Social Media Models

Context

- **This preliminary analysis** was made to show the value of building effective models that allow us to **increase engagement,** using data from google analytics and BBVA’s blog information.
- In this report we will explain the steps that we followed, and we will suggest further analysis**.**

Objectives

- Present the descriptive analysis of the data.
- Describe the models used to solve the problem.
- Suggest next steps.

Phases of the analysis

The analysis and modelization of the data have several phases that it is necessary to consider.

These steps can vary depending of the problem

Conceptualisation phase

- Workshops with decision makers to determine priorities, share information and define **success metrics**.

- Review of academic sources, data available sources, propose ideas.

- **Definition of deliverables.**

Analysis and data preparation

- Selection of the databases to be used.

- Verification of the quality of the data.

- Aggregation of the data to the desired granularity.

- Analysis of the feasibility of the deliverables.

Modeling phase

Data modeling

- Definition of the methodology to be used.

- Modeling and visualization.

- Descriptive or/and predictive analysis according to the objectives of the project.

- Monitoring results

Diagnosis of the results

- Refinement of the model and definition of period of the monitoring and evaluation process.

- Validation of the model and, application of the metrics defined in the conceptualization phase.

This is planned depending of the final product

Prototype development

- Definition of the prototype's functionality.

- Creation of dashboards, heatmaps and visualization of the results.

Knowledge transfer

- Delivery of the results, reports, apps or other output previously defined in the conceptualization phase.

- Transfer of knowledge and communications for its use.

Productization plan

Tools to use in the phase of exploration

- We use **Jupyter Notebooks because are easy to read for non-programmer's thanks to the possibility of include code, images and text.** It can be transformed in a script for its deployment in any cloud platforms as AWS, Google Cloud or Azure or even in your personal laptop or server. **It is used by companies as Netflix for their data analysis and experiments.**
- **We use Python and R for data treatment, getting the best of two great languages.** Jupyter Notebooks allows to use any language that we want as: scala, spark, javascript, between others.

Tools to use in the phase of production

- Now, that we have our first prototype, we can suggest paths to follow, in order to introduce new tools that help to get the best of our data. **The tools to choose would depend on the budget of our project and the use that will be given to the tool.**
- First, it is necessary to create a database to save the data, as the data come from external source a relational database will be enough. Due to the possibility that the data will increase with time, we can consider include spark in our implementations to accelerate the calculations, and we can use some cloud computing as AWS (EC2 AMI/AWS lambda) to execute our final product.
- It is necessary to define when and where our final tool would be deployed. This how we indicate before depend of the budget, frequency of use and the utility of the tool.

Data preparation: scope & cleansing process

Target

- Develop descriptive and predictive models, that allow us to **determine which are the mechanisms that help to increase the engagement with the blog feature**. Then, suggest further analysis.

Datasets and Variables to use

- We have two datasets: **google analytics data and blog scraping data for each post published in the BBVA data analytics apps.** These two datasets were cleaned and merged using the url title of the posts, keeping only the useful data (i.e. ignoring page searches u other web pages not associated with the post, as about page, etc.)

Cleansing process

- The acquired data has been prepared in the following way:

1. **Clean text data:** strings of data were treated in order to eliminate numbers, stops words, special characters, between others in order to treat and made predictive models with this features.
2. **Transform categorical data** (transform this variables to numeric/one hot encoding)**, treat numerical data** (eliminate outliers) and **complete missing data**.
3. **Cleaning inconsistencies:** Some variables were not considered in the model due to inconsistencies or due to poor predictive power or high correlations values (multicollinearity/noise).
4. **Creating new variables:** interactions between couple of variables can help to find more complex hidden patterns in the data that may improve the results also feature engineering (creation of new variables based on raw data) may help to improve results.

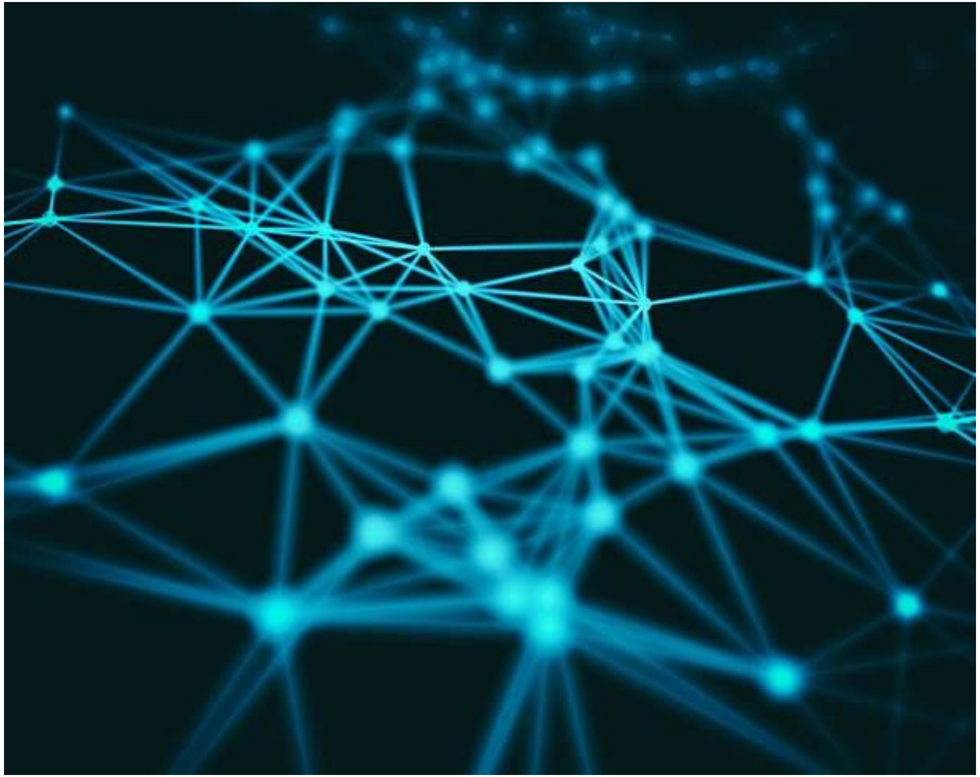
Analysis to implement

Here, a preliminary analysis was made to show the value of building effective models that allow us to **increase the engagement in the BBVA's data science blog.**

- **Identification of topics:** using text analytics methodologies we will try to identify the top spanish topics in the dataset.

- **Recommendation engine**: basic in methodologies like topic modelling and measures of similarity we will present a first prototype of recommendation engine along with improvements in the case of include new data from social networks.

- **Predict average time of each article:** use predictive modelling to determine which is the expected spending time reading a post depending of the topic, device, laguange, length of the text, between others.

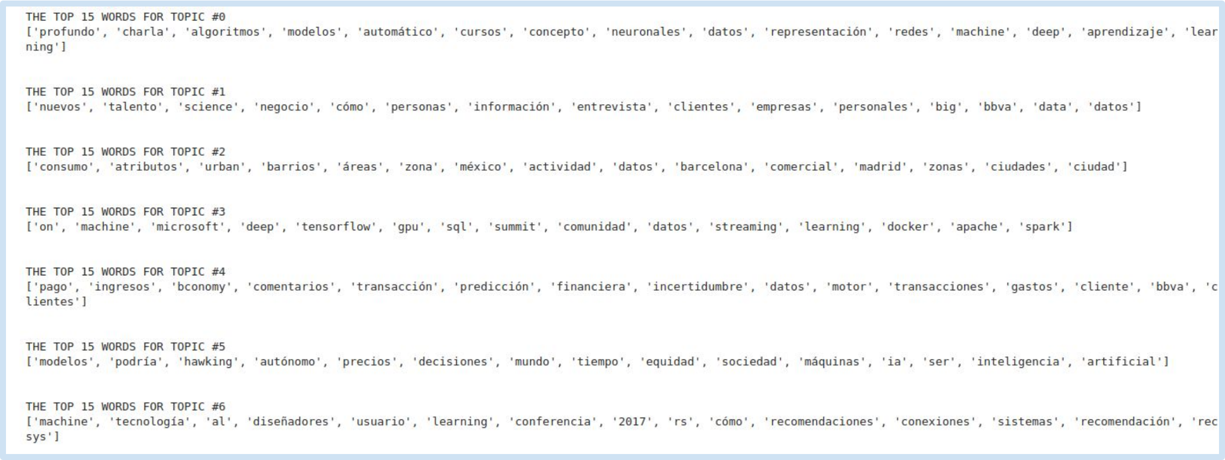


Identification of topics

Topic Modelling

Develop a topic discovery model from post's content allows us to identify semi-automatic topics that are being discussed in a set of documents.

- This methodology allows to classify, in an unsupervised way, text in different categories using methodologies like Latent Dirichlet Allocation (LDA) or **Non-Negative Matrix Factorization (NMF)**.
- To implement the methodology it is necessary to **define the number of topics**. In this case we play with several numbers, but it is possible to have a more automatic solution using a **coherence metric** that measures the probability that a topic have of being a good topic.



```
THE TOP 15 WORDS FOR TOPIC #0
['profundo', 'charla', 'algoritmos', 'modelos', 'automático', 'cursos', 'concepto', 'neuronales', 'datos', 'representación', 'redes', 'machine', 'deep', 'aprendizaje', 'lear
ning']


THE TOP 15 WORDS FOR TOPIC #1
['nuevos', 'talento', 'science', 'negocio', 'cómo', 'personas', 'información', 'entrevista', 'clientes', 'empresas', 'personales', 'big', 'bbva', 'data', 'datos']


THE TOP 15 WORDS FOR TOPIC #2
['consumo', 'atributos', 'urban', 'barrios', 'áreas', 'zona', 'méxico', 'actividad', 'datos', 'barcelona', 'comercial', 'madrid', 'zonas', 'ciudades', 'ciudad']


THE TOP 15 WORDS FOR TOPIC #3
['on', 'machine', 'microsoft', 'deep', 'tensorflow', 'gpu', 'sql', 'summit', 'comunidad', 'datos', 'streaming', 'learning', 'docker', 'apache', 'spark']


THE TOP 15 WORDS FOR TOPIC #4
['pago', 'ingresos', 'bconomy', 'comentarios', 'transacción', 'predicción', 'financiera', 'incertidumbre', 'datos', 'motor', 'transacciones', 'gastos', 'cliente', 'bbva', 'c
lientes']


THE TOP 15 WORDS FOR TOPIC #5
['modelos', 'podría', 'hawking', 'autónomo', 'precios', 'decisiones', 'mundo', 'tiempo', 'equidad', 'sociedad', 'máquinas', 'ia', 'ser', 'inteligencia', 'artificial']


THE TOP 15 WORDS FOR TOPIC #6
['machine', 'tecnología', 'al', 'diseñadores', 'usuario', 'learning', 'conferencia', '2017', 'rs', 'cómo', 'recomendaciones', 'conexiones', 'sistemas', 'recomendación', 'rec
sys']
```

Topics in spanish

TOPIC 0: learning/courses/talks/conferences data science/machine learning
TOPIC 1: digital change in companies/companies and machine learning
TOPIC 2: social research/smart cities
TOPIC 3: tools for machine learning/tech
TOPIC 4: applications of ML in banking
TOPIC 5: future and past of data science/evolution.
TOPIC 6: research, new models

Using this info we could **recommend new topics as: project management in ds, small challenges for data scientists, etc.** This methodology was applied over spanish content, and improvement would be consider other languages.

Text treatments

To make analysis with text data were necessary to make some transformations.

→ Cleaning text

- Remove irrelevant characters (urls, special characters, etc). Convert characters to lowercase, between others.

→ Detect Language

- **No all the contents are in the same language**, so in this analysis we create a feature with the language, in order to analyse those contents that are in the same language. In this case, **we consider only the posts in spanish.**

→ Rearrange the data

- **Stopwords:** eliminate commonly used word (such as “the”, “a”, “an”, “in”) that are used to connect words in the sentence.
- **Lemmatization:** reduce words as ‘am’, ‘are’, and is to a common form such as ‘be’.
- **Part of speech (POS):** take part of the texts, for example only nouns and verbs (possible improvement)

→ Find a good data representation

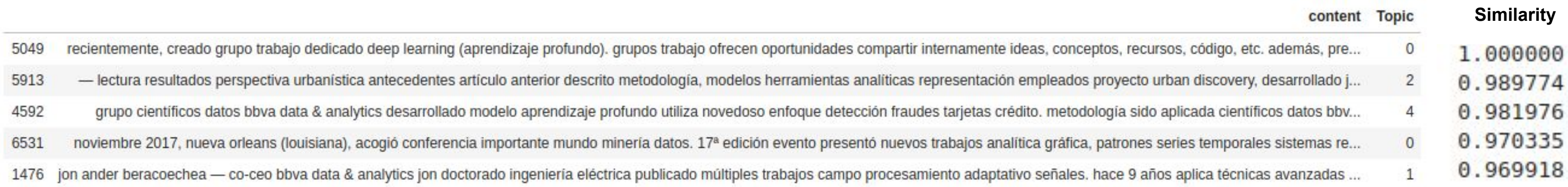
- **One hot encoding (bag of words):** this associate an unique index to each word (or set of words - n-grams) in the sentence.
- **TF-IDF (Term frequency, Inverse Document Frequency):** this aggregates weights to words by how rare they are in the dataset, discounting words that are to frequent (used in this case).
- Other representations like Word2Vec or any other that capture semantic meaning (check **lda2vec**)

Recommendation engine

Recommendation engine

Using methodologies like lda or **Non-Negative Matrix Factorization (NMF)** along with **cosine similarity** we can recommend new content based in the current document that the user is looking.

- This methodology may be **combined with other features in order to improve the recommendation.** Features like length, number of views, time of each article, along with other social media information may complement the results.
- An alternative solution is use **the links inside the content (similar articles will have similar links),** this will lead in a **Supervised** Machine Learning Task: Map articles to Links (target whether or not a particular link was present in a book article)
- In this **first approximation** it is only used **the 'content of the post',** with the data treatment described before.
- For example:



	content	Topic	Similarity
5049	recientemente, creado grupo trabajo dedicado deep learning (aprendizaje profundo). grupos trabajo ofrecen oportunidades compartir internamente ideas, conceptos, recursos, código, etc. además, pre...	0	1.000000
5913	— lectura resultados perspectiva urbanística antecedentes artículo anterior descrito metodología, modelos herramientas analíticas representación empleados proyecto urban discovery, desarrollado j...	2	0.989774
4592	grupo científicos datos bbva data & analytics desarrollado modelo aprendizaje profundo utiliza novedoso enfoque detección fraudes tarjetas crédito. metodología sido aplicada científicos datos bbv...	4	0.981976
6531	noviembre 2017, nueva orleans (louisiana), acogió conferencia importante mundo minería datos. 17ª edición evento presentó nuevos trabajos analítica gráfica, patrones series temporales sistemas re...	0	0.970335
1476	jon ander beracoechea — co-ceo bbva data & analytics jon doctorado ingeniería eléctrica publicado múltiples trabajos campo procesamiento adaptativo señales. hace 9 años aplica técnicas avanzadas ...	1	0.969918

Validation: this model have to be validated. An option is to test this feature over a **random subset of users and measure the number of times that a user click a recommendation**. An **A/B testing** may be applied where the option A are random recommendations and B is the approach described here.

Cosine similarity uses angle to define similarity. Higher values means more similar. Maximum value is 1, when angle is 0˚(another valid option is the Euclidean distance)

* See appendix

Predict average time of each article

Applied methodology

The process to create a powerful and robust predictive model relies on **the following steps:**

Handling data

In order to predict **Avg. time on page** we consider regressions models with the log transformation of the target and the creation of features and interactions.

Algorithms comparison & selection

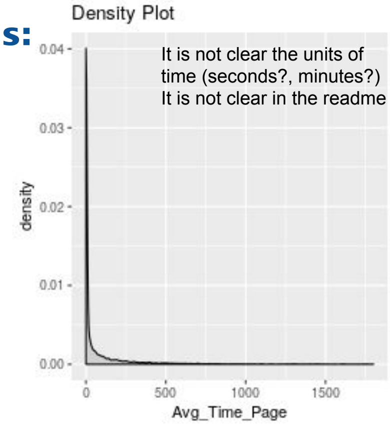
We used **ridge regression, Knn, Bayesian Regression, Decision Tree, SVM, xgb, deep learning (kers) and H2O automl** to predict the average time on page .

Parameters optimization

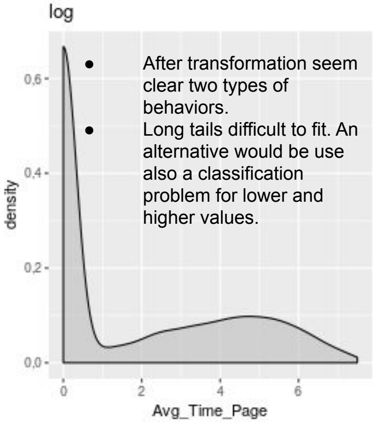
We normally spent some time optimizing the parameters of the algorithms in order to obtain the best results. We do this with random search but **Bayes Search**(GPyOpt package) would be better.

Robust validation

We **tested the model over** an independent test set (20% of the total data), not used for the training to see the quality of the results. Also, cross validation is used in some case.s.



It is not clear the units of time (seconds?, minutes?) It is not clear in the readme



- After transformation seem clear two types of behaviors.
- Long tails difficult to fit. An alternative would be use also a classification problem for lower and higher values.

Applied methodology

Here, we have to predict a variable that have a **long tail and a large frequency of users with close to zero values.** This can be solved through **two alternatives**, the use of one or another would depend of the **requirements of the final user**.

1. Develop **regression models** considering that we can have problems trying to fit the tail. Find features that try to explain the less frequent values or try to increase the sample of the less frequent values.

2. Divide the **continuous target in categories**, this in order to increase the sample of the less frequent values. This may improve the results and depending of the objective or final user this may be a more appropriate alternative.

We develop the alternative #1 with several predictive models.

Numerical treatments

There was necessary some transformations before to apply the methodologies.

→ Manage outliers	● **Understand and modify those values that can distort the final results.** Ex. Eliminate the values or apply log- transformation.
→ Create interactions	● **Create interactions between continuous variables.** We can consider interactions between binary and continuous variables. With deep learning this step is unnecessary.
→ Fill Missing variables	● In the case of missing variables there are **several methodologies that can be used to complete the data** (see Appendix)
→ Scale data	● For some algorithms would be necessary **to scale the data to avoid that large values distort the results.** This is true for methodologies as dimensionality reduction (eX. PCA), clustering, deep learning among others.
→ Avoid multicollinearity and redundant variables	● **Take only those variables that gives information to the target variable** and **eliminate those explanatory variables that share the same information** (high correlation between them). ● **Variance inflation factor** is other alternative to correlation. ● We can use **factor analysis** to identify those variables that shares a great amount of information ● Centering variables is an alternative to reduce collinearity.

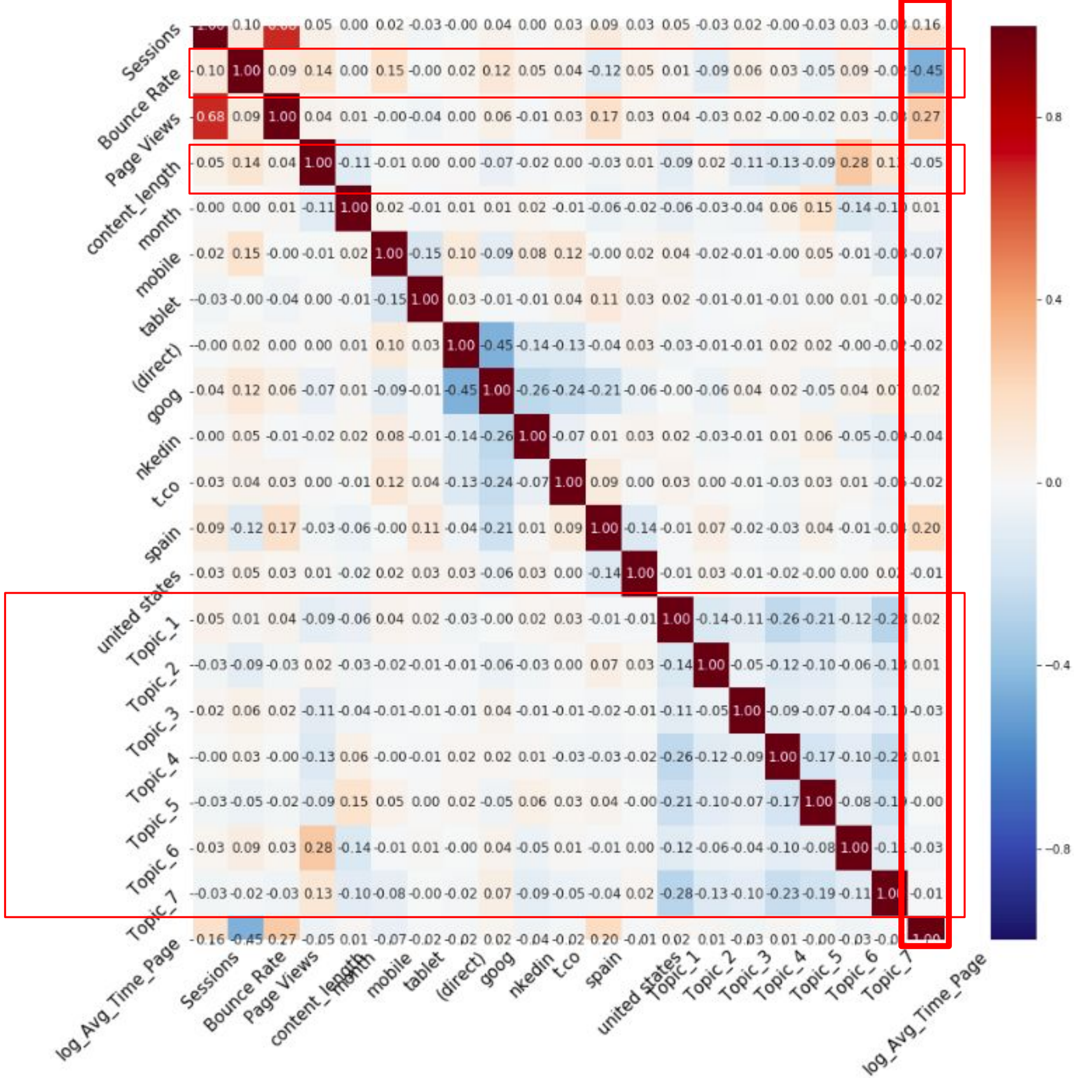
Correlations

- Here we present the correlation of the numerical variables along with some categorical (0/1 variables) (or chi-squared for categorical var./anova for cont.-cat.).

- Here, we use the log transformation of Avg_time_page, due to the large tail of this feature.

- We **eliminate variables that are highly related between themselves (to avoid multicollinearity)**. But we consider other criteria like mutual information to select between the best pair of variables.

- There are variables that are highly correlated between them that can affect the results if they are all included. These are:
  - 'Unique Page Views' vs. Page Views
  - 'Users' vs. Sessions and Page Views.
- We include here the different topics obtained using topic modelling with Non-Negative Matrix Factorization



Categorical treatments

The categorical variables have a different treatment in comparison with continuous variables*.

- ➔ One hot encoding

- We should **transform the categorical variables to 0-1 continuous variables to use them** in the modelling part.

- ➔ Combine levels/categories**
  - ➔ Using Business Logic
  - ➔ Using frequency on response rate

- In our case we could **combine categories considering the number of zeros (threshold > 90%).** Using this rate we combine rare categories together, etc.
- Business logic cannot be applied in this case (it is necessary more business logic).

- ➔ Relation between categorical variables
  - ➔ Chi-squared test

- The Chi-Square test of independence is a statistical test to determine if there is a significant relationship between 2 categorical variables (this can be used to determine which features has dependency between them - further improvement).

- ➔ Check similarity between variables
  - ➔ Cosine similarity/ Jaccard Similarities
  - ➔ Distance measures

- To avoid the problems associated with the redundant variables, we have to **determine if our categorical variables are similar or not to avoid any kind of noise in our model.**
- **In this case we use similarities to create the recommendation engine.**

* We also create interactions between categorical variables or between numerical and continuous variables.

** In the samples of training and testing there could be different categories, so it is necessary to find the way of group this variables.

Data Transformations

After some descriptive analysis we choose to make the following transformations.

To categorical and numerical data

- **Manage Outliers**
- **Fill Missing categorical variables***
- **Fill Missing numerical variables***
- **Include numerical interactions**
- **Apply one-hot encoding** to categorical variables
- **Select variables** based in correlations, similarities and mutual information

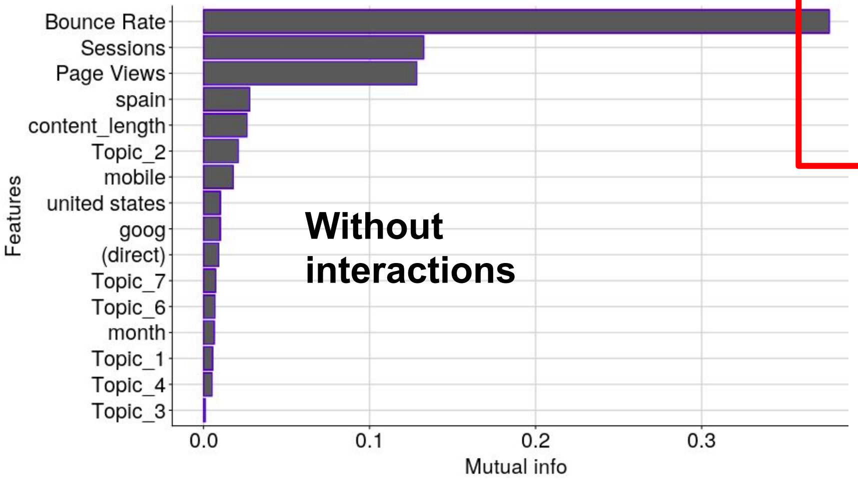
To text data

- **Merge data by url**
- **Eliminate special characters and numbers**
- **Transform to lowercase**
- **Determine language and analyse the majority**
- **Eliminate stopwords, apply lemmatization and Tfidf Vectorizer (term frequency)**
- **Appy Non-Negative Matrix Factorization (NMF) for TOPIC MODELLING and cosine similarities for recommendation engine**

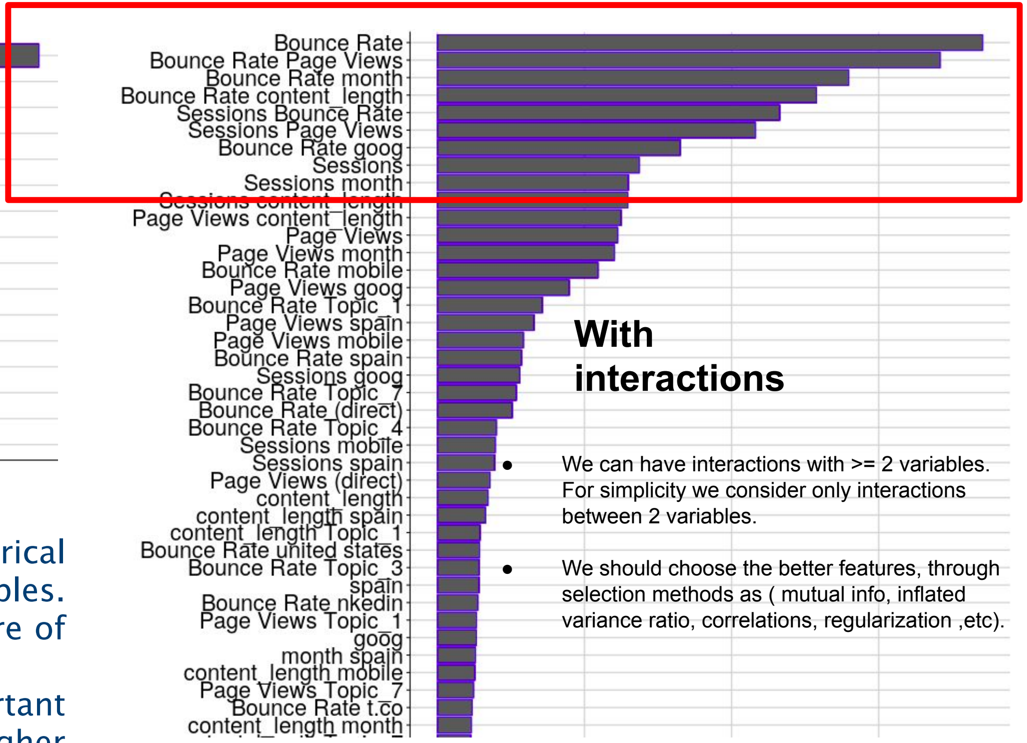
* See appendix

Dataset: Featuring Selection

Using the **Entropy measure*** we can observe, which are the **variables that determine the increase of average time per page**.



Without interactions



With interactions

- We can have interactions with >= 2 variables. For simplicity we consider only interactions between 2 variables.
- We should choose the better features, through selection methods as (mutual info, inflated variance ratio, correlations, regularization ,etc).

- Clearly, the interactions between the numerical variables contribute more than the single variables. We have to select those with higher MI, taking care of not to take pairs with high correlation.
- Also, it is clear that the Bounce rate is a important variable, it has higher mutual information and higher correlation with the target.

***Entropy measure** is used to calculate information gain that you obtain with each variable

Results

- There are several methodologies that we can use to solve the regression problem of estimate the ‘Average time per page’. For simplicity and time, we used: **ridge, knn, bayesian regression, decision tree regressor (DT), svm, xboosted trees (xgb), deep learning (keras)(DL) and H20 autoML** with and without interactions (only the best results are shown).

- We observe that the residuals seem to be correlated with the predicted values. This seems to indicate there are missing or omitted variables in the model (more research is needed)**.**

- Here, we present MAE errors that gives the same weight to small and large errors. RMSE it is also calculated.

- **Xgb, h2o, keras, and Knn regression** seem to show the better results. Although, more work is required (better feature selection, bayesian hiper parametrization, etc).

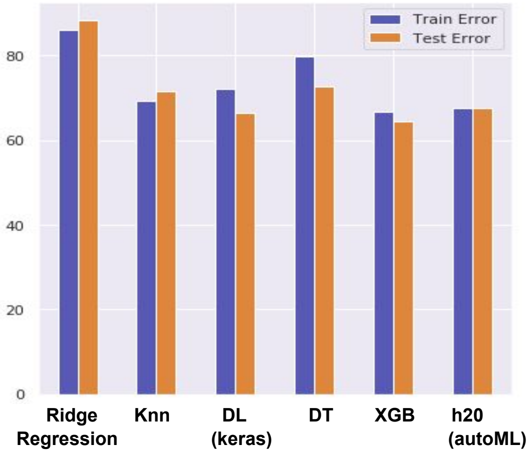
* See appendix and notebooks #4

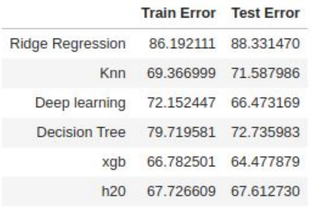
BASELINE

It is very common to create a **baseline model, that represents the most simple model that you can develop with the information you have.** It is a common sense model, that's no involve modelling, just simple calculations. In this case, we do not have much information so will be consider the Random Assignments as the baseline model.

MAE in log.

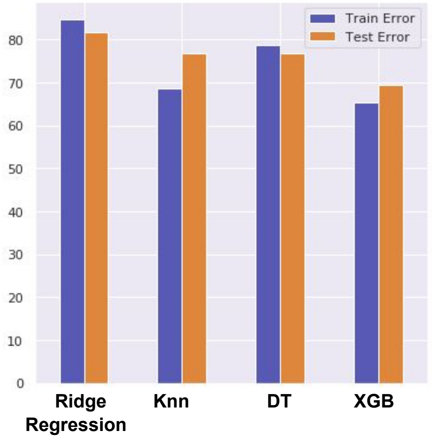
To make more analyses we have to transform to original units (exp(y) -10)



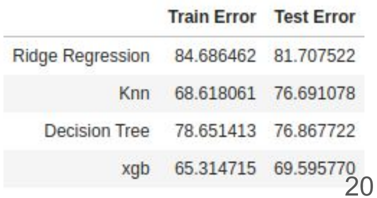


	Train Error	Test Error
Ridge Regression	86.192111	88.331470
Knn	69.366999	71.587986
Deep learning	72.152447	66.473169
Decision Tree	79.719581	72.735983
xgb	66.782501	64.477879
h20	67.726609	67.612730

Without interactions



With interactions



	Train Error	Test Error
Ridge Regression	84.686462	81.707522
Knn	68.618061	76.691078
Decision Tree	78.651413	76.867722
xgb	65.314715	69.595770

Results

BASELINE

It is very common to create a **baseline model, that represents the most simple model that you can develop with the information you have.**

- Random distribution (log_distribution)
- Mean (regression over the response and the intercept)

Solutions:

Observe the distribution of the predictions give us more insights about the results.

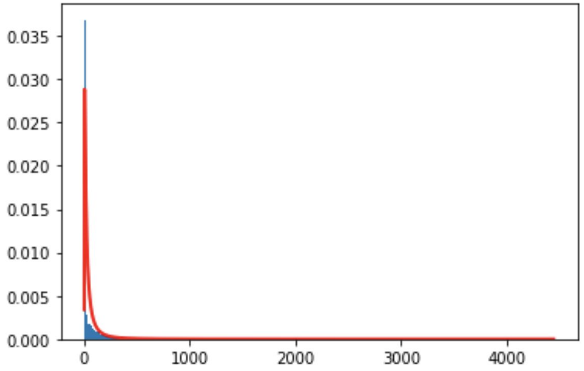
- **Alternatives to the tail:** quantile regression, cut the tail, use Tukey loss function, etc.

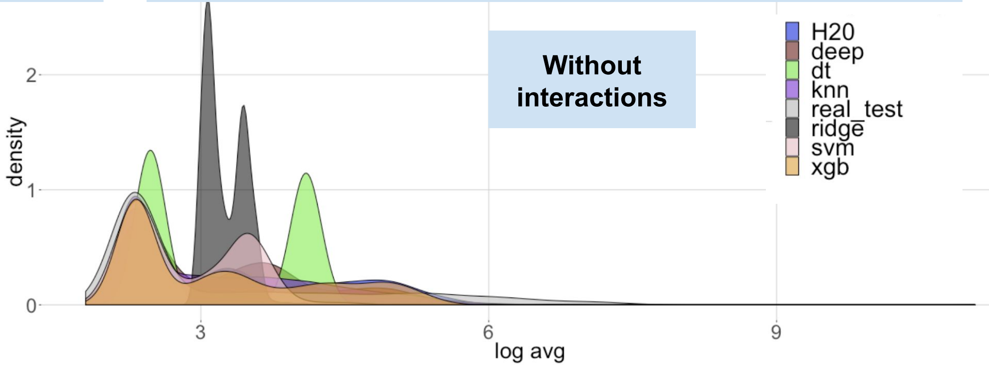
Mean benchmark:

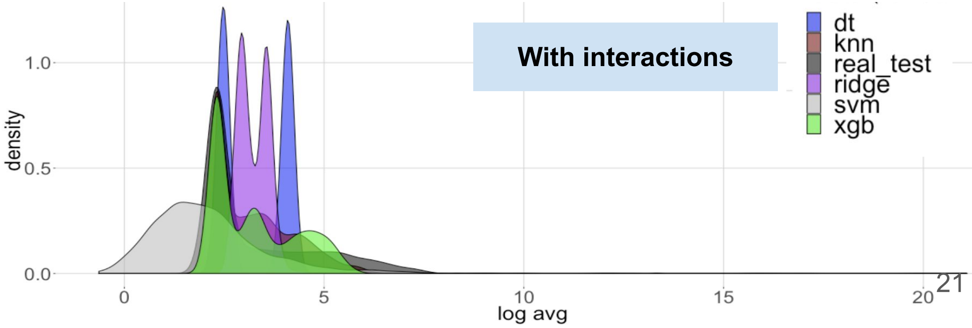
Train error: 85.74

Test error: 87.57

Random log-normal distribution:




Without interactions


With interactions

Conclusions & Recommendations*

*Predict average time of each article

- **Using external variables we can improve the results.** We did several treatments to the variables depending of its nature: numerical or categorical. **Although, more analysis are required to get better results.** A recommendation is to understand the source of the variables, in order to get better insights and get external variables, that can help to improve the solution.

- The **performance of the models could be improved** by better feature selection (reduce the number of variables, to those that are more relevant), because no all the interactions help the model. An alternative is to improve the deep learning model (keras) to let the model find the best interactions without our intervention. The problem is the lack of explanabilty (although alternatives like LIME can help with this), The use of alternative metrics like AIC/BIC can help in the feature and model selection.

- Additional analysis with more advance **hyperparameter tuning** (bayesian optimization) is required to improve analysis. Also, **ensemble methods** (see Appendix) can be useful as additional methodology.

- Improve feature engineering by aggregating more features related with content (ratios related with length and number of paragraphs, punctuations, use keywords, etc) and other variables.

- To understand better the categories inside of each variables (in order to combine the most similar categories together), would be interesting use the T-SNE method and clustering methods to see how the categories work inside of the generated groups.

Conclusions & Recommendations*

*Identification of Topics and Recommendations engine

- **A first simple approach was used** to determine the topics presented in the posts and to recommend other posts based in similarities.

- These outputs may be improved aggregating other features. Features like length, number of views, time of each article along with other social media information may complement the results. **With social media information we can create a recommendation engine based in collaborative filtering**.

- **These models have to be validated.** For the recommendation model an option is to test this feature over a random subset of users and measure the number of times that a user click a recommendation. An **A/B testing** may be applied where the option A are random recommendations and B is the approach described here.

- Use PyMC3 to improve bayesian regression results. This package allows to fix priors.

- More detailed answers to the questions proposed can be revised in the pdf document called: **Answer to questions bbva.pdf**.

Appendix

Topic Modelling

LDA: Latent Dirichlet Allocation

LDA is a probabilistic **three-level bayesian hierarchical model**, where each item of a collection is modelled as a finite mixture over an underlying set of topic probabilities.

- In summary, the steps are random assignment, and calculation of the conditional probabilities P(word | topics) and P(topics | documents)

$$p(\beta_{1:K}, \theta_{1:D}, z_{1:D}, w_{1:D}) = \prod_{i=1}^{K} p(\beta_i) \prod_{d=1}^{D} p(\theta_d) \prod_{n=1}^{N} p(z_{d,n}|\theta_d)\, p(w_{d,n}|\beta_{1:K}, z_{d,n})$$

NMF: Non-Negative Matrix Factorization

NMF is an **linear algebraic optimization algorithm** that consists in produce two matrices W and H.

- The columns of W can be interpreted as documents (bags of words). They represent topics! Sets of words found simultaneously in different documents.
- H tells us how to sum contributions from different topics to reconstruct the word mix of a given original document.

$$\underbrace{X(:,j)}_{j\text{th document}} \approx \sum_{k=1}^{r} \underbrace{W(:,k)}_{k\text{th topic}} \underbrace{H(k,j)}_{\substack{\text{importance of } k\text{th topic} \\ \text{in } j\text{th document}}}, \quad \text{with } W \geq 0 \text{ and } H \geq 0.$$

Therefore, given a set of documents, NMF identifies topics and simultaneously classifies the documents among these different topics.

Although the results depends heavily on the quality of text preprocessing and the strategy of finding the optimal number of topics, some publication indicates that NMF performs better on a smaller number of documents than LDA.

Social Media Metrics

Social media metrics are important because they prove you can measure how successful a post or campaign is, how well your social strategy is performing, and ultimately if you will have an impact with your content.

Some useful metrics will be:

- **Impressions** are how many times a post shows up in someone's timeline
- **Reach** is the potential unique viewers a post could have (usually your follower count plus accounts that shared the post's follower counts - we would need to add up every account that Retweeted it and their follower counts)**.** In **summary, reach** is the number of people who may have seen your content, while **impressions** are the total number of times your content was displayed to people.
- **Engagement:** how much audience is interacting with your account and how often. Engagement may be measure as the sum of retweets, replies, shares, likes and comments.
- **Post engagement rate:** The number of engagements divided by impressions or reach. A high rate means the people who see the post find it interesting.
- **Combinations of impressions, engagement and reach:** a post that has a high impressions count and a low engagement number (and therefore a low engagement rate) would mean that your post wasn't interesting enough for audiences to take action after seeing it in their feed. For a post with a high reach count and high engagement rate, it'll likely mean that the content went viral via Retweets and Shares.
- **Net followers:** #likes - #unlikes
- **Like rate:** divide likes by impressions. This can help you predict the success of a piece of content.
- **Account mentions:** organic mentions, like @mentions, indicate good brand awareness.
- **Conversions** is when someone purchases something from your site (or in this case when someone sent you a cv) .

Methodologies for recommendation systems

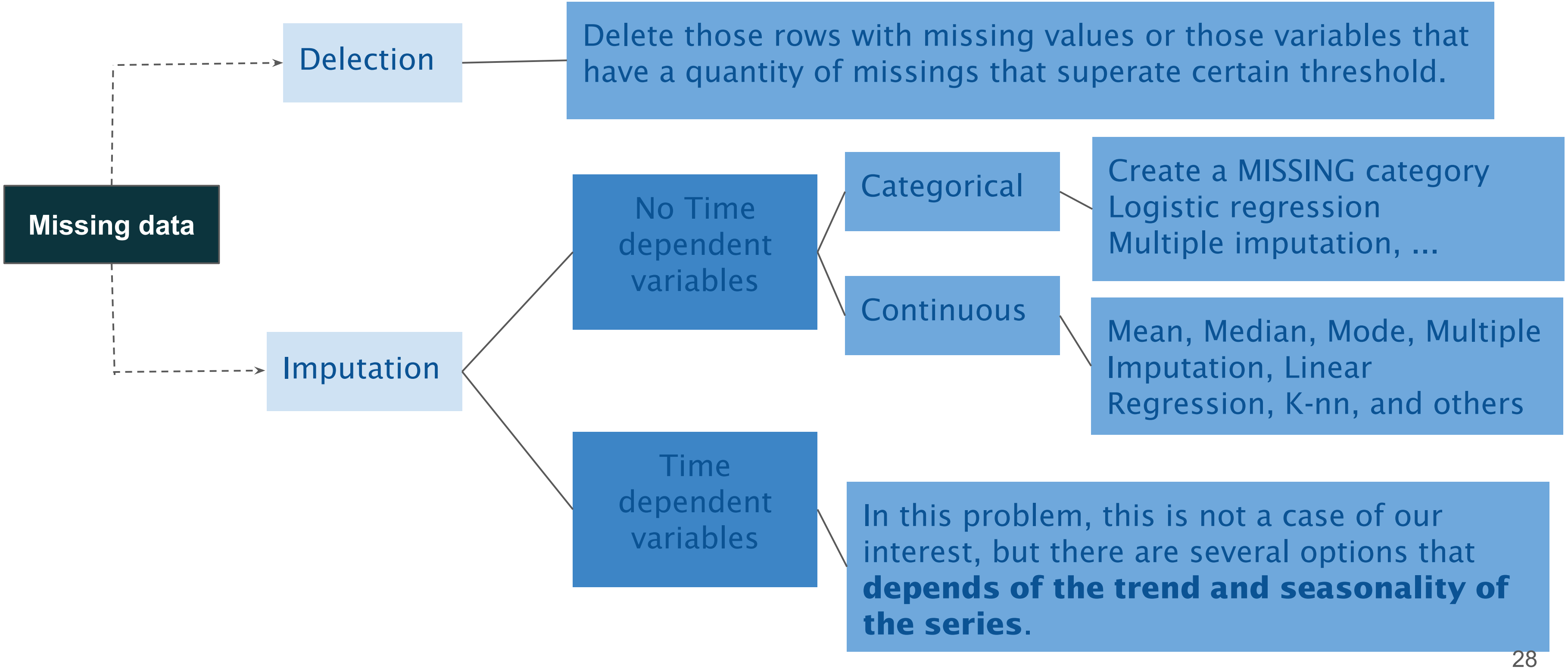
Methodologies to check

Several methodologies to have in consideration for recommendations:

- Clustering: K-nearest neighbour algorithms
- autoencoders
- Best rule recommendation
- Collaborative filtering
- Collaborative deep learning
- Embeddings (method to represent discrete variables as continuous vectors)
    - Neural network embeddings
    - Supervised task

Treatment of missing data

Depending of the data and the feasibility of the solution we have several alternatives:



Missing data

Delection

Delete those rows with missing values or those variables that have a quantity of missings that superate certain threshold.

Imputation

No Time dependent variables

Categorical

Create a MISSING category
Logistic regression
Multiple imputation, ...

Continuous

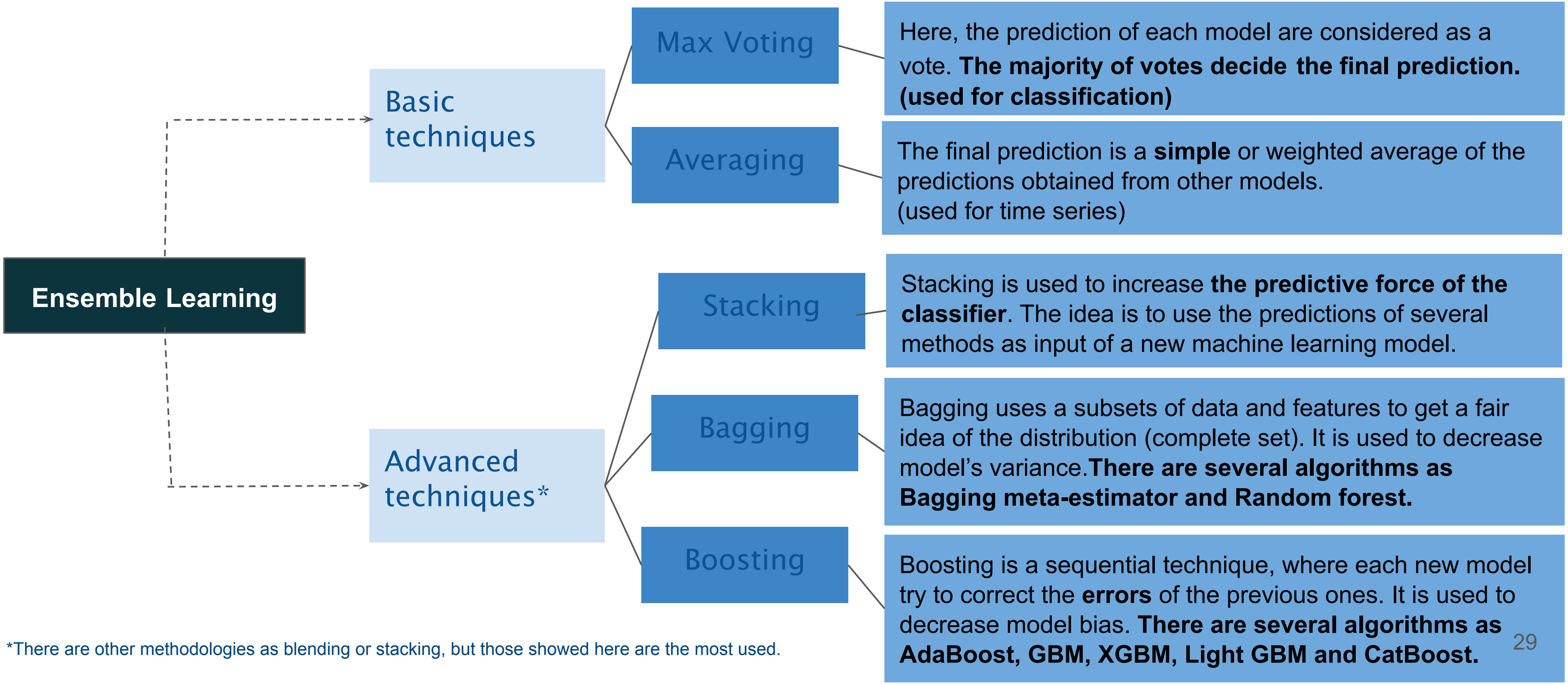
Mean, Median, Mode, Multiple Imputation, Linear Regression, K-nn, and others

Time dependent variables

In this problem, this is not a case of our interest, but there are several options that **depends of the trend and seasonality of the series**.

Ensemble Learning Techniques

A future modelling strategy would be use Ensemble techniques to improve our results.



Ensemble Learning

Basic techniques

Max Voting

Here, the prediction of each model are considered as a vote. **The majority of votes decide the final prediction. (used for classification)**

Averaging

The final prediction is a **simple** or weighted average of the predictions obtained from other models.
(used for time series)

Advanced techniques*

Stacking

Stacking is used to increase **the predictive force of the classifier**. The idea is to use the predictions of several methods as input of a new machine learning model.

Bagging

Bagging uses a subsets of data and features to get a fair idea of the distribution (complete set). It is used to decrease model's variance.**There are several algorithms as Bagging meta-estimator and Random forest.**

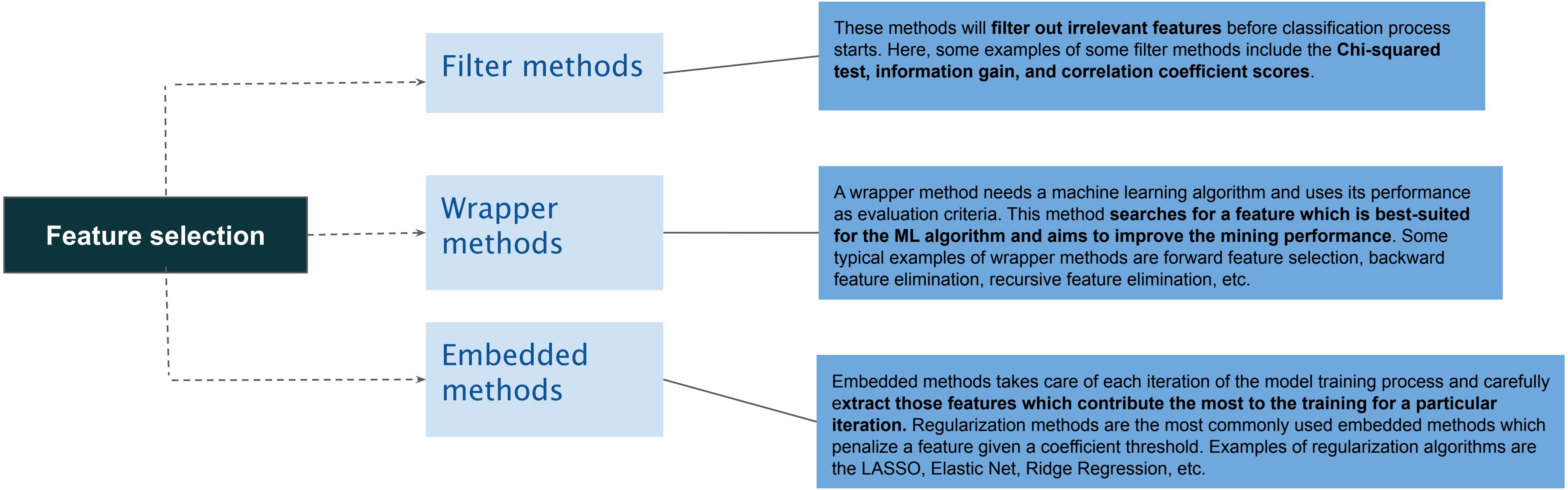
Boosting

Boosting is a sequential technique, where each new model try to correct the **errors** of the previous ones. It is used to decrease model bias. **There are several algorithms as AdaBoost, GBM, XGBM, Light GBM and CatBoost.**

*There are other methodologies as blending or stacking, but those showed here are the most used.

Featuring Selection

Unnecessary features act as a noise for which the machine learning model can perform terribly poorly.



Feature selection

Filter methods

These methods will **filter out irrelevant features** before classification process starts. Here, some examples of some filter methods include the **Chi-squared test, information gain, and correlation coefficient scores**.

Wrapper methods

A wrapper method needs a machine learning algorithm and uses its performance as evaluation criteria. This method **searches for a feature which is best-suited for the ML algorithm and aims to improve the mining performance**. Some typical examples of wrapper methods are forward feature selection, backward feature elimination, recursive feature elimination, etc.

Embedded methods

Embedded methods takes care of each iteration of the model training process and carefully **extract those features which contribute the most to the training for a particular iteration.** Regularization methods are the most commonly used embedded methods which penalize a feature given a coefficient threshold. Examples of regularization algorithms are the LASSO, Elastic Net, Ridge Regression, etc.

Hyperparameters Tuning

A good choice of hyperparameters can really make an algorithm shine. The process of finding the most optimal hyperparameters in machine learning is called hyperparameter optimization.

→ Grid Search	• Search for all possible combinations of parameters and measure its performance using cross-validation.
→ Random search	• Randomly samples the search space and evaluates sets from a specified probability distribution.
→ Bayesian Optimization	• Bayesian optimization typically works by assuming the unknown function was sampled from a Gaussian Process (GP) and maintains a posterior distribution for this function as observations are made.