1. a. The analyst will not get a good result if value of k is equal to the number of instances. It will result in underfitting as it will always give the average of all instances available.

b. It could be useful if weighted knn is used. The instances that are near the query instance gets more weight than the one that are farther. Thus, the prediction will be made sensibly based on the distance metric because of the weight attached to each instance(1/d2).

2.

Answer:

ID	Waves Size	Wave Period	Windspeed	Good surf	Q1	Q2	Q3
1	6	15	5	y	3.60555128	18.493242	4.12310563
2	1	6	9	n	13.3790882	12.083046	8.66025404
3	7	10	4	y	5.47722558	16.1554944	1.41421356
4	7	12	3	y	3.31662479	18.0554701	1.73205081
5	2	2	10	n	16.4012195	10	11.5325626
6	10	2	20	n	22.2934968	2.82842712	18.7882942

Q1 will predict 'yes' as the nearest neighbor with ID-4 is 'yes'
Q2 will predict 'no' as the nearest neighbour with ID 6 is no
Q3 will predict ''yes' as the nearest neighbour with ID 3 is yes

3.

ID	**AGE**	**HEARTRATE**	**OXYCON**	**Predictions**	Error(absolute)	Error_square	ErrorDelta(D,W[0])	ErrorDelta(D,W[1])	ErrorDelta(D,w[2])
1	41	138	37.99	**17.15**	20.84	434.3056	20.84	854.44	2875.92
2	42	153	47.34	**26**	21.34	455.3956	21.34	896.28	3265.02
3	37	151	44.38	**25.55**	18.83	354.5689	18.83	696.71	2843.33
4	46	133	28.17	**13.4**	14.77	218.1529	14.77	679.42	1964.41

5	48	126	27.07	**8.9**	18.17	330.1489	18.17	872.16	2289.42
6	44	145	37.85	**20.9**	16.95	287.3025	16.95	745.8	2457.75
7	43	158	44.72	**28.85**	15.87	251.8569	15.87	682.41	2507.46
8	46	143	36.42	**19.4**	17.02	289.6804	17.02	782.92	2433.86
9	37	138	31.21	**17.75**	13.46	181.1716	13.46	498.02	1857.48
10	38	158	54.85	**29.6**	25.25	637.5625	25.25	959.5	3989.5
11	43	143	39.84	**19.85**	19.99	399.6001	19.99	859.57	2858.57
12	43	138	30.83	**16.85**	13.98	195.4404	13.98	601.14	1929.24
				SUM	216.47	4035.1863	216.47	9128.37	31271.96

b. Sum of squared error= 4035.863/2=**2017.59315**

c. Weights at next iteration is

w[0]=-59.50+ 0.000002*216.47=-59.4996

w[1]=-0.15+0.000002*9128.37=-0.1317

w[2]=0.6+0.000002*31271.96=0.6625

4. a. P(Secondhand=false|purchased=True)=2/4=0.5

   P(Gender=Literature|purchased=True)=1/4=0.25

   P(Cost=expensive|purchased=true)=1/4=0.25

   P(Purchased=true)=4/10=0.4

   P(Secondhand=false|purchased=False)=3/6=0.5

   P(Gender=Literature|purchased=False)=1/6=0.1667

P(Cost=expensive|purchased=False)=1/6=0.1667

P(Purchased=False)=6/10=0.6

Initial Score P(Purchased=True)= 0.5*0.25*0.25*0.4=0.0125

P(Purchased=False)= 0.5*0.1667*0.1667*0.6=0.0083

Normalization constant=0.0125+0.0083=0.0208

Actual probabilities of each outcome

P(Purchased=True)=0.0125/0.0208=0.6010

P(Purchased=FALSE)=0.0083/0.0208 =0.3990

b. NAÏVE Bayes would give the target level with maximum posteriori probability as its outcome. Thus, it would predict the target as **True** as per the calculations done in previous question.

5. As K increases, the model tends to underfit as it smoothens the boundary edges to include more neighbors and model gets more generalized. This results in bias as the model will not capture the subtle patterns in training data.
6. KNN is more sensitive to feature scaling. This is because the KNN makes predictions based on distance metric to measure similarity between data points by measuring the distance from the query instance to the rest of neighbors chosen. If features have different scales, the one with higher scale will dominate the distance calculation and the model will not be able to consider all features equally.
7. Decision trees would be faster in prediction as they are eager learners. Eager learners are algorithms that build a model or construct a representation of the training data during the training phase. This representation is often an abstraction or summary of the data that captures patterns, relationships, and information relevant to making predictions. This abstraction is then used to make predictions for new, unseen data, without directly comparing each new data point to every instance in the training dataset.
   In the case of KNN, the nearest neighbor algorithm delays abstracting from the data until it is asked to make a prediction, making it a lazy learner. At this point the information in the query is used to define neighborhoods in the feature space, and a prediction is made based on the instances in this neighborhood. As the number of instances becomes large, the model will become slower because it has more instances to check when defining the neighborhood.
8. True - basis functions (variable transformations)
9. False - large learning rate may fail to converge at global min
10. False - regularization techniques are used to lower variance (reduce overfitting), i.e., high train accuracy with low test / validation accuracy
11. False - regression is susceptible to large outliers