O presente projeto busca, através de algoritmos de classificação prever a preferência de sabor de um vinho e antecipadamente classificá-lo como ”bom” ou “ruim”. Para tal utilizei uma base de análises físico-químicas de vinhos da região de Porto, Portugal, disponibilizada pela Comissão de Viticultura da Região dos Vinhos Verdes em https://archive.ics.uci.edu/ml/datasets/Wine+Quality.

O projeto seguirá as fases do diagrama abaixo

Os passos que implementei são os seguintes:

1. Entender o problema e analisar o quadro geral
2. Obter os dados
3. Explorar os dados a fim de ter alguns *insights*
4. Preparar os dados para uma melhor exposição aos algoritmos
5. Avaliar os algoritmos de classificação e escolher os melhores para modelagem
6. Apresentação da proposta

1 Entender o problema e analisar o quadro geral

Antes de mais nada, precisei definir o objetivo em termos de negócios. Pois é preciso prever preferências de sabor de vinho baseadas em testes analíticos disponíveis na etapa de certificação do produto.

A abordagem que usei foi construir um sistema de aprendizado baseado em modelo de padrões detectado pelo classificador a fim de construir um modelo preditivo.

A primeira hipótese que assumi foi que a variação da qualidade de um vinho não são explicadas pelas características físico-químicas (H0). Contudo posso assumir que propriedades físico-químicas contribuem para a variação da qualidade e tornam um vinho "bom" ou "ruim" e vice-versa. (H1)

2 Capturar os dados

Como mencionado anteriormente, usei a base real de vinhos tintos da região de Vinho Verde, Portugal.

As variáveis de *input* baseadas nas análises físico-químicas:

1. fixed acidity
2. volatile acidity
3. citric acid
4. residual sugar
5. chlorides
6. free sulfur dioxide
7. total sulfur dioxide
8. density
9. pH
10. sulphates
11. alcohol

E a variável *output*, baseada nas pesquisas sensoriais:

12. quality (em uma escala de 0 a 10)

Após ler o dataset com o pandas, comecei a visualizar algumas características: A amostra é composta por 1599 linhas e 12 colunas, uma por variável. Usando alguns comandos do pandas podemos inspecionar o dataset com um resultado como o mostrado na figura abaixo:

```
RangeIndex: 1599 entries, 0 to 1598
Data columns (total 12 columns):
 #   Column                Non-Null Count  Dtype  
---  ------                --------------  -----  
 0   fixed acidity         1599 non-null   float64
 1   volatile acidity      1599 non-null   float64
 2   citric acid           1599 non-null   float64
 3   residual sugar        1599 non-null   float64
 4   chlorides             1599 non-null   float64
 5   free sulfur dioxide   1599 non-null   float64
 6   total sulfur dioxide  1599 non-null   float64
 7   density               1599 non-null   float64
 8   pH                    1599 non-null   float64
 9   sulphates             1599 non-null   float64
 10  alcohol               1599 non-null   float64
 11  quality               1599 non-null   int64  
dtypes: float64(11), int64(1)
memory usage: 150.0 KB
```

distribuição dos tipos de dados

Em seguida, após ter inspecionado a distribuição, segui visualizando uma amostra do conteúdo do dataset conforme imagem abaixo:

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	quality
0	7.4	0.70	0.00	1.9	0.076	11.0	34.0	0.9978	3.51	0.56	9.4	5
1	7.8	0.88	0.00	2.6	0.098	25.0	67.0	0.9968	3.20	0.68	9.8	5
2	7.8	0.76	0.04	2.3	0.092	15.0	54.0	0.9970	3.26	0.65	9.8	5
3	11.2	0.28	0.56	1.9	0.075	17.0	60.0	0.9980	3.16	0.58	9.8	6
4	7.4	0.70	0.00	1.9	0.076	11.0	34.0	0.9978	3.51	0.56	9.4	5
5	7.4	0.66	0.00	1.8	0.075	13.0	40.0	0.9978	3.51	0.56	9.4	5
6	7.9	0.60	0.06	1.6	0.069	15.0	59.0	0.9964	3.30	0.46	9.4	5
7	7.3	0.65	0.00	1.2	0.065	15.0	21.0	0.9946	3.39	0.47	10.0	7
8	7.8	0.58	0.02	2.0	0.073	9.0	18.0	0.9968	3.36	0.57	9.5	7
9	7.5	0.50	0.36	6.1	0.071	17.0	102.0	0.9978	3.35	0.80	10.5	5

amostra dos dados

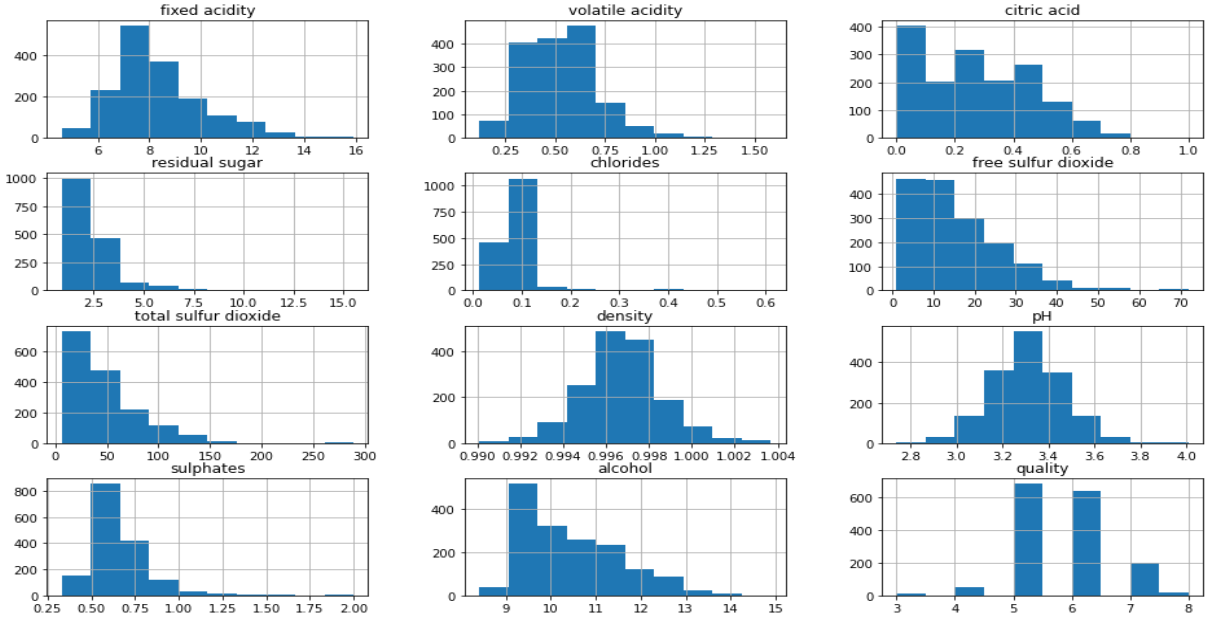
3 Exploração da amostra

A primeira coisa que precisei fazer é analisar a amostra, descrevendo sua distribuição, Informações importantes sobre a amostra podem ser obtidas, médias, desvio padrão e percentis da amostra podem nos dar insights valiosos. A visualização em tabela foi a seguinte:

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol
count	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000
mean	8.319637	0.527821	0.270976	2.538806	0.087467	15.874922	46.467792	0.996747	3.311113	0.658149	10.422983
std	1.741096	0.179060	0.194801	1.409928	0.047065	10.460157	32.895324	0.001887	0.154386	0.169507	1.065668
min	4.600000	0.120000	0.000000	0.900000	0.012000	1.000000	6.000000	0.990070	2.740000	0.330000	8.400000
25%	7.100000	0.390000	0.090000	1.900000	0.070000	7.000000	22.000000	0.995600	3.210000	0.550000	9.500000
50%	7.900000	0.520000	0.260000	2.200000	0.079000	14.000000	38.000000	0.996750	3.310000	0.620000	10.200000
75%	9.200000	0.640000	0.420000	2.600000	0.090000	21.000000	62.000000	0.997835	3.400000	0.730000	11.100000
max	15.900000	1.580000	1.000000	15.500000	0.611000	72.000000	289.000000	1.003690	4.010000	2.000000	14.900000

Visualização tabular

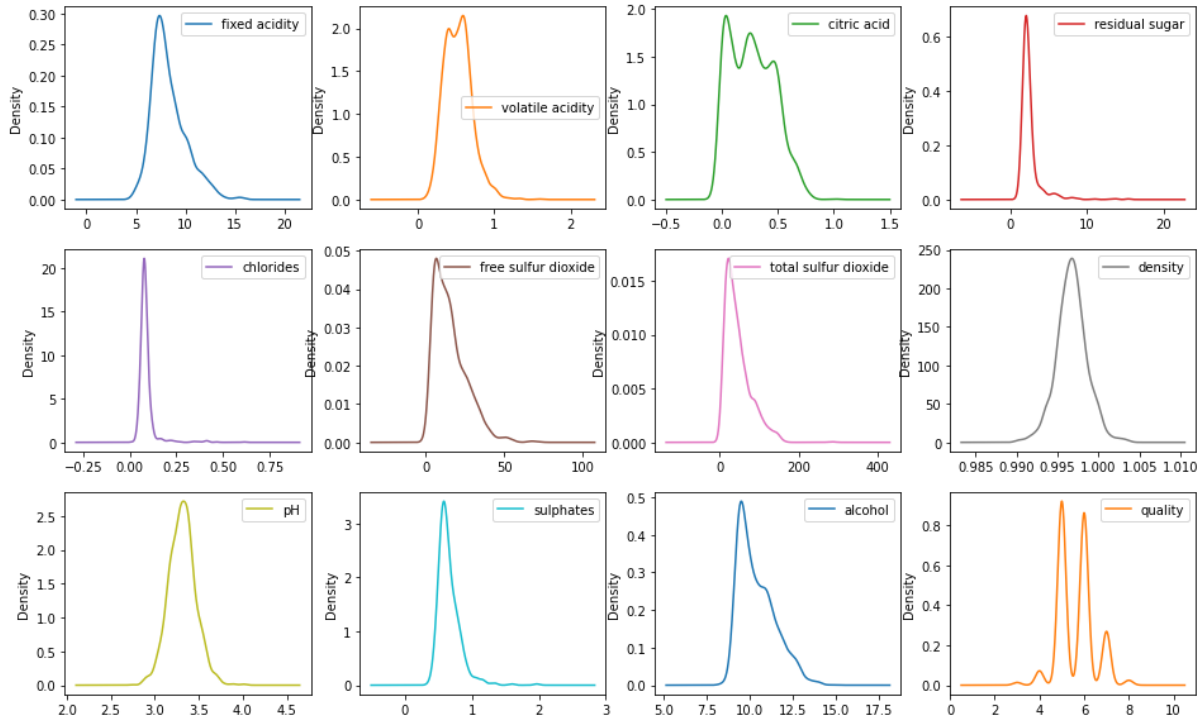
Já com alguns dados a visualização em histograma nos dá uma ideia melhor de como as variáveis estão distribuidas:



É interessante notar a distribuição é bem normalizada com a maioria tendo uma distorção positiva. Sobretudo a variável *alcohol* e *total sulfur dioxide*, podemos notar que as variáveis density e pH estão muito próximas a distribuição normal.

Outro detalhe importante é a variável *quality* que tem uma distribuição bimoda, ou seja, existem mais vinhos com qualidade média do que vinhos com qualidade totalmente “bom” ou totalmente “ruim”.

Essa visualização permite uma melhor visualização na análise da distribuição:



Seguindo a exploração, gostaria de ver como as variáveis se relacionam entre si, ou seja, como a variação de uma interfere na outra, precisei separar em 3 etapas. Gerar uma tabela dinâmica para analisar como as variáveis se distribuem em função da qualidade. Analisar a correlação e finalmente visualizar uma matriz como mapa de calor com o impacto de cada uma.

quality	alcohol	chlorides	citric acid	density	fixed acidity	free sulfur dioxide	pH	residual sugar	sulphates	total sulfur dioxide	volatile acidity
3	9.925	0.0905	0.035	0.997565	7.50	6.0	3.39	2.1	0.545	15.0	0.845
4	10.000	0.0800	0.090	0.996500	7.50	11.0	3.37	2.1	0.560	26.0	0.670
5	9.700	0.0810	0.230	0.997000	7.80	15.0	3.30	2.2	0.580	47.0	0.580
6	10.500	0.0780	0.260	0.996560	7.90	14.0	3.32	2.2	0.640	35.0	0.490
7	11.500	0.0730	0.400	0.995770	8.80	11.0	3.28	2.3	0.740	27.0	0.370
8	12.150	0.0705	0.420	0.994940	8.25	7.5	3.23	2.1	0.740	21.5	0.370

distribuição qualidade x variável

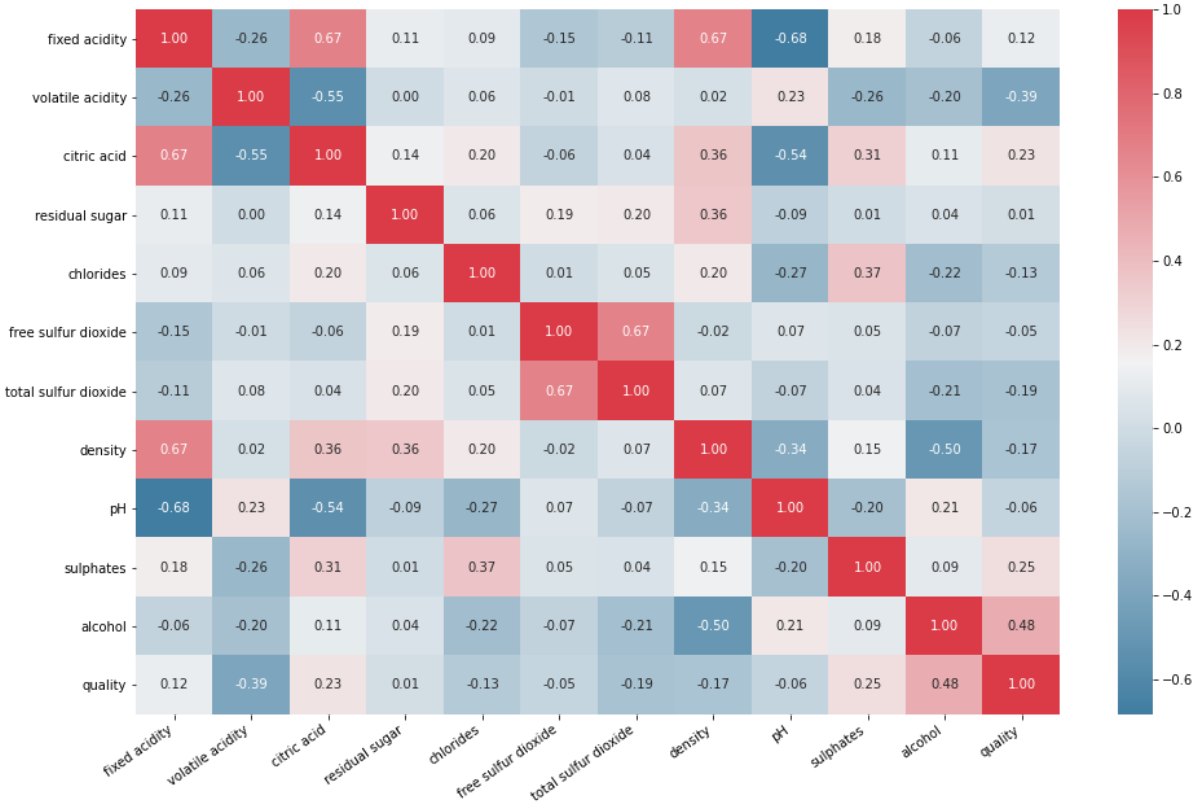
Para entender o quanto cada atributo se correlaciona com grau de qualidade do vinho, é preciso calcular o coeficiente de correlação padrão (também chamado de r de Pearson) entre cada par de atributos.

O pandas já fornece esse cálculo de graça, então o resultado do comando é como na figura a seguir:

```
quality                1.000000
alcohol                0.476166
sulphates              0.251397
citric acid            0.226373
fixed acidity          0.124052
residual sugar         0.013732
free sulfur dioxide   -0.050656
pH                    -0.057731
chlorides             -0.128907
density               -0.174919
total sulfur dioxide  -0.185100
volatile acidity      -0.390558
Name: quality, dtype: float64
```

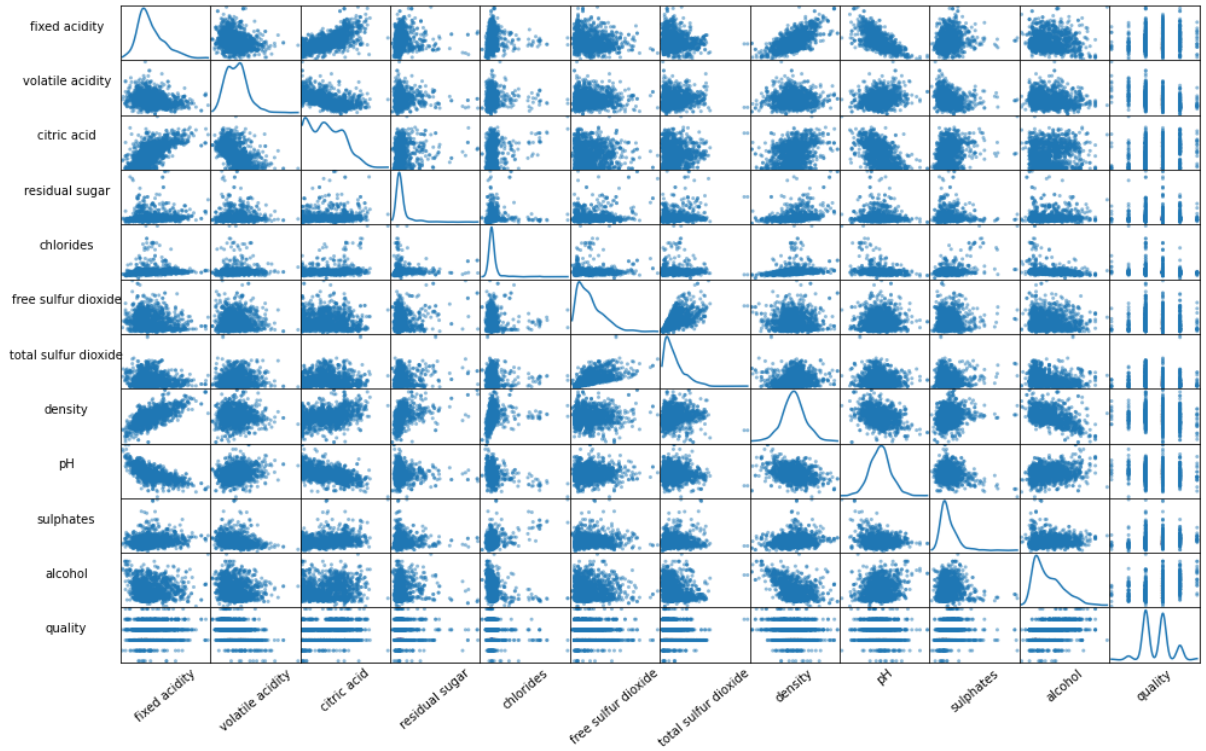
O coeficiente de relação segue uma escala que varia de -1 a 1. Quando mais perto de 1, mais positiva é essa relação. Por exemplo, o valor de qualidade tende aumentar conforme o valor de álcool aumenta. Ao passo que quanto mais próximo de -1 mais negativamente forte é a correlação. Podemos perceber uma pequena correlação negativa entre *quality* e *volatile acidity*. Por fim, quanto mais o valor fica próximo de 0 indica que não há correlação entre as variáveis.

De posse dessa tabela eu pude montar uma matriz de correlação, representada pelo grafico de mapa de calor abaixo:



Nele é possível perceber visualmente a correlação positiva para cada uma das variáveis entre si.

Por fim, decidi visualizar a dispersão de cada uma das variáveis entre si. Por sorte o pandas fornece essa matriz de dispersão. e o resultado foi o seguinte:



Interessante destacar a correlação positiva entre *density* e *fixed acidity* que têm um coeficiente de relação no valor de 0.67. O gráfico de dispersão comprova isso dada a tendência de alta e a pouca dispersão entre os pontos quando comparamos esses dois atributos.

4 Preparar os dados para uma melhor exposição aos algoritmos

Aqui precisei ajustar os valores de *quality* de maneira que pudesse fazer uma análise binária dos valores. alguns comandos do panda e do scikit learn me facilitaram isso e a amostra passou a ser representada assim:

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	quality
0	7.4	0.700	0.00	1.9	0.076	11.0	34.0	0.9978	3.51	0.56	9.4	ruim
1	7.8	0.880	0.00	2.6	0.098	25.0	67.0	0.9968	3.20	0.68	9.8	ruim
2	7.8	0.760	0.04	2.3	0.092	15.0	54.0	0.9970	3.26	0.65	9.8	ruim
3	11.2	0.280	0.56	1.9	0.075	17.0	60.0	0.9980	3.16	0.58	9.8	ruim
4	7.4	0.700	0.00	1.9	0.076	11.0	34.0	0.9978	3.51	0.56	9.4	ruim
5	7.4	0.660	0.00	1.8	0.075	13.0	40.0	0.9978	3.51	0.56	9.4	ruim
6	7.9	0.600	0.06	1.6	0.069	15.0	59.0	0.9964	3.30	0.46	9.4	ruim
7	7.3	0.650	0.00	1.2	0.065	15.0	21.0	0.9946	3.39	0.47	10.0	bom
8	7.8	0.580	0.02	2.0	0.073	9.0	18.0	0.9968	3.36	0.57	9.5	bom
9	7.5	0.500	0.36	6.1	0.071	17.0	102.0	0.9978	3.35	0.80	10.5	ruim

Essencialmente o que fiz foi dividir o intervalo de valores de *quality* em uma escala binária, afinal algoritmos de máquina entendem binário. O intervalo ia de 3 a 8, gerei uma saída mais perto de 8 como “bom” e mais perto de 2 como “ruim”. Ao passo que o valor 0 seria “ruim” e 1 seria “bom”. Após as transformações um total de 1382 foram classificados como “bom” e 217 com “ruim”.

5 Avaliar os algoritmos de classificação e escolher os melhores

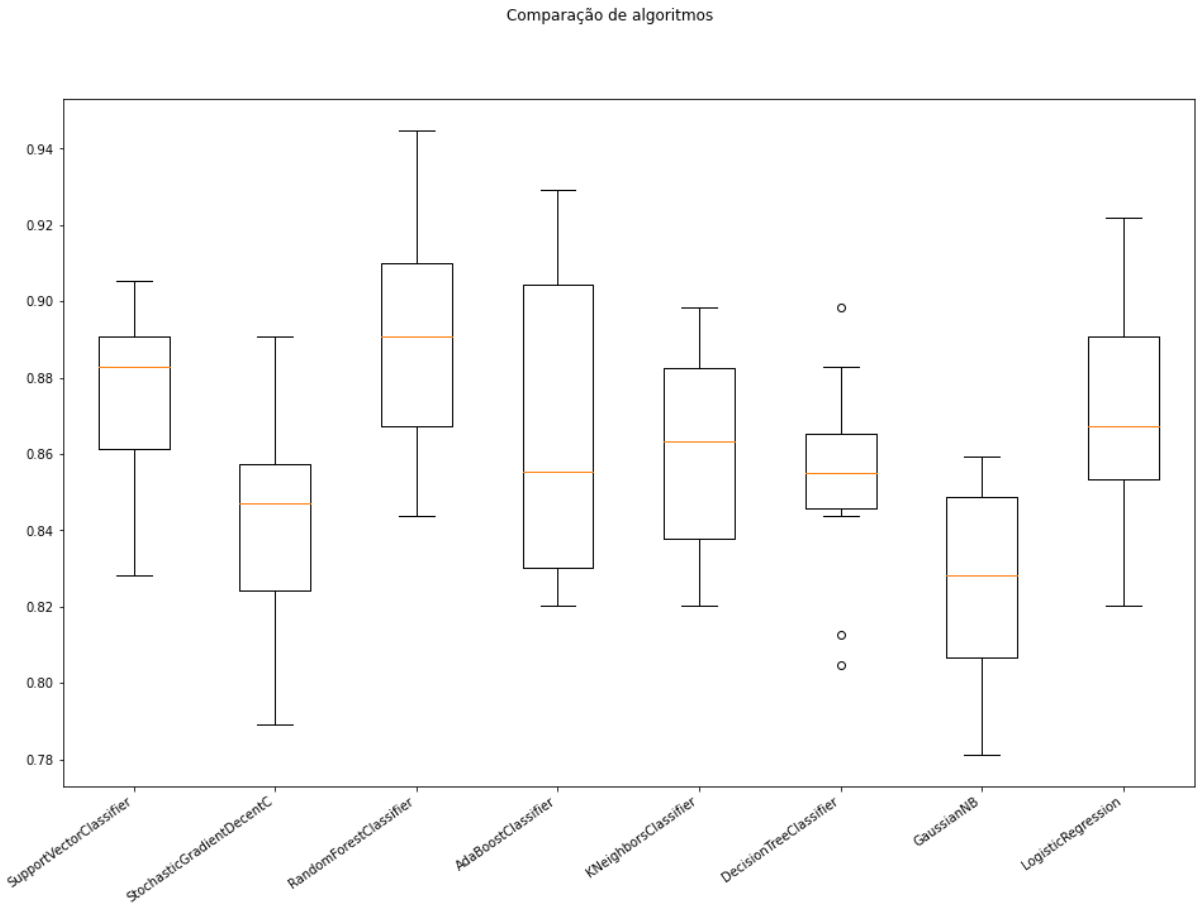
Um pouco do meu background como programador python me permitiu fazer um algoritmo dinâmico que calculasse a performance dos vários algoritmos de classificação disponibilizados pelo scikit learn. Dentre os quais escolhi:

- SupportVectorClassifier
- StochasticGradientDecentC
- RandomForestClassifier
- AdaBoostClassifier
- KNeighborsClassifier
- DecisionTreeClassifier
- GaussianNB
- Logistic Regression

Fiz um algoritmo que dinamicamente executasse cada um deles, passando 20% da amostra como base de treino, fazendo uma avaliação cruzada e agrupando o “score” com base na *accuracy* ou precisão; Ao final da execução o resultado foi o seguinte

algoritmo	**média**	**desvio padrão**
Support Vector Classifier	0.873364	0.024056
Stochastic Gradient Decent C	0.843633	0.029411
Random Forest Classifier	0.890582	0.028730
Ada Boost Classifier	0.866351	0.039970
K Neighbors Classifier	0.860845	0.026717
Decision Tree Classifier	0.853014	0.026995
GaussianNB	0.826446	0.025753
Logistic Regression	0.871014	0.028362

Aproveitei e gerei um gráfico de candlestick para visualizar melhor as médias e comparar cada algoritmo. O resultado foi a figura a seguir:



Depois treinar e executar cada um dos algoritmos pude avaliar melhor a performance resultando na seguinte tabela comparativa

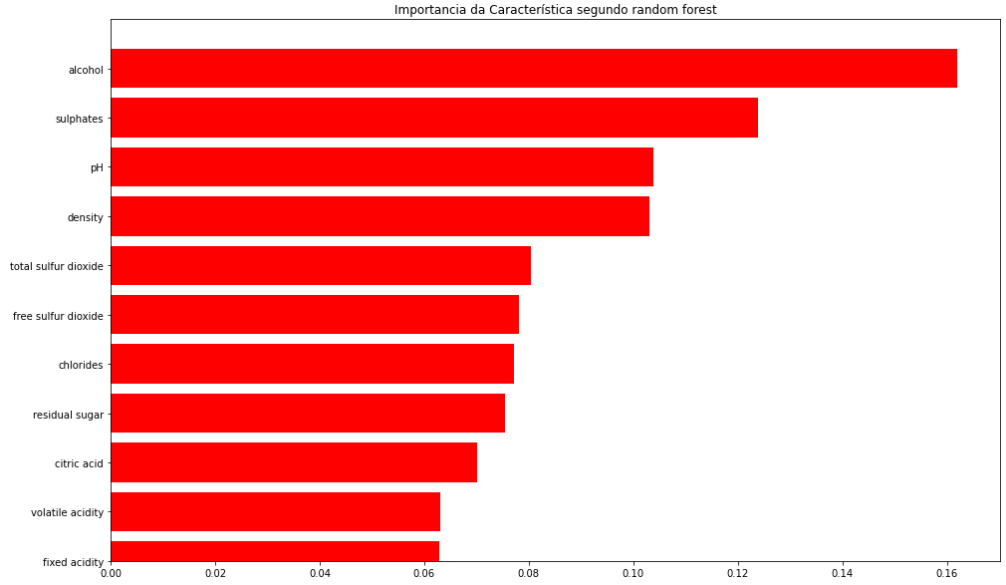
Algoritmo	**Média de erro absoluta**	**Precisão**	**Média de pontuação no Score**
svc 'default'	0.0781	92.19%	0.9
svc otimizado	0.0625	93.75%	0.91875
sgd	0.1156	88.44%	0.878125
random forest	0.0625	93.75%	0.91875
ada	0.0781	92.19%	0.896875

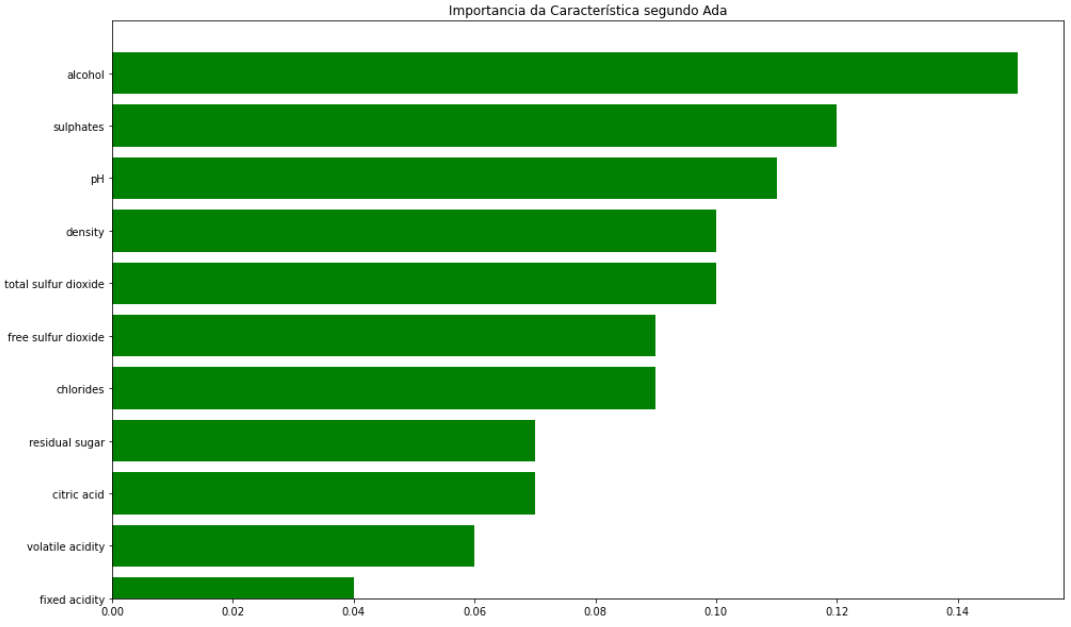
6 Apresentação da proposta

Dado o estudo das características, os *insights* na fase de exploração e a avaliação dos algoritmos, escolhi representar a proposta de importância da característica entre os 2 melhores, random forest e ada.

Escolhi ambos pois eles são os únicos em que pude captar a importância das variáveis no processo de avaliação através do atributo interno do modelo dentro do scikit learn chamado *feature_importances_*.

Fica comprovada que a relação positiva entre a *quality* e o *alcohol* é o principal atributo a ser considerado para prever a qualidade dos vinhos tintos. A importância entre as variáveis e pode ser representada para cada um dos algoritmos conforme os gráficos a seguir:





7 Avaliação e possíveis usos da predição

Como treinamos e testamos os algoritmos usando um grau de precisão de 90%, podemos prever, dada novas amostras, a qualidade em que o novo vinho virá a se enquadrar apenas fazendo a relação de sua graduação alcoólica.

O valor previsto pode ser usado para projetar novos tipos de vinho, definir a política de preços ou apoiar a tomada de decisões em sistemas de consultoria ou rotação de estoque por exemplo.