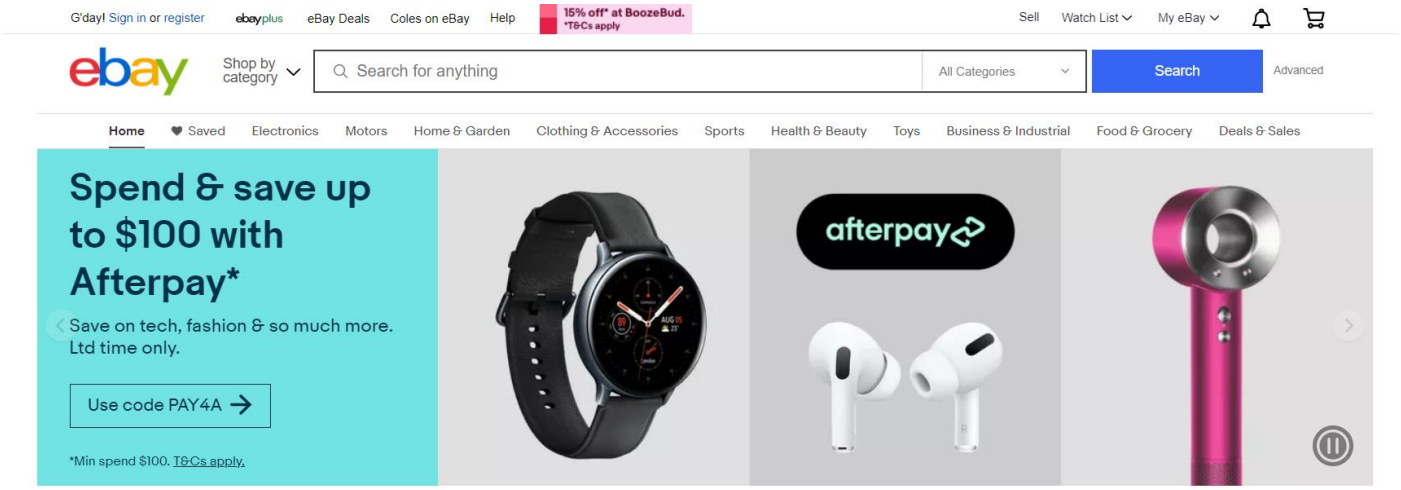
INFS4203 ASSIGNMENT 4

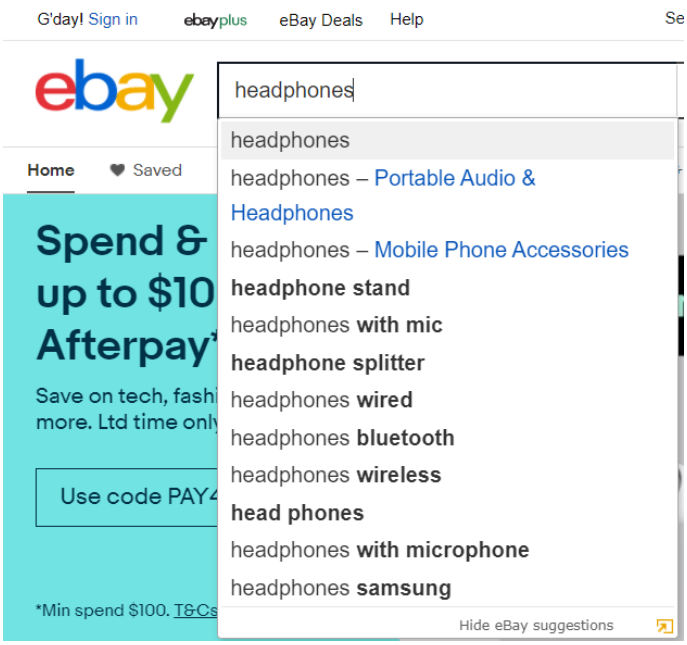
Student Number: 4536071

By Michael Delmastro

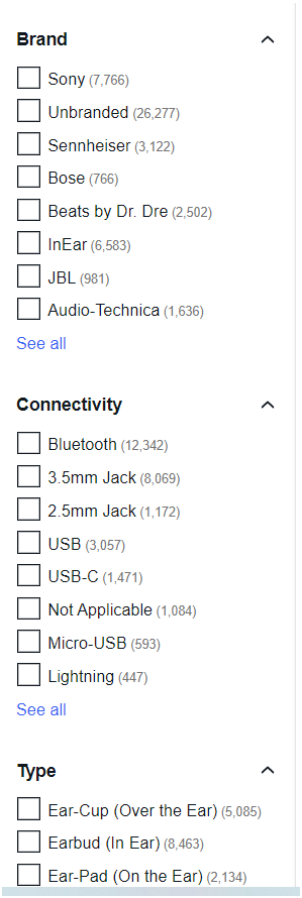
Q1.1: Classification methods are applied to eBay's website so that users can search easily and narrow down their choices with categories.



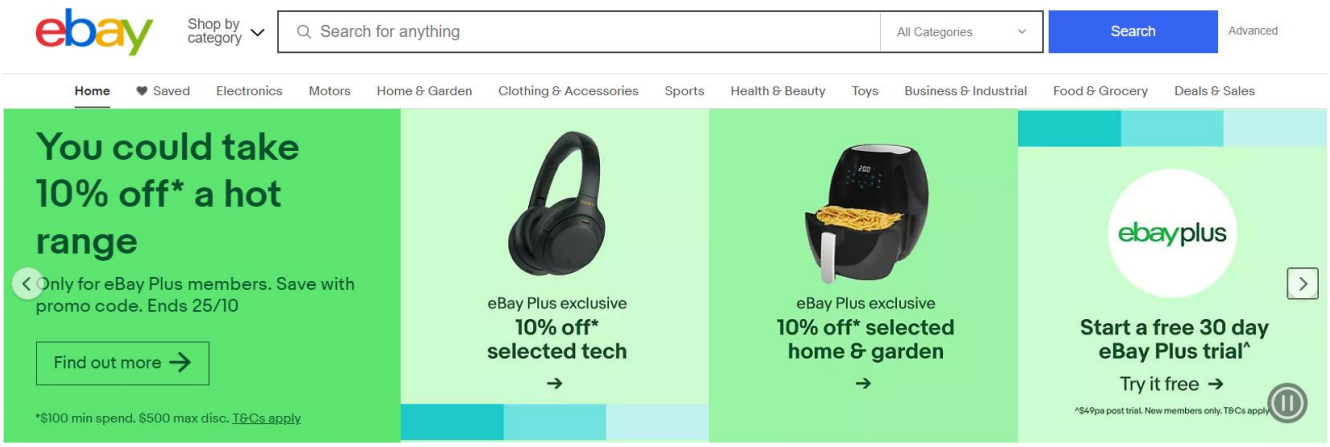
By searching headphones, eBay provides a list of recommendations which accompany the headphones.



After searching headphones, provides a list of recommendations which are categories within the headphone branch. These are labels that are provided which the training data collection process will use to help users purchase things more efficiently and effectively. This is based off Naïve Bayes Algorithm. It is like a decision tree essentially, where the user has the option on which brand to choose, connectivity, and type of device. Basically going further and further down the decision tree until the user gets exactly the specifications they require. By collecting data on the users' movements on the website, how they interact with the labels (brand, connectivity, type) and searching suggestions will help the company get people to buy more. By utilising classification techniques on this data, better UI, UX design features could be created.

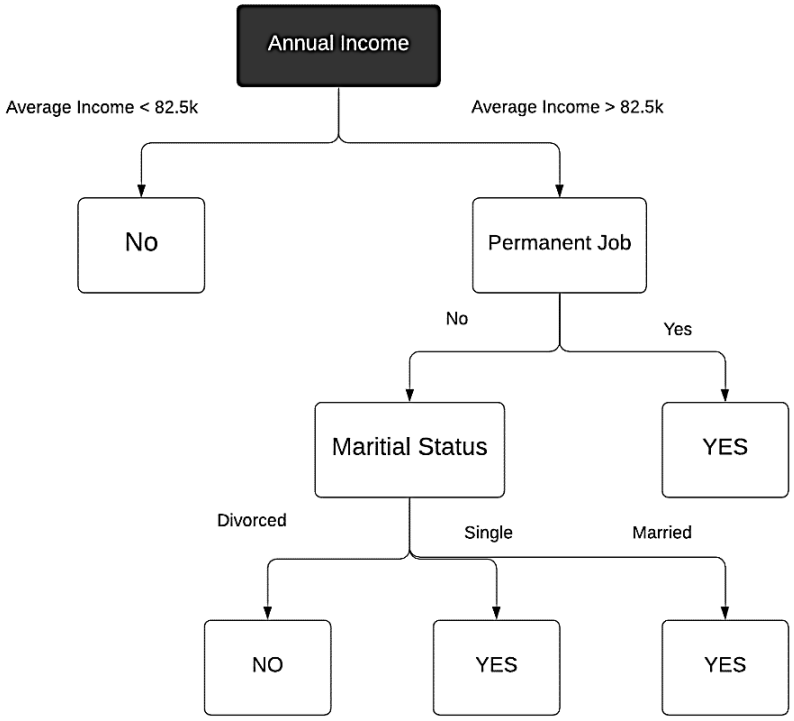


Q1.2: Training data that eBay collects is essentially how the user interacts on their website. This includes where their mouse moves, their previous purchases, shopping cart items and searches. After searching for a product, eBay will recommend and advertise similar products. Consequently, classification techniques such as decision trees and Naive Bayes allow the company to tailor their content towards individual users a lot better than without. These classification techniques make it more user friendly and more efficient from a backend programming aspect since the searching that the user has to do is minimised.



Q1.3: One thing about websites, online stores and social media applications is that they tailor their content based on how users interact with their platforms. However, a main ethical issue is concerned about user privacy. By using these classification techniques, users information is stored and users may not want their data being collected. So privacy is a main concern when classification methods are used.

Q2.1: Decision Tree



Description of the tree:

The decision tree above can predict whether the bank will approve a credit card given the three attributes: Permanent Job, Marital Status and Annual Income. By calculating the Gain_Ratio for each attribute we can see that:

$$\text{Max}(\text{Gain Ratio}(AnnualIncome)) > \text{Gain\_Ratio}(PermanentJob) > \text{Gain\_Ratio}(MaritalStatus)$$

And consequently, a tree with Annual income being the first attribute is split into two categories where a credit card application is not approved for those who earn less than 82.5k (Marked with NO in the decision tree). This can also be confirmed with the training data to make sure the tree follows the data:

Tuple ID:	Permanent Job	Marital Status	Annual Income	Approved?
2	No	Married	80K	No
5	No	Single	60K	No

Then the gain ratio of both the other attributes was calculated on the conditions that the Annual Income had to be greater than 82.5k. And we found out that the following condition still applied:

$$\text{Gain\_Ratio}(PermanentJob) > \text{Gain\_Ratio}(MaritalStatus)$$

So, Permanent Job is the second attribute to be selected to categorise the data. According to the training data, if the person has a permanent job then it is always approved:

Tuple ID:	Permanent Job	Marital Status	Annual Income	Approved?
1	Yes	Single	130K	Yes
4	Yes	Divorced	90K	Yes
6	Yes	Married	120K	Yes
7	Yes	Single	85K	Yes
9	Yes	Married	95K	Yes

Note: If Permanent Job = Yes & Annual Income > 82.5K then credit card is approved

Now, if Permanent Job = No, we need to look at Marital Status.

Tuple ID:	Permanent Job	Marital Status	Annual Income	Approved?
8	No	Divorced	110K	No

The only case when a credit card is not approved is when Permanent Job = No & Annual Income > 82.5K & Marital Status = Divorced as above. So this was labelled accordingly with a No in the decision tree as above and the others were assigned a yes since the following tuples are prevalent:

Tuple ID:	Permanent Job	Marital Status	Annual Income	Approved?
3	No	Single	100K	Yes
10	No	Married	125K	Yes

Naive Bayes

The mean, variance was calculated for both of the classes, yes and no, and probabilities. For each tuple in the table below, the max value among p(C1)p(x | C2)…p(Cm)p(x | Cm) was selected.

Table 2 filled in based on the two classifiers (Decision Tree and Naïve Bayes):

Permanent Job	Marital Status	Annual Income	Prediction (Approved?)	
			Decision Tree	Naïve Bayes
No	Single	60k	No	No
Yes	Married	100k	Yes	Yes
Yes	Single	90k	Yes	Yes
No	Divorced	95k	No	No
No	Married	85k	Yes	No

Q2.2: Table 3 Data

Permanent Job	Marital Status	Annual Income	Training Data Approved?	Prediction (Approved?)	
				Decision Tree	Naïve Bayes
No	Single	60k	No	No	No
Yes	Married	100k	Yes	Yes	Yes
Yes	Single	90k	No	Yes	Yes
No	Divorced	95k	Yes	No	No
No	Married	85k	No	Yes	No

ACCURACY:

Decision Tree Accuracy = 2/5

Naïve Bayes Accuracy = 3/5

F1-Measure:

Decision Tree

	Predicted positive	Predicted negative
Actual positive	1	1
Actual negative	2	1

Naïve Bayes

	Predicted positive	Predicted negative
Actual positive	1	1
Actual negative	1	2

Decision Tree

$$Precision = \frac{1}{3}$$

$$Recall = \frac{1}{2}$$

$$F1 = \frac{2}{((1/1/3) + (1/1/2))} = \frac{2}{5}$$

Naïve Bayes

$$Precision = \frac{1}{2}$$

$$Recall = \frac{1}{2}$$

$$F1 = \frac{2}{((1/0.5) + (1/0.5))} = \frac{1}{2}$$

It is clear, that the Naïve Bayes is more accurate than the decision tree. Since, the Naïve Bayes classifier has a great F1 value, it is a more appropriate method to predict whether the table 3 tuples will be approved or not.

Q2.1: Calculations:

Entropy of the whole dataset with tuple Identifiers of 1 – 10:

$$\text{Ent}([1\ -\ 10]) = -\frac{3}{10}\log_2\left(\frac{3}{10}\right) - \frac{7}{10}\log_2\left(\frac{7}{10}\right) = 0.8813$$

Entropy, Gain and IV calculations for Permanent Job Attribute:

(YES) $\text{Ent}(PermanentJob = Yes) = -\frac{5}{5}\log_2\left(\frac{5}{5}\right) - \frac{0}{5}\log_2\left(\frac{0}{5}\right) = 0$

(No) $\text{Ent}(PermanentJob = No) = -\frac{2}{5}\log_2\left(\frac{2}{5}\right) - \frac{3}{5}\log_2\left(\frac{3}{5}\right) = 0.9710$

(Gain) $\text{Gain}([1-10],\ PermanentJob) = 0.8813 - 0.9710 \cdot \left(\frac{5}{10}\right) = 0.3958$

(IV) $\text{IV}(PermanentJob) = -\frac{5}{5}\log_2\left(\frac{5}{5}\right) - \frac{5}{5}\log_2\left(\frac{5}{5}\right) = 1$

Permanent Job Gain Ratio:

$$\text{Gain\_Ratio}(PermanentJob) = 0.3958$$

Entropy, Gain and IV calculations for Marital Status Attribute:

(SINGLE) $\text{Ent}(MaritalStatus\ = Single) = -\frac{3}{4}\log_2\left(\frac{3}{4}\right) - \frac{1}{4}\log_2\left(\frac{1}{4}\right) = 0.8113$

(MARRIED) $\text{Ent}(MaritalStatus\ = Married) = -\frac{3}{4}\log_2\left(\frac{1}{4}\right) - \frac{1}{4}\log_2\left(\frac{1}{4}\right) = 0.8113$

(Divorced) $\text{Ent}(MaritalStatus\ = Divorced) = -\frac{1}{2}\log_2\left(\frac{1}{4}\right) - \frac{1}{2}\log_2\left(\frac{1}{2}\right) = 1$

(Gain) $\text{Gain}([1-10],\ MaritalStatus) = 0.8813 - 0.8113 \cdot \left(\frac{4}{10}\right) - 0.8113 \cdot \left(\frac{4}{10}\right) - 1 \cdot \left(\frac{2}{10}\right) = 0.0323$

(IV) $\text{IV}(PermanentJob) = -\frac{4}{10}\log_2\left(\frac{4}{10}\right) - \frac{4}{10}\log_2\left(\frac{4}{10}\right) - \frac{2}{10}\log_2\left(\frac{2}{10}\right) = 1.5219$

Marital Status Gain Ratio:

$$\text{Gain\_Ratio}(MaritalStatus) = \frac{0.0323}{1.519} = 0.0213$$

Bi-Partition of Annual Income Attribute:

Since, this is a continuous variable, a Bi-Partition will be used to find the Gain Ratio.

Sorted Values	60	80	85	90	95	100	110	120	125	130

Sorted Values	60	80	85	90	95	100	110	120	125	130

Candidate Split Values	70	82.5	87.5	92.5	97.5	105	115	122.5	127.5
Gain Ratios	0.412	0.618	0.217	0.094	0.035	0.006	0.218	0.163	0.117

Entropy, Gain and IV calculations for Annual Income Attribute (Highest Gain):

(YES) $\text{Ent}(Annual\ Income > 82.5\ ) = -\frac{7}{8}\log_2\left(\frac{7}{8}\right) - \frac{1}{8}\log_2\left(\frac{1}{8}\right) = 0.5436$

(No) $\text{Ent}(Annual\ Income\ < 82.5\ ) = -\frac{0}{2}\log_2\left(\frac{0}{2}\right) - \frac{2}{2}\log_2\left(\frac{2}{2}\right) = 0$

(Gain) $\text{Gain}([1-10],\ Annual\ Income) = 0.8813 - 0.5436\left(\frac{8}{10}\right) - 0 = 0.4464$

(IV) $\text{IV}(Annual\ Income = 82.5) = -\frac{8}{10}\log_2\left(\frac{8}{10}\right) - \frac{2}{10}\log_2\left(\frac{2}{10}\right) = 0.7219$

Annual Income Gain Ratio:

$$\text{Gain\_Ratio}(AnnualIncome\ (82.5\ Split)) = \frac{0.4464}{0.7219} = 0.6184$$

Consequently, when we split values, the largest Gain Ratio is generated from 82.5k Annual Income split. Thus, we will set 0.6184 to be the Gain Ratio. (Note all other calculations for Gain_ratios were done on paper).

Since the following holds true:

$$\text{Gain Ratio}(AnnualIncome\ ) > \text{Gain\_Ratio}(PermanentJob) > \text{Gain\_Ratio}(MaritalStatus)$$

The Annual Income will be the first feature in the Decision Tree with Annual Income < 82.5k and Annual Income > 82.5K as branches. Also, if Annual Income < 82.5k, it is never approved (a No label will be assigned). So now we calculate Gain Ratio of the other attributes based on the Annual Income < 82.5k constraint. These calculations were completed on paper.