

## FLASK SERVER/ DASH CODE \*Highlighted Code is not my own\*

```
1 # Imports needed to run Dash, Flask and plotly, along with pandas to read in the /csv files
2 import dash
3 import dash_core_components as dcc
4 import dash_html_components as html
5 import pandas as pd
6 import plotly.graph_objs as go
7
8 # Read in CSV file
9 df = pd.read_csv('./Data/Comparison.csv')
10
11 # External Stylesheet was used for this Prototype as it worked well with this layout as well as it saving time.
12 external_stylesheets = ['https://codepen.io/chriddyp/pen/bWLwgP.css']
13
14 # Declare the server type
15 app = dash.Dash(__name__, external_stylesheets=external_stylesheets)
16
17 # This section is the beginning of the HTML section. Dash incorporates this in through Python code.
18 app.layout = html.Div(children=[
19     html.H1(children='Wine Review Analysis'),
20
21     html.Div(children='''
22         Welcome to my Final Year Project on Wine Reviews Analysis.
23     '''),
24     html.Div(children='''
25         The first visualisation is the actual wine prices in relation to the age and rating of a wine. Ratings in the world of wine tasters and reviews is referred to as
26     '''),
27     # First Visualisation Graph below.
28     dcc.Graph(
29         id='Real Wine Reviews',
30         figure={
31             'data': [
32                 go.Scatter(
33                     x=df[df['price'] == 1]['Age'], # This section puts the Age column as the X axis and the points column as the Y axis.
34                     y=df[df['price'] == 1]['points'], # The price then shows in relation to both axis columns
35                     mode='markers',
36                     opacity=0.7, # Declaring opacity and marker styling parameters
37                     marker={
38                         'size': 15,
39                         'line': {'width': 0.5, 'color': 'white'}
40                     },
41                     name=1
42                 ) for i in df.price[10:] # This for loop will show the first 10 prices in the dataset.
43             ],
44             # This section is the layout for the graphs.
45             'layout': go.Layout(
46                 xaxis={'type': 'log', 'title': 'Age'},
47                 yaxis={'title': 'points'},
48                 margin={'l': 40, 'b': 40, 't': 10, 'r': 10},
49                 legend={'x': 0, 'y': 1},
50                 hovermode='closest'
51             )
52         ),
53     ),
54     # Below is the second graph, same information as above.
55     html.Div(children='''
56         The next visualisation is the corresponding KNN Predictions.
57     '''),
58     dcc.Graph(
59         id='Predicted Wine Reviews',
60         figure={
61             'data': [
62                 go.Scatter(
63                     x=df[df['Predictions'] == 1]['Age'], # The only difference in this graph is it calls the predictive model price rather than the actual price.
64                     y=df[df['Predictions'] == 1]['points'],
65                     mode='markers',
66                     opacity=0.7,
67                     marker={
68                         'size': 15,
69                         'line': {'width': 0.5, 'color': 'white'}
70                     },
71                     name=1
72                 ) for i in df.Predictions[0:10]
73             ],
74             'layout': go.Layout(
75                 xaxis={'type': 'log', 'title': 'Age'},
76                 yaxis={'title': 'points'},
77                 margin={'l': 40, 'b': 40, 't': 10, 'r': 10},
78                 legend={'x': 0, 'y': 1},
79                 hovermode='closest'
80             )
81         ),
82     )
83 ]
84
85 # This call runs the server.
86 if __name__ == '__main__':
87     app.run_server(debug=True)
```

All the code on the Jupyter Notebooks is my own. Although it does use SkLearn libraries to help to make a prediction.