Anàlisis de la relació entre les emisions de gasos hivernacle i l'augment de la temperatura, mitjançant tècniques i tecnologies del Big Data

Miquel Freixes Faya

Resum– Resum del projecte, màxim 10 línies

Paraules clau– Paraules clau del treball, màxim 2 línies

Abstract– Versió en anglès del resum

Keywords– Versió en anglès de les paraules clau

TAULA 1: TAULA DE TEMPERATURES

dt	Temp	Average	Country
01-01-1750	1.12	2.245	Germany
01-02-1750	2.234	1.3	Germany

1 OBJECTIUS

2 ESTAT DE L'ART

3 METODOLOGIA

4 DESENVOLUPAMENT

4.1 Preprocessament de dades

Per començar s'han importat els dos *DataSets* des del seu csv corresponent: El primer està format de quatre columnes, tal com es mostra a la taula 1 hi ha una amb la data, en format String amb dies mesos i anys. El segon valor és la temperatura mesurada amb un marge d'error del 95%, serà el valor que es donarà per vàlid al llarg del treball. El tercer és el marge d'error que pot tenir la temperatura de la segona columna i l'última el nom del país al qual s'està mesurant les temperatures. Aquest *DataSet* està extret de Kaggle, però és una recopilació d'investigacions fetes per Berkeley Earth [2]. És una empresa sense ànim de lucre destinada a revisar i documentar l'increment de temperatura al llarg dels anys [3]. Aquest *DataSet* és una compressió de dades extretes de més d'1,6 bilions de registres entre els anys 1750 i el 2013 de tot el món per cada mes de l'any.

El segon està compost pel mateix nombre de columnes. Com es pot veure a la taula 2, és bastant semblant a l'anterior. La primera és el nom del país on s'està mesurant, la segona és l'any on s'ha mesurat, aquest cop en format d'enter. En la tercera tenim el valor de la mitjana de Kilotones de gas produït per capita en el país, i en l'última tenim el tipus de gas que s'ha mesurat i quins paràmetres s'han utilitzat per fer-ho. En aquest *DataSet* s'han extret les dades de les emissions de cada país, dels diferents gasos hivernacle des de l'any 1990 al 2013 [4]. La majoria de mesures estan extretes sense considerar les activitats d'ús de la terra o d'intercanvi de recursos amb la terra, anomenades en anglès com a LULUCF (*Land Use, Land-Use Change and Forestry*). Bàsicament aquestes activitats comprenen alguns sectors de l'agricultura i/o obtenció de matèries primeres [5]. Aquestes activitats representen un 7% del total d'emissions, per tant s'analitzarà en el treball el 93% restant que representa tots els altres sectors com el de la indústria, el turisme, el consum particular, entre d'altres [6].

Un cop entesos els *DataSets* es pot començar amb el seu tractament. Per fer-lo s'ha utilitzat l'eina en línia de Databricks, optimitzada per treballar amb Spark, a més de permetre treballar amb Pandas i d'altres frameworks o llibre-

• E-mail de contacte: m.freixes.faya@gmail.com
• Menció realitzada: Tecnologies de la Informació
• Treball tutoritzat per: Jordi Casas Roma (departament)
• Curs 2018/19

TAULA 2: TAULA DE GASOS

Country	Year	Value	Type
Germany	1990	432012.435	Co2_lulucf
Germany	1990	474532.8236	CH4_lulucf
Germany	1990	29645.74	N2O_lulucf

ries de processament de dades. Aquesta eina té una versió de pagament i una gratuïta, en aquest treball s'ha escollit la gratuïta que proporciona un clúster amb un màxim de 6Gb de RAM [7]. S'ha preferit utilitzar aquesta opció, ja que ha evitat tot el procés de muntar un sistema en una màquina per treballar amb Spark, permetent dedicar més temps al processament de les dades. En aquesta eina s'ha utilitzat el *framework* de pySpark, i diverses llibreries com Pandas, Numpy i SciPy.

Per una banda tenim Pandas, una llibreria de codi obert amb llicència de *Berkeley Distributed Systems* optimitzada per al processament de grans volums de dades. Aquesta llibreria proporciona estructures de dades que són flexibles i ràpides que permet treballar amb elles d'una forma fàcil. El seu element bàsic és el *DataFrame*, que es pot representar com una taula amb files i columnes. Aquest element és molt més fàcil i eficient que un diccionari o una llista gràcies al fet que no cal implementar estructures de bucles molt complicades per anar iterant sobre els seus valors [8].

Per l'altra banda tenim Spark, un entorn creat per obtenir un processament i una anàlisi de grans quantitats de dades en entorns distribuïts, obtenint un alt rendiment i una gran velocitat. Això ho aconsegueix a partir de distribuir les dades pel clúster i treballant en memòria, no en disc. En programar el treball en Python s'ha utilitzat el framework pySpark, bàsicament ofereix totes les eines de Spark pel llenguatge. L'element més bàsic i important en Spark és el *resilient distributed dataset* o RDD, és la base de la seva estructura de paral·lelització. El RDD és un objecte abstracte que representa un conjunt de dades, que estan distribuïdes pel clúster. Aquests objectes són immutables i poden emmagatzemar-se en memòria [9]. A banda dels RDD, un altre element important és el *DataFrame*. Aquest element és un concepte pràcticament idèntic al *DataFrame* utilitzat per Pandas. Utilitzant DataBricks els *DataFrames* de Spark es poden visualitzar de moltes maneres diferents gràcies a l'interfície que ofereix l'eina.

Amb aquestes dues opcions sobre la taula, s'ha preferit començar utilitzant Spark gràcies al fet que l'eina de DataBricks estava optimitzada per ella. Primer s'ha importat els dos *DataSets* i se'ls ha transformat a *DataFrames*. Un cop fet iniciat el procediment de canviar les dades, tant el nombre de dades com el format, s'ha començat a trobar alguns problemes ja que no s'acabava de trobar la manera de fer alguns canvis necessaris per treure el format correcte. Per exemple, era difícil iterar cada fila canviant valors específics en cada una d'elles. Per poder fer aquests processos s'ha decidit passar els *DataFrames* a la llibreria Pandas. Aquesta decisió ha anat condicionada pel gran volum d'informació que hi ha a Internet sobre com fer el tractament de dades en Pandas i també per la facilitat que proporciona la llibreria amb algunes llibreries científiques com Numpy i SciPy.

Un cop passats a *DataFrames*de Pandas s'han modificat i adaptat cada un per després unir-los en un de sol:

- Amb el de les temperatures, s'ha començat descartant tots els anys que no té el *DataSet* de les emissions, a causa del fet que l'algorisme de *Machine Learning* ha d'agafar el mateix període de temps per poder comparar els valors. Un cop descartats s'ha quedat el període de 1990 al 2012, amb dotze mesures mensuals per cada any. A partir d'aquí, s'ha descartat tots els països que no estiguin dins de la Unió Europea i s'ha canviat el format de les dates, passant-les de String a enter i eliminant els dies.

- El de les emissions ha estat una mica més complicat de modificar. Primer s'ha hagut de filtrar els gasos i països desitjats, per descartar totes les dades innecessàries. Després, ha aparegut el principal problema, en aquest *DataFrame* les mesures es fan per any, no per mes com en el de les temperatures. Per solucionar-ho primer s'ha plantejat el fet de canviar les temperatures a mesures anuals, fent la mitjana de les mensuals, però el nombre de files restants no arribava a les mínimes per treure resultats bons amb els algorismes de *Machine Learning*. Com l'única solució restant era transformar una mesura per any a una per més, ha calgut fer una interpolació de les dades. Una interpolació és el mètode matemàtic que permet construir un conjunt de punts a partir d'uns de coneguts. Existeixen diversos tipus d'interpolació, d'entre les quals s'ha considerat aplicar en aquest treball la lineal, la polinòmica i la de traçadors. La primera s'ha descartat a causa de la poca precisió que aporta en la creació de nous punts. La segona s'ha considerat com una opció viable, però té un hàndicap molt gran amb polinomis de graus elevats anomenat el Síndrome de Runge, en el cas d'aquest treball és de grau 12. Aquest fenomen provoca una gran desviació de les dades als extrems i al centre de la interpolació, afectant greument a la precisió de les dades [10]. Per evitar aquest fenomen s'ha aplicat el tercer mètode d'interpolació. Aquest consisteix a dividir els punts coneguts en polinomis de grau tres, aquests polinomis es poden interpolar amb la interpolació polinòmica, evitant el fenomen de Runge gràcies al seu grau baix. Un cop interpolats, es van encadenant un darrere l'altre aconseguint la gràfica completa amb tots els punts desitjats. Amb aquest mètode s'aconsegueix evitar les desviacions a més d'aconseguir una bona precisió en els punts creats [11]. Havent escollit el mètode d'interpolació, s'ha buscat la millor manera de programar-ho en Python i s'ha utilitzat un mètode de la llibreria SciPy [12] on només cal enviar-li els punts coneguts i el nombre de punts que es vol aconseguir per cada interpolació. Per crear els punts s'ha utilitzat una funció d'una altra llibreria, Numpy, la qual ha transformat els anys i les quantitats en les coordenades x i y d'un punt.

Un cop editats els dos *DataFrames* per separat s'han unit en un de sol i s'ha passat a un *DataFrame* de Spark per poder visualitzar-lo millor. Gràcies a les eines que ofereix Databricks per fer múltiples gràfiques a partir de les dades en Spark es pot observar bé si hi ha hagut errors en aquest

procés, o si hi ha algun tipus d'inconsistència en les dades. Un cop revisades les dades d'aquest últim *DataFrame* s'ha exportat a un fitxer csv que serà el que utilitzarà l'algorisme de *Machine Learning*.

4.2 Entrenament i avaluació dels algorismes

Aquesta fase s'ha centrat a provar diferents algorismes de regressió de la llibreria Scikit-Learn per determinar quin pot aconseguir predir millor les temperatures segons les diferents emissions de gasos hivernacle. Abans de començar a entrenar algorismes, calia entendre bé quines eren les dades que formaven el *DataSet*. Encara que en la fase anterior s'hagués creat el *DataSet* calia veure si les temperatures variaven molt entre diferents anys, i si era una variació constant o amb una certa aleatorietat. Per fer-ho s'ha tret uns gràfics amb la temperatura de Gener al llarg de diversos anys. En la figura 1 es pot veure les temperatures del mes de Gener a Austria. És visible que la temperatura no presenta cap patró que pugui facilitar l'obtenció de bons resultats, ja que hi ha variacions d'1 o 2 graus però també de 4 o 5 en alguns anys. A més d'analitzar el *DataSet* també ha

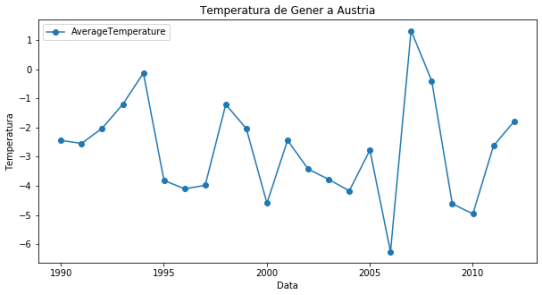

Fig. 1: Temperatures

estat necessari passar els països a dades numèriques, ja que Scikit-Learn no permet *Strings* per entrenar els seus algorismes [2]. Aquesta acció s'havia de fer d'una manera que no convertís les variables en categòriques, és a dir, que l'algorisme no entengués que ser d'un país o d'un altre tenia una relació numèrica. Per fer-ho s'ha utilitzat un mètode de la mateixa llibreria de SciKit-Learn que transforma els *String* en variables binàries, per tant si és d'un país, serà un 1 i si no ho és, serà 0.

Un cop sabent que el *DataSet* era adequat per fer aquestes anàlisis, s'ha començat a escollir i entrenar diversos algorismes de regressió. Els algorismes de regressió són tasques d'aprenentatge inductiu [2]. Es diferencien en les tasques de classificació a causa del fet que en la regressió es prediuen valors numèrics, en canvi en la classificació es prediuen etiquetes de classe. La tasca de regressió consisteix en l'assignació de variables sobre un domini donat, aquestes variables estan descrites per atributs discrets. A partir d'aquest conjunt de dades, els algorismes de regressió assignen valors numèrics a instàncies d'aquest domini amb la finalitat d'aconseguir una aproximació a partir de valors que es donen [3].

Dins de la regressió s'ha decidit agafar un conjunt de mètodes que es considerin dins de l'estat de l'art actual en temes de *Machine Learning* (ML).

- **Arbres de Decisió:** Els arbres de decisió són models predictius no linears que serveixen per fer tasques de regressió i tasques de classificació. La idea és simple, si es vol predir una classe o resposta Y a partir d'entrades X cal fer créixer un arbre binari. En els nodes d'aquest arbre binari s'aplica un test sobre l'input que es consideri més rellevant, per exemple en la figura 2 es comença preguntant sobre la contaminació de CO_2. En aquest test es fa una pregunta binària de si o no, si la resposta és si es passarà al fill de la dreta, si la resposta és no, es passarà al fill de l'esquerra. A partir d'aquí es van fent tests en cada un dels nodes, l'input que s'agafa per formular la pregunta pot variar segons les dades que s'estiguin treballant. Com es mostra a la figura 2 si s'envà cap a l'esquerra, es passa a testejar l'input de les emissions de CH_4. A la figura també es pot observar com les prediccions sobre quina és la temperatura final s'acaben fent en arribar a les fulles de l'arbre. Aquestes prediccions vénen donades pel conjunt de totes les preguntes que s'han anat formulant en els seus antecessors [4]. Quan hi ha molts inputs a l'hora d'en-

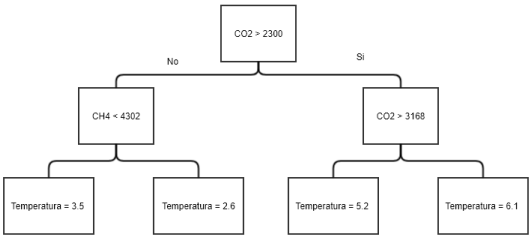

Fig. 2: Arbre de Decisió Bàsic

trenar l'arbre de decisió, el mateix arbre ha d'escollir quin paràmetre dels inputs cal avaluar primer. Una mala elecció d'aquest input pot comportar que es comenci a fer comprovacions innecessàries, causant un impacte negatiu en l'eficiència de l'entrenament. L'arbre de decisió fa aquesta elecció a partir d'una recerca mitjançant la força bruta, en aquesta s'intenta buscar quina és la pregunta que pot aportar més informació de cara als següents nodes. Aquesta decisió la fa en cada un dels nodes i va variant fins a arribar a una fulla. Si en aquesta recerca es trobessin dues preguntes sobre dos inputs, que donen la mateixa quantitat d'informació, l'arbre escull una arbitràriament [4].

Per acabar, els arbres de decisions tenen molts paràmetres editables com la seva longitud màxima, el nombre màxim de fulles, el mínim nombre de paràmetres per considerar el node com una fulla... La majoria d'aquests paràmetres limiten el creixement de l'arbre, ja que per defecte creix fins a fer totes les preguntes possibles. Aquests paràmetres són molt importants en l'obtenció de bons resultats, parar el creixement d'un arbre, en una profunditat equivocada, pot fer que l'arbre acabi descartant gran part d'informació, ja que no tots els nodes donen la mateixa quantitat d'informació a l'arbre[4].

- **Support Vector Machines:** Support Vector Machines o SVM és un mètode d'aprenentatge supervisat que

es poden utilitzar per a la classificació i la regressió. Aquest té com a objectiu principal definir una funció f(x) que tingui com a molt una desviació o error per sota d'un límit establert en la creació del mètode. Per exemple, si s'estableix un error igual a 1, es descartaran tots els objectius amb un error superior a 1, i es consideraran tots els que tinguin un valor inferior. El SVM treballa amb productes sobre punts dins de dimensions definides, això troba les familiaritats entre dos vectors. SVM tenen un paràmetre essencial anomenat kernel, bàsicament són un conjunt de fórmules matemàtiques que defineixen com es processarà les dades que entren com a inputs. Els kernels són els encarregats de modificar el producte de punts establert pel SVM i adaptar-lo al seu espai. Per defecte el SVM té com a kernel el lineal, que limita el producte de punts original en dues dimensions. Si en lloc del lineal s'utilitzes un kernel com el polinòmic, aquest espai acceptaria també combinacions polinòmiques fins a certs graus, aconseguint fer paràboles [5]. A més d'aquests dos, hi ha un de molt utilitzat anomenat Radial Basis Function o RBF. Aquest utilitza un espai limitat per distribucions Gaussianes, en la majoria d'ocasions aquest kernel és el que pot arribar a obtenir els millors resultats [6].

- **Random Forest:** El Random Forest és un dels mètodes de ML que s'està utilitzant més en l'actualitat, com els altres dos, es pot utilitzar tant per classificació com per regressió. La idea principal del Random Forest és la combinació de molts arbres de decisió iguals relacionats entre ells, això ho fa gràcies al fet que es basa en la idea que la combinació de molts algoritmes iguals aconsegueix millors resultats que un de sol molt potent. Com es pot veure a la figura 3, que representa un Random Forest amb dos arbres, el mètode crea dos arbres de decisió i després combina els resultats de cada un d'ells mitjançant el Bagging Method o un altre algoritme segons la implementació que s'esculli [7].

  Aquest mètode té pràcticament els mateixos paràmetres que un arbre de decisió però amb petites diferències que optimitzen cada una de les decisions preses per l'arbre. Com s'ha dit anteriorment els arbres necessiten escollir quina pregunta s'ha de fer en cada node per aconseguir informació a l'hora de separar les variables, el Random Forest modifica aquesta elecció. En lloc d'escollir la millor opció de totes agafa un subconjunt dels inputs que arriben al node i escull la millor dins d'aquest subconjunt [8]. El Random Forest permet editar molts paràmetres entre els quals destaquen el nombre d'arbres de decisió que es combinaran, el nombre d'inputs que s'inclouran en el subconjunt aleatori, també es pot escollir tots els paràmetres per editar els arbres de decisió o fins i tot l'algorisme que combinarà els seus resultats.

D'aquests algorismes s'ha fet un entrenament amb els paràmetres per defecte marcats per la llibreria de Scikit-Learn i després s'ha utilitzat dues tècniques, molt utilitzades en l'àmbit del ML per trobar el conjunt de paràmetres més òptim per cada algorisme. Aquestes tècniques tenen com a objectiu esprémer al màxim els algorismes i aconseguir l'aproximació més precisa de la temperatura. Aquestes tècniques han estat:

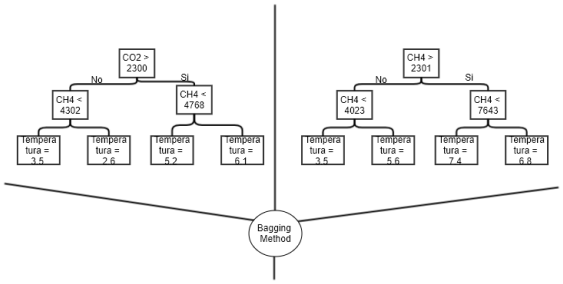

Fig. 3: Random Forest amb 2 arbres

- **Grid SearchCV:**)Implementa una recerca exhaustiva dels paràmetres editables d'un mètode de predicció. Un cop fet aquest anàlisis entrena l'algorisme amb la millor combinació de paràmetres que ha trobat. Amb cada una de les combinacions fa 3 iteracions i després utilitza una fórmula per avaluar el resultat obtingut, aquest pot ser el Mean Absolute Error o MAE, Mean Squared Error [9]... Les iteracions dels paràmetres les fa en ordre ascendent, comença pels paràmetres més petits i va augmentant-los fins al límit que se l'hi ha establert.

- **Randomized SearchCV:** És un mètode molt semblant al Grid Search. També fa una recerca sobre els millors paràmetres que aconsegueixin treure el millor resultat del mètode. La principal diferència és que en lloc de fer-ho de forma iterativa i ascendent ho fa de forma iterativa però aleatòriament. Això permet que es puguin provar valors molt elevats sense necessitat de què la recerca augmenti exponencialment el temps necessari per acabar [10].

5 RESULTATS

Per aconseguir els resultats s'ha hagut d'aplicar tant el Grid Search com el Randomized Search amb les mateixes iteracions als tres algorismes escollits, amb els mateixos conjunts de dades per cada un d'ells. En tots els casos els millors resultats els ha donat la tècnica de Randomized Search. Això s'ha donat a causa del fet que la tècnica pot editar paràmetres molt més grans i diversos. El Grid Search no pot fer el mateix pel fet que el Randomized utilitza iteracions de valors aleatoris, dins d'un rang molt més gran de valors. En canvi, el Grid Search està limitat a anar fent proves seqüencialment, començant pels valors més petits i fins a arribar al màxim del rang que s'ha establert. Al cap i a la fi el Grid Search està limitat per la quantitat elevada de càlculs que necessita fer. Per exemple, ficant el mateix nombre de variables, amb el mateix rang de valors en cada una d'elles, per arribar al valor màxim el Grid Search ha de fer X iteracions prèvies (on X pot ser un número molt elevat d'iteracions), en canvi el Randomized ho pot fer a la primera sense passar per totes les anteriors.

Amb el Randomized s'ha aconseguit els millors resultats per cada un dels algorismes aplicats. Com es pot veure a la figura 4 l'algorisme que ha aconseguit aproximar-se més als valors originals ha estat el Random Forest, en canvi el pitjor ha estat l'algorisme de les Support Vector Machine. Veient

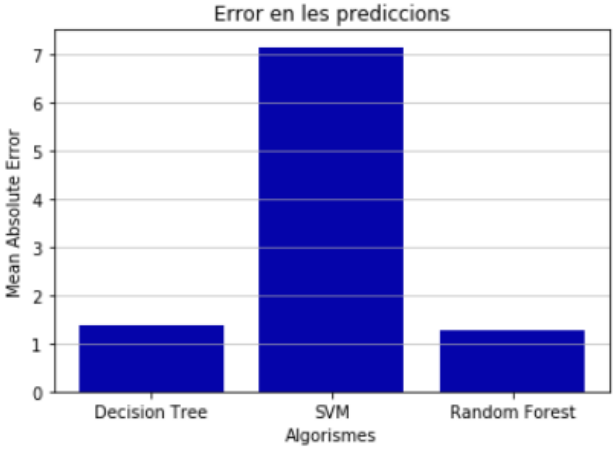

Fig. 4: MAE dels mètodes

les comparacions amb les temperatures reals de la figura 5 es pot veure com les SVM no s'està adaptant bé a cap canvi de temperatura, bàsicament es manté constant en 10 graus excepte en tres ocasions que sembla que intenta aproximar-se més a la temperatura real. Això pot ser degut a l'elecció de kernel o del paràmetre C. Tal com es mostra a la gràfica les SVM no estan tenint un resultat gens esperat, els resultats poden venir donats també per una incompatibilitat entre el conjunt de dades i l'algorisme. Els altres dos algorismes si que s'aproximen molt en les seves prediccions, fallen entre 1 i 1.5 graus en la majoria de temperatures.

Aquests resultats són molt bons a causa del fet que l'error és bastant baix, considerant que la temperatura varia d'un any per l'altre entre 2 i 4 graus. No obstant això, els dos mètodes tenen un comportament una mica diferent a l'hora de fer prediccions. El Random Forest manté bastant la constància amb les temperatures que no varien massa, en canvi sí que falla bastant quan hi ha un canvi brusc de temperatura. Per l'altra banda els Decision Tree fallen en la gran majoria de prediccions però amb un error baix en tots ells, això fa que el MAE d'ambdós sigui molt semblant, encara que el Random Forest acabi tenint un més baix.

Com a idees per continuar aquest TFG es podria fer la mateixa anàlisi de les dades però en un àmbit mundial en lloc d'Europeu. Segurament els resultats siguin encara millors gràcies a un nombre més gran de dades a analitzar. Una altra possible continuació pot ser l'increment d'algorismes de ML amb els que analitzar les dades, ja que en aquest treball s'han obviat alguns mètodes com les Xarxes Neuronals, que actualment estan donant resultats molt impressionants en l'àmbit del ML i del Deep Learning.

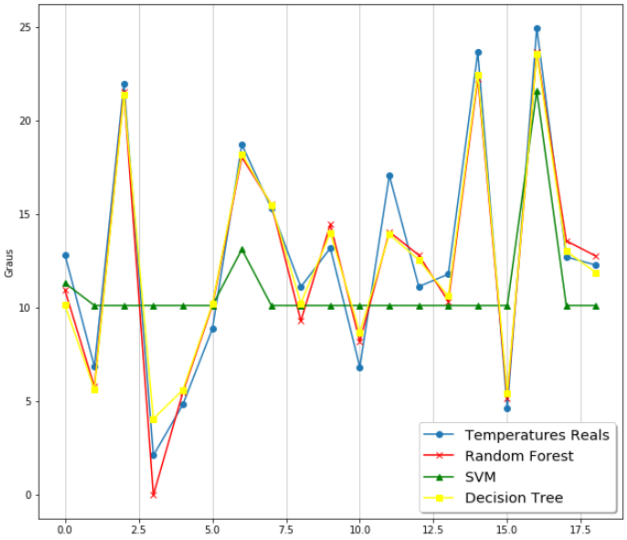

Fig. 5: Comportament dels mètodes

6 Conclusions

Referències

[1] "Climate Change: Earth Surface Temperature Data", *Berkeley Earth*. [En línia]. Disponible a: https://www.kaggle.com/unitednations/international-greenhouse-gas-emissions. [Accedit Novembre 9, 2018].

[2] "About Berkjeley Earth", *Berkeley Earth*. [En línia]. Disponible a: http://berkeleyearth.org/about/. [Accedit Novembre 9, 2018].

[3] "International Greenhouse Gas Emissions", *United Nations*. [En línia]. Disponible a: https://www.kaggle.com/unitednations/international-greenhouse-gas-emissions. [Accedit Novembre 9, 2018].

[4] "Land Use, Land-Use Change and Forestry (LULUCF)", *United Nations*. [En línia]. Disponible a: https://unfccc.int/topics/land-use/workstreams/land-use--land-use-change-lulucf. [Accedit Novembre 7, 2018].

[5] Federica Pozzi, "Importancia del recuento del UTCUTS (LULUCF) para el éxito del Acuerdo de París". *Carbon Market Watch*. [En línia]. Disponible a: https://carbonmarketwatch.org/es/2017/07/18/29671/. [Accedit Novembre 7, 2018].

[6] "Databricks Unified Analytics", *DataBricks*. [En línia]. Disponible a: https://databricks.com/. [Accedit Novembre 6, 2018].

[7] "Python Data Analysis Library", *Pandas*. [En línia]. Disponible a: https://pandas.pydata.org/. [Accedit Novembre 7, 2018].

[8] F.Julbe, *Anàlisi de dades massives: Tècniques fonamentals*. Barcelona, UOC. Pàgines: 30-32.

[9] B.Fornberg, J.Zuev. *The Runge phenomenon and spatially variable shape parameters in RBF interpolation*. University Of Colorado.

[10] "Interpolation", *Wikipedia*. [En línia]. Disponible a: https://en.wikipedia.org/wiki/

Interpolation#Spline_interpolation. [Accedit Novembre 10, 2018].

[11] "scipy.interpolate.InterpolatedUnivariateSpline", *SciPy*. [En línia]. Disponible a: https://docs.scipy.org/doc/scipy/reference/generated/scipy.interpolate.InterpolatedUnivariateSpline.html. [Accedit Novembre 10, 2018].

[12] "Encoding Categorical Features", *Scikit-Learn*. [En línia]. Disponible a:https://scikit-learn.org/stable/modules/preprocessing.html#encoding-categorical-features. [Accedit Desembre 22, 2018].

[13] F.Julbe, *Anàlisi de dades massives: Tècniques fonamentals*. Barcelona, UOC. Pàgines: 43-45.

[14] R. Tibshirani, *Classi*

cation and Regression Trees. Machine Learning Department, Carnegie Mellon University. 2009.

[15] Alex J. Smola, B. Schölkopf, *A tutorial on support vector regression*. RSISE, Australian National University. 2003.

[16] https://data-flair.training/blogs/svm-kernel-functions/

[17] "Kernel Functions-Introduction to SVM Kernel & Examples", *Data Flair*. [En línia]. Disponible a:https://towardsdatascience.com/the-random-forest-algorithm-d457d499ffcd. [Accedit Desembre 22, 2018].

[18] L. Breiman, *Random Forests*. Statistics Department, University of California Berkeley. 2001.

[19] "Grid Search CV", *Scikit-Learn*. [En línia]. Disponible a:https://scikit-learn.org/0.15/modules/generated/sklearn.grid_search.GridSearchCV.html#sklearn.grid_search.GridSearchCV. [Accedit Desembre 22, 2018].

[20] "Randomized Search CV", *Scikit-Learn*. [En línia]. Disponible a:https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.RandomizedSearchCV.html. [Accedit Desembre 22, 2018].

7 ANNEX

- **Temperatures Espanya:**

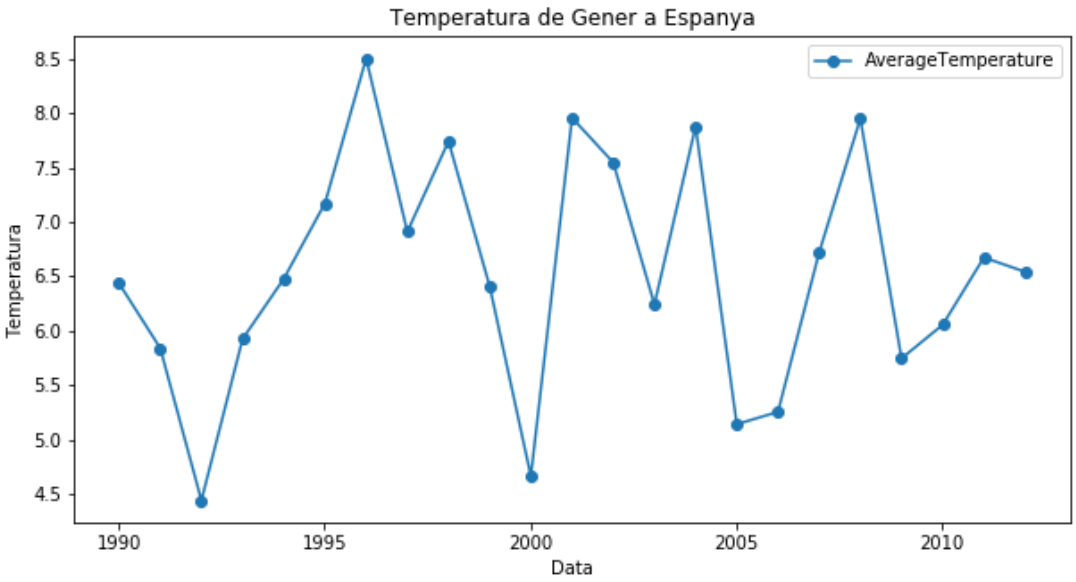

Fig. 6: Temperatura a Espanya

- **Temperatures Alemanya:**

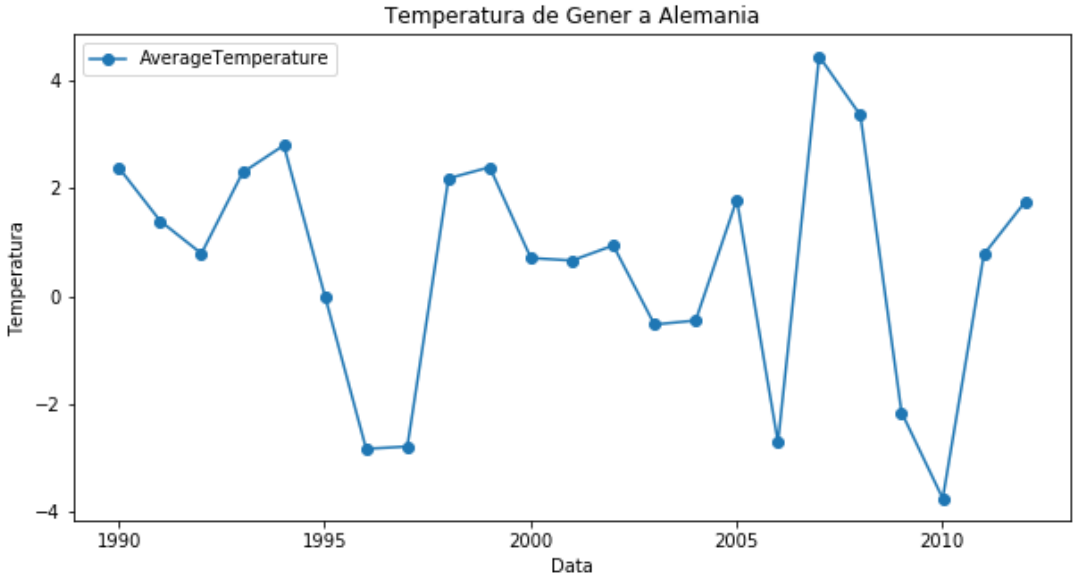

Fig. 7: Temperatura a Alemanya