Usando análises químicas, determine a origem dos vinhos

Esses dados são o resultado de uma análise química de vinhos cultivados na mesma região da Itália, mas derivados de três cultivares diferentes. A análise determinou as quantidades de 13 constituintes encontradas em cada um dos três tipos de vinhos.

Classe: Se refere a classificação do Vinho.

Álcool: O álcool no vinho é resultado da fermentação do açúcar presente nas uvas. Ele afeta o sabor, a textura e o corpo do vinho, bem como tem um efeito psicoativo.

Ácido málico: Um dos ácidos presentes nas uvas. Seu nível afeta a acidez do vinho, contribuindo para o sabor fresco e a sensação de boca.

Cinza: Também conhecida como "minerais", a cinza refere-se a uma variedade de minerais presentes no vinho que podem afetar sua estrutura e sabor.

Alcalinidade das cinzas: Mede a quantidade de minerais alcalinos na cinza do vinho, o que pode influenciar a sensação de boca.

Magnésio: Um dos minerais presentes no vinho, que pode ter efeitos sutis no sabor e na textura.

Fenóis totais: Um grupo de compostos antioxidantes que afetam a cor, sabor e estrutura do vinho, contribuindo para o seu envelhecimento.

Flavanoides: Subgrupo de fenóis que contribuem para a cor e os sabores do vinho, além de terem propriedades antioxidantes.

Fenóis não flavonoides: Outro grupo de fenóis que podem ter efeitos sobre o sabor, a textura e a longevidade do vinho.

Proantocianinas: São uma classe específica de flavonoides que contribuem para a cor e para a adstringência do vinho.

Intensidade da cor: Mede a profundidade da cor do vinho, o que pode indicar o tipo de uvas utilizadas e o potencial de envelhecimento.

Tonalidade: Refere-se à tonalidade da cor do vinho, que pode fornecer informações sobre seu estágio de maturidade e variedade de uva.

OD280/OD315 dos vinhos diluídos: Essa relação de absorbância de luz em diferentes comprimentos de onda pode ser usada para estimar a concentração de compostos específicos, como proteínas e fenóis.

Prolina: Um aminoácido que pode ser usado como indicador da maturidade das uvas e da quantidade de nitrogênio no mosto, que afeta o desenvolvimento das leveduras na fermentação.

Primeira etapa será de importação da biblioteca Pandas e da base de dados dos vinhos.

In [2]:
```
import pandas as pd
df = pd.read_csv(r"C:\Users\Dell\Desktop\Ciencia de Dados\Projetos\wine.data")
columns = ['class', 'alcool', 'acido_malico','cinzas', 'alcalinidade_das_cinzas','magnesio','total_fenol', 'flavenoides', 'Nãoflavanóides_f
df.columns = columns
```

Visualizando o tamanho do Dataframe junto com as 5 primeiras e 5 ultimas linhas.

In [3]:
```
print(f'Linhas: {df.shape[0]} \nColunas: {df.shape[1]}')
display(df.head())
display(df.tail())
```

```
Linhas: 177
Colunas: 14
```

	class	alcool	acido_malico	cinzas	alcalinidade_das_cinzas	magnesio	total_fenol	flavenoides	Nãoflavanóides_fenóis	proantocianidinas	Intensidade_cor	cor	dil
0	1	13.20	1.78	2.14	11.2	100	2.65	2.76	0.26	1.28	4.38	1.05	
1	1	13.16	2.36	2.67	18.6	101	2.80	3.24	0.30	2.81	5.68	1.03	
2	1	14.37	1.95	2.50	16.8	113	3.85	3.49	0.24	2.18	7.80	0.86	
3	1	13.24	2.59	2.87	21.0	118	2.80	2.69	0.39	1.82	4.32	1.04	
4	1	14.20	1.76	2.45	15.2	112	3.27	3.39	0.34	1.97	6.75	1.05	

	class	alcool	acido_malico	cinzas	alcalinidade_das_cinzas	magnesio	total_fenol	flavenoides	Nãoflavanóides_fenóis	proantocianidinas	Intensidade_cor	cor	
172	3	13.71	5.65	2.45	20.5	95	1.68	0.61	0.52	1.06	7.7	0.64	
173	3	13.40	3.91	2.48	23.0	102	1.80	0.75	0.43	1.41	7.3	0.70	
174	3	13.27	4.28	2.26	20.0	120	1.59	0.69	0.43	1.35	10.2	0.59	
175	3	13.17	2.59	2.37	20.0	120	1.65	0.68	0.53	1.46	9.3	0.60	
176	3	14.13	4.10	2.74	24.5	96	2.05	0.76	0.56	1.35	9.2	0.61	

A seguir é possivel verificar o tipo de dados de cada coluna e quantidade de valores nulos.

A base de dados possui um total de 177 linhas e nenhum valor nulo.

In [4]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 177 entries, 0 to 176
Data columns (total 14 columns):
 #   Column                   Non-Null Count  Dtype
---  ------                   --------------  -----
 0   class                    177 non-null    int64
 1   alcool                   177 non-null    float64
 2   acido_malico             177 non-null    float64
 3   cinzas                   177 non-null    float64
 4   alcalinidade_das_cinzas  177 non-null    float64
 5   magnesio                 177 non-null    int64
 6   total_fenol              177 non-null    float64
 7   flavenoides              177 non-null    float64
 8   Nãoflavanóides_fenóis    177 non-null    float64
 9   proantocianidinas        177 non-null    float64
 10  Intensidade_cor          177 non-null    float64
 11  cor                      177 non-null    float64
 12  diluicao                 177 non-null    float64
 13  prolina                  177 non-null    int64
dtypes: float64(11), int64(3)
memory usage: 19.5 KB
```

Visualizando dados estatísticos

In [5]:
```
df.describe()
```

Out[5]:

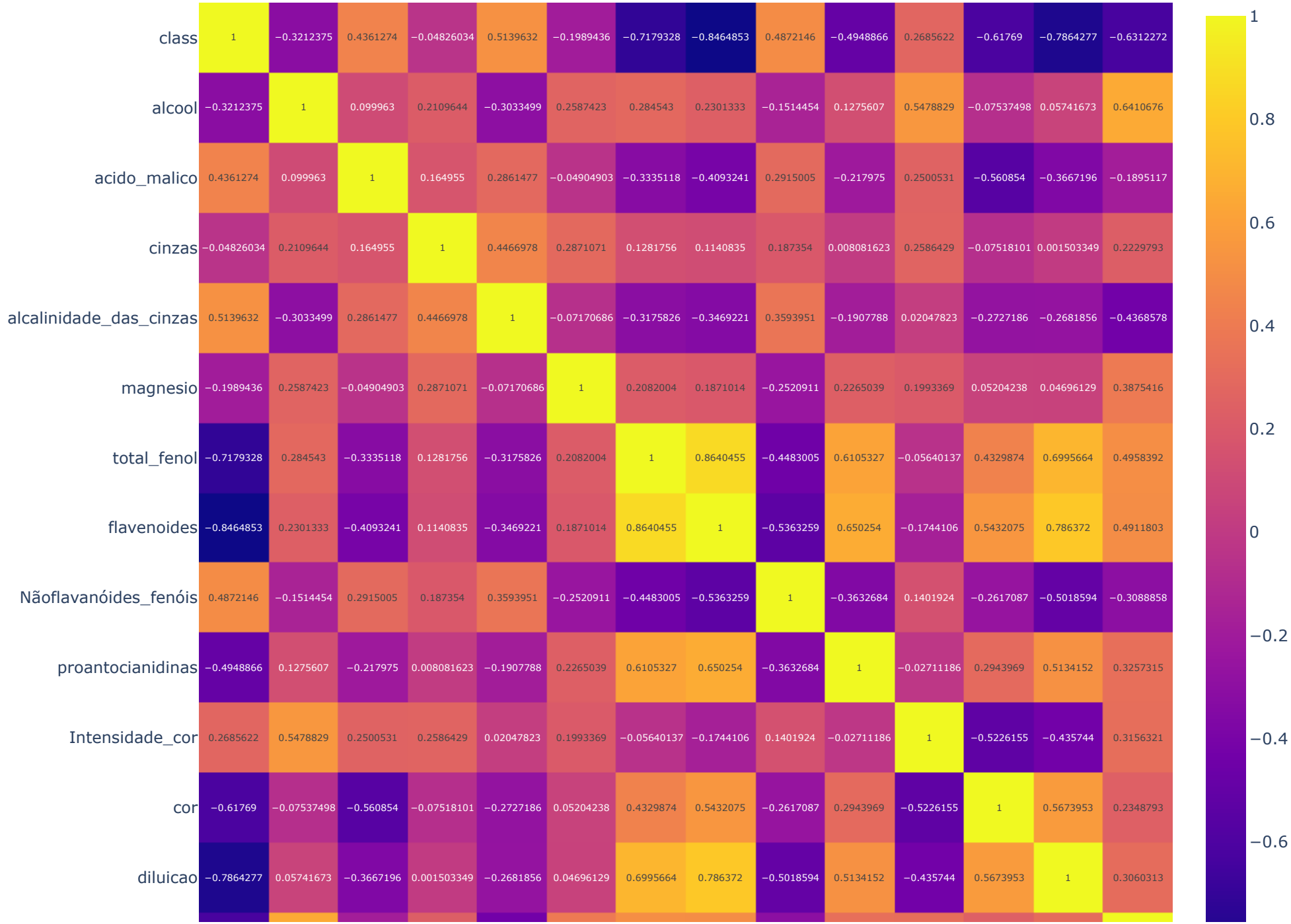
	class	alcool	acido_malico	cinzas	alcalinidade_das_cinzas	magnesio	total_fenol	flavenoides	Nãoflavanóides_fenóis	proantocianidinas	Inten
count	177.000000	177.000000	177.000000	177.000000	177.000000	177.000000	177.000000	177.000000	177.000000	177.000000	
mean	1.943503	12.993672	2.339887	2.366158	19.516949	99.587571	2.292260	2.023446	0.362316	1.586949	
std	0.773991	0.808808	1.119314	0.275080	3.336071	14.174018	0.626465	0.998658	0.124653	0.571545	
min	1.000000	11.030000	0.740000	1.360000	10.600000	70.000000	0.980000	0.340000	0.130000	0.410000	
25%	1.000000	12.360000	1.600000	2.210000	17.200000	88.000000	1.740000	1.200000	0.270000	1.250000	
50%	2.000000	13.050000	1.870000	2.360000	19.500000	98.000000	2.350000	2.130000	0.340000	1.550000	
75%	3.000000	13.670000	3.100000	2.560000	21.500000	107.000000	2.800000	2.860000	0.440000	1.950000	
max	3.000000	14.830000	5.800000	3.230000	30.000000	162.000000	3.880000	5.080000	0.660000	3.580000	

Como vimos anteriormente, nosso Datraframe possui um total de 14 colunas, é preciso descobrir quais valores serão importantes para o modelo.

A seguir foi plotado um mapa de calor com a correlação de cada variavel.

In [6]:
```
import plotly.express as px

fig = px.imshow(df.corr(), text_auto= True , aspect='auto')
fig.update_layout( width=1000, height = 900)
fig.show()
```


class 1 −0.3212375 0.4361274 −0.04826034 0.5139632 −0.1989436 −0.7179328 −0.8464853 0.4872146 −0.4948866 0.2685622 −0.61769 −0.7864277 −0.6312272
alcool −0.3212375 1 0.099963 0.2109644 −0.3033499 0.2587423 0.284543 0.2301333 −0.1514454 0.1275607 0.5478829 −0.07537498 0.05741673 0.6410676
acido_malico 0.4361274 0.099963 1 0.164955 0.2861477 −0.04904903 −0.3335118 −0.4093241 0.2915005 −0.217975 0.2500531 −0.560854 −0.3667196 −0.1895117
cinzas −0.04826034 0.2109644 0.164955 1 0.4466978 0.2871071 0.1281756 0.1140835 0.187354 0.008081623 0.2586429 −0.07518101 0.001503349 0.2229793
alcalinidade_das_cinzas 0.5139632 −0.3033499 0.2861477 0.4466978 1 −0.07170686 −0.3175826 −0.3469221 0.3593951 −0.1907788 0.02047823 −0.2727186 −0.2681856 −0.4368578
magnesio −0.1989436 0.2587423 −0.04904903 0.2871071 −0.07170686 1 0.2082004 0.1871014 −0.2520911 0.2265039 0.1993369 0.05204238 0.04696129 0.3875416
total_fenol −0.7179328 0.284543 −0.3335118 0.1281756 −0.3175826 0.2082004 1 0.8640455 −0.4483005 0.6105327 −0.05640137 0.4329874 0.6995664 0.4958392
flavenoides −0.8464853 0.2301333 −0.4093241 0.1140835 −0.3469221 0.1871014 0.8640455 1 −0.5363259 0.650254 −0.1744106 0.5432075 0.786372 0.4911803
Nãoflavanóides_fenóis 0.4872146 −0.1514454 0.2915005 0.187354 0.3593951 −0.2520911 −0.4483005 −0.5363259 1 −0.3632684 0.1401924 −0.2617087 −0.5018594 −0.3088858
proantocianidinas −0.4948866 0.1275607 −0.217975 0.008081623 −0.1907788 0.2265039 0.6105327 0.650254 −0.3632684 1 −0.02711186 0.2943969 0.5134152 0.3257315
Intensidade_cor 0.2685622 0.5478829 0.2500531 0.2586429 0.02047823 0.1993369 −0.05640137 −0.1744106 0.1401924 −0.02711186 1 −0.5226155 −0.435744 0.3156321
cor −0.61769 −0.07537498 −0.560854 −0.07518101 −0.2727186 0.05204238 0.4329874 0.5432075 −0.2617087 0.2943969 −0.5226155 1 0.5673953 0.2348793
diluicao −0.7864277 0.05741673 −0.3667196 0.001503349 −0.2681856 0.04696129 0.6995664 0.786372 −0.5018594 0.5134152 −0.435744 0.5673953 1 0.3060313
1
0.8
0.6
0.4
0.2
0
−0.2
−0.4
−0.6

Para o modelo iremos considerar apenas as colunas que possui um grau de correlação acima de 0.5 (Positivo e Negativo)

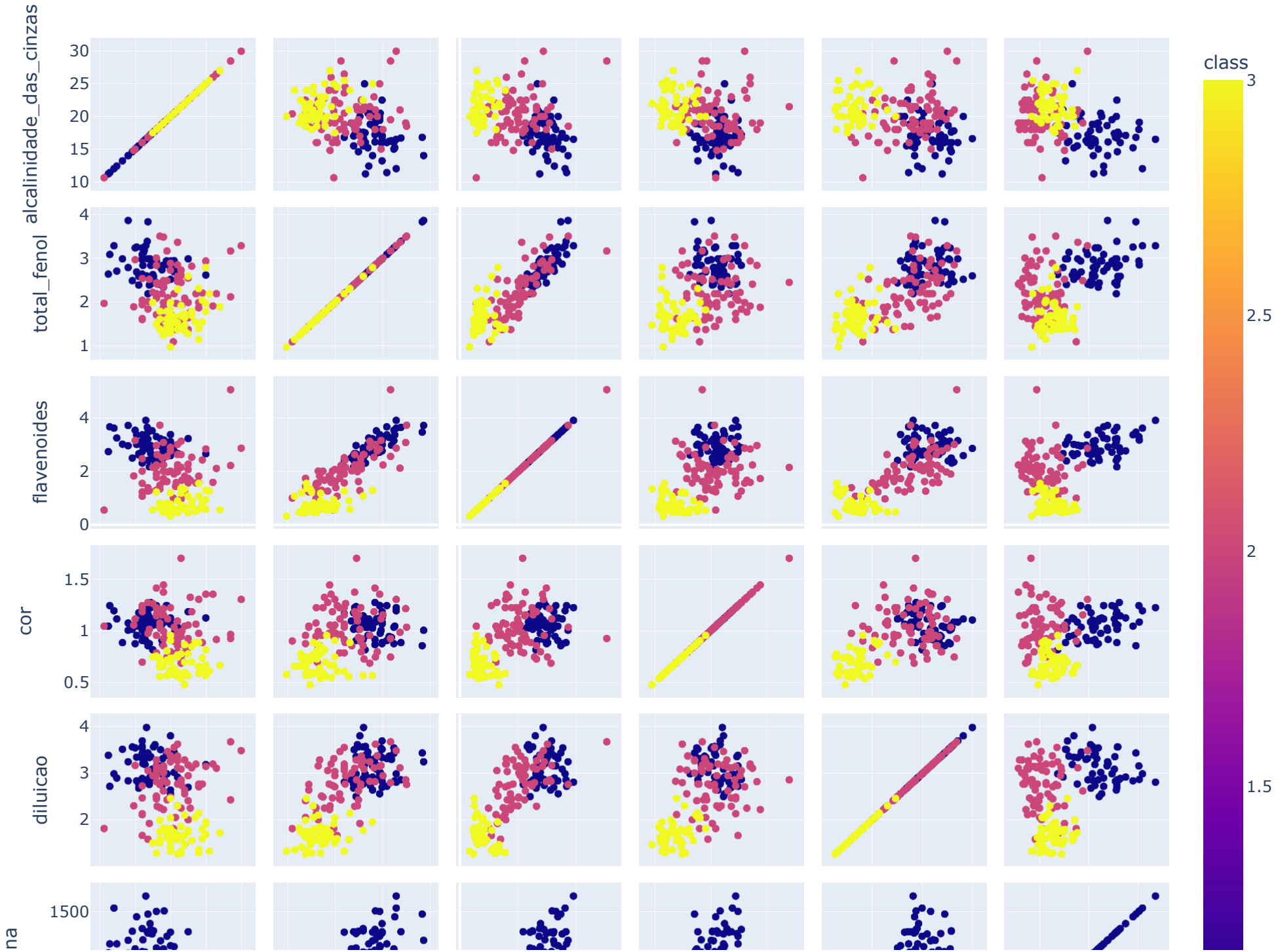
In [7]:
```
corr = df.corr()
top_features = corr[abs(corr['class']) > 0.5].index # Considerado valores absolutos (+, -)
dados = df[top_features]
```

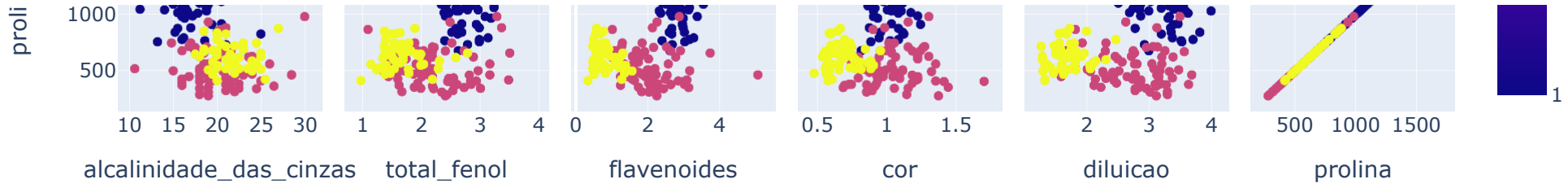
Imprimindo um gráfico de disperção para melhor visualização

In [8]:
```
dimensoes_excluidas = [coluna for coluna in dados.columns if coluna != "class"] # oculta a coluna class
fig = px.scatter_matrix(dados, color='class', dimensions= dimensoes_excluidas)
fig.update_layout( width=1000, height = 900)
fig.show()
```

```
C:\Users\Dell\anaconda3\lib\site-packages\plotly\express\_core.py:279: FutureWarning:

iteritems is deprecated and will be removed in a future version. Use .items instead.
```


alcalinidade_das_cinzas
total_fenol
flavenoides
cor
diluicao
na
class
30
25
20
15
10
4
3
2
1
0
1.5
0.5
1500
2.5



In [9]:
```
x = dados.drop('class', axis=1)
y = dados['class']
```

In [10]:
```
from sklearn.model_selection import train_test_split

x_train, x_test, y_train, y_test = train_test_split(x, y , test_size=0.33 , random_state=42)
```

In [11]:
```
from sklearn.naive_bayes import GaussianNB

clf = GaussianNB()
clf.fit(x_train, y_train)
```

Out[11]:
```
▾ GaussianNB
GaussianNB()
```

In [12]:
```
clf.score(x_test, y_test)
```

Out[12]:
```
0.9322033898305084
```

In [14]:
```
import numpy as np
def modelo_previsao(alcalinidade_das_cinzas, total_fenol, flavenoides, cor, diluicao, prolina):
    x = [alcalinidade_das_cinzas, total_fenol, flavenoides, cor, diluicao, prolina]
    return clf.predict([x])[0]
```

In [ ]:
```
df.rename(columns={'class': 'y_true'}, inplace=True)
df['y_predict'] = df.apply(lambda x:modelo_previsao(x['alcalinidade_das_cinzas'],x['total_fenol'], x['flavenoides'],x['cor'], x['diluicao']
```

In [16]:
```
from sklearn.metrics import confusion_matrix
y_true = df['y_true']
y_predict = df['y_predict']

confusion_matrix(y_true , y_predict, labels=[1,2,3])
```

Out[16]:
```
array([[56,  2,  0],
       [ 4, 66,  1],
       [ 0,  0, 48]], dtype=int64)
```

- V1 = 56 F1 = 2 F1 = 0
- F2 = 4 V2 = 66 F2 = 1
- F3 = 0 F3 = 0 V3 = 48

In [17]:
```
# Acuracidade do modelo
from sklearn.metrics import accuracy_score
accuracy_score(y_true , y_predict)
```

Out[17]:
```
0.96045197740113
```

In [18]:
```
# Precisao do modelo
from sklearn.metrics import precision_score

precision_score(y_true , y_predict, average= None, labels=[1,2,3])
```

Out[18]:
```
array([0.93333333, 0.97058824, 0.97959184])
```

In [19]:
```
teste = df[['y_predict','y_true']]
teste = teste[teste['y_predict'] != 2]
display(teste.loc[teste['y_true'] == 2])
```

	y_predict	y_true
65	1	2
69	3	2
73	1	2
94	1	2
108	1	2

In [20]:
```
# recall do modelo

from sklearn.metrics import recall_score

recall_score(y_true, y_predict, average=None, labels=[1,2,3])
```

Out[20]: array([0.96551724, 0.92957746, 1.   ])