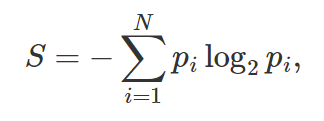
1. Link - <https://mlcourse.ai/book/topic04/topic04_intro.html> and https://habr.com/ru/company/ods/blog/323890/  
   Env:

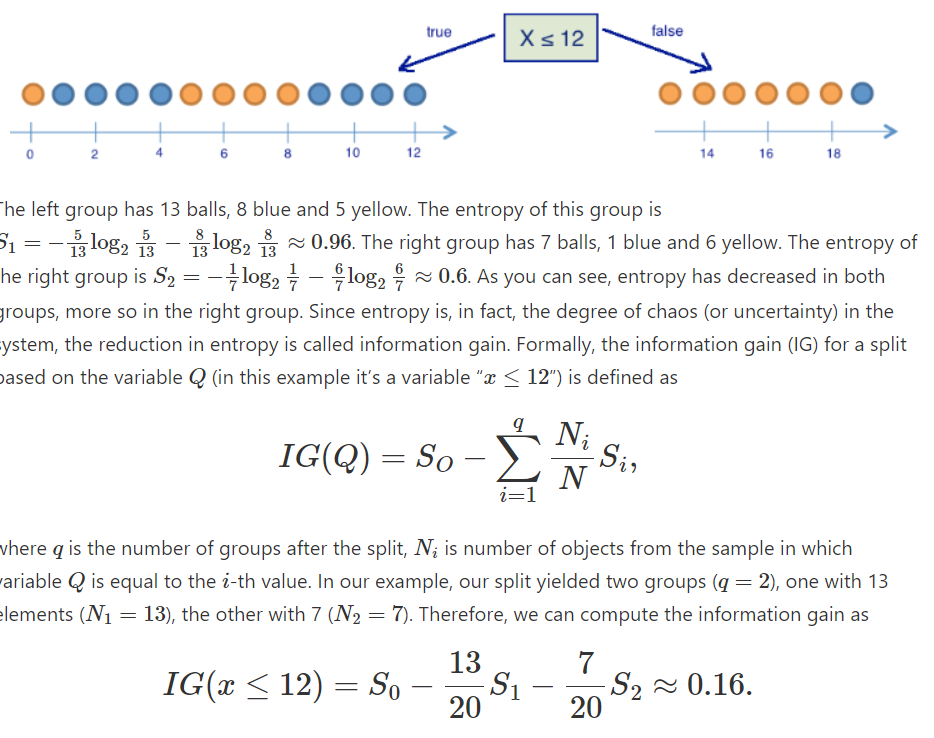
PS C:\Users\admin> cd 'E:\KULIKOV\ML\MLCourseAI\'   
PS E:\KULIKOV\ML\MLCourseAI> python -m venv .venv   
PS E:\KULIKOV\ML\MLCourseAI> .venv/Scripts/activate /deactivate

1. Try this - <https://www.coursera.org/learn/machine-learning>
2. Forum with problems - https://stats.stackexchange.com/
3. SetUp - Python extension for Visual Studio Code, Python Snippets, Python Docstring Generator, Python Test Explorer, Python Preview, Python Type Hint
4. Подсказки – Crtl+Space
5. DataFrames can be indexed by column name (label) or row name (index) or by the serial number of a row. The **loc** method is used for indexing by **name**, while **iloc()** is used for indexing by **number.**
6. Метрики - <https://habr.com/ru/company/ods/blog/328372/>
7. Correlation matrix: correlations among the numerical variables in our dataset. This information is important to know as there are Machine Learning algorithms (for example, linear and logistic regression) that do not handle highly correlated input variables well.
8. *Principal Component Analysis* (PCA) - limitation is that it is a *linear* algorithm that implies certain restrictions on the data.
9. There are also many non-linear methods, collectively called *Manifold Learning*. One of the best-known of them is *t-SNE (t-distributed Stochastic Neighbor Embedding.)*. dimensionality reduction methods convert the high-dimensional data set X = {x1, x2,..., xn} into two or three-dimensional data Y = {y1, y2,..., yn} that can be displayed in a scatterplot. Его основная идея проста: найдите проекцию для пространства объектов высокой размерности на плоскость (или 3D-гиперплоскость, но это почти всегда 2D) таким образом, чтобы те точки, которые были далеко друг от друга в исходном n-мерном пространстве, оказались далеко друг от друга на плоскости. Те, которые изначально были близки, останутся близкими друг к другу. Это поиск нового и менее размерного представления данных, которое сохраняет соседство примеров. If you have a large number of samples, you should try Multicore-TSNE instead.
10. ML definition: A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E.
11. Entropy

Shannon’s entropy is defined for a system with N possible states as follows:



where is the probability of finding the system in the -th state. This is a very important concept used in physics, information theory, and other areas. Entropy can be described as the degree of chaos in the system. The higher the entropy, the less ordered the system and vice versa. This will help us formalize “effective data splitting”, which we alluded to in the context of “20 Questions”.

1. 
2. At the heart of the popular algorithms for decision tree construction, such as ID3 or C4.5, lies the principle of greedy maximization of information gain: at each step, the algorithm chooses the variable that gives the greatest information gain upon splitting. Then the procedure is repeated recursively until the entropy is zero (or some small value to account for overfitting). Different algorithms use different heuristics for “early stopping” or “cut-off” to avoid constructing an overfitted tree.
3. There are two exceptions where the trees are built to the maximum depth:

Random Forest (a group of trees) averages the responses from individual trees that are built to the maximum depth (we will talk later on why you should do this)

Pruning trees. In this approach, the tree is first constructed to the maximum depth. Then, from the bottom up, some nodes of the tree are removed by comparing the quality of the tree with and without that partition (comparison is performed using cross-validation, more on this below).

The most common ways to deal with overfitting in decision trees are as follows:

* artificial limitation of the depth or a minimum number of samples in the leaves: the construction of a tree just stops at some point;
* pruning the tree.

1. Nearest Neighbors Method

The nearest neighbors method (k-Nearest Neighbors, or k-NN) is another very popular classification method that is also sometimes used in regression problems. This, like decision trees, is one of the most comprehensible approaches to classification. The underlying intuition is that you look like your neighbors. More formally, the method follows the compactness hypothesis: if the distance between the examples is measured well enough, then similar examples are much more likely to belong to the same class.

To classify each sample from the test set, one needs to perform the following operations in order:

* + Calculate the distance to each of the samples in the training set.
  + Select samples from the training set with the minimal distance to them.
  + The class of the test sample will be the most frequent class among those nearest neighbors.

1. Nearest Neighbors Method in Real Applications

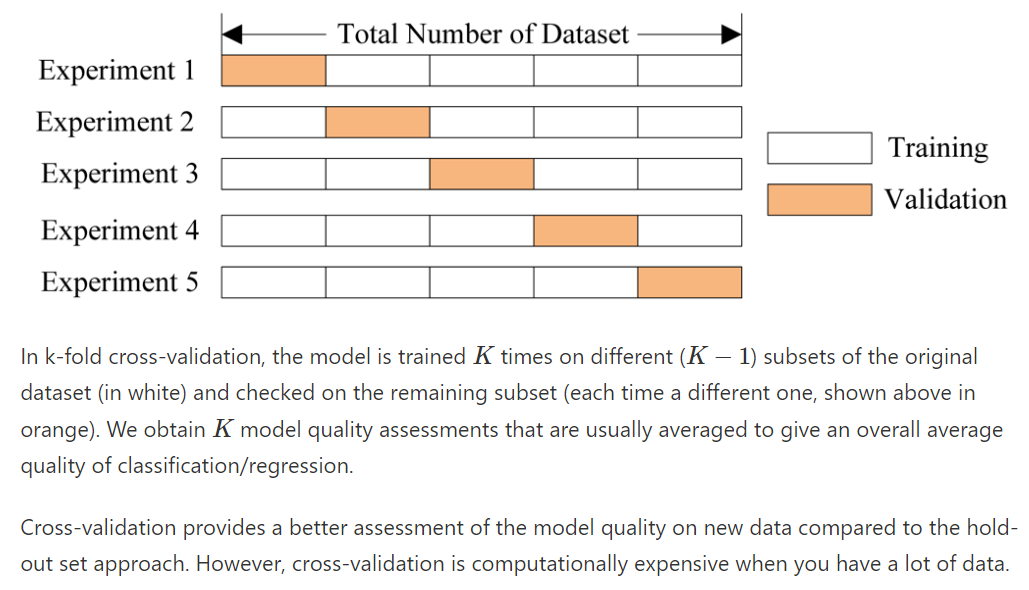
k-NN can serve as a good starting point (baseline) in some cases;

In Kaggle competitions, k-NN is often used for the construction of meta-features (i.e. k-NN predictions as input to other models) or for stacking/blending;

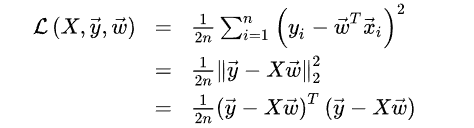
The nearest neighbors method extends to other tasks like recommendation systems. The initial decision could be a recommendation of a product (or service) that is popular among the closest neighbors of the person for whom we want to make a recommendation;

In practice, on large datasets, approximate methods of search are often used for nearest neighbors. There is a number of open source libraries that implement such algorithms; check out Spotify’s library Annoy.

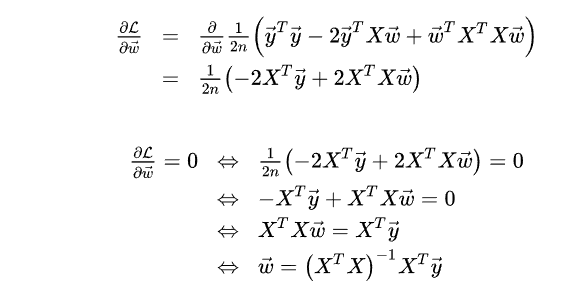
1. Cross validation:

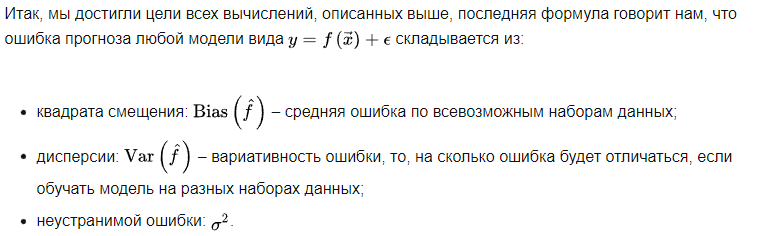


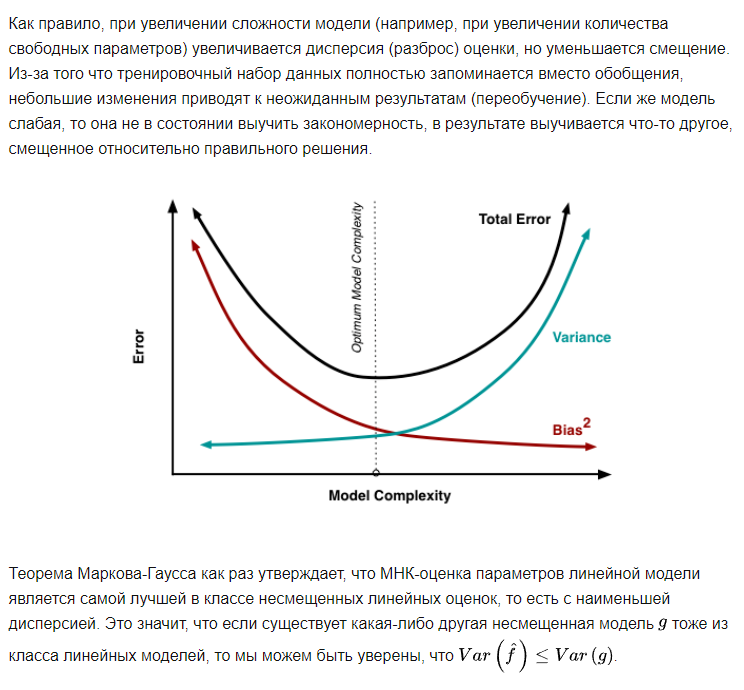
1. K-NN for mnist shows better result, then Tree od RandomForest
2. Linear regression и метод наименьших квадратов (МНК):
3. Один из способов вычислить значения параметров модели является метод наименьших квадратов (МНК, ordinary least squares method (OLS)), который минимизирует среднеквадратичную ошибку между реальным значением зависимой переменной и прогнозом, выданным моделью:



Для решения данной оптимизационной задачи необходимо вычислить производные по параметрам модели, приравнять их к нулю и решить полученные уравнения относительно $\vec w$ (матричное дифференцирование неподготовленному читателю может показаться затруднительным, попробуйте расписать все через суммы, чтобы убедиться в ответе):



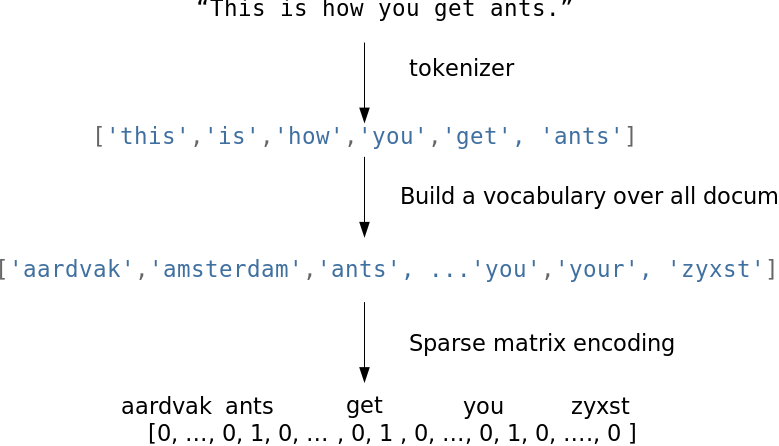




1. Регрессио́нный анализ — набор статистических методов исследования влияния одной или нескольких независимых переменных X\_{1},X\_{2},...,X\_{p} на зависимую переменную Y. Независимые переменные иначе называют регрессорами или предикторами, а зависимые переменные — критериальными или регрессантами. Терминология зависимых и независимых переменных отражает лишь математическую зависимость переменных (см. Корреляция), а не причинно-следственные отношения. Наиболее распространённый вид регрессионного анализа — линейная регрессия, когда находят линейную функцию, которая, согласно определённым математическим критериям, наиболее соответствует данным. Например, в методе наименьших квадратов вычисляется прямая(или гиперплоскость), сумма квадратов между которой и данными минимальна.
2. Регуляризация линейной регрессии

Иногда бывают ситуации, когда мы намеренно увеличиваем смещенность модели ради ее стабильности, т.е. ради уменьшения дисперсии модели $\text{Var}\left(\hat{f}\right)$. Одним из условий теоремы Маркова-Гаусса является полный столбцовый ранг матрицы $X$. В противном случае решение МНК $\vec{w} = \left(X^T X\right)^{-1} X^T \vec{y}$ не существует, т.к. не будет существовать обратная матрица $\left(X^T X\right)^{-1}.$ Другими словами, матрица $X^T X$ будет сингулярна, или вырожденна. Такая задача называется некорректно поставленной. Задачу нужно скорректировать, а именно, сделать матрицу $X^TX$ невырожденной, или регулярной (именно поэтому этот процесс называется регуляризацией). Чаще в данных мы можем наблюдать так называемую мультиколлинеарность — когда два или несколько признаков сильно коррелированы, в матрице $X$ это проявляется в виде "почти" линейной зависимости столбцов. Например, в задаче прогнозирования цены квартиры по ее параметрам "почти" линейная зависимость будет у признаков "площадь с учетом балкона" и "площадь без учета балкона". Формально для таких данных матрица $X^T X$ будет обратима, но из-за мультиколлинеарности у матрицы $X^T X$ некоторые собственные значения будут близки к нулю, а в обратной матрице $\left(X^T X\right)^{-1}$ появятся экстремально большие собственные значения, т.к. собственные значения обратной матрицы – это $\frac{1}{\lambda\_i}$. Итогом такого шатания собственных значений станет нестабильная оценка параметров модели, т.е. добавление нового наблюдения в набор тренировочных данных приведёт к совершенно другому решению. Иллюстрации роста коэффициентов вы найдете в одном из наших прошлых постов. Одним из способов регуляризации является регуляризация Тихонова, которая в общем виде выглядит как добавление нового члена к среднеквадратичной ошибке:

1. мешок слов ("Bag of words"). При таком подходе признаками отзыва будут индикаторы наличия в нем каждого слова из всего корпуса, где корпус – это множество всех отзывов. Идея иллюстрируется картинкой



1. Выводы по кривым валидации и обучения

Ошибка на обучающей выборке сама по себе ничего не говорит о качестве модели

Кросс-валидационная ошибка показывает, насколько хорошо модель подстраивается под данные (имеющийся тренд в данных), сохраняя при этом способность обобщения на новые данные

Валидационная кривая представляет собой график, показывающий результат на тренировочной и валидационной выборке в зависимости от сложности модели:

если две кривые распологаются близко, и обе ошибки велики, — это признак недообучения

если две кривые далеко друг от друга, — это показатель переобучения

Кривая обучения — это график, показывающий результаты на валидации и тренировочной подвыборке в зависимости от количества наблюдений:

если кривые сошлись друг к другу, добавление новых данных не поможет – надо менять сложность модели

если кривые еще не сошлись, добавление новых данных может улучшить результат.

1. Плюсы и минусы линейных моделей в задачах машинного обучения

**Плюсы:**  
Хорошо изучены

Очень быстрые, могут работать на очень больших выборках

Практически вне конкуренции, когда признаков очень много (от сотен тысяч и более), и они разреженные (хотя есть еще факторизационные машины)

Коэффициенты перед признаками могут интерпретироваться (при условии что признаки масштабированы) – в линейной регрессии как частные производные зависимой переменной от признаков, в логистической – как изменение шансов на отнесение к одному из классов в $\exp^{\beta\_i}$ раз при изменении признака $x\_i$ на 1 ед., подробнее тут

Логистическая регрессия выдает вероятности отнесения к разным классам (это очень ценится, например, в кредитном скоринге)

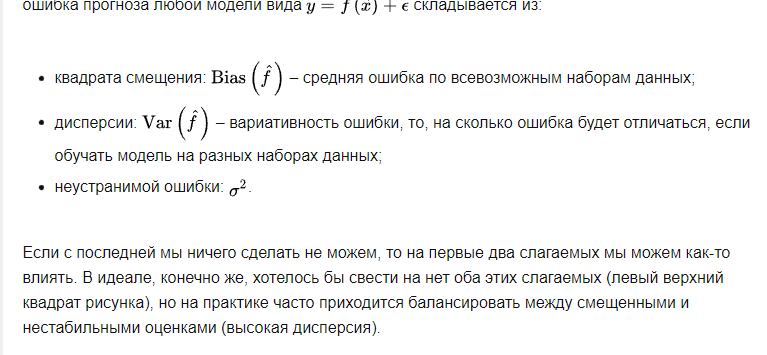
Модель может строить и нелинейную границу, если на вход подать полиномиальные признаки

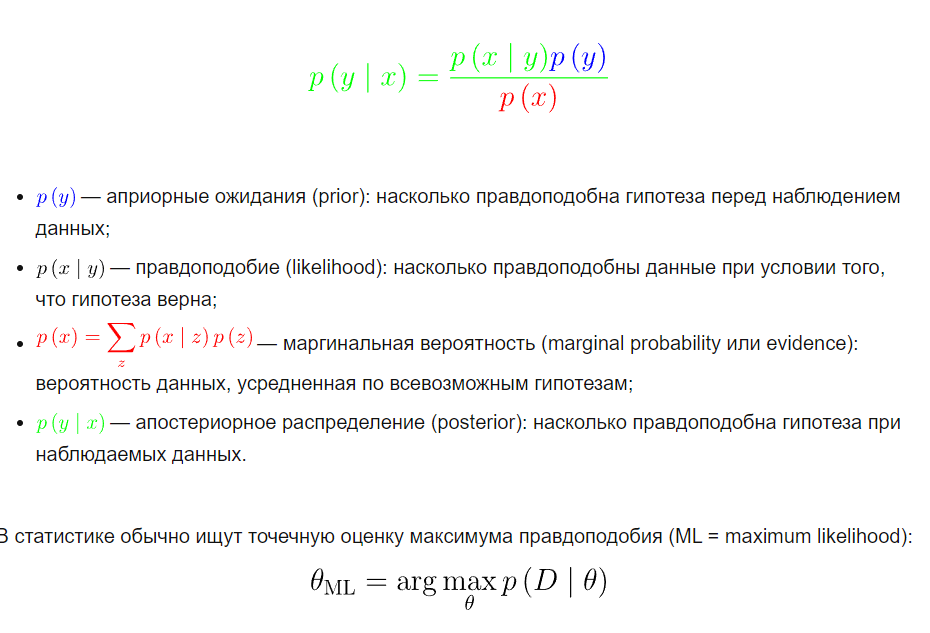
**Минусы:**

Плохо работают в задачах, в которых зависимость ответов от признаков сложная, нелинейная

На практике предположения теоремы Маркова-Гаусса почти никогда не выполняются, поэтому чаще линейные методы работают хуже, чем, например, SVM и ансамбли (по качеству решения задачи классификации/регрессии)

1. максимизация правдоподобия данных – это то же самое, что и минимизация среднеквадратичной ошибки. Получается, что именно такая функция стоимости является следствием того, что ошибка распределена нормально, а не как-то по-другому.



1. ML: программа обучается на опыте $E$ относительно класса задач $T$ в смысле меры качества $L$, если при решении задачи $T$ качество, измеряемое мерой $L$, возрастает при демонстрации нового опыта $E$.
2. Переобученная модель не обладает обобщающей способностью, т.е. на обучающем наборе данных ошибка мала, а на тестовом наборе данных ошибка существенно больше.
3. Регуляризация — это способ уменьшить сложность модели чтобы предотвратить переобучение или исправить некорректно поставленную задачу.
4. Байес:
5. 
6. оценка, полученная методом максимального правдоподобия, – это то же самое, что и оценка, полученная методом наименьших квадратов.
7. Бэггинг позволяет снизить дисперсию (variance) обучаемого классификатора, уменьшая величину, на сколько ошибка будет отличаться, если обучать модель на разных наборах данных, или другими словами, предотвращает переобучение. Эффективность бэггинга достигается благодаря тому, что базовые алгоритмы, обученные по различным подвыборкам, получаются достаточно различными, и их ошибки взаимно компенсируются при голосовании, а также за счёт того, что объекты-выбросы могут не попадать в некоторые обучающие подвыборки. Результат - https://scikit-learn.org/stable/auto\_examples/ensemble/plot\_bias\_variance.html#sphx-glr-auto-examples-ensemble-plot-bias-variance-py
8. Out-of-Bag оценка — это усредненная оценка базовых алгоритмов на тех ~37% данных, на которых они не обучались.
9. Случайный лес — это бэггинг над решающими деревьями, при обучении которых для каждого разбиения признаки выбираются из некоторого случайного подмножества признаков.

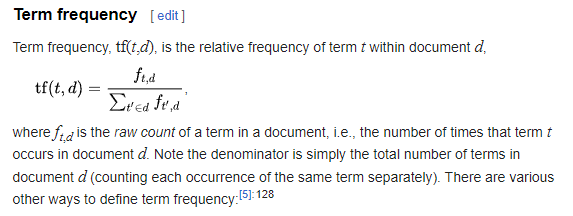
Как мы видим из графиков и значений ошибки MSE, случайный лес из 10 деревьев дает лучший результат, чем одно дерево или бэггинг из 10 деревьев решений. Основное различие случайного леса и бэггинга на деревьях решений заключается в том, что в случайном лесе выбирается случайное подмножество признаков, и лучший признак для разделения узла определяется из подвыборки признаков, в отличие от бэггинга, где все функции рассматриваются для разделения в узле.

1. Gini - Среднее уменьшение неопределенности Джини (или ошибки mse в задачах регрессии) является мерой того, как каждая переменная способствует однородности узлов и листьев в окончательной модели случайного леса. Каждый раз, когда отдельная переменная используется для разбиения узла, неопределенность Джини для дочерних узлов рассчитывается и сравнивается с коэффициентом исходного узла. Неопределенность Джини является мерой однородности от 0 (однородной) до 1 (гетерогенной). Изменения в значении критерия разделения суммируются для каждой переменной и нормируются в конце вычисления. Переменные, которые приводят к узлам с более высокой чистотой, имеют более высокое снижение коэффициента Джини.
2. Bag of words and N-grams:

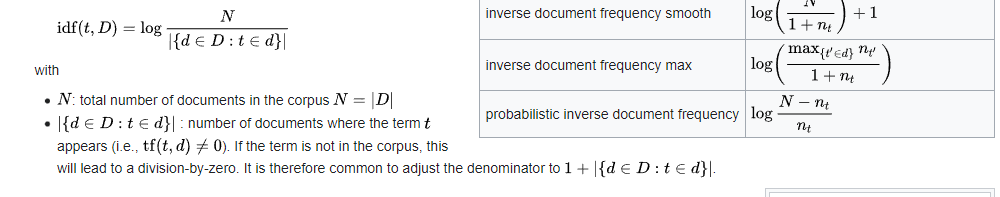
When using algorithms like Bag of Words, we lose the order of the words in the text, which means that the texts “i have no cows” and “no, i have cows” will appear identical after vectorization when, in fact, they have the opposite meaning. To avoid this problem, we can revisit our tokenization step and use N-grams (the sequence of N consecutive tokens) instead.

1. TF-IDF :

TF: The weight of a term that occurs in a document is simply proportional to the term frequency. Term frequency, tf(t,d), is the relative frequency of term t within document d.



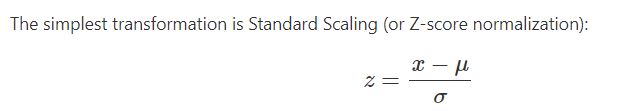
IDF: The specificity of a term can be quantified as an inverse function of the number of documents in which it occurs. he inverse document frequency is a measure of how much information the word provides, i.e., if it is common or rare across all documents. It is the logarithmically scaled inverse fraction of the documents that contain the word (obtained by dividing the total number of documents by the number of documents containing the term, and then taking the logarithm of that quotient):

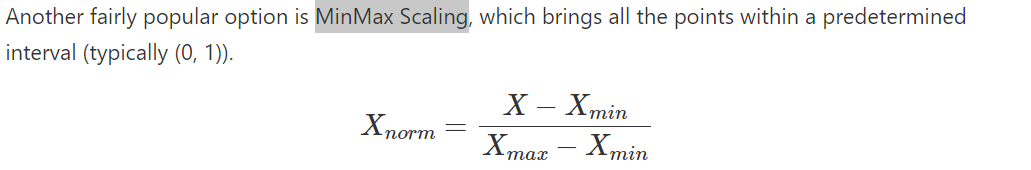


1. Word2vec: Word2Vec is a special case of the word embedding algorithms. Using Word2Vec and similar models, we can not only vectorize words in a high-dimensional space (typically a few hundred dimensions) but also compare their semantic similarity. This is a classic example of operations that can be performed on vectorized concepts: king - man + woman = queen.

Pretrained-model - <https://github.com/3Top/word2vec-api#where-to-get-a-pretrained-model>

1. Normalization:





Standard Scaling and Min Max Scaling have similar applications and are often more or less interchangeable. However, if the algorithm involves the calculation of distances between points or vectors, the default choice is StandardScaling. But Min Max Scaling is useful for visualization by bringing features within the interval (0, 255).

1. Easy feature selection <https://www.kaggle.com/code/arsenyinfo/easy-feature-selection-pipeline-0-55-at-lb/script>
2. K-Means - В теории вероятностей и статистике ковариация является мерой совместной изменчивости двух случайных величин. Если большие значения одной переменной в основном соответствуют большим значениям другой переменной, и то же самое верно для меньших значений (то есть переменные имеют тенденцию демонстрировать одинаковое поведение), ковариация положительна.В противоположном случае, когда большие значения одной переменной в основном соответствуют меньшим значениям другой (т. е. переменные имеют тенденцию показывать противоположное поведение), ковариация отрицательна. Таким образом, знак ковариации показывает тенденцию линейной зависимости между переменными. Величину ковариации нелегко интерпретировать, поскольку она не нормирована и, следовательно, зависит от величин переменных. Однако нормализованная версия ковариации, коэффициент корреляции, своей величиной показывает силу линейной зависимости.

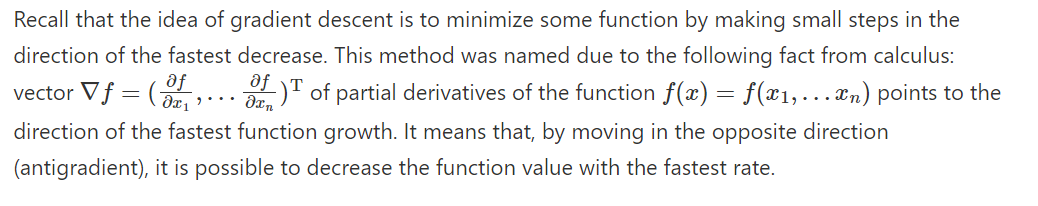
K-means algorithm is the most popular and yet simplest of all the clustering algorithms. Here is how it works:

* + Select the number of clusters that you think is the optimal number.
  + Initialize points as “centroids” randomly within the space of our data.-
  + Attribute each observation to its closest centroid.
  + Update the centroids to the center of all the attributed set of observations.
  + Repeat steps 3 and 4 a fixed number of times or until all of the centroids are stable (i.e. no longer change in step 4).

When working with k-means, we optimize the sum of squared distances between the observations and their centroids. But, there is a problem – the optimum is reached when the number of centroids is equal to the number of observations, so you would end up with every single observation as its own separate cluster.

In order to avoid that case, we should choose a number of clusters after which a function

is decreasing less rapidly.

1. For classification, use the support vector machine – class sklearn.svm.LinearSVC.
2. SGD: 

Если вы хотите ехать как можно быстрее, вам нужно выбрать путь самого крутого спуска. Вычисление антиградиентов можно рассматривать как оценку наклона в различных точках.

1. Time series is a series of data points indexed (or listed or graphed) in time order.
2. Normalization - <https://tungmphung.com/deep-learning-normalization-methods/>
3. Time Series - Авторегрессионное интегрированное скользящее среднее, или ARIMA, является одним из наиболее широко используемых методов прогнозирования для однофакторного прогнозирования данных временных рядов.

Хотя метод может обрабатывать данные с трендом, он не поддерживает временные ряды с сезонным компонентом.

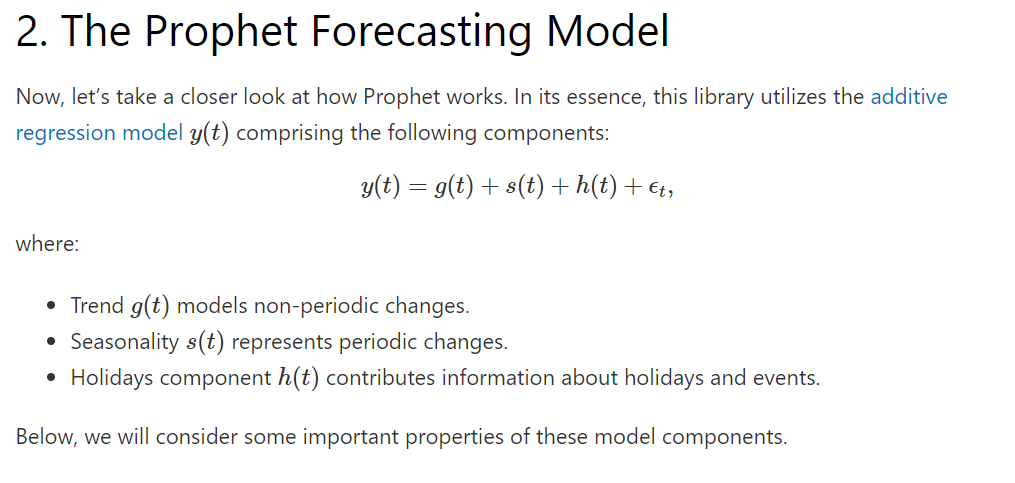
Расширение ARIMA, которое поддерживает прямое моделирование сезонного компонента ряда, называется SARIMA.

ARIMA model which can handle non-stationary data with the help of nonseasonal differences.

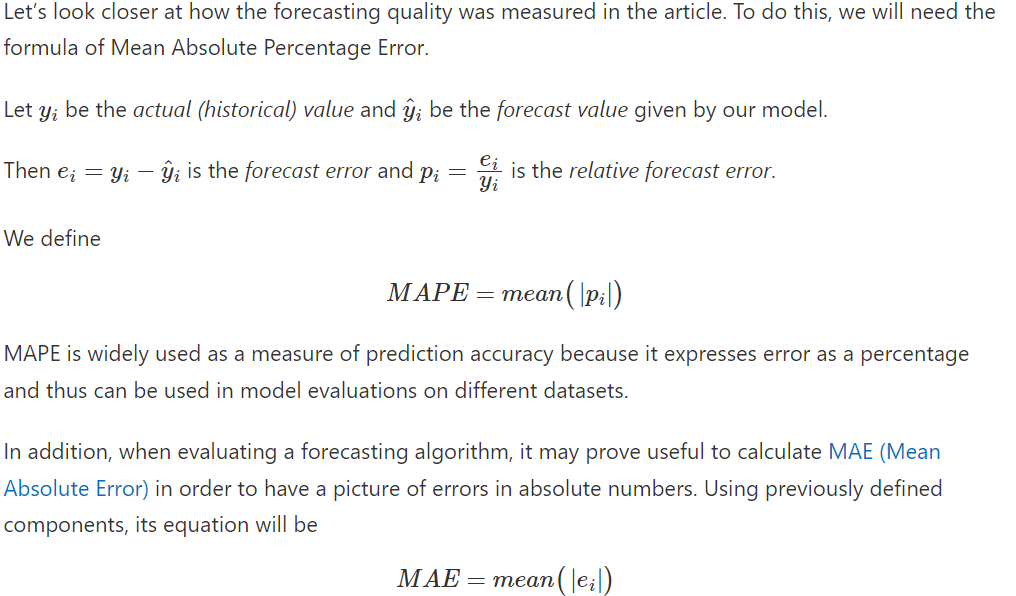
1. We can fight non-stationarity using different approaches: various order differences, trend and seasonality removal, smoothing, and transformations like Box-Cox or logarithmic.
2. In the case of Ridge regression, those constraints are the sum of squares of the coefficients multiplied by the regularization coefficient. The bigger the coefficient a feature has, the bigger our loss will be. Hence, we will try to optimize the model while keeping the coefficients fairly low.

As a result of **this L2 regularization**, we will have higher bias and lower variance, so the model will generalize better (at least that’s what we hope will happen).

The second regression model, Lasso regression, adds to the loss function, not squares, but absolute values of the coefficients. As a result, during the optimization process, coefficients of unimportant features may become zeroes, which allows for automated feature selection. This regularization type is **called L1.**



1. MAPE in forecasting:



1. A few words about the algorithms that Prophet was compared with. Most of them are quite simple and often are used as a baseline for other models:

naive is a simplistic forecasting approach where we predict all future values relying solely on the observation at the last available point of time.

snaive (seasonal naive) is a model that makes constant predictions taking into account information about seasonality. For instance, in the case of weekly seasonal data for each future Monday, we would predict the value from the last Monday, and for all future Tuesdays we would use the value from the last Tuesday and so on.

mean uses the averaged value of data as a forecast.

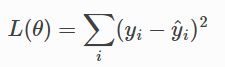
arima stands for Autoregressive Integrated Moving Average, see [Wikipedia](https://en.wikipedia.org/wiki/Autoregressive_integrated_moving_average) for details.

ets stands for Exponential Smoothing, see [Wikipedia](https://en.wikipedia.org/wiki/Exponential_smoothing) for more.

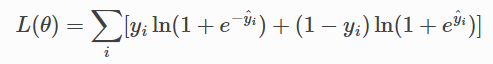
1. **XGBOOST https://xgboost.readthedocs.io/en/stable/tutorials/model.html**

In order to train the model, we need to define the objective function to measure how well the model fit the training data. A salient characteristic of objective functions is that they consist of two parts: training loss and regularization term: obj(Q) = l(Q) + Omega(Q). Omega -> .

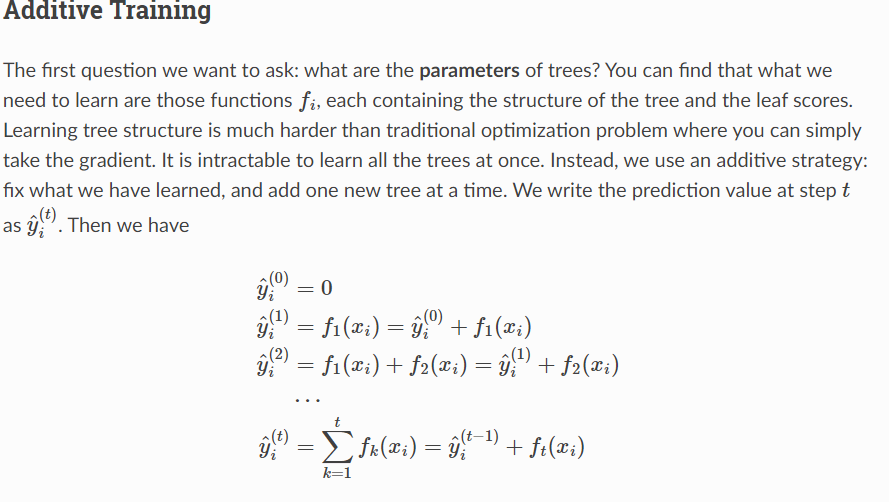
l – training loss function, Omega – regularization term.

L – is the mean squared error - 

Or for logistic regression is used logistic loss =

  
The regularization term is what people usually forget to add. The regularization term controls the complexity of the model, which helps us to avoid overfitting.

Now here comes a trick question: what is the model used in random forests? Tree ensembles! So random forests and boosted trees are really the same models; the difference arises from how we train them. This means that, if you write a predictive service for tree ensembles, you only need to write one and it should work for both random forests and gradient boosted trees. (See Treelite for an actual example.) One example of why elements of supervised learning rock.

****

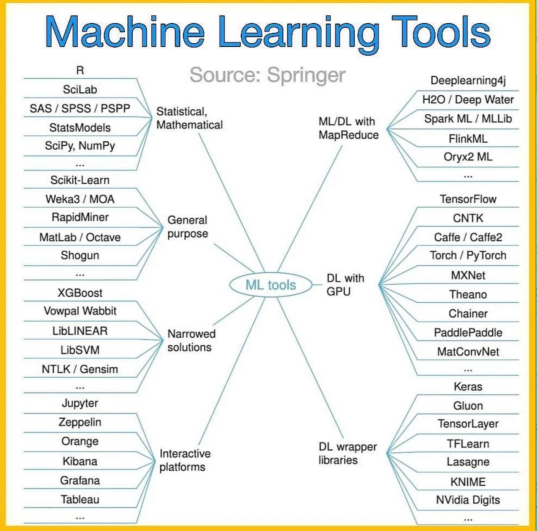
1. Материалы разные - <https://habr.com/ru/company/selectel/blog/723854/?telegram_habr=#7>
2. Feature engineering <https://habr.com/ru/company/ruvds/blog/680498/>
3. Много материалов - <https://machinelearningmastery.com/feature-selection-for-regression-data/>
4. sample\_weights is used to provide a weight for each training sample. That means that you should pass a 1D array with the same number of elements as your training samples (indicating the weight for each of those samples). In case you are using temporal data you may instead pass a 2D array, enabling you to give weight to each timestep of each sample.
5. Ridge regression is a method of estimating the coefficients of multiple-regression models in scenarios where the independent variables are highly correlated
6. How to delete categorical:

def recover\_dtypes(df):    for col in df.columns:

        if df[col].dtype == 'object' or df[col].dtype == 'datetime64[ns]':

            del df[col]

    return df



1. Same as leetcode, but more structured - https://neetcode.io/practice