## Predictive Analysis

Predictive analytics encompasses a variety of statistical techniques but we chose to use [machine learning](https://en.wikipedia.org/wiki/Machine_learning) to analyze current and historical facts, which would allow us to make predictions about future or otherwise unknown events.

Our choice regarding this project was between supervised and unsupervised learning, since this project is made for learning purpose we chose to implement both, to see what each of them are good at and why we should consider them.

**Supervised learning**

Overview

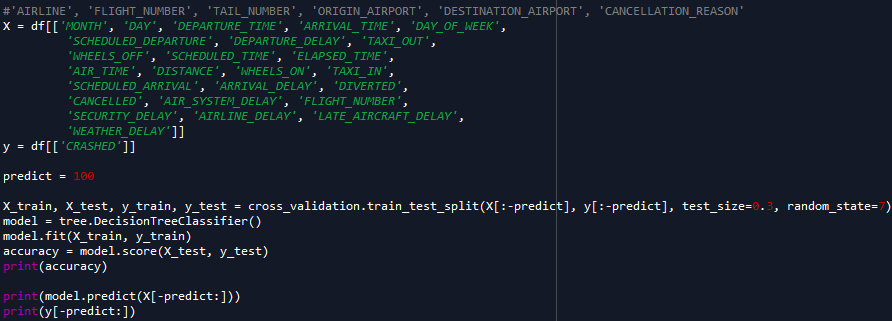
Supervised learning maps an input to an output based on example input-output pairs. It uses labeled data consisting of features (X- input object) and labels (y- a desired output). A supervised learning algorithm analyzes the training data and produces a function, which can be used for mapping new examples. In best case scenario this will allow for the algorithm to correctly determine the class labels for unseen instances. This requires the learning algorithm to generalize from the training data to unseen situations in a "reasonable" way.

* Classification: A classification problem is when the output variable is a category, such as “red” or “blue” or “disease” and “no disease”.
* Regression: A regression problem is when the output variable is a real value, such as “dollars” or “weight”.

Examples of Supervised Learning:

* Linear Regression
* Logistic Regression
* Decision Tree
* SVM (Support Vector Machine)
* Naive Bayes
* kNN (k- Nearest Neighbors)
* Random Forest

Our code:



As shown above we started by separating data frame into features and labels. Than we decided how many rows of this dataset we would like to predict, then followed by using cross\_vaidation to split this for training purpose (70% training/30% testing). After this we trained the algorithm and then tested the results receiving the score of 99,53% accuracy. Although the resulting score may seem very accurate, we determined that it is this precise only because the possibility of animal collision is tiny, making the algorithm very biased towards plain not crashing, but still we decided to predict and used last 10 rows of dataset to base our predictions on and in the end compared the received results to expected ones.

We decided to use decision tree, because it is good solution to solve classification problems, which we are facing, as we want to know whether the plain will or will not crash on a specific flight.

Some other algorithms we investigated were:

**Naive Bayes**, which we opted not to use, as although is fast and highly performing algorithm for machine learning, it is really biased to often recurring results, which would mean the algorithm would just consider all flights to not crash, as it is most represented result.

**Logistic Regression,** it is used to estimate discrete values (like 0/1, yes/no, true/false) based on given set of independent variables. In simple words, it predicts the probability of occurrence of an event. Since, it predicts the probability, its output values lie between 0 and 1. We considered this but opted to use decision tree as we wanted to see clear answer yes or no.

**Unsupervised learning**

Overview

Unsupervised learningdescribes hidden structure of "unlabeled" data. It can be divided in following categories: [Clustering](https://en.wikipedia.org/wiki/Data_clustering), [Anomaly detection](https://en.wikipedia.org/wiki/Anomaly_detection) and [Neural Networks](https://en.wikipedia.org/wiki/Artificial_neural_network). Since the examples given to the learner are unlabeled, there is no evaluation of the accuracy of the structure that is output by the algorithm, it is used to predict pattern in data —which is one way of distinguishing from supervised.

* Clustering is the task of grouping a set of objects in such a way that objects in the same group (called a cluster) are more similar (in some sense) to each other than to those in other groups (clusters)
* Anomaly detection is the identification of items, events or observations which do not conform to an expected pattern or other items in a dataset. Typically, the anomalous items will translate to problem such as bank fraud, a structural defect, medical problems or errors in a text. Anomalies are also referred to as outliers, novelties, noise, deviations and exceptions.
* Neural networks are computing systems inspired by the biological neural networks that constitute animal brains. Such systems "learn" (i.e. progressively improve performance on) tasks by considering examples, generally without task-specific programming. For example, in image recognition, they might learn to identify images that contain cats by analyzing example images that have been manually labeled as "cat" or "no cat" and using the results to identify cats in other images. They do this without any a priori knowledge about cats, e.g., that they have fur, tails, whiskers and cat-like faces. Instead, they evolve their own set of relevant characteristics from the learning material that they process.

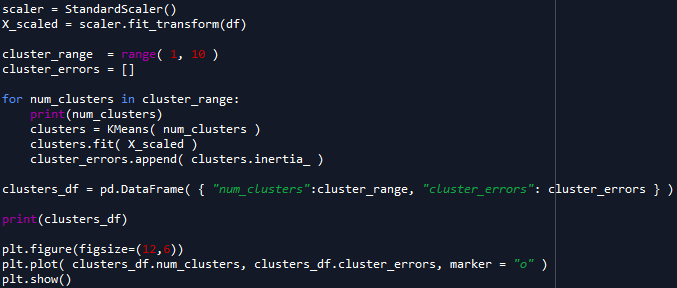
Examples of Unsupervised Learning:

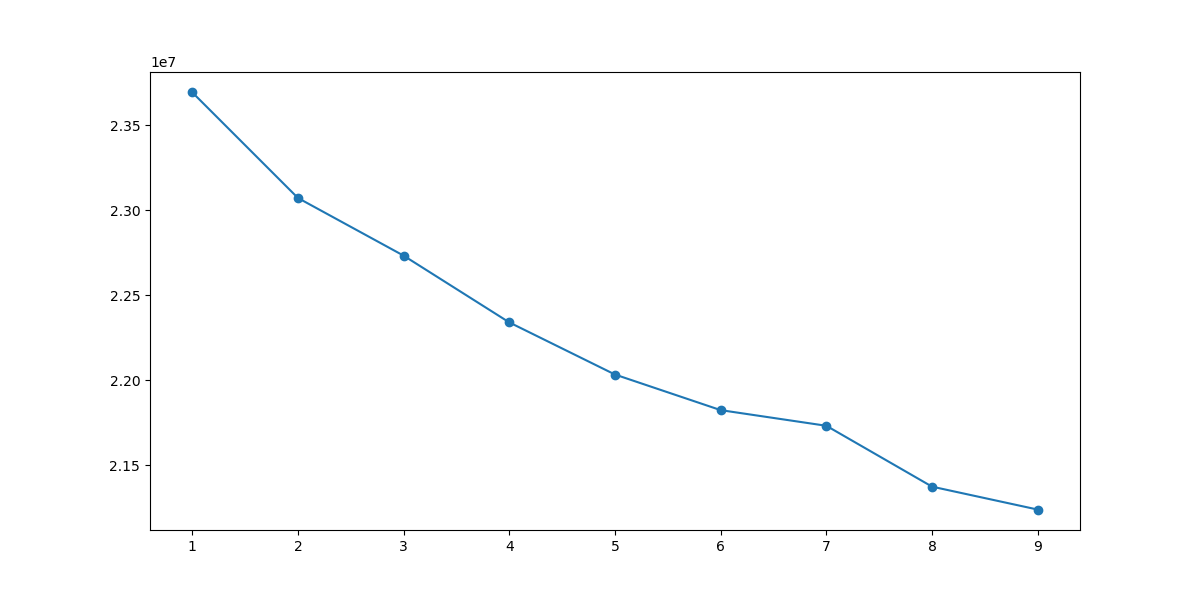
* K-Means
* hierarchical clustering
* Deep Belief Nets
* Generative Adversarial Networks
* Hebbian Learning

Deciding number of clusters:

To decide the number of clusters we had choice of using either Silhouette Analysis or Elbow Analysis, we chose latter, as it was both simple to implement and work with.

Chosen method looks at the percentage of variance with different number of clusters: The person should choose highest number of clusters, so that adding another cluster doesn't give much better modeling of the data.

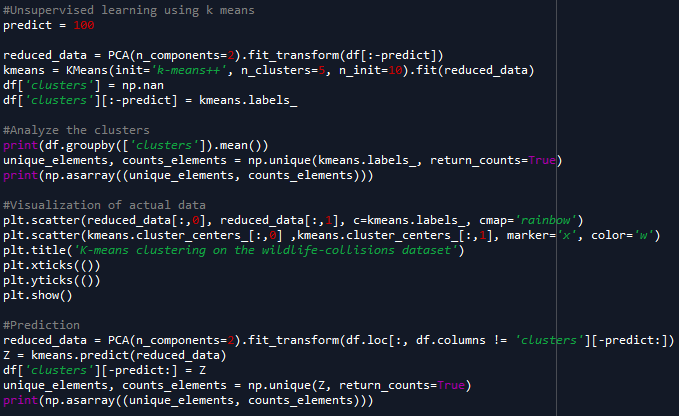


As the code above shows, we started work by scaling our dataset, so our date isn’t affected by outliners, making mean value 0 and standard deviation of 1. Following this step, we executed the algorithm 10 times with different number of clusters and in the end the percentage of variance was plotted to visualize the results.

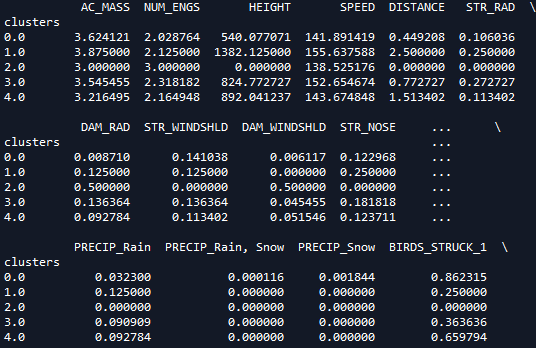
As can be seen in the image, the variance starts dropping slower at algorithm which has 5 cluster, this led us to decide to use this number for further tasks when deciding the number of clusters necessary for the dataset.

We wanted to increase this test further, but found it hard to do so, as time spent on clustering increases inspirationally resulting in approximately 1 hour spent on running existing test, as a result we decided to limit test size just to 10, as we believe it accurately represents dataset and further conclusions can be further extracted from this result.

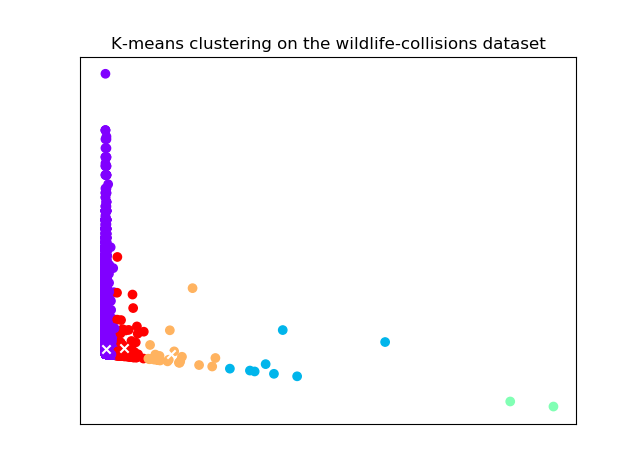
Our code:



We started the work on unsupervised learning, by deciding the amount of values to be predicted, then we decomposed the data into 2 components and proceeded to teach the algorithm using the extracted information. Than we added the “clusters” column to data frame, so we can store the results of clustering. The we proceeded to display information about each cluster (img 1) and visualized the clusters in easily understandable way (img 2). At the end we continued by predicting in what cluster the new values would fit. After executing this the results are following:



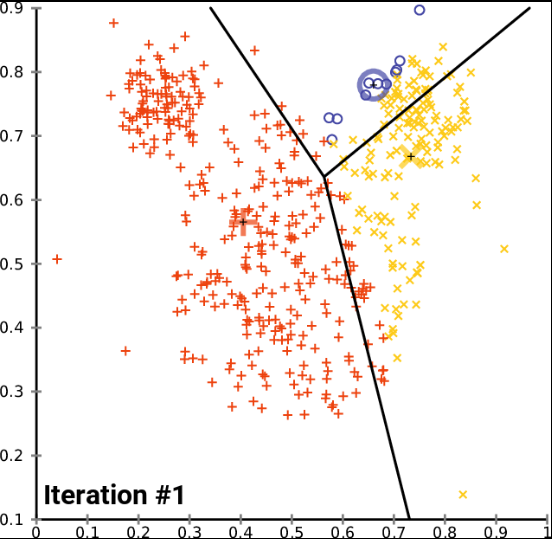
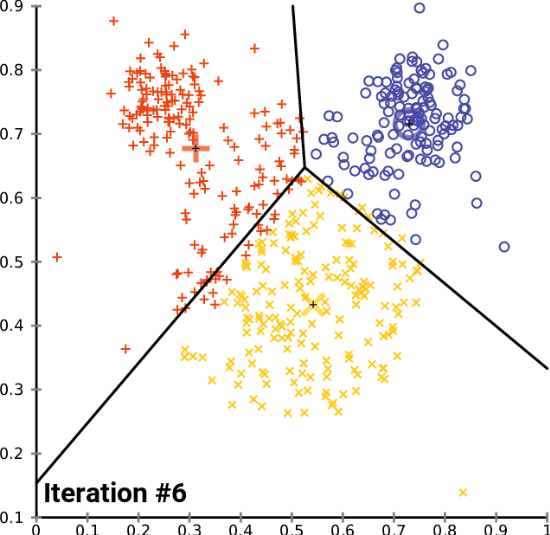
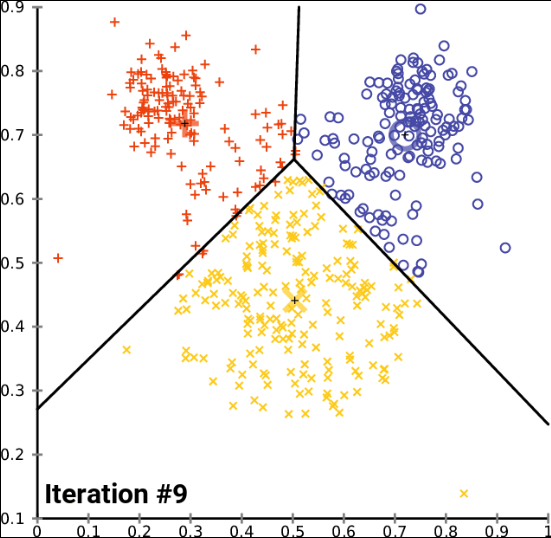
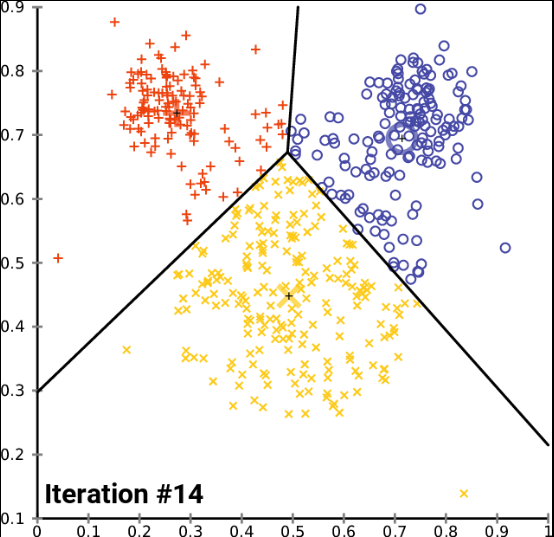
Img 1



img 2



As can be seen from results shown above, some cluster are bigger than others, for example cluster nr. 1 has almost 99% of all values, which tells use that there is something drastically different between them and could be looked into when continuing work un project

We decided to use K-Means, because it is good solution to solve problems related to unsupervised learning, as it clusters data to allow us to see correlations which otherwise we could have looked over.

Some other algorithms we investigated were:

**Hierarchical clustering** is a method of [cluster analysis](https://en.wikipedia.org/wiki/Cluster_analysis) which seeks to build a [hierarchy](https://en.wikipedia.org/wiki/Hierarchy) of clusters. Strategies for hierarchical clustering generally fall into two types:

* Agglomerative: This is a "bottom up" approach: each observation starts in its own cluster, and pairs of clusters are merged as one moves up the hierarchy.
* Divisive: This is a "top down" approach: all observations start in one cluster, and splits are performed recursively as one moves down the hierarchy.

Results of hierarchical clustering are usually presented in a [dendrogram](https://en.wikipedia.org/wiki/Dendrogram). The standard algorithm for hierarchical clustering is too slow for even medium data sets and has great memory requirements, that’s why we couldn’t afford to use it, as we encountered memory issues even during the data wrangling part.

**Deep belief network** (DBN) is a generative graphical model, or alternatively a class of deep neural network, composed of multiple layers of hidden units, with connections between the layers but not between units within each layer. When trained on a set of examples without supervision, a DBN can learn to reconstruct its inputs. The layers then act as feature detectors. After this learning step, a DBN can be further trained with supervision to perform classification. This could have been used, but we determined that we will have determined supervised learning algorithm for predicting and unsupervised algorithm for showing before unseen correlations, removing the need to implement already mentioned algorithm.

**Final thoughts**

We overestimated our skills when choosing the projects idea, even though we achieved our goals for this project we should have done more research regarding our idea. The result of 99.5% accuracy may seem high but we determined that this is basically the same as ratio of flight crushing to not crushing, to make our prediction more accurate we would need more informative datasets and other data which we didn’t use in making of this project(some of the data we could have used to increase the efficiency of prediction could be: weather patterns, information about birds and they’re flight patterns and many more)