

UC3MDeep at PoliticEs 2022: Exploring Traditional Machine Learning Algorithms for Political Ideology Detection

Álvaro García-Ochoa Martín-Forero^{1,†}, Alejandro Massotti López^{1,†} and Isabel Segura-Bedmar¹

¹Universidad Carlos III de Madrid (UC3M University), Av. de la Universidad, 30, 28911 Leganés, Spain

Abstract

Social media has played an important role in shaping political discourse over the last decade. It is often perceived to have increased political polarization, thanks to the scale of discussions and their public nature. Automatic political ideology detection allows us to identify the bias of information sources as well as professionals such as journalists. It has also become a relevant area due to its successful application to user behaviour analysis and prediction of malicious user versus legitimate user. This paper describes our participation at PoliticEs@IberLEF2022 shared task, whose goal is classify the political ideology of a person as well as other related information such as profession or gender. We explore several machine learning models to address the task of political ideology detection.

Keywords

Political ideology detection, Logistic Regression, Random Forest, kNN, Text classification

1. Introduction

The political ideology is the combination of beliefs, values and ideas that define a person individually and socially. [1] Political ideology can be used to understand the individual and social behaviour. Several previous works have studied the relationship between personality traits and ideology [2, 3]. For example, the study described in [4] analyzed why liberals are happier than conservatives. Moreover, our political ideology has a great influence in our daily lives. The study [5] describes a correlation between political ideology and citizens attitude towards vaccination campaigns, which can help to identify those groups where the promotion of these campaigns could be more challenging.

Furthermore, the number of people who get the news from social media is increasing. According to a recent survey by the Pew Research Center, more than 80% of American adults get the news from social media. Although social media have brought benefits, they also promoted echo chambers. In [6], the authors describe a correlation between the increase in political polarization and the echo chambers created in social media. It is in these echo chambers where fake news, ideologically acceptable for its members, are propagated and distributed, fostering


IberLEF 2022, September 2022, A Coruña, Spain.

[†]These authors contributed equally.

✉ agarciao@inf.uc3m.es (G. Martín-Forero); amassott@inf.uc3m.es (A. M. López); isegura@inf.uc3m.es (I. Segura-Bedmar)



© 2022 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

 CEUR Workshop Proceedings (CEUR-WS.org)

intolerance and affecting voting behavior. Therefore, the detection of possible bias in the different information sources can help to ensure important values in a democratic society such as objectivity, unbiased reporting, promotion of plurality, or democracy.

So far, most attempts at political ideology detection have focused on English [7]. Indeed, the number of Twitter datasets in a language other than English is very scarce [7].

The PoliticEs@IberLEF2022 [8] aims to classify the political ideology of a person from tweets written in Spanish.

This paper describes our participation at the PoliticES@IberLef 2022 shared task [9]. This work explores different machine learning algorithms to automatically detect the political ideology of Twitter users based on their tweets

This paper is organized as follows: Section 2 describes the state-of-art of this task. Section 3 describes the dataset and methods used in this work. Section 4 describes the evaluation of proposed methods and discusses the results. Finally, Section 5 describes the conclusions of the work.

2. Related work

The use of traditional machine learning methods for the automatic detection of political ideology has been used since its inception until today. Some common techniques used for political ideology detection are Logistic Regression, Support Vector Machine [10], Naive Bayes [11], K-Nearest Neighbors [12] and Random Forest [13].

As early as 2006, the study described in [14] the used SVM and Naive Bayes classifier to classify political speeches in the US Congress in 2005.

In [15], the authors analyzed political speeches held in the German Bundestag in order to classify politicians according to their political party (five class problem), their affiliation in the government (binary problem) and their political opinions (56 class problem). To carry out this work the authors used the Logistic Regression technique. The dataset has two parts: the first part is composed of all political speeches made in the German Bundestag during the 17th and 18th legislatures. The second part contains all manifesto texts of parties running for election in the German parliament during the same legislatures. In total 22,784 speeches from the 17th legislative period and 11,317 speeches from the 18th period. For the political party detection, the work achieves an average precision of 0.62 in parliament speeches and an average precision of 0.51 in party manifestos. For the binary problem, the model obtains an average precision of 0.85 in parliament speeches and an average precision of 0.65 in party manifestos. In the political views task the model achieves an average precision of 0.47.

In [16], the authors use a set of algorithms to classify political orientation on Twitter during the 2015 Spanish elections. In this work, they use several algorithms such as Naive Bayes, Support Vector Machine, k-Nearest Neighbors and Random Forest. The dataset they used contains 24,900 tweets: 14,297 tweets with the hashtag #24M and 10,603 tweets with the hashtag #Elecciones2015. The authors grouped the tweets in three classes: progressive ideological trend, conservative ideological trend and no political orientation. They trained five models: Naive Bayes (accuracy of 0.67), Random forest (accuracy of 0.77), k-Nearest Neighbors (Average accuracy of 0.71), Linear SVM (accuracy of 0.76), Logistic Regression (accuracy of 0.71).

Recently, several deep Learning techniques such as LSTM (Long Short Term Memory) [17] and transformers models have been used for the political ideology detection.

LSTM is a type of recurrent neural network that is able to "remember" and exploits the previous states of the network to predict the next state. The main characteristic of the LSTM networks is the that their cells contain three different gates: input gate, forget gate and output gate, which allows to regulate the information flow in the LSTM network. In [18], the authors use an LSTM network to classify tweets according to whether they are democratic or republican. The authors obtained an average accuracy of 0.87. The dataset used contains 1,417,723 of training samples and 354,431 of test samples.

The Transformer architecture was first described in [19]. This architecture is based on the use of multiple Attention mechanisms. An Attention mechanism allows us to identify the features more relevant parts of a text. In this work, the authors described that the use of multiple interconnected attention mechanisms can be used to successfully perform several Natural Language Processing (NLP) tasks such as text classificaton or named entity recognition. One of the most widely used architectures today that use transformers is BERT [20]. In [21], the authors use the BERT architecture to detect political ideology in news articles. The authors use a dataset of 34,737 manually labelled articles to train the network. With this network the authors obtained an accuracy of 72% and a F1 macro of 54.29%.

3. Approaches

3.1. Dataset

In this section, we describe the dataset [22] provided by the organizers of PoliticEs@IberLEF 2022 shared task. It contains 37,560 tweets in Spanish, belonging to a total of 313 users of different ideologies (left, moderate left, right or moderate right). In addition, it provides a set of extra labels such as gender or profession that can be interesting for the analysis of the ideological situation of the country. Moreover, these features can help to find certain relationships that at first would seem unconnected. We now describe in more detail the fields of each instance in the dataset:

- **User:** Predefined username that maintains the user's privacy. This will follow the structure of "@userX", where X corresponds to a unique identifier number of the user. There are a total of 313 users with 120 tweets each one.
- **Gender:** Determine the user's gender using the labels male or female.
- **Profession:** Determines the user's profession distinguishing between journalist and politician.
- **Ideology binary:** Determines the user's ideology using the left or right political labels.
- **Ideology multiclass:** Determines the user's ideology using the political labels left, moderate left, right or moderate right.
- **Tweet:** Messages written by the user on the social network Twitter. These tweets are tagged to generalize political parties, proper nouns (person), etc.

In Figure 1, we can see a predominance of men with 56.54% compared to women, with 43.45%. Regarding the profession, we can see that politicians (80.35%) are more frequent than journalists

(19.65%). This extreme unbalanced distribution can cause serious problems when training the model and lead to overfitting.

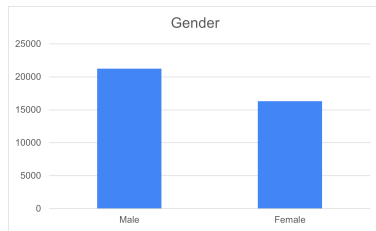


Figure 1: Gender analysis

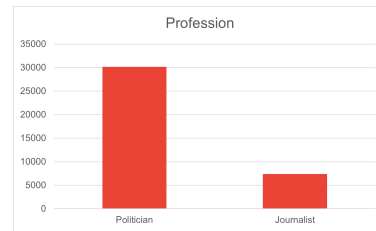


Figure 2: Profession analysis

Looking at the binary ideology (see Fig. 3), we can see a similar unbalance with respect to gender. It is not as severe as the profession of the users, so it should not cause so many problems when training a deep learning model.

Finally, in the analysis of multiclass ideology (see Fig. 4), it can be observed that users tend to have a moderate ideology, with the left and moderate right tags occupying the largest share of tweets. Furthermore, it can be observed that the right ideology has the least number of tweets, while the left ideology has a similar number of tweets than the moderate ideologies.

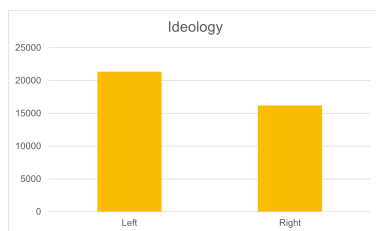


Figure 3: Ideology binary analysis

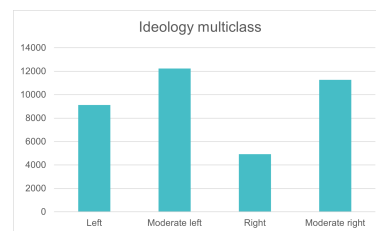


Figure 4: Ideology multiclass analysis

3.2. Methods

The following section will define all the trained models and the preprocessing performed on the corpus.

Preprocessing the text corpus is an essential task for text classification models to work properly. Performing data cleaning allows the creation of a more reliable corpus, since it ensures that the training text does not contain duplicated, irrelevant or incorrect information. This preprocessing is usually performed once the text has been tokenized using different techniques, of which the following have been used in all the trained models:

- **Normalization:** Normalization of the training text is helpful in removing noise from the corpus. In natural language processing, noise is understood as all extraneous characters, punctuation marks, capitalization, numbers, etc. In the models, all training tweets have been normalized by removing all characters that could affect the classification.

- **Stop words:** Stop words are those terms in a sentence that do not provide relevant information to the text as a whole. To avoid noise in the corpus they should be eliminated.
- **Stemming:** Finally, the preprocessing of the text should analyze all the words by eliminating the root of the words.

Once the text has been preprocessed, the architectures of the models to be trained must be chosen. In our project, the following architectures have been used:

- **K-Nearest Neighbors:** As mentioned above, one of the most widely used classifiers is K-Nearest Neighbors. For this, different number of neighbors has been used to observe the performance. The neighbors used have been 3, 5, 10, 20, 40.
- **Random Forest:** Another architecture has been Random Forest, which uses decision trees in sub samples of the training corpus. Different numbers of estimators (10,100,200) have been used, which determine the number of decision trees in the forest and various splitting criterion (gini and entropy).
- **Logistic Regression:** Logistic Regression has also been used with the penalties none, l1, l2 and elasticnet combined with the solvers newton-cg, lbfgs, liblinear, sag, saga.

4. Results

A total of 100 models have been trained with the different combinations of parameters mentioned in the previous section. They can be divided into groups of 25 models, each group intended to predict gender, profession, binary ideology and multiclass ideology.

Next, we will briefly discuss the results obtained, which can be consulted in the appendix.

Attending to the profession, we can see that there is a large disproportion between F1 metrics of politician and journalist. This is due to the imbalance of the data, with politicians occupying 0.80 versus 0.19 for journalists. An increasing improvement can be observed with the increase in the number of neighbors in K-Nearest Neighbors, however the results for journalists are still very low. The most compensated models are offered by Logistic Regression, more in particular, Logistic Regression without penalty and with the saga solver, which has an F1 of 0.88 for politicians and 0.47 for journalists.

In the analysis of gender classification, the results are much more balanced than in the profession models, although there are still better metrics for the male gender prediction. In these models it can be observed more clearly that the higher the number of neighbors in the K-Nearest Neighbors, the better the results. However, the number of estimators does not seem to be relevant for Random Forest. Finally, even though the metrics between the different architectures differ much, Logistic Regression without penalty and with the saga solver is the one that offers the best results.

The binary ideology classification models perform similarly to the gender classifiers, with an F1 of 0.73 for the left, and 0.63 for the right, there is a small imbalance in the data.

Finally, the multiclass ideology ranking obtains the worse results. Moderate ideologies have better metrics in line with their higher number of training instances, while the more extreme ones offer metrics around an F1 of 0.50.

5. Conclusion

This work has explored several algorithms to study a multi-classification problem and a binary classification problem in the field of political ideology detection using Spanish tweets. A binary classification study for gender and profession detection has also been carried out using the same dataset.

Several algorithms have been compared: Decision Trees, K-nearest Neighbors, Random Forest and Logistic Regression. The best model was Logistic Regression with the saga solver for all tasks, obtaining an average F1-score of 0.67 for profession classification, an average F1-score of 0.59 for gender classification, an average F1-score of 0.71 for binary classification of political ideology and an average F1-score of 0.53 for multi-classification of political ideology. We ranked 18th in the shared task.

We plan to experiment with different variants of transformers language models such as Maria [23] or RigoBERTa [24].

References

- [1] J. T. Jost, C. M. Federico, J. L. Napier, Political ideology: Its structure, functions, and elective affinities, *Annual review of psychology* 60 (2009) 307–337.
- [2] B. Verhulst, L. J. Eaves, P. K. Hatemi, Correlation not causation: The relationship between personality traits and political ideologies, *American journal of political science* 56 (2012) 34–51.
- [3] B. Verhulst, P. K. Hatemi, N. G. Martin, The nature of the relationship between personality traits and political attitudes, *Personality and Individual Differences* 49 (2010) 306–316.
- [4] B. R. Schlenker, J. R. Chambers, B. M. Le, Conservatives are happier than liberals, but why? Political ideology, personality, and life satisfaction, *Journal of Research in Personality* 46 (2012) 127–146. URL: <https://www.sciencedirect.com/science/article/pii/S009265661100170X>. doi:<https://doi.org/10.1016/j.jrp.2011.12.009>.
- [5] B. Baumgaertner, J. E. Carlisle, F. Justwan, The influence of political ideology and trust on willingness to vaccinate, *PLOS ONE* 13 (2018) e0191728. URL: <https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0191728>. doi:10.1371/journal.pone.0191728, publisher: Public Library of Science.
- [6] S. Du, S. Gregory, The Echo Chamber Effect in Twitter: does community polarization increase?, in: H. Cherifi, S. Gaito, W. Quattrociocchi, A. Sala (Eds.), *Complex Networks & Their Applications V, Studies in Computational Intelligence*, Springer International Publishing, Cham, 2017, pp. 373–378. doi:10.1007/978-3-319-50901-3_30.
- [7] T. M. Doan, J. A. Gulla, A Survey on Political Viewpoints Identification, *Online Social Networks and Media* 30 (2022) 100208. URL: <https://www.sciencedirect.com/science/article/pii/S246869642200012X>. doi:10.1016/j.osnem.2022.100208.
- [8] IberLEF 2022, 2022. URL: <https://sites.google.com/view/iberlef2022>.
- [9] J. A. García-Díaz, S. M. Jiménez-Zafra, M. T. Martín-Valdivia, F. García-Sánchez, L. A. Ureña-López, R. Valencia-García, Overview of PoliticES 2022: Spanish Author Profiling for Political Ideology, *Procesamiento del Lenguaje Natural* 69 (2022).

- [10] C. Cortes, V. Vapnik, Support-vector networks, *Machine Learning* 20 (1995) 273–297. URL: <http://link.springer.com/10.1007/BF00994018>. doi:10.1007/BF00994018.
- [11] G. I. Webb, Naïve Bayes, in: C. Sammut, G. I. Webb (Eds.), *Encyclopedia of Machine Learning*, Springer US, Boston, MA, 2010, pp. 713–714. URL: https://doi.org/10.1007/978-0-387-30164-8_576. doi:10.1007/978-0-387-30164-8_576.
- [12] B. W. Silverman, M. C. Jones, E. fix and j.l. hedges (1951): An important contribution to nonparametric discriminant analysis and density estimation: Commentary on fix and hedges (1951), *International Statistical Review / Revue Internationale de Statistique* 57 (1989) 233–238. URL: <http://www.jstor.org/stable/1403796>.
- [13] L. Breiman, Random Forests, *Machine Learning* 45 (2001) 5–32. URL: <https://doi.org/10.1023/A:1010933404324>. doi:10.1023/A:1010933404324.
- [14] B. Yu, S. Kaufmann, D. Diermeier, Classifying Party Affiliation from Political Speech, *Journal of Information Technology & Politics* 5 (2008) 33–48. URL: <https://doi.org/10.1080/19331680802149608>. doi:10.1080/19331680802149608, publisher: Routledge_eprint: <https://doi.org/10.1080/19331680802149608>.
- [15] F. Biessmann, Automating Political Bias Prediction, Technical Report arXiv:1608.02195, arXiv, 2016. URL: <http://arxiv.org/abs/1608.02195>, arXiv:1608.02195 [cs] type: article.
- [16] R. C. Prati, E. Said-Hung, Predicting the ideological orientation during the Spanish 24M elections in Twitter using machine learning, *AI & SOCIETY* 34 (2019) 589–598. URL: <http://link.springer.com/10.1007/s00146-017-0761-0>. doi:10.1007/s00146-017-0761-0.
- [17] S. Hochreiter, J. Schmidhuber, Long short-term memory, *Neural Computation* 9 (1997) 1735–1780. doi:10.1162/neco.1997.9.8.1735.
- [18] A. Rao, N. Spasojevic, Actionable and political text classification using word embeddings and lstm, 2016. URL: <https://arxiv.org/abs/1607.02501>. doi:10.48550/ARXIV.1607.02501.
- [19] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, I. Polosukhin, Attention Is All You Need, arXiv:1706.03762 [cs] (2017). URL: <http://arxiv.org/abs/1706.03762>, arXiv: 1706.03762.
- [20] M.-W. C. Jacob Devlin, Open Sourcing BERT: State-of-the-Art Pre-training for Natural Language Processing, 2018. URL: <http://ai.googleblog.com/2018/11/open-sourcing-bert-state-of-art-pre.html>.
- [21] R. Baly, G. D. S. Martino, J. Glass, P. Nakov, We can detect your bias: Predicting the political ideology of news articles (2020). URL: <https://arxiv.org/abs/2010.05338>. doi:10.48550/ARXIV.2010.05338.
- [22] J. A. García-Díaz, R. Colomo-Palacios, R. Valencia-García, Psychographic traits identification based on political ideology: An author analysis study on Spanish politicians’ tweets posted in 2020, *Future Generation Computer Systems* 130 (2022) 59–74.
- [23] A. Gutiérrez Fandiño, J. Armengol Estapé, M. Pàmies, J. Llop Palao, J. Silveira Ocampo, C. Pio Carrino, C. Armentano Oller, C. Rodríguez Penagos, A. Gonzalez Agirre, M. Villegas, Maria: Spanish language models, *Procesamiento del Lenguaje Natural* 68 (2022).
- [24] A. Vaca Serrano, G. Garcia Subies, H. Montoro Zamorano, N. Aldama Garcia, D. Samy, D. Betancur Sanchez, A. Moreno Sandoval, M. Guerrero Nieto, A. Barbero Jimenez, Rigoberta: A state-of-the-art language model for spanish, arXiv e-prints (2022) arXiv–2205.

6. Appendix

Profession	F1 politician	F1 journalist
Decision Tree: [gini, best]	0.84	0.34
Decision Tree: [entropy, best]	0.84	0.35
Decision Tree: [entropy, random]	0.85	0.36
K-Nearest neighbors: 3	0.08	0.34
K-Nearest neighbors: 5	0.88	0.03
K-Nearest neighbors: 10	0.89	0.28
K-Nearest neighbors: 20	0.89	0.17
K-Nearest neighbors: 40	0.88	0.07
Random Forest: [10, gini]	0.88	0.30
Random Forest: [100, gini]	0.89	0.19
Random Forest: [200, gini]	0.89	0.19
Random Forest: [10, entropy]	0.89	0.33
Random Forest: [100, entropy]	0.89	0.20
Random Forest: [200, entropy]	0.89	0.19
Logistic Regression: [l1, liblinear]	0.89	0.32
Logistic Regression: [l1, saga]	0.89	0.32
Logistic Regression: [l2, lbfgs]	0.89	0.28
Logistic Regression: [l2, liblinear]	0.89	0.28
Logistic Regression: [l2, newton-cg]	0.89	0.28
Logistic Regression: [l2, sag]	0.89	0.28
Logistic Regression: [l2, saga]	0.89	0.28
Logistic Regression: [none, lbfgs]	0.87	0.48
Logistic Regression: [none, newton-cg]	0.86	0.46
Logistic Regression: [none, sag]	0.87	0.47
Logistic Regression: [none, saga]	0.88	0.47

Gender	F1 male	F1 female
Decision Tree: [gini, best]	0.61	0.50
Decision Tree: [gini, random]	0.60	0.50
Decision Tree: [entropy, best]	0.61	0.50
Decision Tree: [entropy, random]	0.60	0.49
K-Nearest neighbors: 3	0.06	0.59
K-Nearest neighbors: 5	0.02	0.59
K-Nearest neighbors: 10	0.01	0.59
K-Nearest neighbors: 20	0.38	0.57
K-Nearest neighbors: 40	0.57	0.55
Random Forest: [10, gini]	0.62	0.53
Random Forest: [100, gini]	0.69	0.52
Random Forest: [200, gini]	0.69	0.52
Random Forest: [10, entropy]	0.62	0.53
Random Forest: [100, entropy]	0.69	0.52
Random Forest: [200, entropy]	0.69	0.52
Logistic Regression: [l1, liblinear]	0.70	0.49
Logistic Regression: [l1, saga]	0.70	0.49
Logistic Regression: [l2, lbfgs]	0.70	0.50
Logistic Regression: [l2, liblinear]	0.70	0.50
Logistic Regression: [l2, newton-cg]	0.70	0.50
Logistic Regression: [l2, sag]	0.70	0.50
Logistic Regression: [l2, saga]	0.70	0.50
Logistic Regression: [none, lbfgs]	0.65	0.55
Logistic Regression: [none, newton-cg]	0.63	0.52
Logistic Regression: [none, sag]	0.64	0.54
Logistic Regression: [none, saga]	0.65	0.54

Ideology binary	F1 left	F1 right
Decision Tree: [gini, best]	0.67	0.58
Decision Tree: [gini, random]	0.68	0.58
Decision Tree: [entropy, best]	0.67	0.58
Decision Tree: [entropy, random]	0.68	0.59
K-Nearest neighbors: 3	0.08	0.61
K-Nearest neighbors: 5	0.05	0.61
K-Nearest neighbors: 10	0.65	0.63
K-Nearest neighbors: 20	0.73	0.62
K-Nearest neighbors: 40	0.76	0.59
Random Forest: [10, gini]	0.75	0.57
Random Forest: [100, gini]	0.77	0.62
Random Forest: [200, gini]	0.78	0.63
Random Forest: [10, entropy]	0.74	0.56
Random Forest: [100, entropy]	0.78	0.63
Random Forest: [200, entropy]	0.78	0.63
Logistic Regresion: [l1, liblinear]	0.77	0.65
Logistic Regresion: [l1, saga]	0.77	0.65
Logistic Regresion: [l2, lbfgs]	0.78	0.66
Logistic Regresion: [l2, liblinear]	0.78	0.66
Logistic Regresion: [l2, newton-cg]	0.78	0.66
Logistic Regresion: [l2, sag]	0.78	0.66
Logistic Regresion: [l2, saga]	0.78	0.66
Logistic Regresion: [none, lbfgs]	0.75	0.67
Logistic Regresion: [none, newton-cg]	0.72	0.64
Logistic Regresion: [none, sag]	0.75	0.67
Logistic Regresion: [none, saga]	0.75	0.67

Ideology multiclass	F1 left	F1 moderate left	F1 moderate right	F1 right
K-Nearest neighbors: 3	0.04	0.08	0.47	0.01
K-Nearest neighbors: 5	0.01	0.05	0.47	0.00
K-Nearest neighbors: 10	0.07	0.46	0.49	0.00
K-Nearest neighbors: 20	0.28	0.53	0.53	0.10
K-Nearest neighbors: 40	0.31	0.56	0.53	0.11
Random Forest: [10, gini]	0.45	0.52	0.48	0.20
Random Forest: [100, gini]	0.48	0.57	0.56	0.21
Random Forest: [200, gini]	0.49	0.58	0.56	0.22
Random Forest: [10, entropy]	0.44	0.52	0.48	0.22
Random Forest: [100, entropy]	0.49	0.57	0.55	0.22
Random Forest: [200, entropy]	0.47	0.58	0.56	0.22
Logistic Regresion: [l1, liblinear]	0.54	0.58	0.59	0.35
Logistic Regresion: [l1, saga]	0.54	0.58	0.58	0.36
Logistic Regresion: [l2, lbfgs]	0.57	0.61	0.61	0.38
Logistic Regresion: [l2, liblinear]	0.57	0.60	0.60	0.34
Logistic Regresion: [l2, newton-cg]	0.57	0.61	0.61	0.38
Logistic Regresion: [l2, sag]	0.57	0.61	0.61	0.38
Logistic Regresion: [l2, saga]	0.57	0.61	0.61	0.38
Logistic Regresion: [none, lbfgs]	0.55	0.59	0.58	0.43
Logistic Regresion: [none, newton-cg]	0.51	0.55	0.55	0.38
Logistic Regresion: [none, sag]	0.54	0.58	0.58	0.43
Logistic Regresion: [none, saga]	0.54	0.58	0.58	0.43