Universidade Estadual de Feira de Santana
EXA 864 - Mineração de Dados
Prof. Rodrigo Tripodi Calumby

Projeto Prático 2 - There and back again

Discentes: Ramon de Cerqueira Silva
Adlla Katarine Aragão Cruz Passos
Daniel Alves Costa

Fonte: <https://docmanagement.com.br/03/13/2014/tecnologia-de-analise-de-dados-da-ibm-auxilia-a-reduzir-ciclos-de-planejamento//> Acesso em 04 set .2019

Sumário

Introdução

- Aplicação de técnicas adquiridas em sala de aula para criação de uma avaliação experimental.
- Objetivo é predizer, a partir de algoritmos, determinadas características e superpoderes dos personagens disponibilizados.
- Utilização de técnicas de classificação de dados com modelos de Ensemble Learning.
- Objetivo do grupo é atingir 70% de acurácia na previsão de todos os atributos.

Algoritmos e Ferramentas Utilizadas

- Tecnologias
- Ferramentas

Tecnologias

- Python em sua versão 3.6
- IDE Spyder 3.3.4
- Anaconda Navigator
- Ambientação para utilizar o pacote Fancyimpute

Fonte: <https://www.google.com/search> Acesso em 04 set .2019

Ferramentas

- Leitura de arquivos .csv, com a biblioteca Pandas:
  - Dataframes

- Algoritmos para classificação e métodos de Ensemble com a biblioteca Sklearn:
  - Árvore de Decisão
  - Naive Bayes
  - Random Forest
  - VotingClassifier
  - Bagging
  - Adaboost

Fonte: <https://www.google.com/search> Acesso em 04 set .2019

Ferramentas

- Tratamento de valores NaN, com a biblioteca Numpy

- Avaliação da qualidade dos modelos de predição, com Sklearn:
  - Matriz de Confusão
  - Recall
  - Precision
  - Validação Cruzada
  - Acurácia
  - F1
  - Curva ROC

Fonte: <https://www.google.com/search> Acesso em 04 set .2019

Análise das Propostas e Metodologia

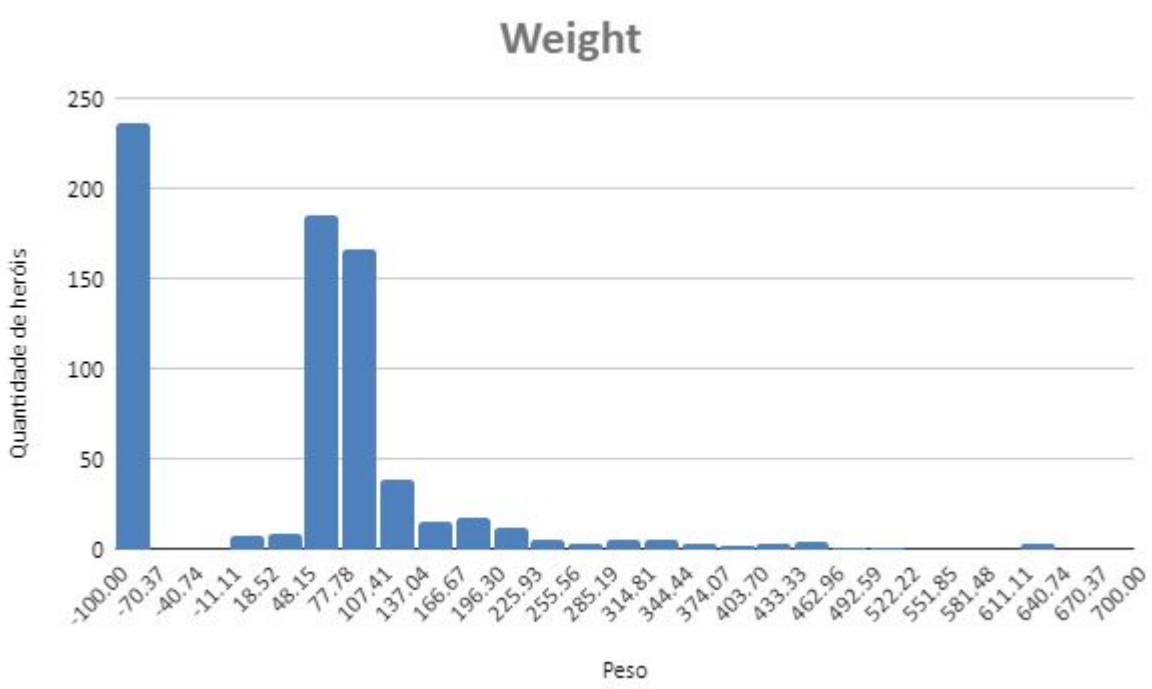
- Base de Dados
- Pré-processamento
- Classificadores
- Ensemble Learning
- Validação

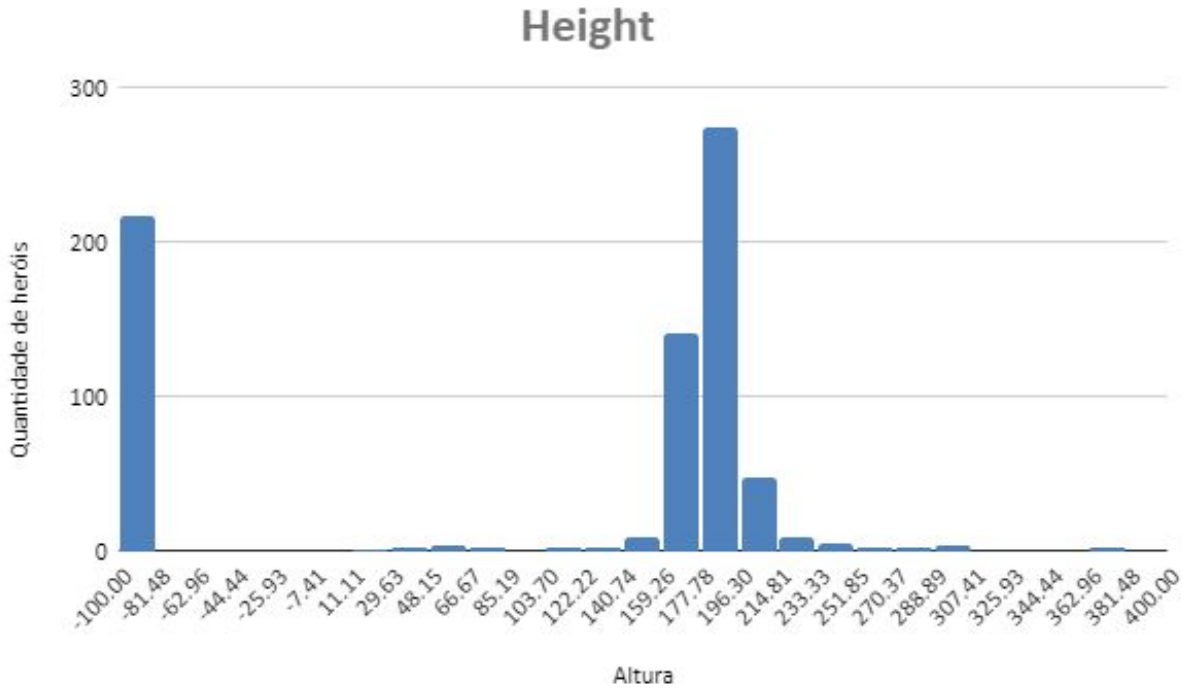
Base de Dados

- 2 datasets em formato **csv**
- Dataset “herois”
  - 734 nomes de heróis
  - 9 características
  - Atributos categóricos e numéricos
  - Valores ausentes e **NaN**

- Dataset “superpoderes”
  - 667 nomes de heróis
  - 167 poderes
  - Atributos binários
  - Sem valores ausentes ou **NaN**

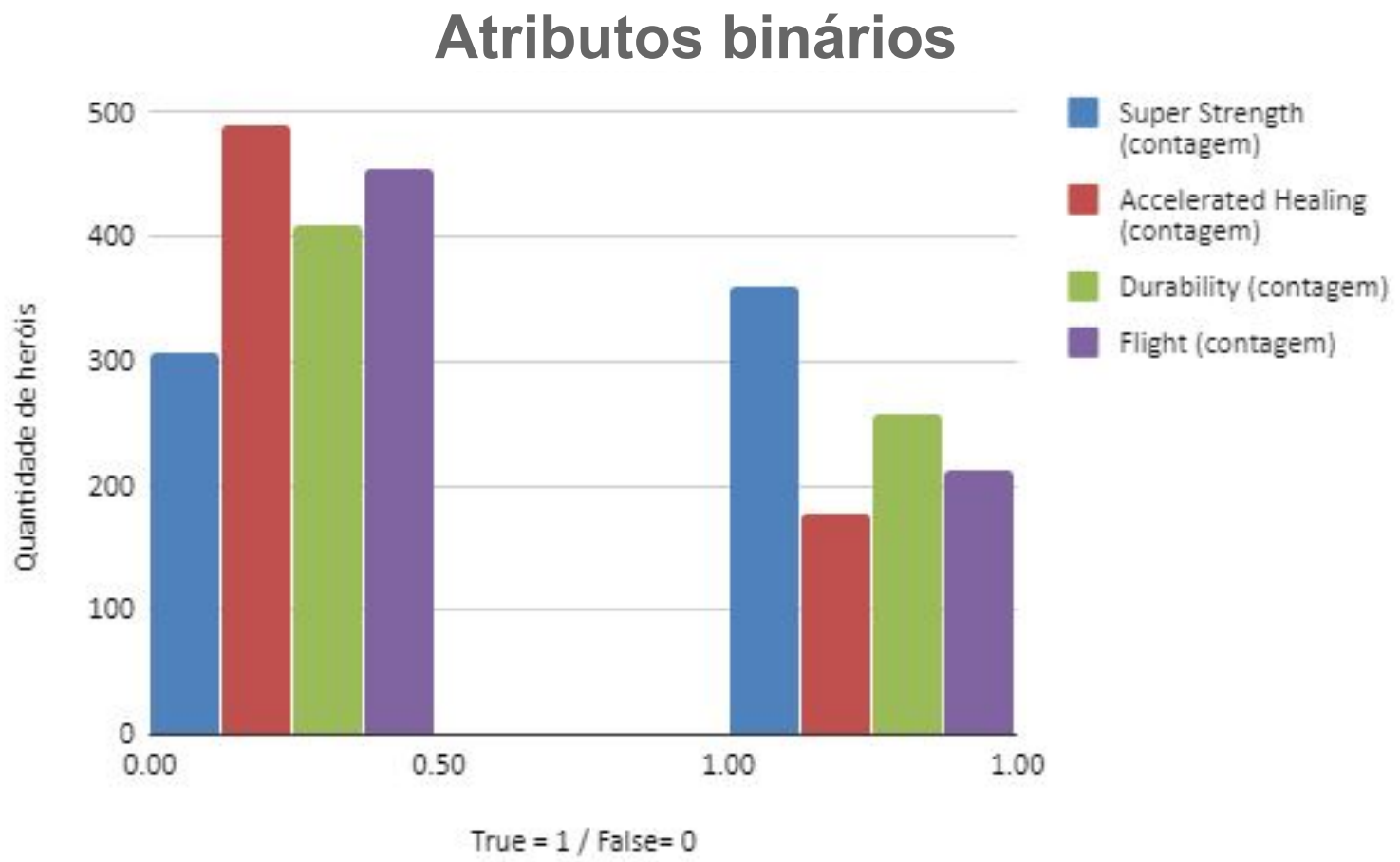
Base de Dados





Fonte: Autor Próprio

Base de Dados



Fonte: Autor Próprio

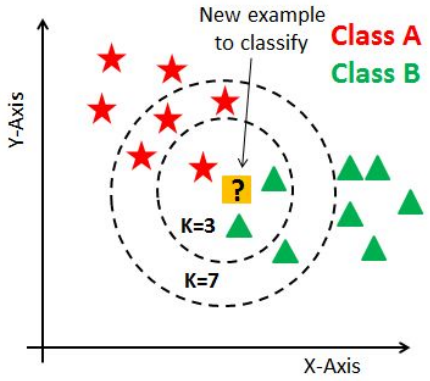
Pré-processamento

- Fase mais importante de um projeto de aprendizagem
- Integração dos datasets
- Tratamento dos dados
- Transformação de atributos multiclasses em binários
- Tratamento de personagens duplicados e sem características

name	Gender	Eye color	Race	Hair color	Height	Publisher	Skin color	Alignment	Weight
Goliath	Male	-	-	-	-99	Marvel Comics	-	good	-99
Goliath	Male	-	Human	-	-99	Marvel Comics	-	good	-99
Goliath	Male	-	Human	-	-99	Marvel Comics	-	good	-99

Fonte: Autor Próprio

- Preenchimento de valores ausentes e **NaN**
- Uso da biblioteca Fancyimpute para tratamento com KNN
  - Valores ’-’ foram trocados por np.nan.
  - K vizinhos, escolhidos com base em alguma medida de distância.
  - A média é usada como estimativa de imputação.
    - Atributos discretos: Hamming
    - Atributos contínuos: Euclidean, Manhattan e Cosine.
  - Parâmetro usado: 3-NN ou 5-NN, e Hamming.



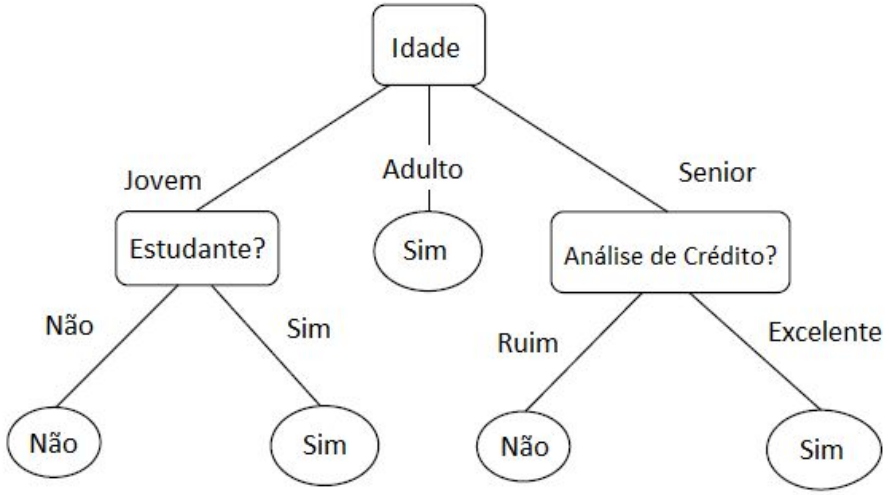
Fonte: <https://www.datacamp.com/community/tutorials/k-nearest-neighbor-classification-scikit-learn> Acesso em 04 set. 2019

Classificadores

- Naive Bayes

$$P(A|B) = \frac{P(B|A)\ P(A)}{P(B)}$$

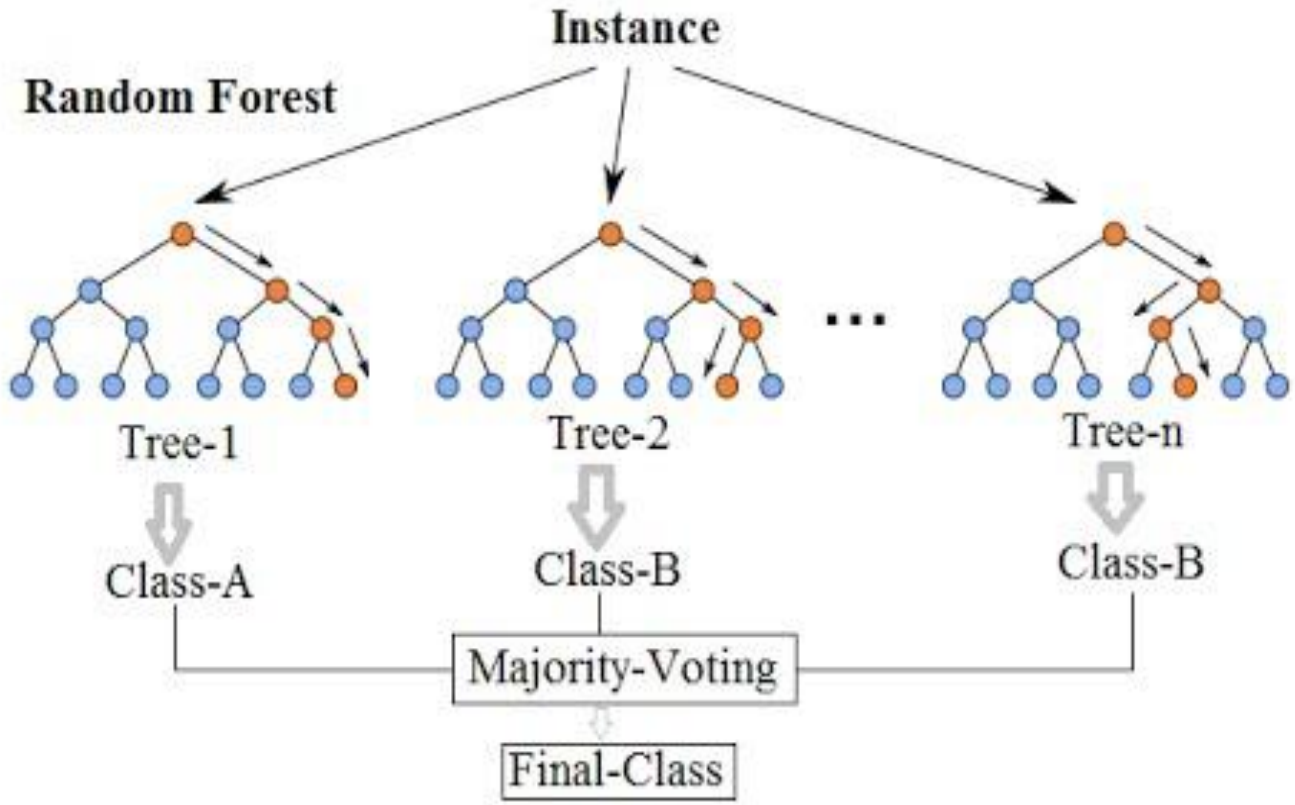
- Decision Tree
  - hiperparâmetros
  - *GridSearchCV*



Fonte: <https://www.google.com/search> Acesso em 04 set .2019

Classificadores

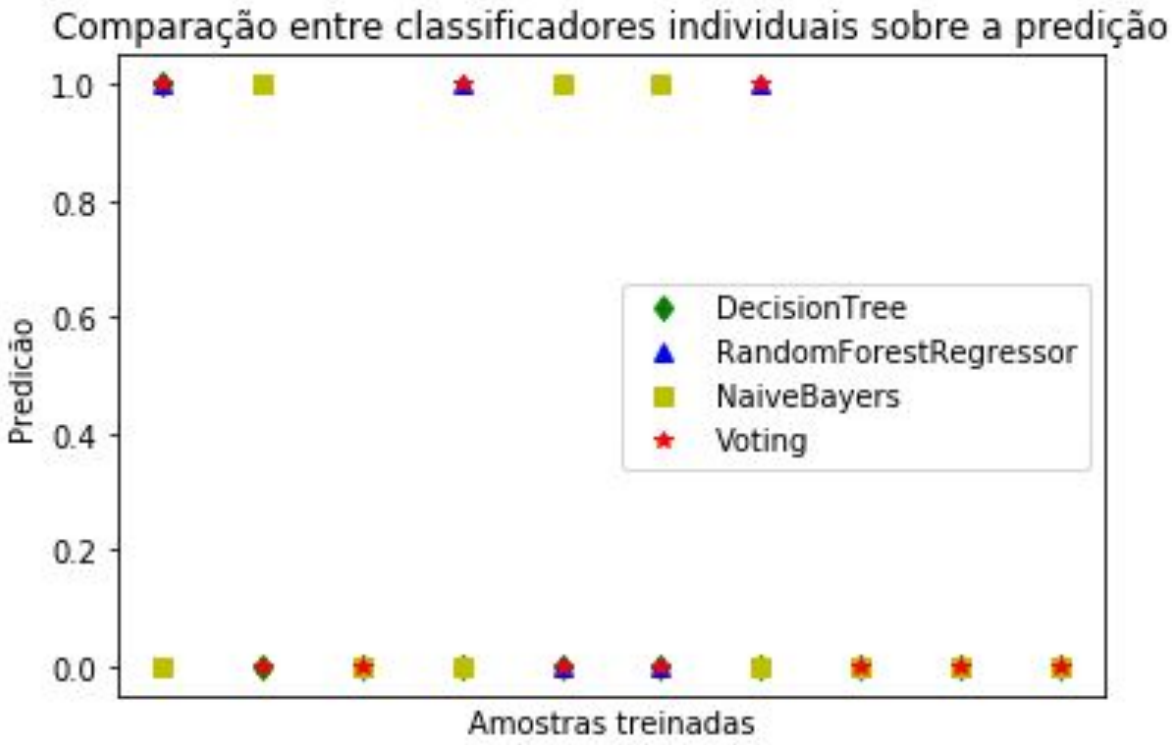
- Random Forest
  - parâmetro: N_samples:50



Fonte: <https://www.google.com/search> Acesso em 04 set .2019

Ensemble Learning

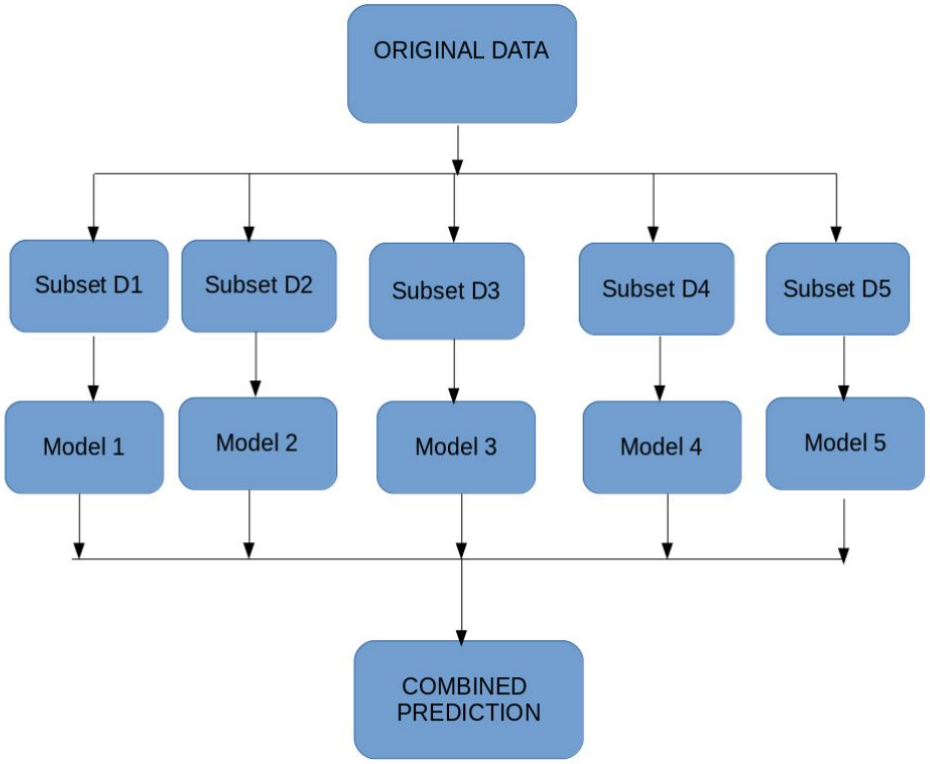
- Bagging
- Adaboost
- Voting-Classifier



Fonte: Autor Próprio

Ensemble Learning

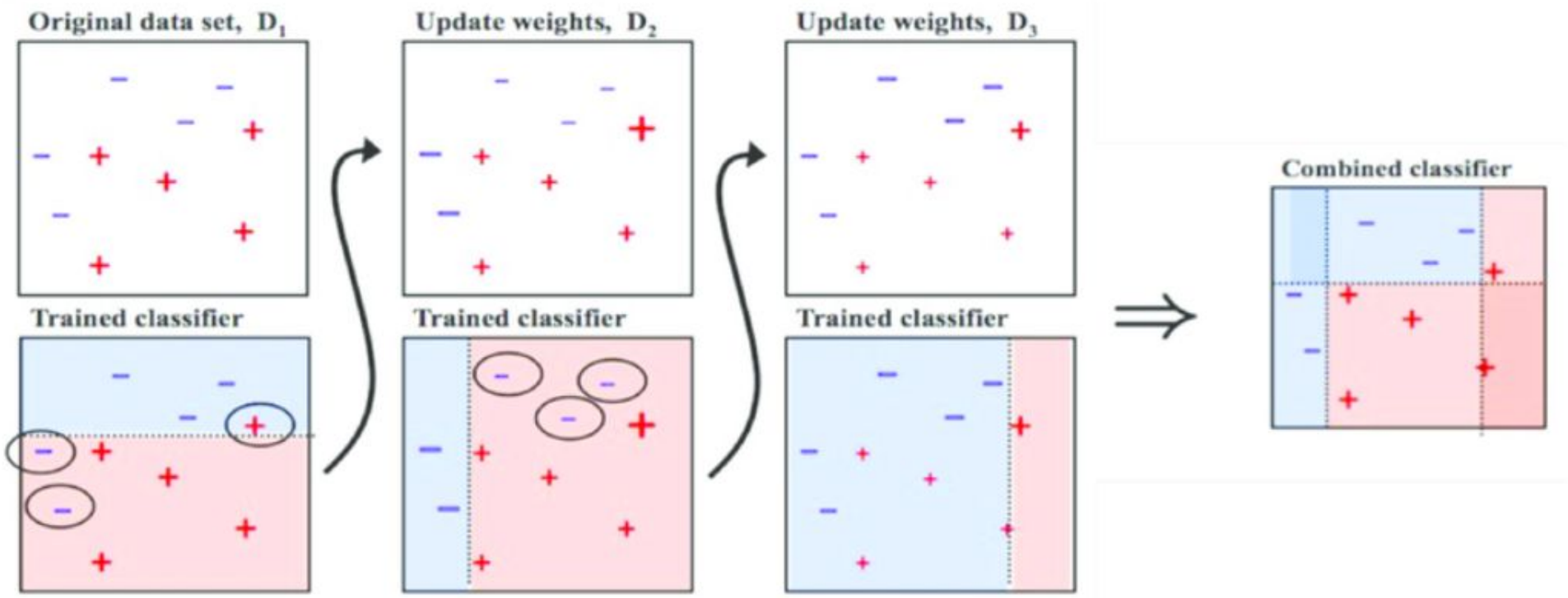
Bagging



Fonte: <https://medium.com/swlh/difference-between-bagging-and-boosting-f996253acd22> Acesso em 04 set. 2019

Ensemble Learning

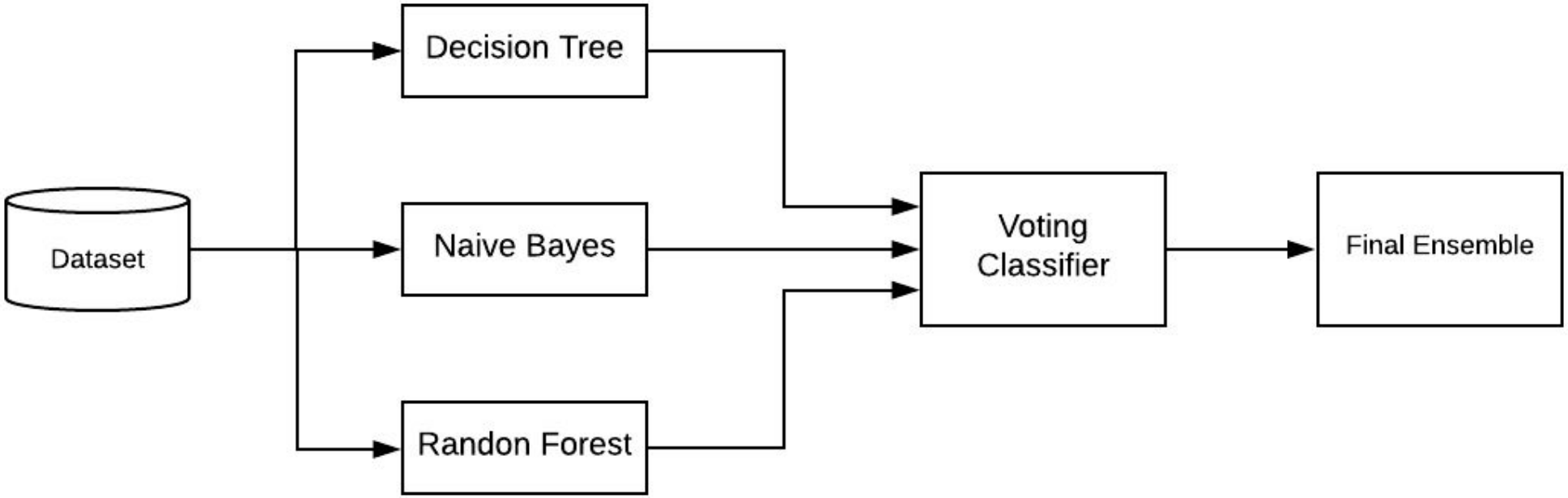
AdaBoost



Fonte: <https://medium.com/diogo-menezes-borges/boosting-with-adaboost-and-gradient-boosting-9cbab2a1af81> Acesso em 04 set. 2019

Ensemble Learning

Voting Classifier



Fonte: Autor Próprio

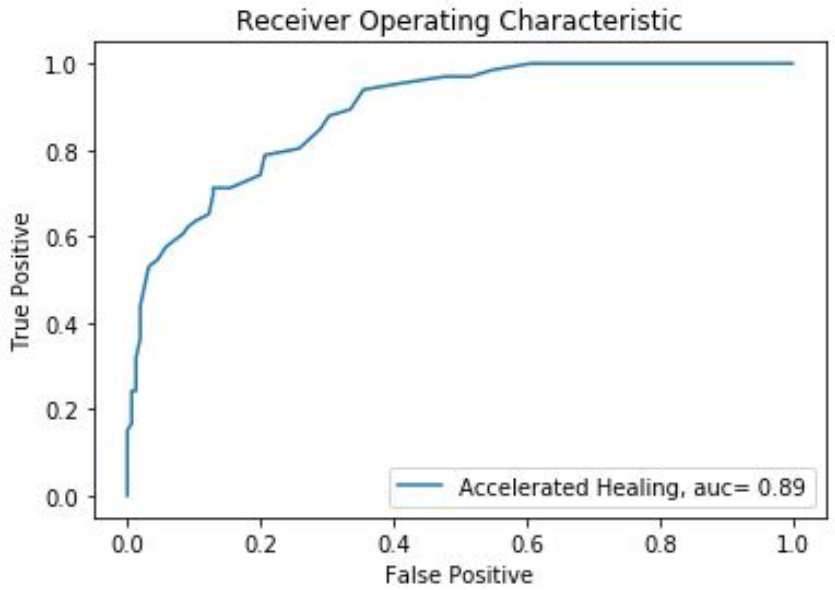
Validação

- Avaliar sua performance em diferentes métricas de desempenho

- Matriz de confusão
- Validação Cruzada
- Acurácia
- Recall
- Precision
- Curva ROC
- F1

$$ACC = \frac{VP + VN}{VP + FP + FN + VN}$$

$$F1 = \frac{2 * (PRE * REC)}{(PRE + REC)}$$



Fonte: Autor Próprio

Resultado e Discussão

- Comparação entre pré-processamentos
- Divisão treinamento e teste
- Classes
  - Alignment
  - Durability
  - Super Strength
  - Flight
  - Gender
  - Publisher
  - Accelerated Healing

ENSEMBLE LEARNING - KNN			
Ensemble	Voting Ensemble	Bagging	Ada-Boost
Score	86%	84%	84,10%

Evaluation - KNN preprocessing - SuperStrength			
Classifier	Naive Bayers	Decision Tree	Random Forest
Accurancy	73%	77,00%	81%
Score	79%	83,20%	85%
Recall	80,60%	87%	87%
Precision	81,20%	87,10%	87%
AUC	88%	89%	92%
F1	79%	84,60%	86%

Fonte: Autor Proprio

Comparação entre pré-processamentos

- Comparação entre pré-processamentos utilizando a classe *Alignment*

ENSEMBLE LEARNING - SuperV			
Ensemble	Voting Ensemble	Bagging	Ada-Boost
Score	69%	66,00%	66,00%

Avaliation - SuperV preprocessing - Alignment			
Classifier	Naive Bayers	Random Forest	Decision Tree
Accurancy	65,00%	68%	66.2%
Score	67,40%	67%	67,60%
Recall	62,40%	58%	53%
Precision	65,00%	64,40%	66,40%
AUC	67%	63%	56%
F1	67,40%	67%	66,00%

ENSEMBLE LEARNING - KNN			
Ensemble	Voting Ensemble	Bagging	Ada-Boost
Score	85%	81%	82,00%

Avaliation - KNN preprocessing - Alignment			
Classifier	Naive Bayers	Random Forest	Decision Tree
Accurancy	78%	84,00%	85,00%
Score	72%	82%	86,00%
Recall	60,80%	51%	58%
Precision	65,00%	59%	92,00%
AUC	62%	70%	54%
F1	72%	82%	83,00%

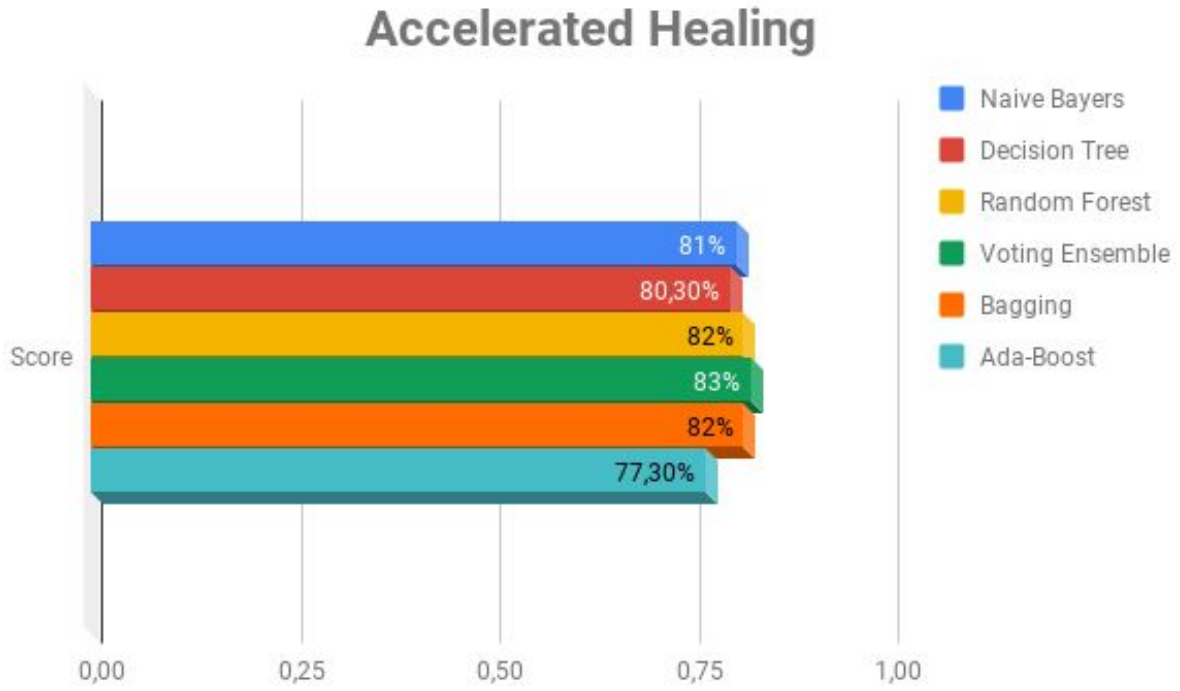
Fonte: Autor Próprio

Divisão entre treinamento e teste

- Divisão de 70% para treinamento e 30% para teste.
- Uso do *GridSearchCV* na Decision Tree para evitar *overfitting*.

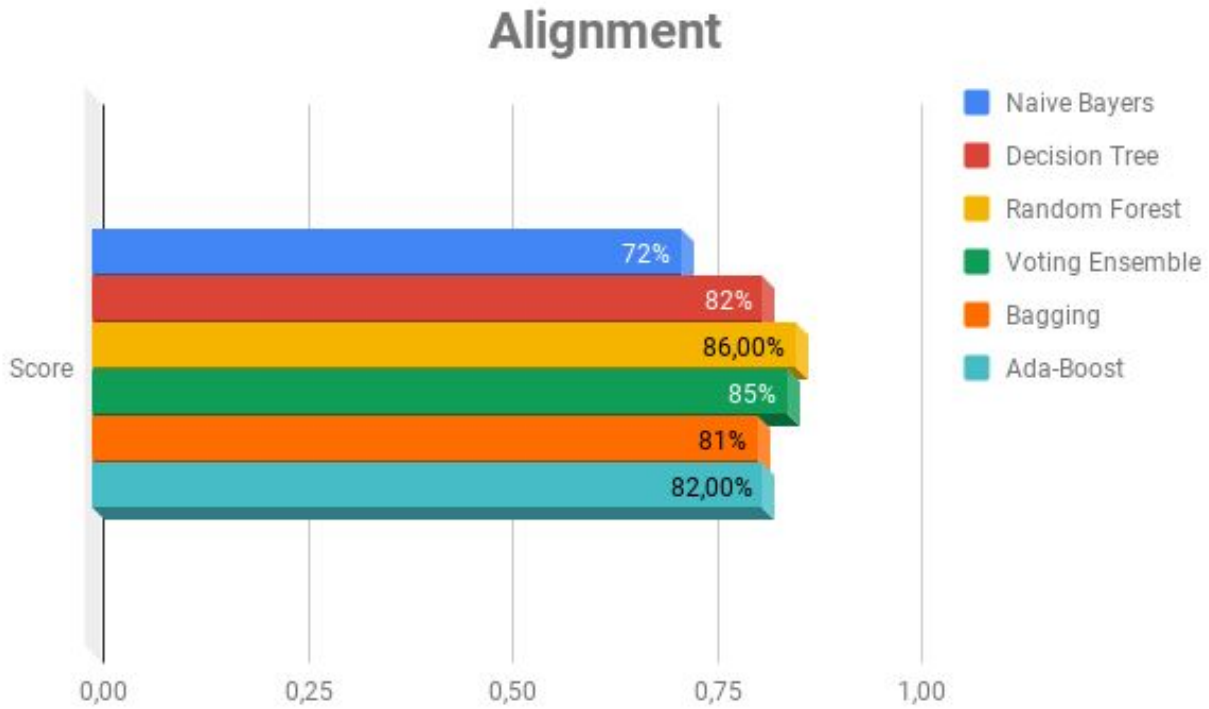
- Utilizou-se *cross_val_score* para gerar resultados.
- Parâmetros: 5 cross-validation e escore pela acurácia.

Fonte: Autor Próprio

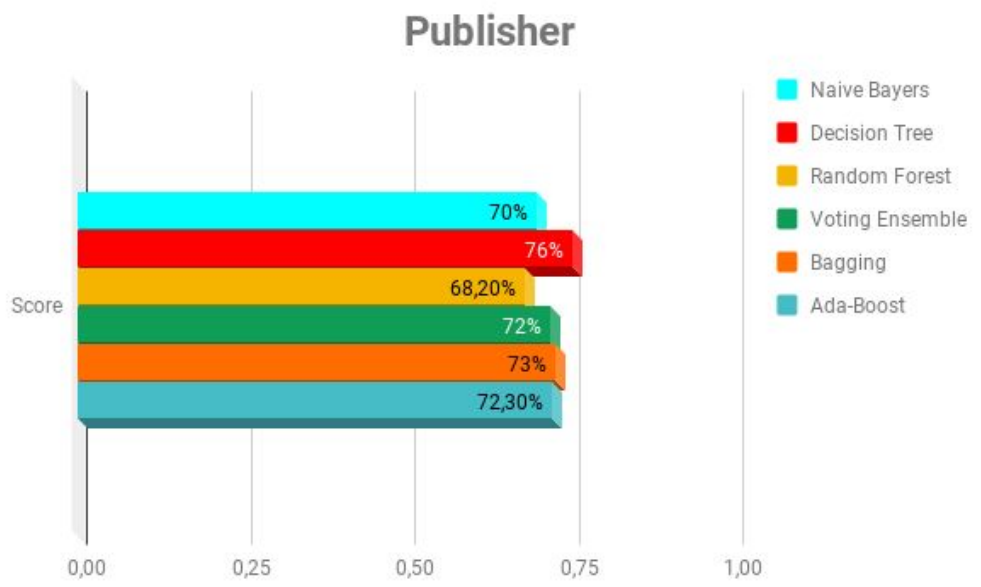


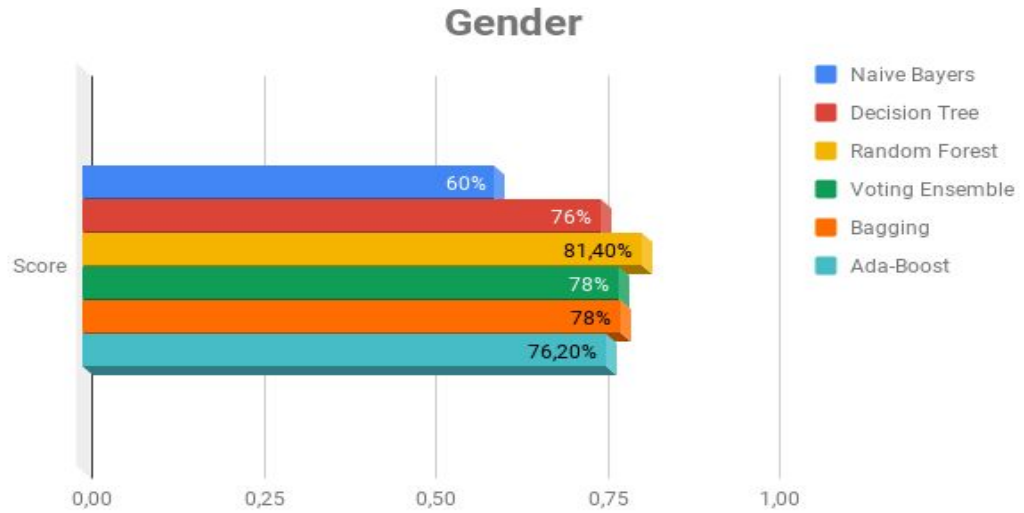
- Atingiu o objetivo proposto pela equipe.
- Todos os modelos de classificação, o atributo obteve valores maiores que 80% de acurácia
- Destaque para Voting Ensemble que se beneficia da boa acurácia dos outros modelos.

Fonte: Autor Próprio

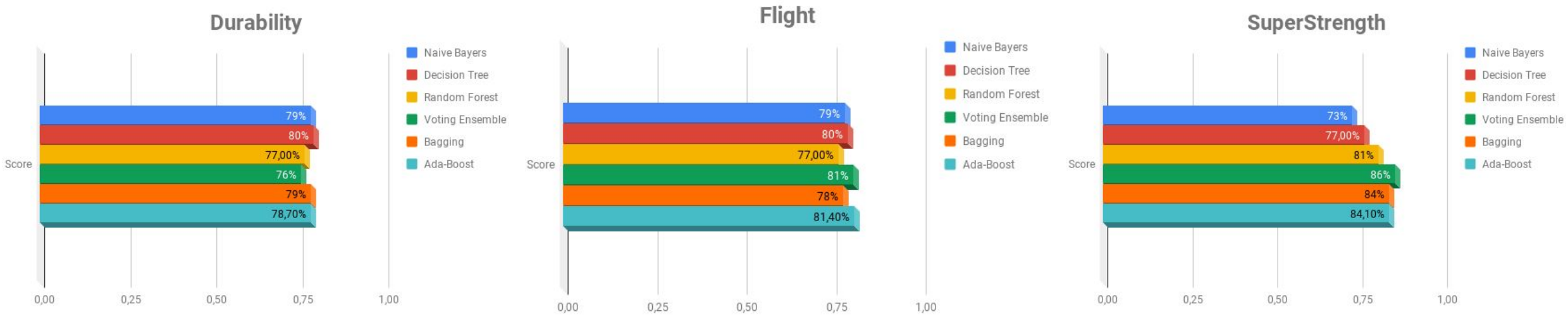


- Atingiu o objetivo proposto pela equipe.
- Eliminou-se a classe ”bad” transformando-o em atributo vago.
- Atributo binário, ”good”e ”neutral”.
- Com o uso de métodos de Ensemble, chega a uma acurácia de 85%.
- Destaque para o modelo Forest Random





- Ambos tiveram acurácia abaixo de 70% em dois modelos.
- *Gender* obteve 60% de acurácia no modelo Naive Bayes
- *Publisher*, teve baixa no Random Forest atingindo 68% de acurácia.



- Durability , SuperStrength e Flight que representam superpoderes, obtiveram taxas satisfatórias em relação a todos os modelos.
- Atinge o objetivo da equipe.

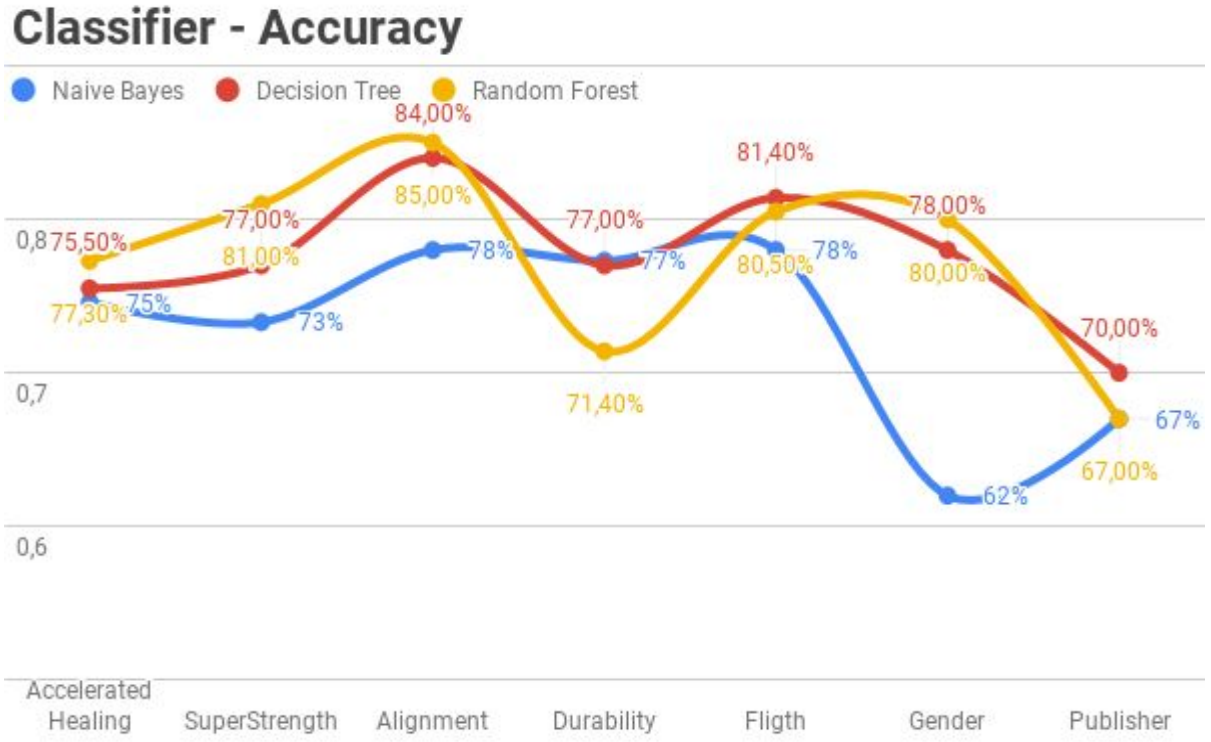
- Sobressaem-se nos métodos de Ensemble.
- Destaque sobre o Adaboost, quase se comparando com o Voting.

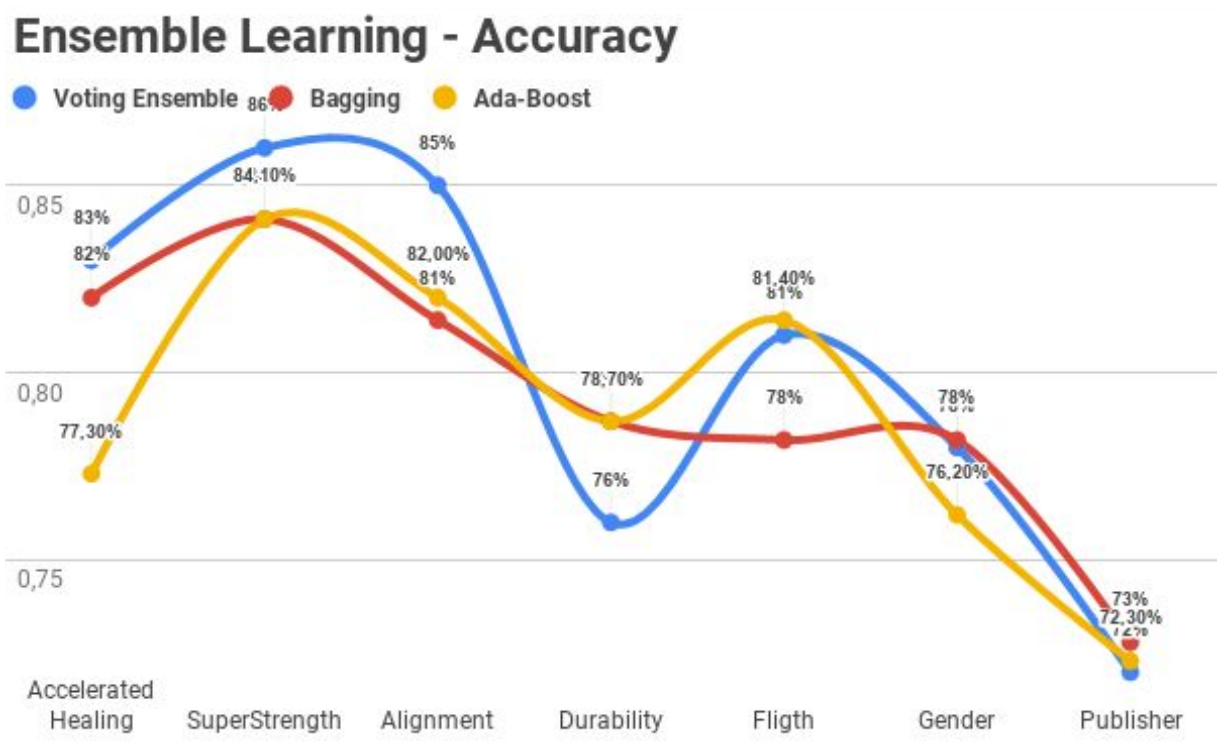
Conclusão

- O requisito de aplicação de múltiplos algoritmos de classificação e a execução de modelos de Ensemble Learning foi devidamente atendido, bem como o objetivo da equipe.
- Pré-processamento, os testes e análises das melhores configurações, auxiliaram a melhorar os resultados obtidos.
- Houve melhora significativa nos resultados utilizando o pré-processamento com KNN.

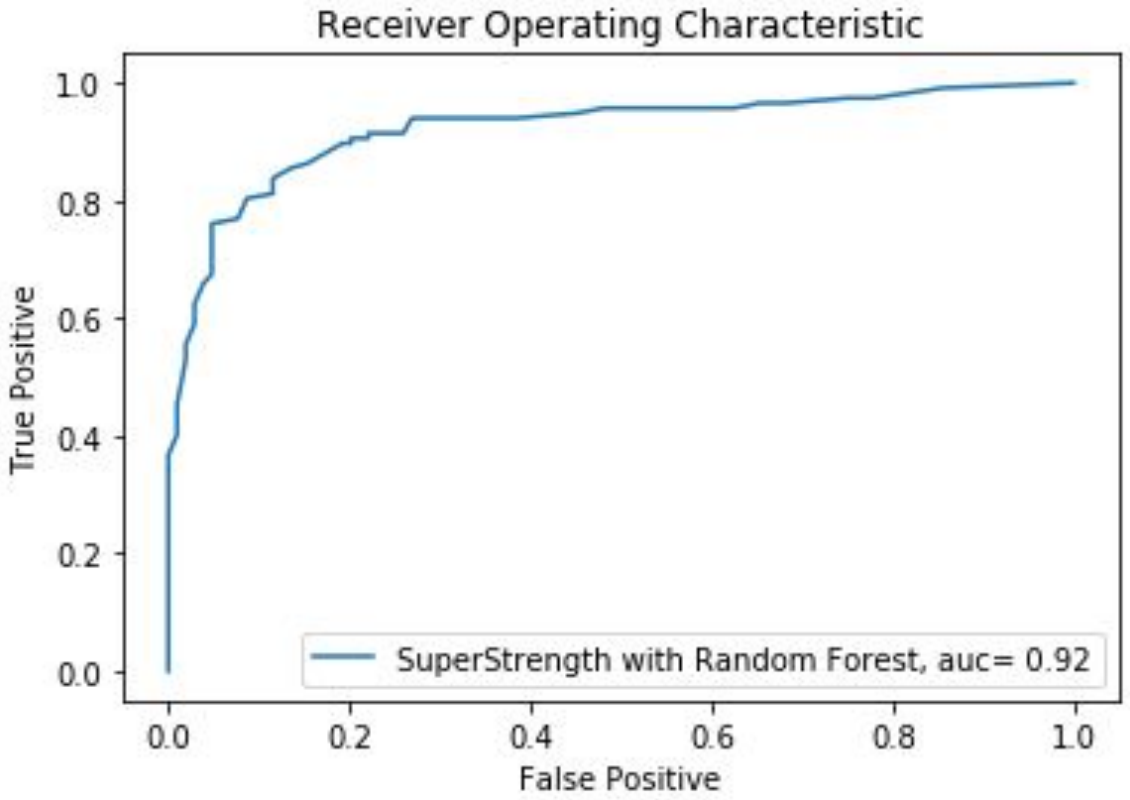
- Modelos de Ensemble melhoraram significativamente as predições dos classificadores.

Fonte: Autor Proprio





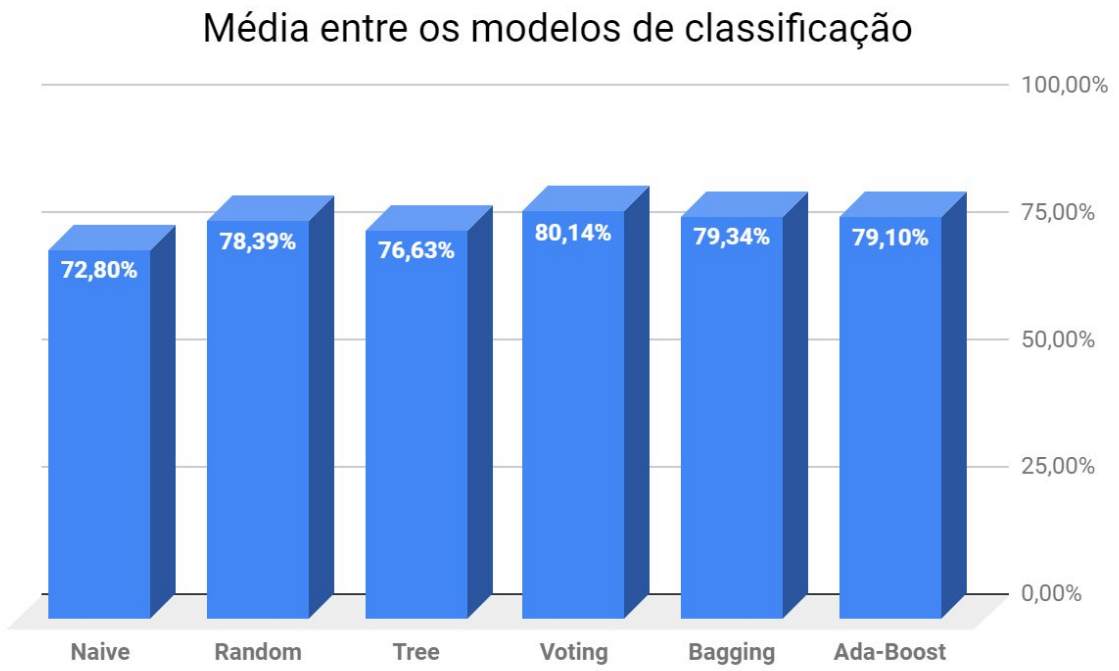
Fonte: Autor Próprio



- Melhor poder de discriminação (ROC) do classificador Random Forest no atributo *SuperStrength*.
- Melhor recall nos classificadores Random Forest e Decision Tree (Recall 87%), confirmando assim, a eficácia dos modelos classificatórios desenvolvidos.

Fonte: Autor Próprio

- Portanto, utilizando da média da acurácia entre os classificadores, podemos concluir a efetividade geral ao usar algoritmos Ensemble.



Fonte: Autor Próprio

Referências

- Freund, Y., e R. E. Schapire. “A short introduction to Boosting.” Journal of Japanese Society for Artificial Intelligence, Vol. 14, No5., 1999: 771 - 780.
- CHAVES, Bruno Butilhao. Estudo do algoritmo AdaBoost de aprendizagem de máquina aplicado a sensores e sistemas embarcados. Dissertação de Mestrado, Escola Politecnica, Universidade de São Paulo, 2011.
- Georgios Drakos. Handling Missing Values in Machine Learning: Part 2. Disponível em: <https://towardsdatascience.com/handling-missingvalues-in-machine-learning-part-2-222154b4b58e>. Acesso em Agosto de 2019.

- Documentação Pandas. Disponível em: <https://pandas.pydata.org>. Acesso em Julho de 2019.
- Documentação Numpy. Disponível em: <https://numpy.org>. Acesso em Julho de 2019.
- Documentação scikit-learn. Disponível em: <https://scikitlearn.org/stable/>. Acesso em Julho de 2019.
- Jones Granatyr. Machine Learning e Data Science com Python de A a Z. Disponível em: <https://www.udemy.com/machine-learning-e-datascience-com-python-y/>. Acesso em Agosto de 2019.
- Nicolas L. Gentille. Aprendizado de máquina e caracterização de aterosclerose subclínica: um estudo de caso. Disponíivel em: <http://cassiopea.ipt.br/teses/2017 EC Nicolas-Gentille.pdf>. Acesso em Julho de 2019.

- Documentação Python. Disponível em: <https://www.python.org/doc/>. Acesso em Julho de 2019.
- Babenko, D., e H. Marmanis. Algorithms of the Intelligence Web. Manning Publications, 2009.
- Bernardini, F. C. Combinac¸ao de classificação simbólicos para melhorar o poder preditivo e descritivo de ensembles. Dissertação de Mestrado, São Carlos: Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, 2002.
- Breiman, L. “Bagging Predictors.” Machine Learning 24, 1996: 123 - 140.