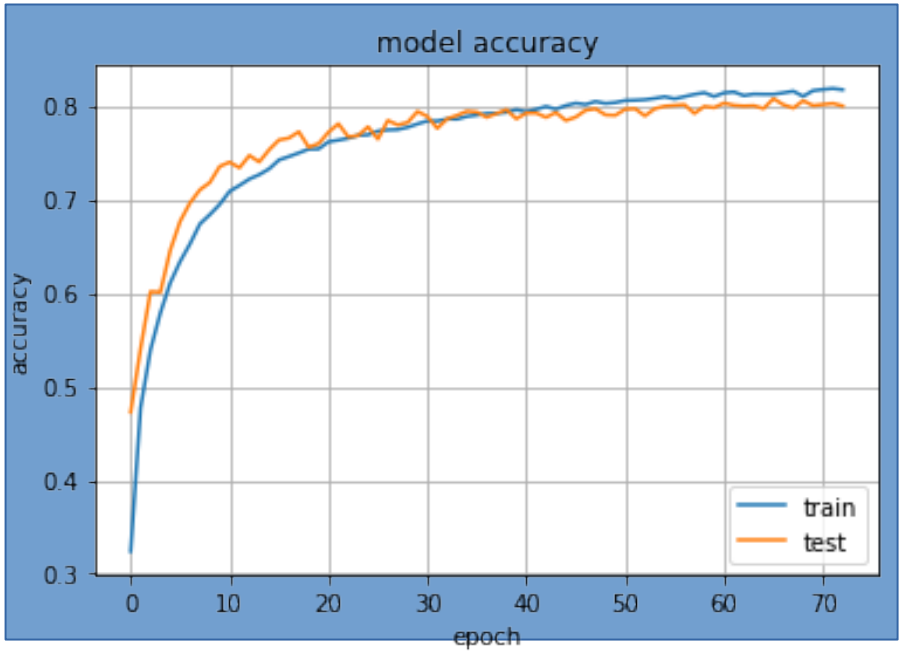
train samples	40000
test samples	10000
classes	10
learning rate	0.0005
batch_size	32
Validation split	0.2



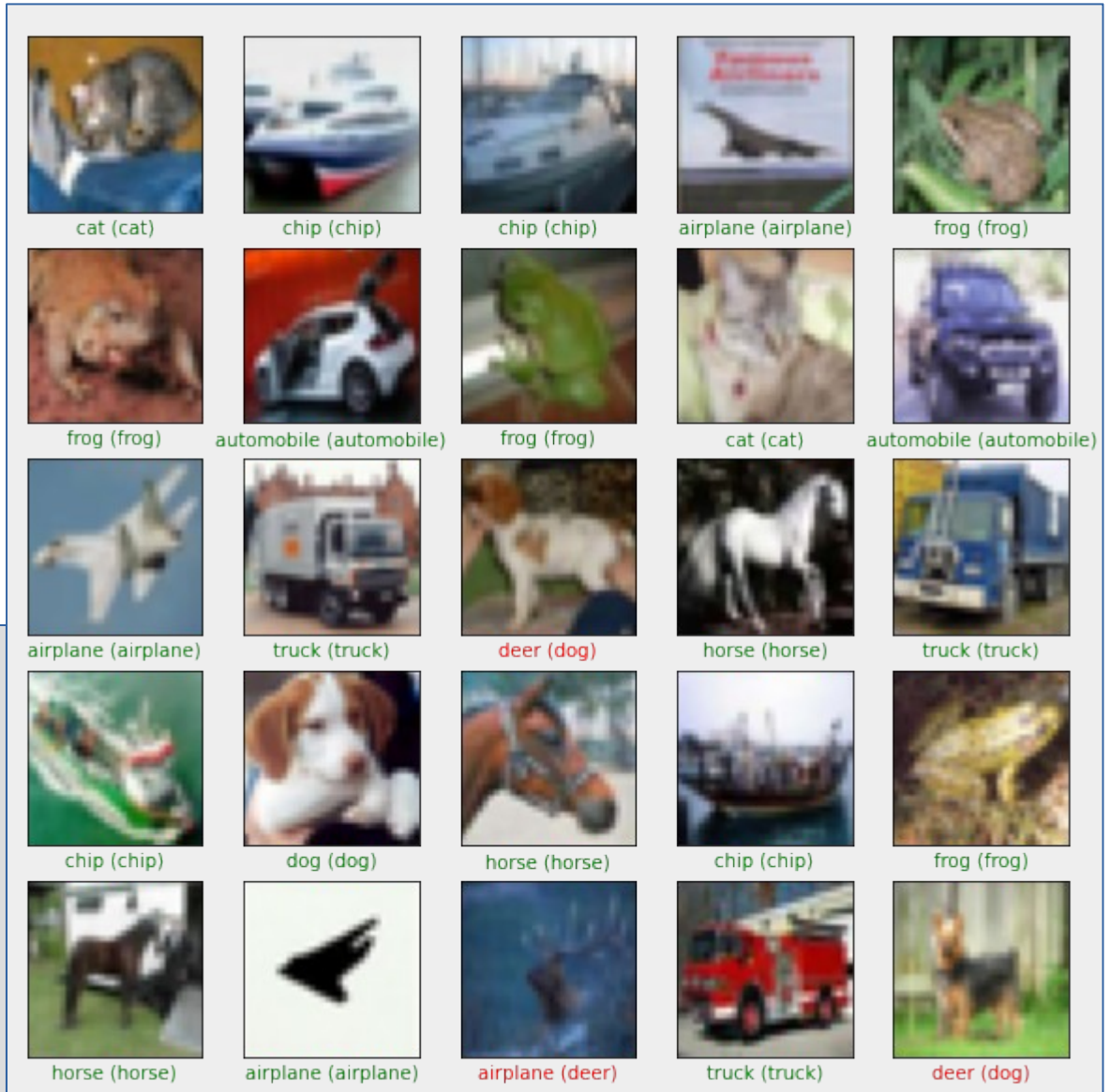

```
Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_1 (Conv2D)            (None, 32, 32, 16)        448
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 32, 32, 16)        2320
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 16, 16, 16)        0
_________________________________________________________________
dropout_1 (Dropout)          (None, 16, 16, 16)        0
_________________________________________________________________
conv2d_3 (Conv2D)            (None, 16, 16, 32)        4640
_________________________________________________________________
conv2d_4 (Conv2D)            (None, 14, 14, 32)        9248
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 7, 7, 32)          0
_________________________________________________________________
dropout_2 (Dropout)          (None, 7, 7, 32)          0
_________________________________________________________________
conv2d_5 (Conv2D)            (None, 7, 7, 64)          18496
_________________________________________________________________
conv2d_6 (Conv2D)            (None, 5, 5, 64)          36928
_________________________________________________________________
max_pooling2d_3 (MaxPooling2 (None, 2, 2, 64)          0
_________________________________________________________________
dropout_3 (Dropout)          (None, 2, 2, 64)          0
_________________________________________________________________
flatten_1 (Flatten)          (None, 256)               0
_________________________________________________________________
dense_1 (Dense)              (None, 512)               131584
_________________________________________________________________
dropout_4 (Dropout)          (None, 512)               0
_________________________________________________________________
dense_2 (Dense)              (None, 10)                5130
_________________________________________________________________
activation_1 (Activation)    (None, 10)                0
=================================================================
Total params: 208,794
Trainable params: 208,794
Non-trainable params: 0
_________________________________________________________________
```

Applied AI and IMAGE PROCESSING

CONVNET : CIFAR10 CLASSIFICATION OF IMAGES

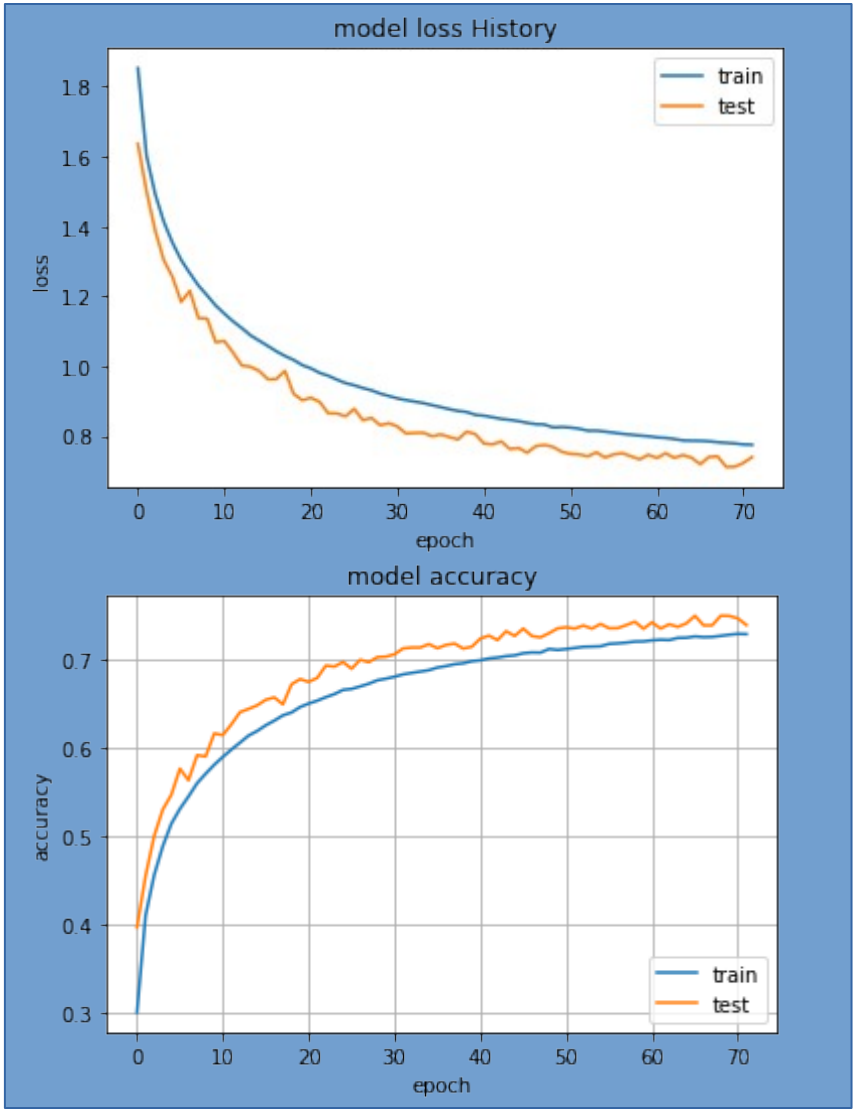


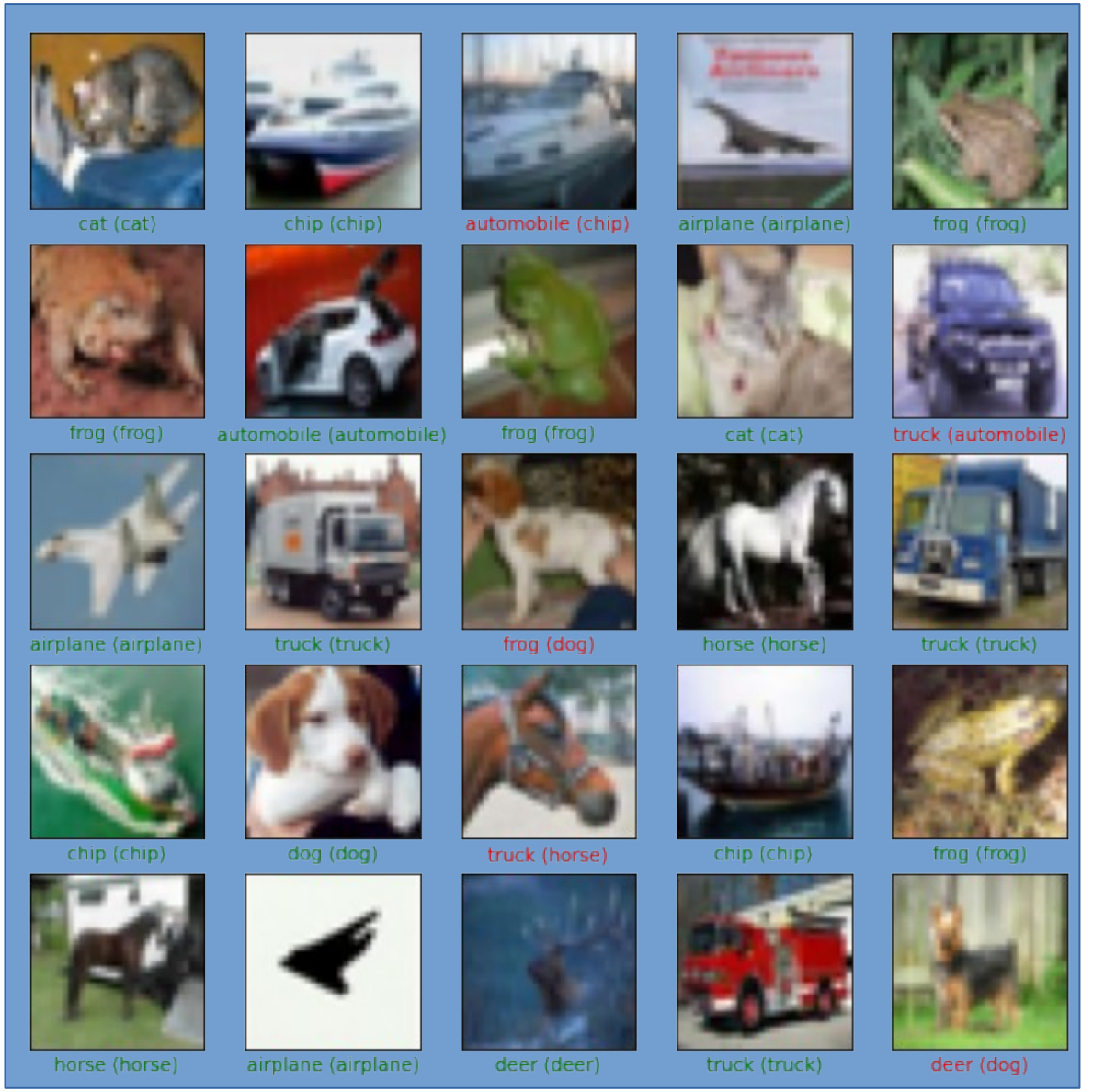

```
10000/10000
[===]-1s 87us/step
Test score:
60.954413199424 74
Test accuracy:
79.62999399185 18
```



Applied AI and IMAGE PROCESSING

CONVNET : CIFAR10 CLASSIFICATION OF IMAGES
DATA AUGMENTEDx5 lr=2*10^-4 BATCH_SIZE=128



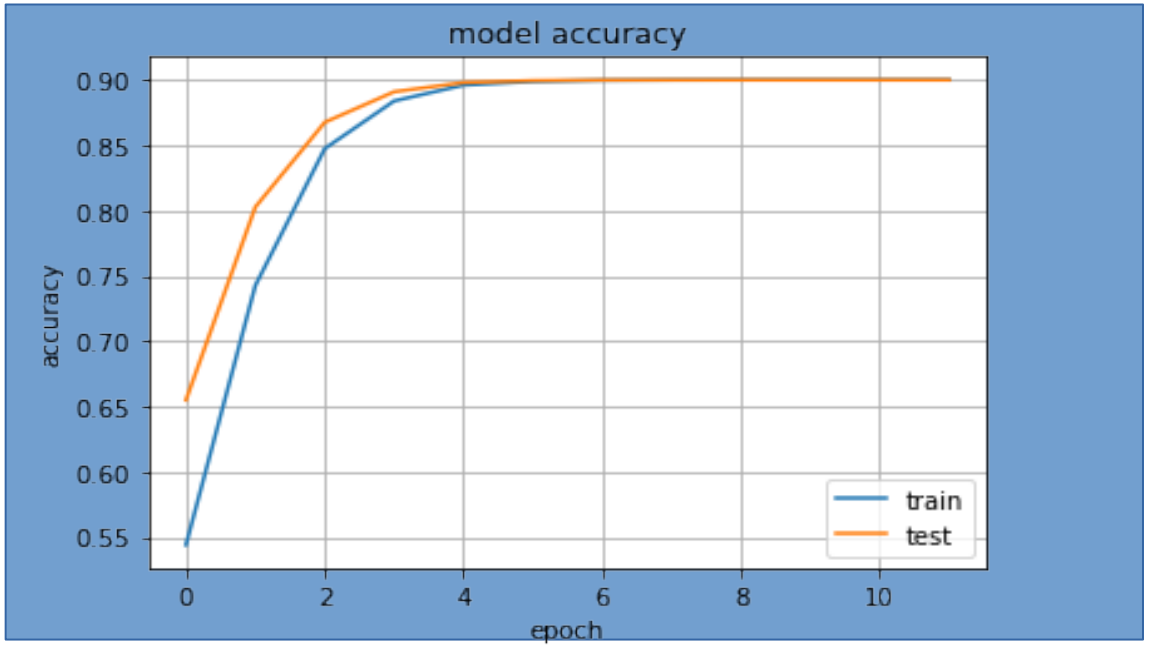


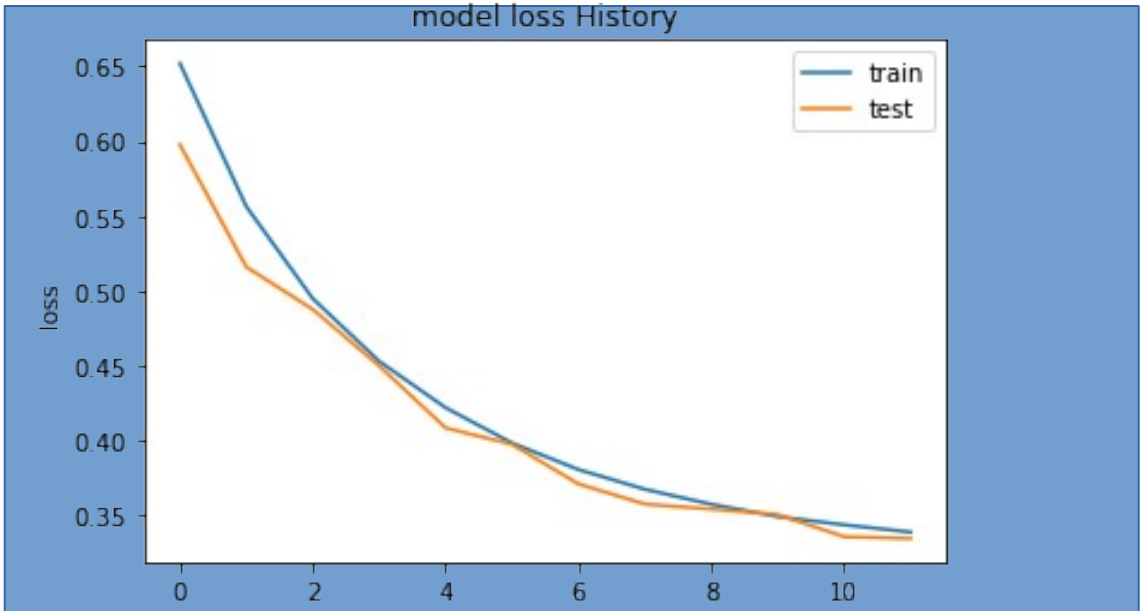

```
10000/10000 [==============================] - 1s 113us/step
Test score: 82.35761503219604
Test accuracy: 71.90999984741211
```

Applied AI and IMAGE PROCESSING

CONVNET : STL10 CLASSIFICATION OF IMAGES

Image DataGenerator + PRE-TRAINED MODEL VGG16[-3 layers]



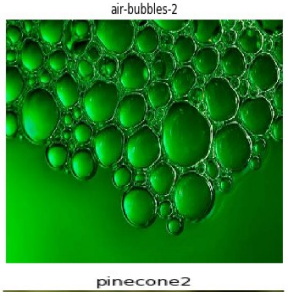
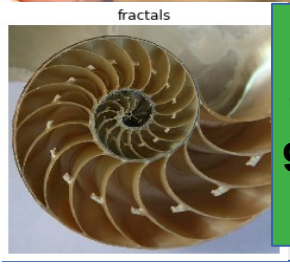



```
Model: "model_7"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_4 (InputLayer)         (None, 96, 96, 3)         0
_________________________________________________________________
block1_conv1 (Conv2D)        (None, 96, 96, 64)        1792
_________________________________________________________________
block1_conv2 (Conv2D)        (None, 96, 96, 64)        36928
_________________________________________________________________
block1_pool (MaxPooling2D)   (None, 48, 48, 64)        0
_________________________________________________________________
block2_conv1 (Conv2D)        (None, 48, 48, 128)       73856
_________________________________________________________________
block2_conv2 (Conv2D)        (None, 48, 48, 128)       147584
_________________________________________________________________
block2_pool (MaxPooling2D)   (None, 24, 24, 128)       0
_________________________________________________________________
block3_conv1 (Conv2D)        (None, 24, 24, 256)       295168
_________________________________________________________________
block3_conv2 (Conv2D)        (None, 24, 24, 256)       590080
_________________________________________________________________
block3_conv3 (Conv2D)        (None, 24, 24, 256)       590080
_________________________________________________________________
block3_pool (MaxPooling2D)   (None, 12, 12, 256)       0
_________________________________________________________________
block4_conv1 (Conv2D)        (None, 12, 12, 512)       1180160
_________________________________________________________________
block4_conv2 (Conv2D)        (None, 12, 12, 512)       2359808
_________________________________________________________________
block4_conv3 (Conv2D)        (None, 12, 12, 512)       2359808
_________________________________________________________________
block4_pool (MaxPooling2D)   (None, 6, 6, 512)         0
_________________________________________________________________
block5_conv1 (Conv2D)        (None, 6, 6, 512)         2359808
_________________________________________________________________
block5_conv2 (Conv2D)        (None, 6, 6, 512)         2359808
_________________________________________________________________
block5_conv3 (Conv2D)        (None, 6, 6, 512)         2359808
_________________________________________________________________
block5_pool (MaxPooling2D)   (None, 3, 3, 512)         0
_________________________________________________________________
flatten_7 (Flatten)          (None, 4608)              0
_________________________________________________________________
dense_19 (Dense)             (None, 128)               589952
_________________________________________________________________
dense_20 (Dense)             (None, 128)               16512
_________________________________________________________________
dense_21 (Dense)             (None, 10)                1290
=================================================================
Total params: 15,322,442
Trainable params: 5,327,370
Non-trainable params: 9,995,072
```

Applied AI and IMAGE PROCESSING

PRE-TRAINED MODELS PREDICTION :CLASSIFICATION OF IMAGES

FRACTALS IN NATURE

	ResNet50	VGG16	VGG19	InceptionV3
	'coil' 99.28587675094604	'coil' 85.86033582687378	'coil' 99.2819607257843	'coil' 84.0158104896545
	'broccoli' 66.43117070198059	'cauliflower' 43.12864542007446	'cauliflower' 85.71616411209106	'cauliflower' 91.8545663356781
	'bubble' 45.6044465303421	'chain_mail' 31.28640353679657	'necklace' 31.665799021720886	'honeycomb' 75.99433660507202
	'sea_cucumber' 27.89953649044037	'honeycomb' 25.767436623573303	'acorn' 84.96872782707214	'pineapple' 96.2532997131347
	'chambered_nautilus' 99.94205236434937	'chambered_nautilus' 99.7738242149353	'chambered_nautilus' 99.5328068733215 3	'chambered_nautilus' 99.82097148895264

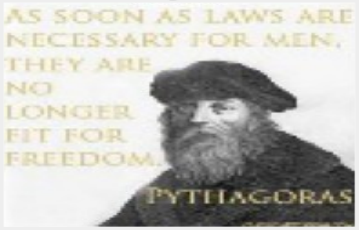
Applied AI and IMAGE PROCESSING

FACENET: FACE RECOGNITION TASK
FAMOUS MATHEMATICIANS

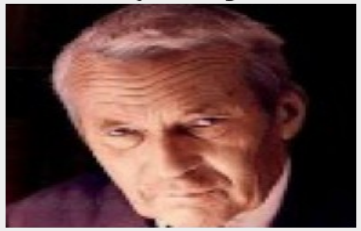
Bernhard-Riemann

CHristiaan-Huygens

Pitagoras



Andrey-Kolmogorov



isaac-newton

Andrew-Wiles

CHristiaan-Huygens

Vladimir-Arnold

Stephen-Hawking

marjorie-lee-browne

Benoit_Mandelbrot

Leonardo-da-Vinci

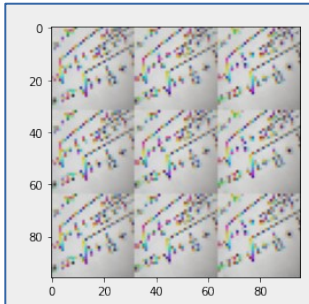
euphemia-haynes

Srinivasa-Ramanujan

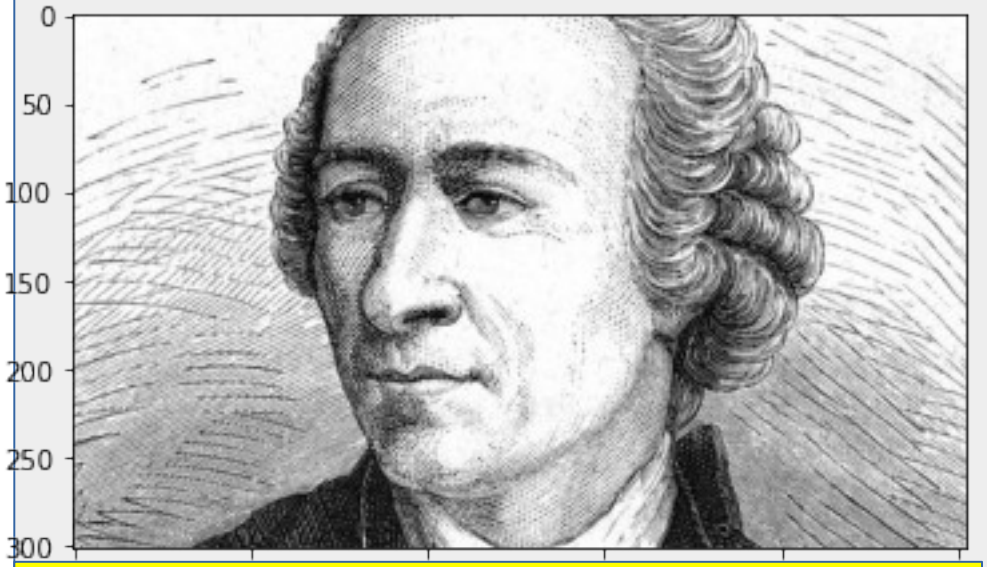
Gottfried-Leibniz

Applied AI and IMAGE PROCESSING

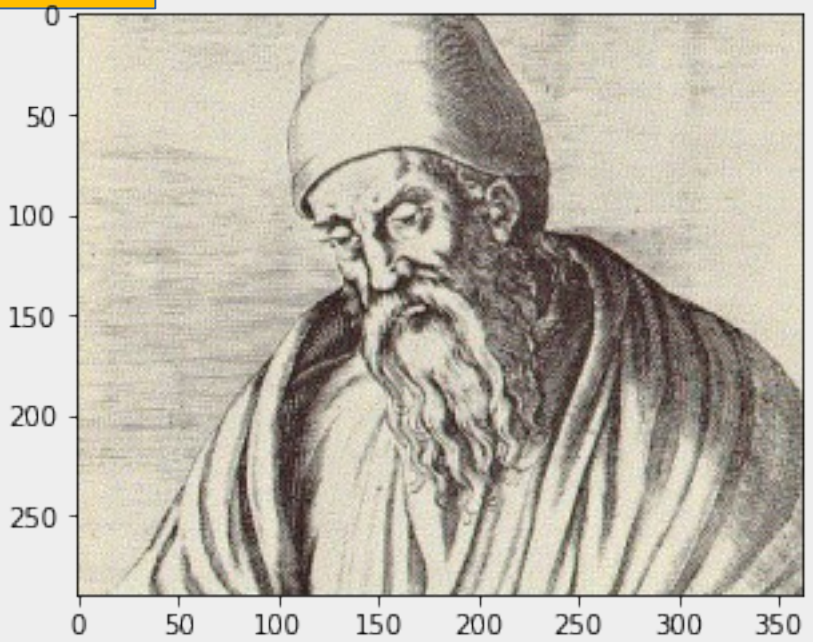
FACENET: FACE RECOGNITION TASK
FAMOUS MATHEMATICIANS



```
It's not Rene-Descartes, please
have to continue studying
Out[28]:(0.9089534, False)
```



who_is_it("images96/Euler.jpg", database, FRmodel)
It's Euler, the distance is 0.0



who_is_it("images96/Euclid.jpg", database, FRmodel)
It's Euclid, the distance is 0.0

Applied AI and NLP
NATURAL LANGUAGE PROCESSING

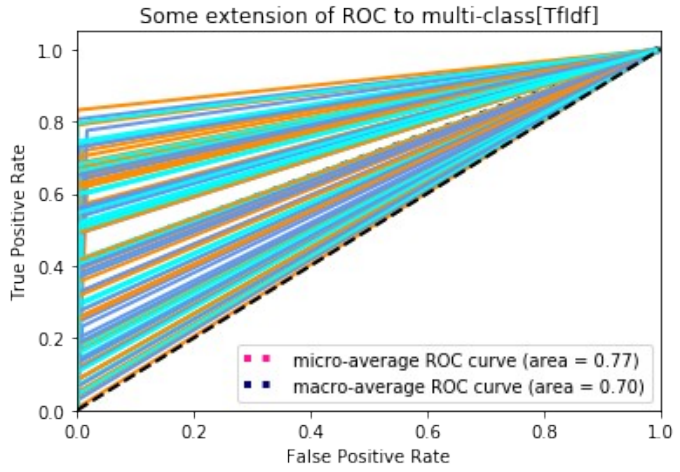
Predict tags on StackOverflow with linear models

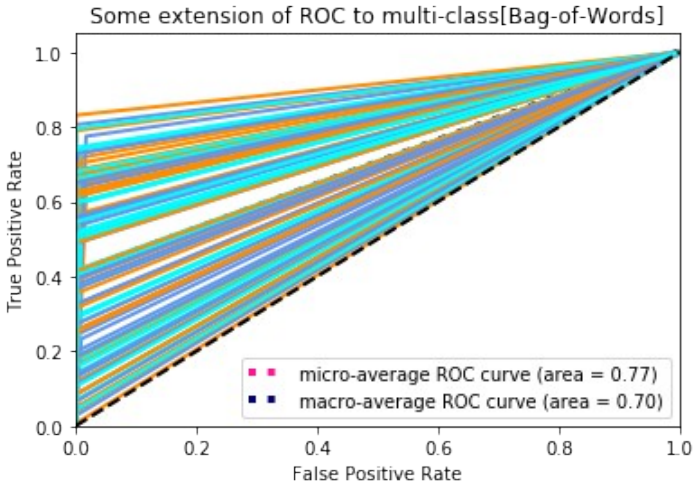
PARAMETERS

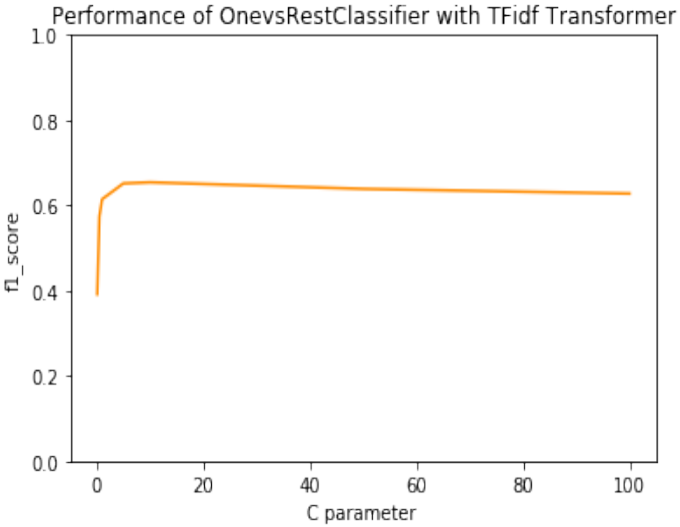
DICT_SIZE	5000
Validation split	0.2
Testing split	0.13
Bag of Words	5000
TF-IDF	18300
num_classes	100

```
LogisticRegression
Bag-of-words
0.6514944359375615
LogisticRegression
Tfidf 0.6540109906751699
Support Vector Machine
Bag-of-words
0.6581060304888663
Support Vector Machine
Tfidf 0.6614169003852886
MultinomialNB
Bag-of-words
0.6296148179299079
MultinomialNB
Tfidf 0.398129431834211
```

```
Task Multilabel
     Classification:

0       mysql,php
1       javascript
2       python
3       javascript,jquery
4       android,java
5       php,xml
6       json
7       java,swing
```





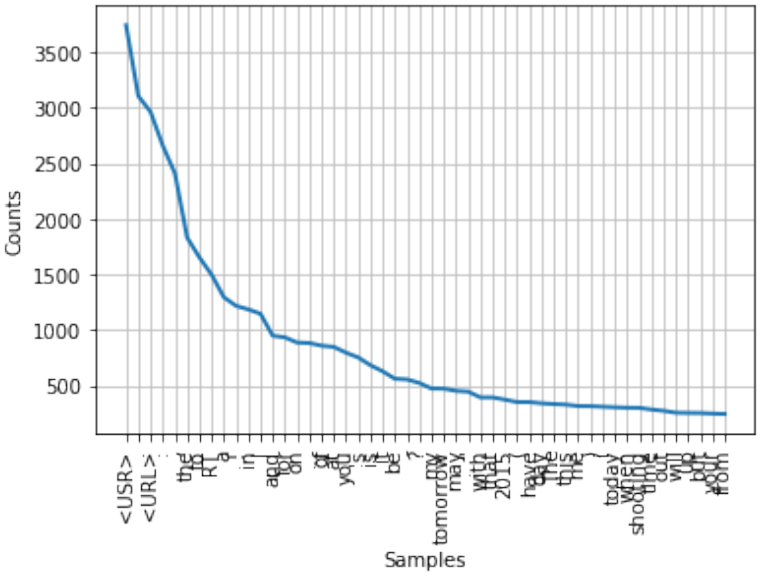


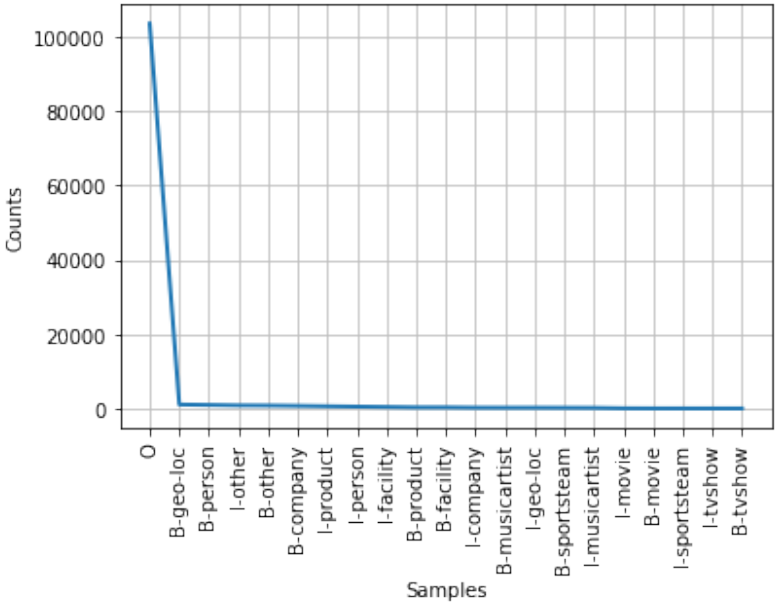
Applied AI and NLP
NATURAL LANGUAGE PROCESSING
Recognize named entities on Twitter with LSTMs

B-PER I-PER B-PRODUCT I-PRODUCT B-ORG I-ORG

PARAMETERS

vocabulary_size	112095
words_tokens	20503
sequence length	41
embedding_dim	200
n_hidden_rnn	200
PAD_index	1
batch_size	32
n_epochs	4
learning_rate	0.0075
learning_rate_decay	2.0
dropout_keep_probability	0.7





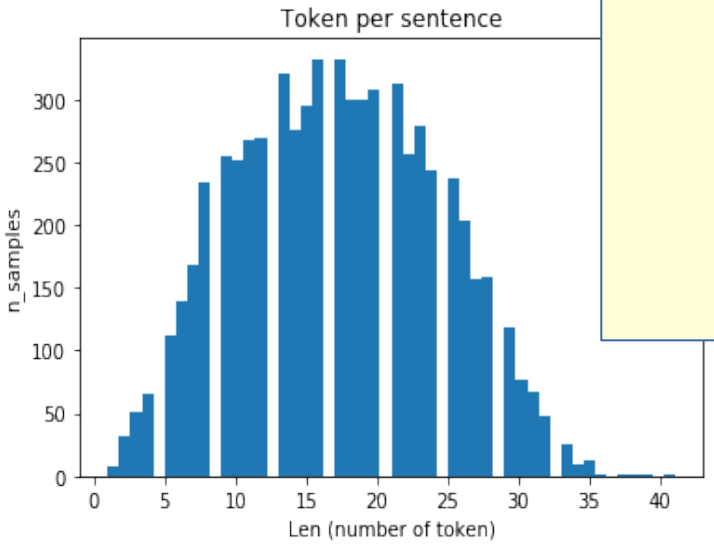

```
------------------- Train set quality: --------------------
processed 105778 tokens with 4489 phrases; found: 4651 phrases; correct: 4174.
precision:  89.74%; recall:  92.98%; F1:  91.33
             company: precision:   91.72%; recall:   94.71%; F1:   93.19; predicted:   664
            facility: precision:   89.44%; recall:   91.72%; F1:   90.57; predicted:   322
             geo-loc: precision:   93.10%; recall:   97.59%; F1:   95.29; predicted:  1044
               movie: precision:   50.00%; recall:   57.35%; F1:   53.42; predicted:    78
        musicartist: precision:   80.57%; recall:   85.78%; F1:   83.09; predicted:   247
               other: precision:   88.96%; recall:   93.66%; F1:   91.25; predicted:   797
              person: precision:   93.54%; recall:   96.50%; F1:   95.00; predicted:   914
             product: precision:   89.33%; recall:   92.14%; F1:   90.71; predicted:   328
          sportsteam: precision:   85.07%; recall:   86.64%; F1:   85.84; predicted:   221
              tvshow: precision:   61.11%; recall:   37.93%; F1:   46.81; predicted:    36
```

Applied AI and NLP
NATURAL LANGUAGE PROCESSING
Recognize named entities on Twitter with LSTMs

B-PER I-PER B-PRODUCT I-PRODUCT B-ORG I-ORG

PARAMETERS

vocabulary_size	112095
words_tokens	20503
sequence length	41
embedding_dim	200
n_hidden_rnn	200
PAD_index	1
batch_size	32
n_epochs	4
learning_rate	0.0075
learning_rate_decay	2.0
dropout_keep_probability	0.7




```
-------------------- Validation set quality: --------------------
processed 12836 tokens with 537 phrases; found: 401 phrases; correct: 192.
precision:  47.88%; recall:  35.75%; F1:  40.94
              company: precision:   59.34%; recall:   51.92%; F1:   55.38; predicted:    91
             facility: precision:   46.15%; recall:   35.29%; F1:   40.00; predicted:    26
              geo-loc: precision:   64.84%; recall:   52.21%; F1:   57.84; predicted:    91
                movie: precision:    0.00%; recall:    0.00%; F1:    0.00; predicted:     3
          musicartist: precision:   25.00%; recall:   14.29%; F1:   18.18; predicted:    16
                other: precision:   40.62%; recall:   32.10%; F1:   35.86; predicted:    64
               person: precision:   45.31%; recall:   25.89%; F1:   32.95; predicted:    64
              product: precision:   13.79%; recall:   11.76%; F1:   12.70; predicted:    29
           sportsteam: precision:   23.53%; recall:   20.00%; F1:   21.62; predicted:    17
               tvshow: precision:    0.00%; recall:    0.00%; F1:    0.00; predicted:     0
```

```
-------------------- Test set quality: --------------------
processed 13258 tokens with 604 phrases; found: 526 phrases; correct: 254.
precision:  48.29%; recall:  42.05%; F1:  44.96
              company: precision:   50.00%; recall:   46.43%; F1:   48.15; predicted:    78
             facility: precision:   51.35%; recall:   40.43%; F1:   45.24; predicted:    37
              geo-loc: precision:   73.44%; recall:   56.97%; F1:   64.16; predicted:   128
                movie: precision:    0.00%; recall:    0.00%; F1:    0.00; predicted:     5
          musicartist: precision:    9.52%; recall:    7.41%; F1:    8.33; predicted:    21
                other: precision:   39.00%; recall:   37.86%; F1:   38.42; predicted:   100
               person: precision:   55.81%; recall:   46.15%; F1:   50.53; predicted:    86
              product: precision:   10.53%; recall:   14.29%; F1:   12.12; predicted:    38
           sportsteam: precision:   27.27%; recall:   29.03%; F1:   28.12; predicted:    33
               tvshow: precision:    0.00%; recall:    0.00%; F1:    0.00; predicted:     0
```

Applied AI and NLP
NATURAL LANGUAGE PROCESSING
Learn to calculate with seq2seq model¶
Dataset of generated equations with the operators addition and substraction

PARAMETERS

vocab_size 15
embeddings_size 20
max_iter 7
hidden_size 512
start_symbol_id=word2id["^"]
end_symbol_id=word2id["$"]
padding_symbol_id=word2id["#"]
batch_size = 128
n_epochs = 10
learning_rate = 0.01
dropout_keep_probability = 0.5
max_len =20

ENCODER - DECODER ARCHITECTURE

Train: epoch 10

Epoch: [10/10],step: [1/625],l oss: 1.120122

Epoch: [10/10], step: [201/625], loss: 1.166973

Epoch: [10/10], step: [401/625], loss: 1.115895

Epoch: [10/10], step: [601/625], loss: 1.134763

Test: epoch 10 loss: 1.1185263

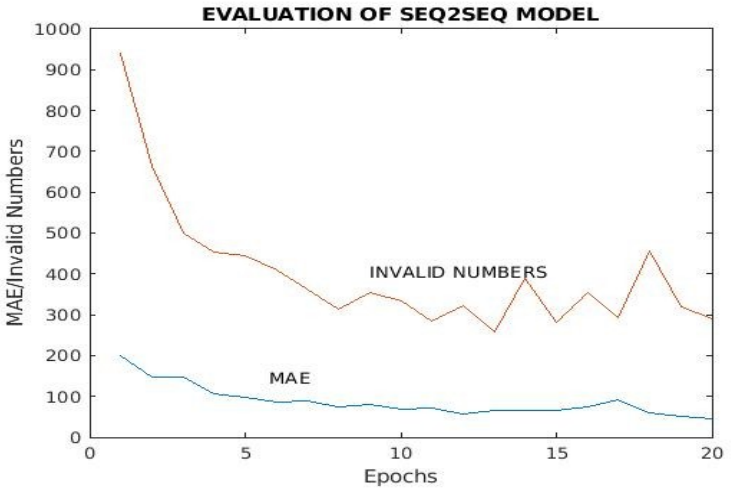
X: 8160-8424$
Y: -264$#
O: -288$#

X: 1249+7575$
Y: 8824$#
O: 8887$#

X: 565-567$##
Y: -2$###
O: -111$#

EVALUATE RESULTS

Epoch: 1, MAE: 284.882726, Invalid numbers: 1232
Epoch: 2, MAE: 151.816467, Invalid numbers: 494
Epoch: 3, MAE: 122.000563, Invalid numbers: 454
Epoch: 4, MAE: 119.092943, Invalid numbers: 472
Epoch: 5, MAE: 112.519414, Invalid numbers: 555
Epoch: 6, MAE: 97.322286, Invalid numbers: 387
Epoch: 7, MAE: 102.530941, Invalid numbers: 382
Epoch: 8, MAE: 76.065245, Invalid numbers: 305
Epoch: 9, MAE: 75.231431, Invalid numbers: 344
Epoch: 10, MAE: 69.462707, Invalid numbers: 318



Applied AI and NLP
NATURAL LANGUAGE PROCESSING
TEXT-SUMMARIZATION WITH AMAZON REVIEWS WITH ATTENTION LAYER

MODEL (NO-FILTERING) PARAMETERS
54,018,847

vocab_size	56957
embeddings_size	500
max_len_text	80
max_len_summary	10
encoder	LSTM
decoder	LSTM
batch_size	64
rnn_len	256
n_layers	2
learning_rate	0.005
dropout_keep_probability	0.75
max_len =20	

MODEL PARAMETERS

FILTERING + EMB GLOV.6B.50d

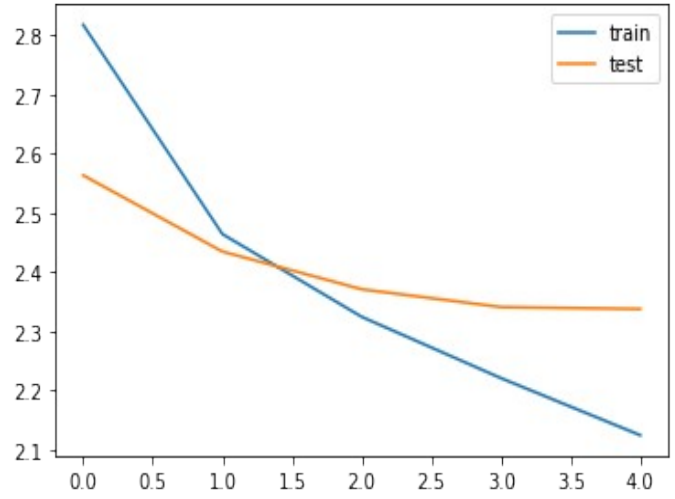
len_word_emb	36805
text/sum filtered	27125
embeddings_size	50
encoder/decoder	GRUCELL+ DROP + BDIRECT-DYNAMIC RNN
batch_size	64
rnn_len	128
n_layers	2
learning_rate	0.005
dropout_keep_probability	0.75

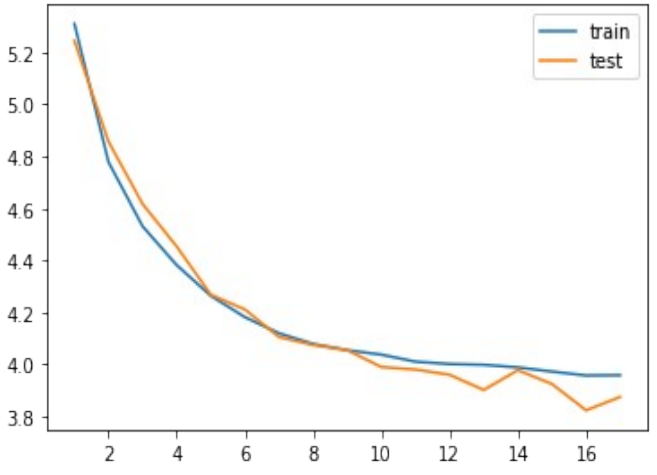
MODEL PARAMETERS

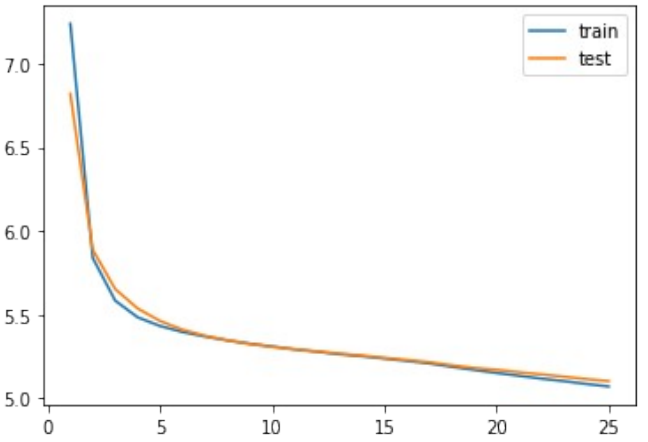
FILTERING + EMB CC.EN.300.VEC

len_word_emb	43127
text/sum filtered	27128
embeddings_size	300
encoder/decoder	GRUCELL+ DROP + BDIRECT-DYNAMIC RNN
batch_size	64
rnn_len	128
n_layers	2
learning_rate	0.00005
dropout_keep_probability	0.75

LOSS FUNCTIONS





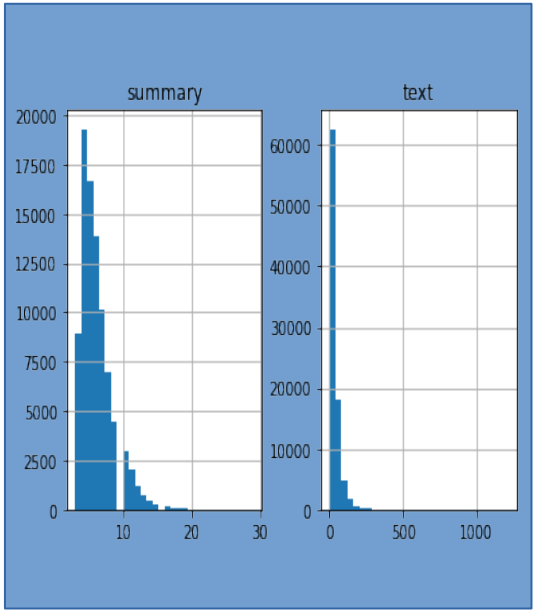


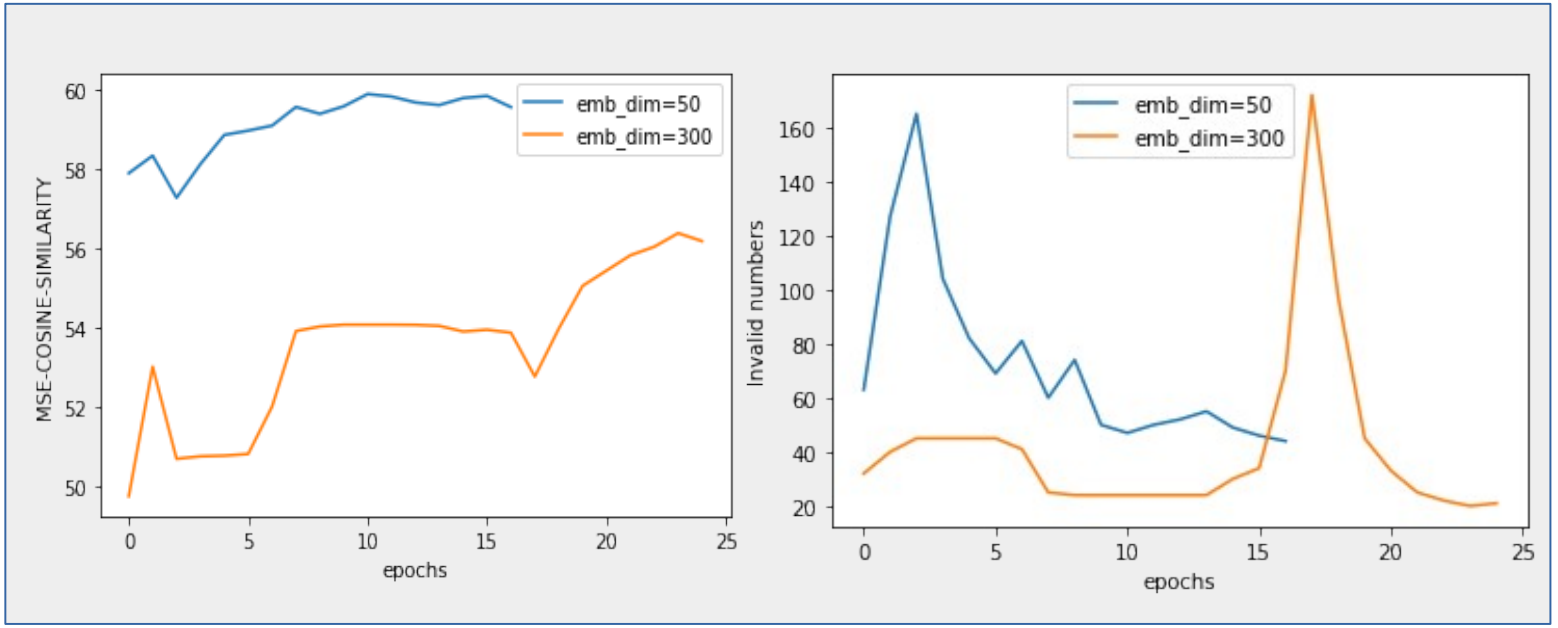
Applied AI and NLP

NATURAL LANGUAGE PROCESSING

TEXT-SUMMARIZATION WITH AMAZON REVIEWS

EVALUATION OF RESULTS






```
Embedding_dim=50
Text
  Word Ids:    [0, 721, 179, 6983, 399, 106, 460, 781, 2291, 3503, 5983, 1528, 5320, 359, 664, 71, 1738, 2802, 1646, 36803]
  Input Words: good strong coffee modified cup pack works well keurig brewer suppose less wasteful cannot beat price especially
subscribe save
Summary
  Word Ids:       [0, 179, 137, 71]
  Response Words: good coffee at price
 Ground Truth: good coffee good price
```

Applied AI and NLP

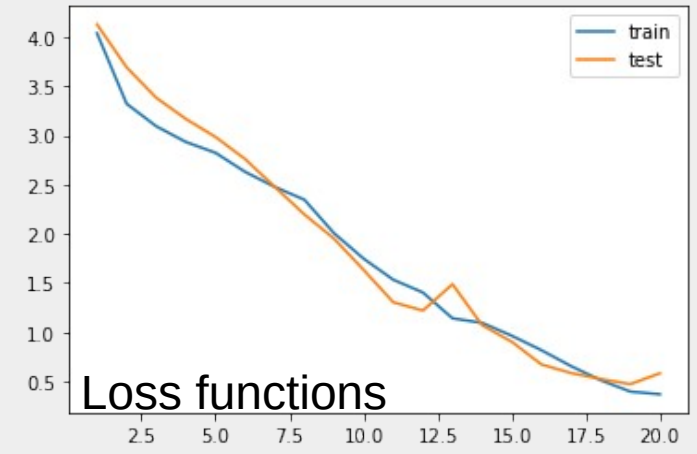
NATURAL LANGUAGE PROCESSING

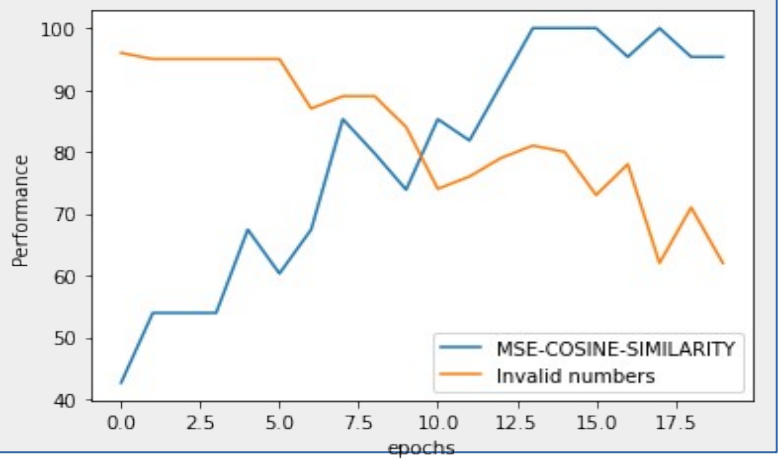
ENGLISH-CATALÀ TRANSLATION

USING FILTER + PRE-TRAINED EMBEDDINGS

PARAMETERS

eng vocab_size	991
Cat vocab _size	1166
embeddings_size	300
FILTERED WORDS	78.71
rnn_len	256
n_layers	1
batch_size	32
learning_rate	0.0025
dropout_keep_probability =	0.75





Input Language; index to word mapping

11 ----> too
12 ----> late
7 ----> .
3 ----> <EOS>
3 ----> <EOS>

Target Language; index to word mapping
12 ----> massa
13 ----> tard
11 ----> .
3 ----> <EOS>

```
Text
  Word Ids:    [0, 305, 47, 81, 618, 213, 73, 619, 7, 3, 3]
  Input Words: tell me your plans for the future .
Summary
  Word Ids:       [692, 56, 123, 693, 694, 149]
  Response Words: explica m els teus plans per
 Ground Truth: explica m els teus plans per al futur .

                                    Text
                                      Word Ids:    [0, 16, 15, 70, 644, 189, 16, 645, 49, 7, 3, 3]
                                      Input Words: i can t remember where i bought it .
                                    Summary
                                      Word Ids:       [82, 724, 210, 19, 71, 306]
                                      Response Words: no recorde on el vaig comprar
                                     Ground Truth: no recordo on el vaig comprar .
Text
  Word Ids:    [0, 632, 67, 633, 80, 634, 635, 7, 3, 3]
  Input Words: bangkok is thailand s capital city .
Summary
  Word Ids:       [712, 40, 98, 713]
  Response Words: bangkok es la capital
 Ground Truth: bangkok es la capital de tailandia .
```

Applied AI and NLP
NATURAL LANGUAGE PROCESSING
StackOverflow assistant bot

Dataset size 40000 Intent recognizer Test accuracy = 0.991795454545454546
Test size 11% Tag classifier Test accuracy = 0.800625

RANKING QUESTIONS WITH EMBEDDINGS

‘c#’:	394451
'java':	383456
'javascript':	375867
'php':	321752
'c_cpp':	281300
'python':	208607
'r':	36359
'ruby':	99930
'vb':	35044
'swift':	34809