1. **Discretisation**
   1. Arbitrary
   2. Equal-frequency discretisation
   3. Equal-width discretisation
   4. K-means discretisation
   5. Discretisation with trees
   6. Discretisation with Scikit-learn
   7. Discretisation with Feature-engine

Process of transforming- continuous variable – discrete variable

By creating a set of continuous intervals(that span the range of the variables values)

Also called binning

Bin is alternative to interval

Also handle skewness

Handle : outlier(outlier places either outer or lower intervals)

2 types of discretization:

Supervised

Unsupervised

Supervised: informed by target (decision tree based)

Unsupervised:

Equal weight

Eql freq

Discretization by kmeans

--

Equal width:

Devides variable into n bins with same width

Width(max value- min value)/N

Where N = num bins

Min age =0

Max age=73

Width=73-0/10

0-7,7-17,14-21

Y axix: number of passenger

X: bins

Almost similar distribution

Does not improve value spread

Handles outliers

Creates discrete variable

Combine with categorical encoding

## **Discretisation**

Discretisation is the process of **transforming continuous variables** **into discrete variables** by creating a set of **contiguous intervals that span** the range of the **variable's values**.

Discretisation is also called **binning**, where bin is an alternative name for interval.

**Discretisation helps handle outliers and may improve value spread in skewed variables**

Discretisation helps **handle outliers** by placing these values into **the lower or higher intervals,** together with the remaining inlier values of the distribution.

Thus, **these outlier observations no longer differ from the rest of the values at the tails of the distribution**, as they are now all together in the same interval / bucket.

In addition, by creating appropriate bins or intervals, discretisation **can help spread the values of a skewed variable across a set of bins** with equal number of observations.

**Discretisation approaches**

There are several approaches to transform continuous variables into discrete ones. Discretisation methods fall into 2 categories:

**supervised and unsupervised**.

Unsupervised methods do **not use any information**, other than the variable distribution, to create the contiguous bins in which the values will be placed.

Supervised methods typically use target information in order to create the bins or intervals.

**Unsupervised discretisation methods**

* Equal width discretisation
* Equal frequency discretisation
* K-means discretisation

**Supervised discretisation methods**

* Discretisation using decision trees

In this lecture, I will describe **equal width discretisation**.

**Equal width discretisation**

Equal width discretisation divides the scope of possible values into **N bins of the same width**.The width is determined by the range of values in the variable and the number of bins we wish to use to divide the variable:

width = (max value - min value) / N

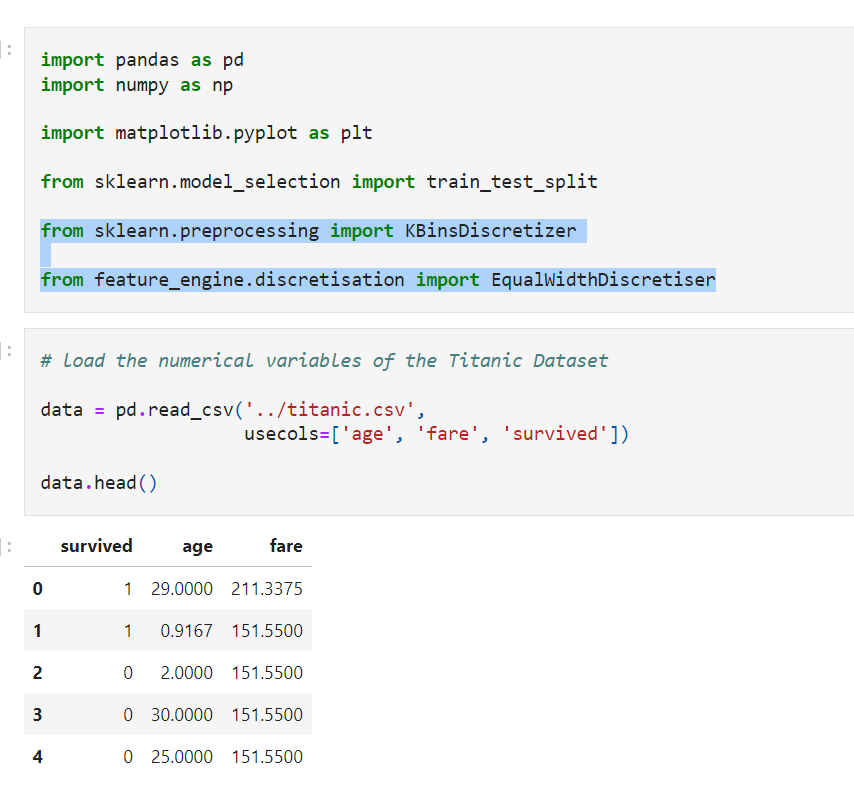
where N is the number of bins or intervals.

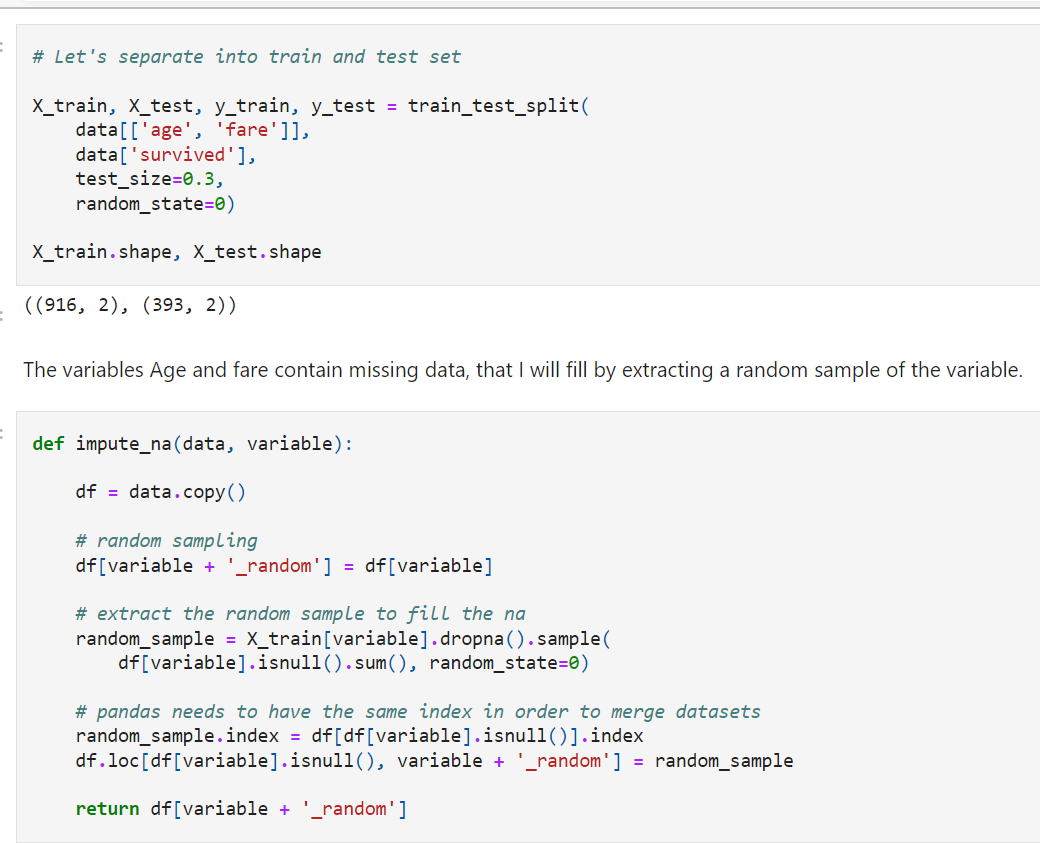
For example if the values of the variable vary between 0 **and 100, we create 5 bins like this: width = (100-0) / 5 = 20.** The bins thus are 0**-20, 20-40, 40-60, 80-100.** The first and final bins (0-20 and 80-100) can be expanded to accommodate outliers (that is, values under 0 or greater than 100 would be placed in those bins as well).

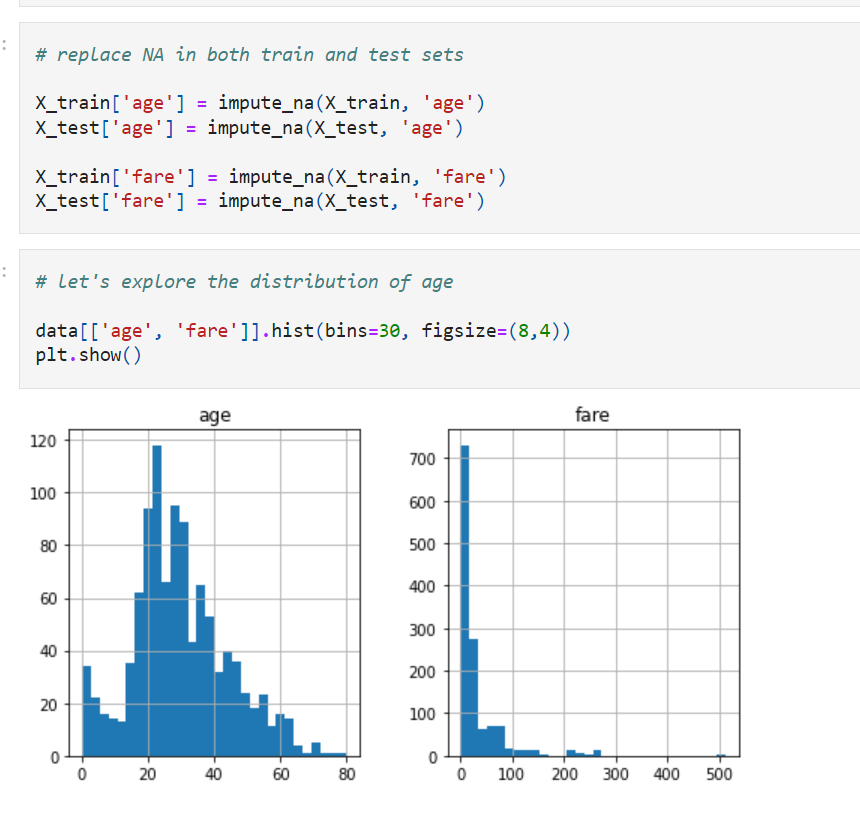
There is no rule of thumb to define N, that is something to determine experimentally.

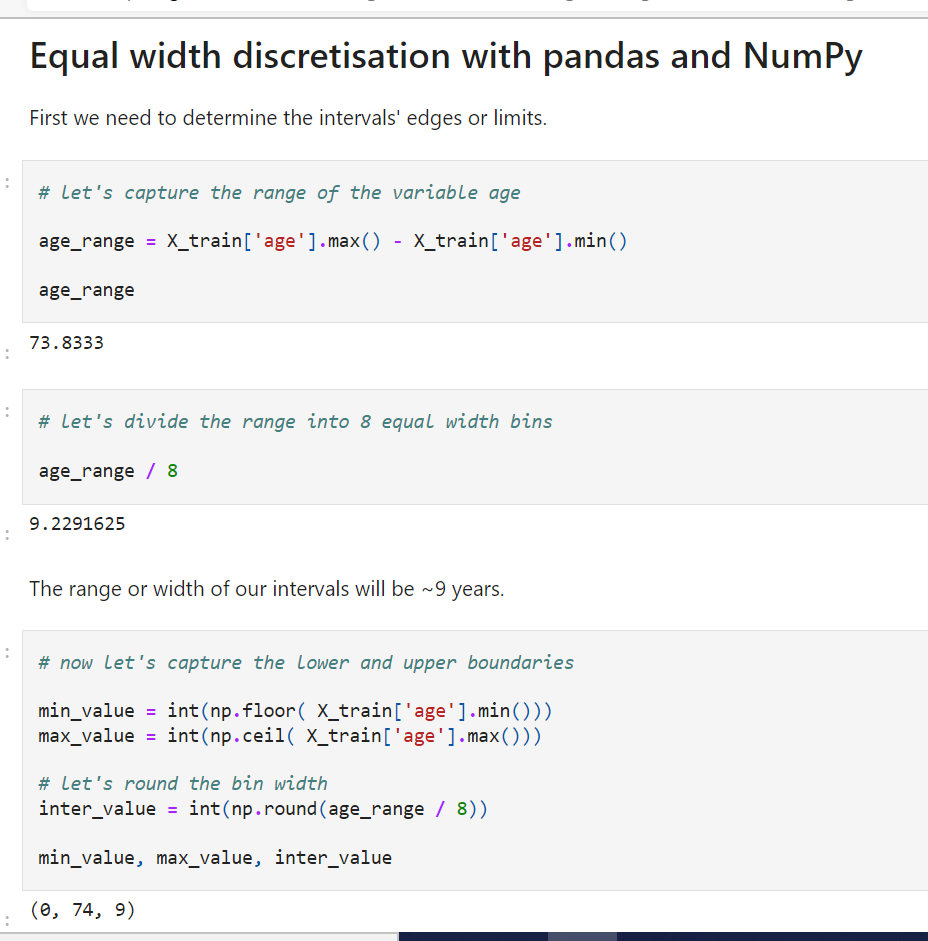
**from sklearn.preprocessing import KBinsDiscretizer**

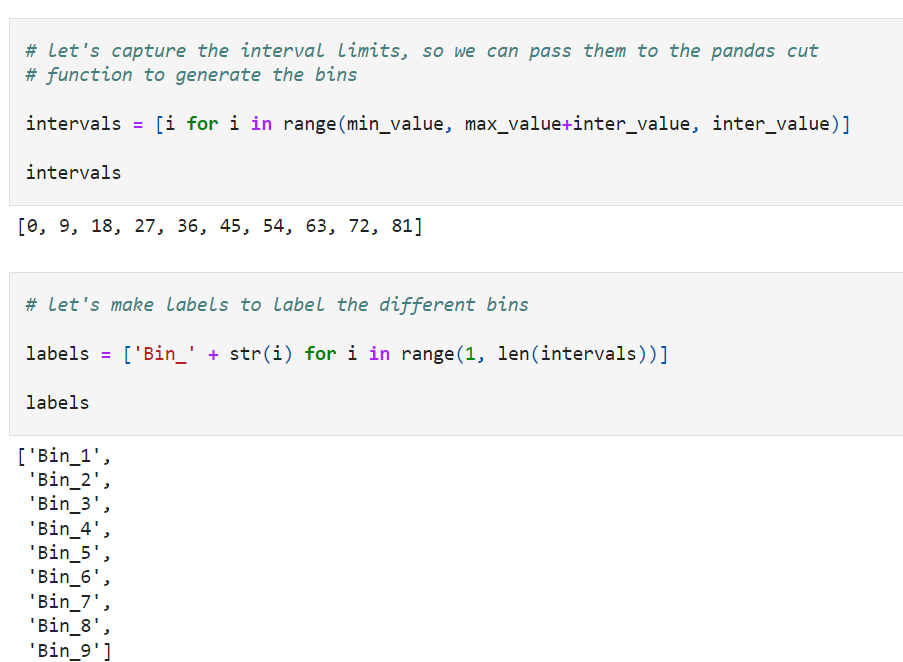
**from feature\_engine.discretisation import EqualWidthDiscretiser**







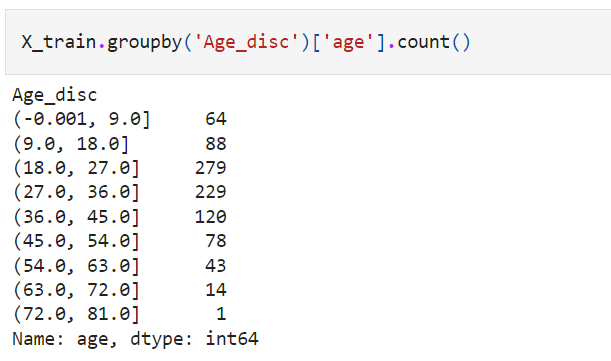


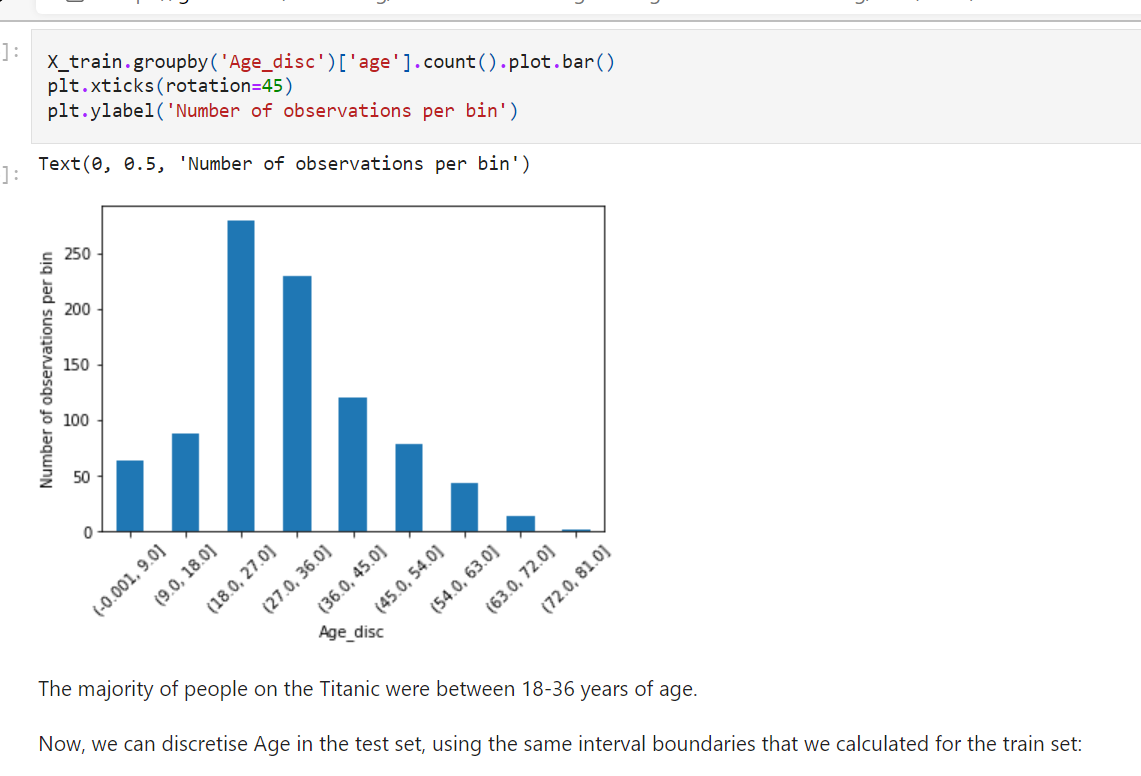


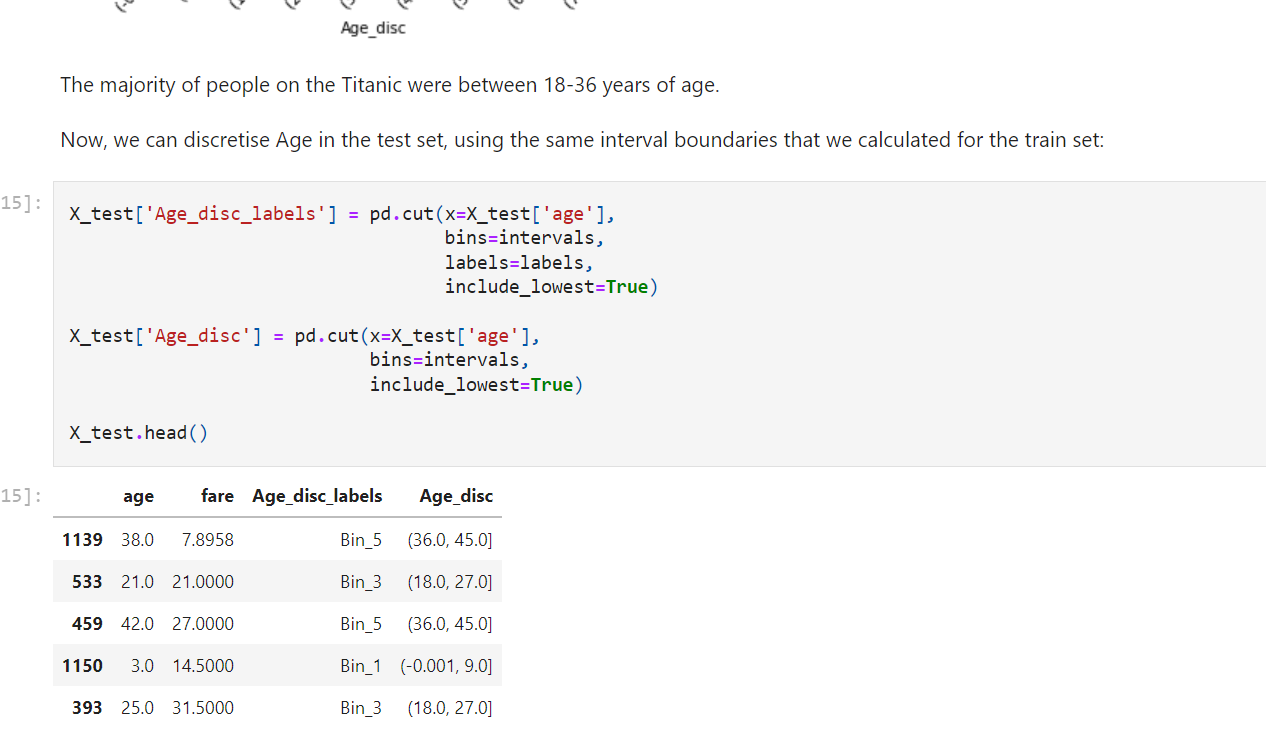


We can see in the above output how by discretising using equal width, we placed each Age observation within one interval / bin. For example, age=13 was placed in the 9-18 interval, whereas age 30 was placed into the 27-36 interval.

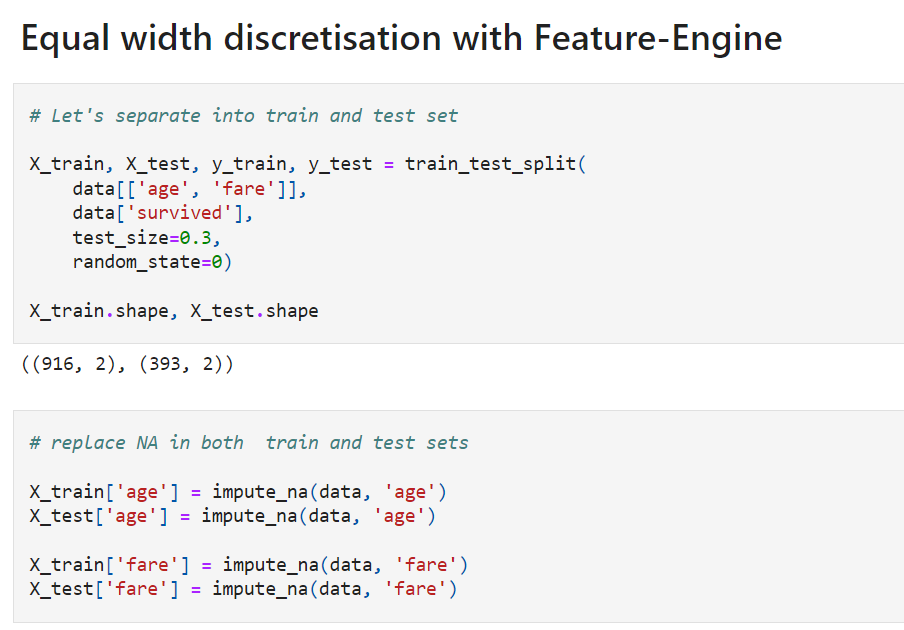
When performing equal width discretisation, we guarantee that the intervals are all of the same length, however there won't necessarily be the same number of observations in each of the intervals. See below:

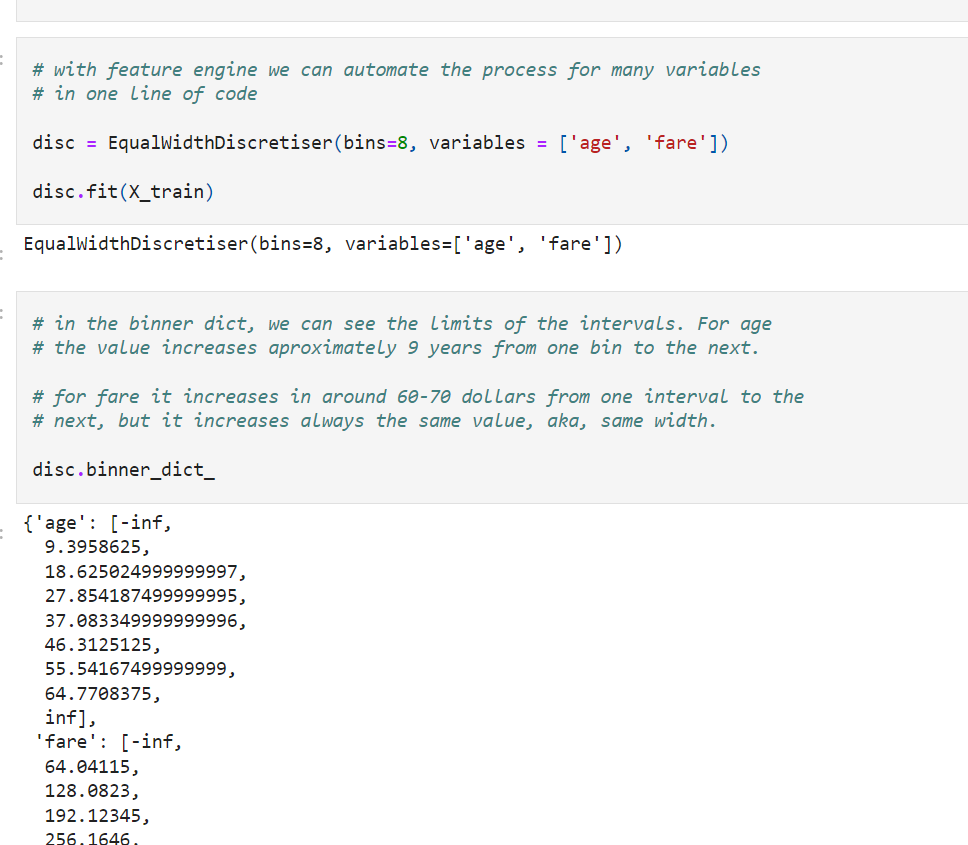


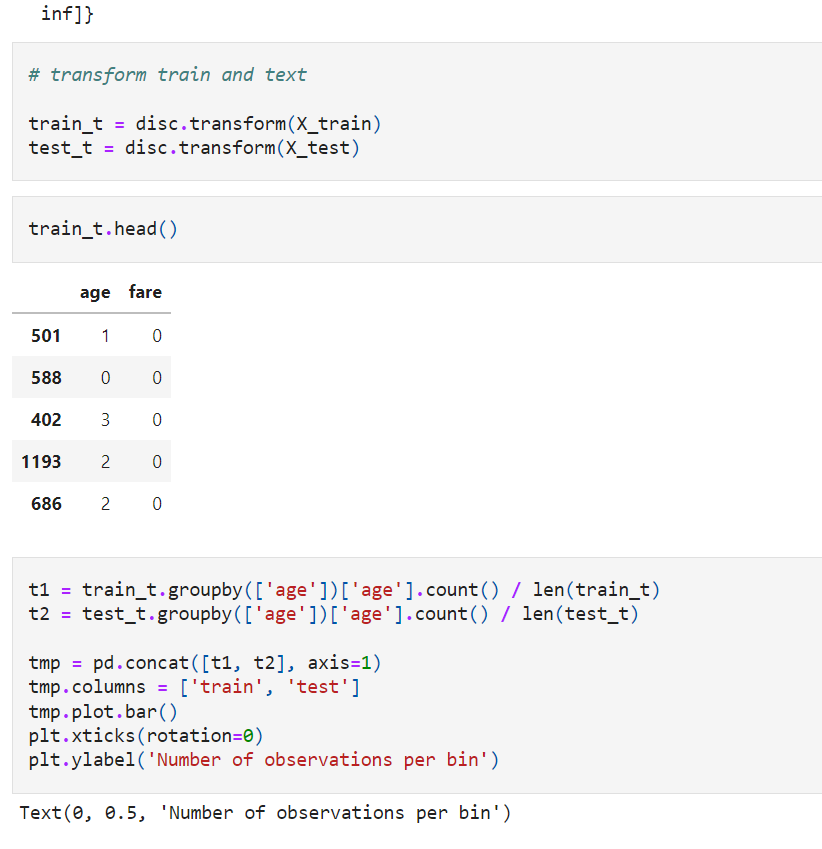


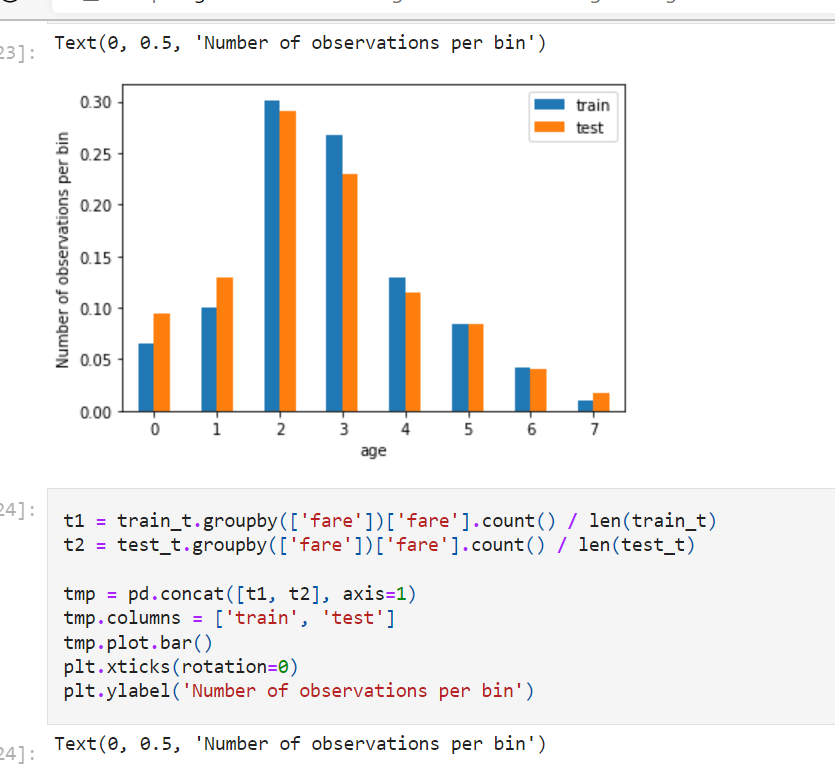


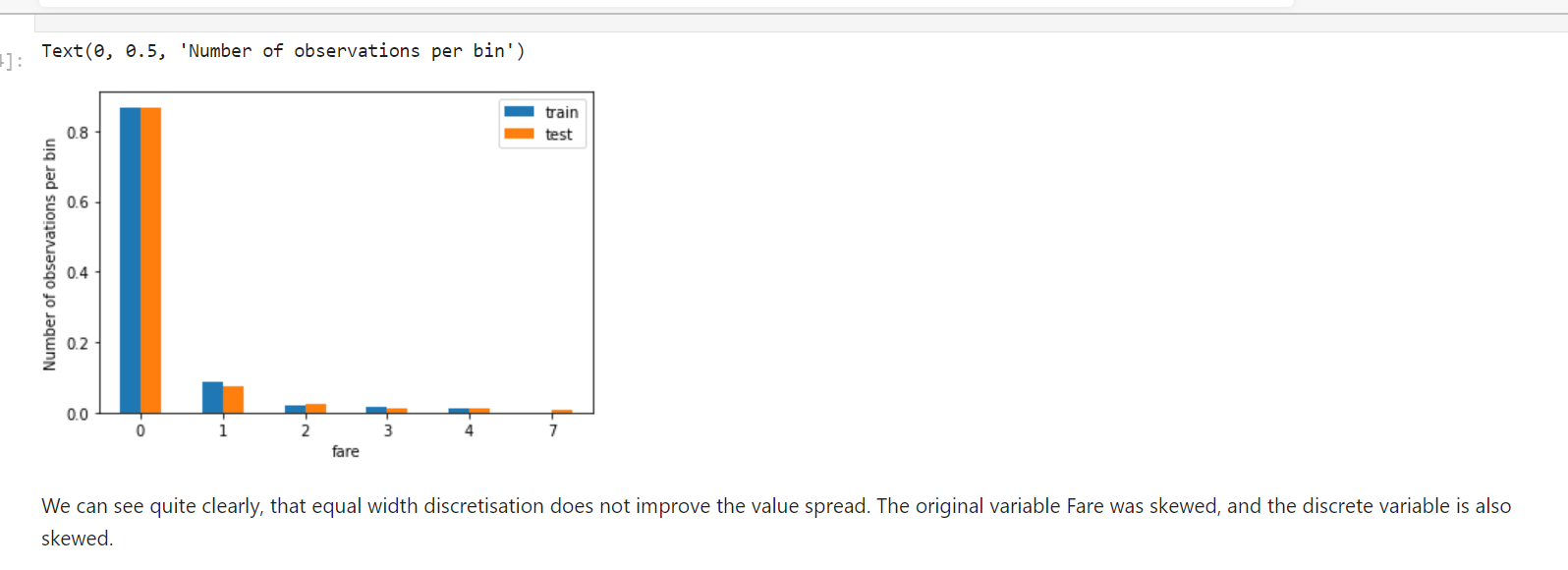


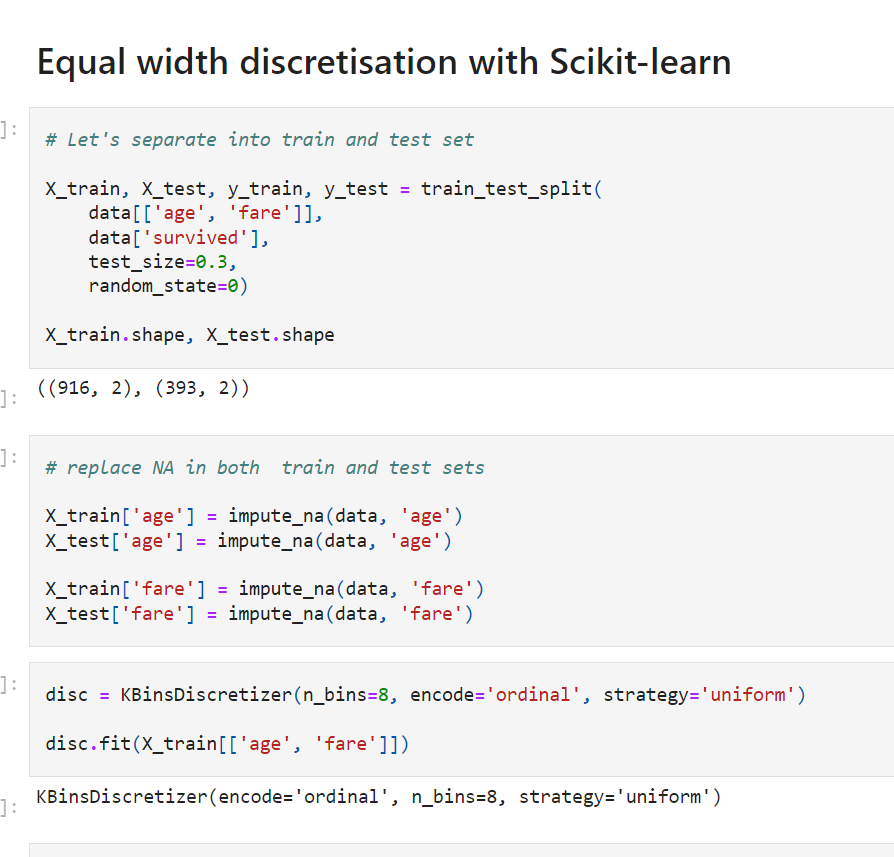


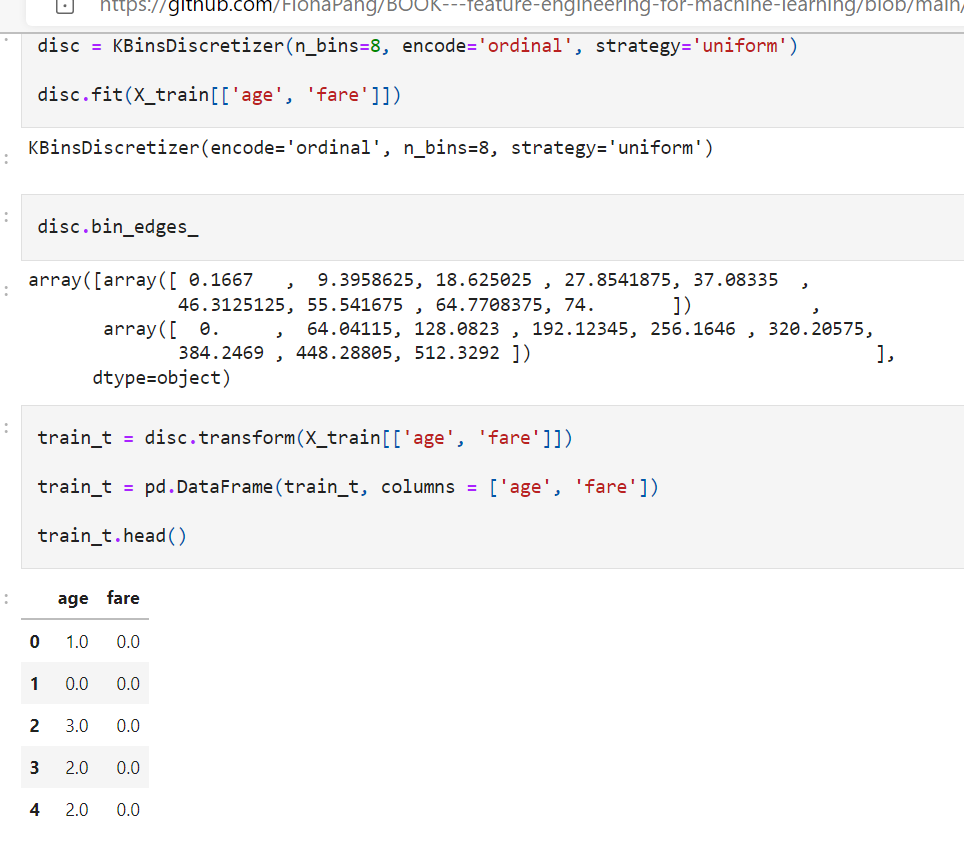




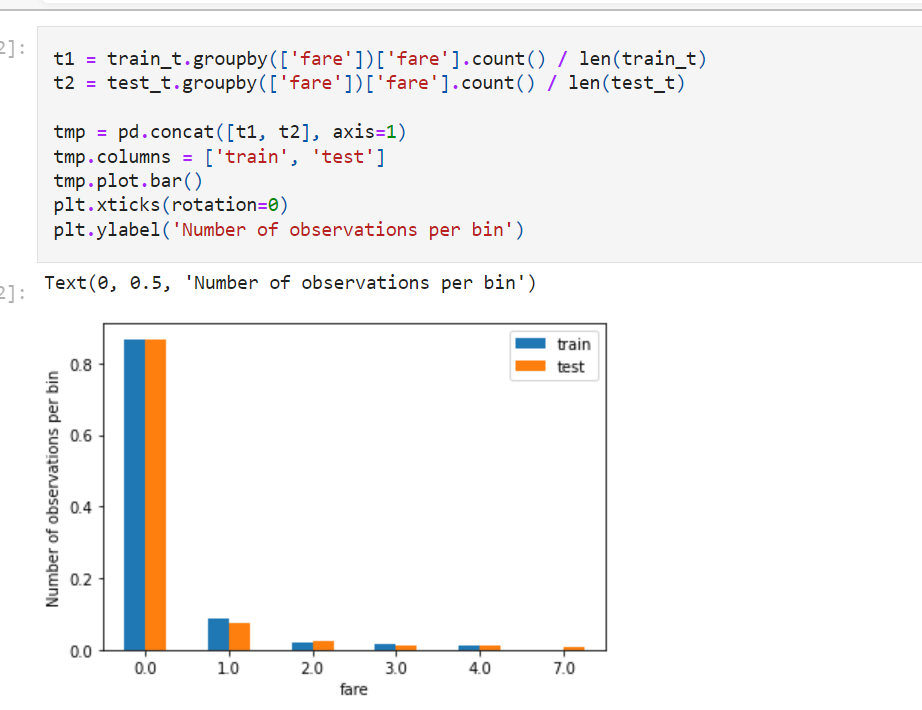












**Equal frequency discretisation**

Equal frequency discretisation divides the scope of possible values of the variable into **N bins**

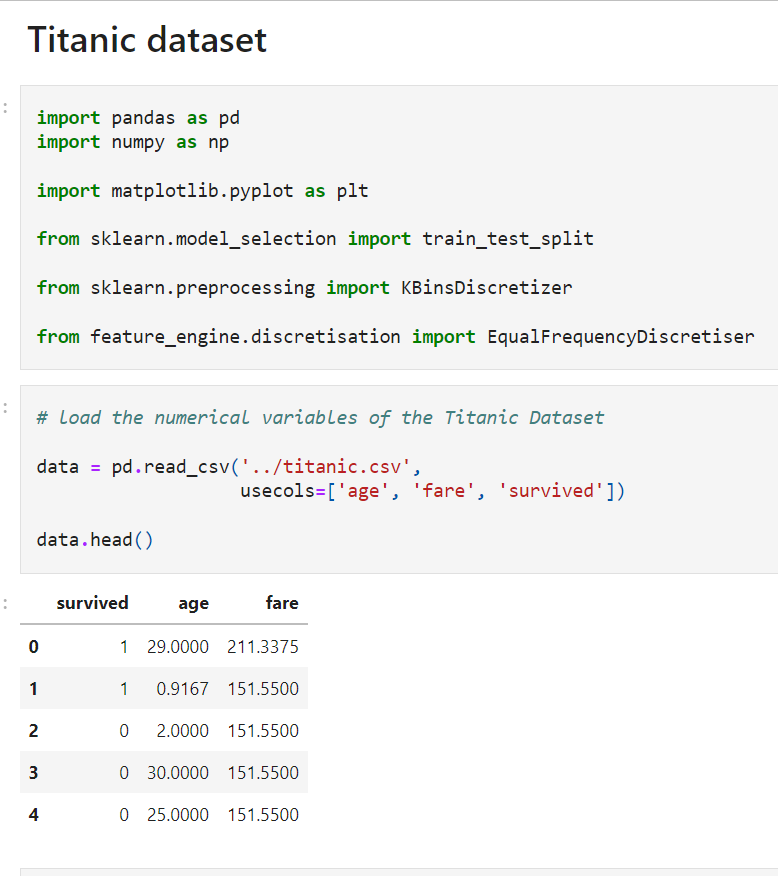
, where **each bin carries the same amount of observations**.

This is particularly **useful for skewed variables** as it spreads the observations over the different bins equally.

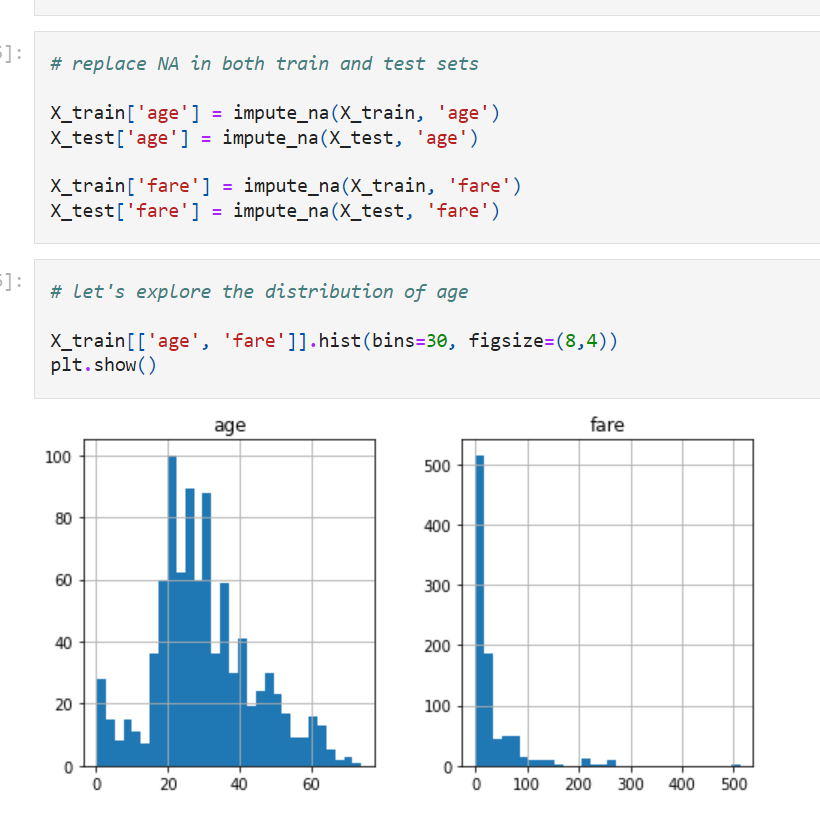
We find the **interval boundaries by determining the quantiles.**

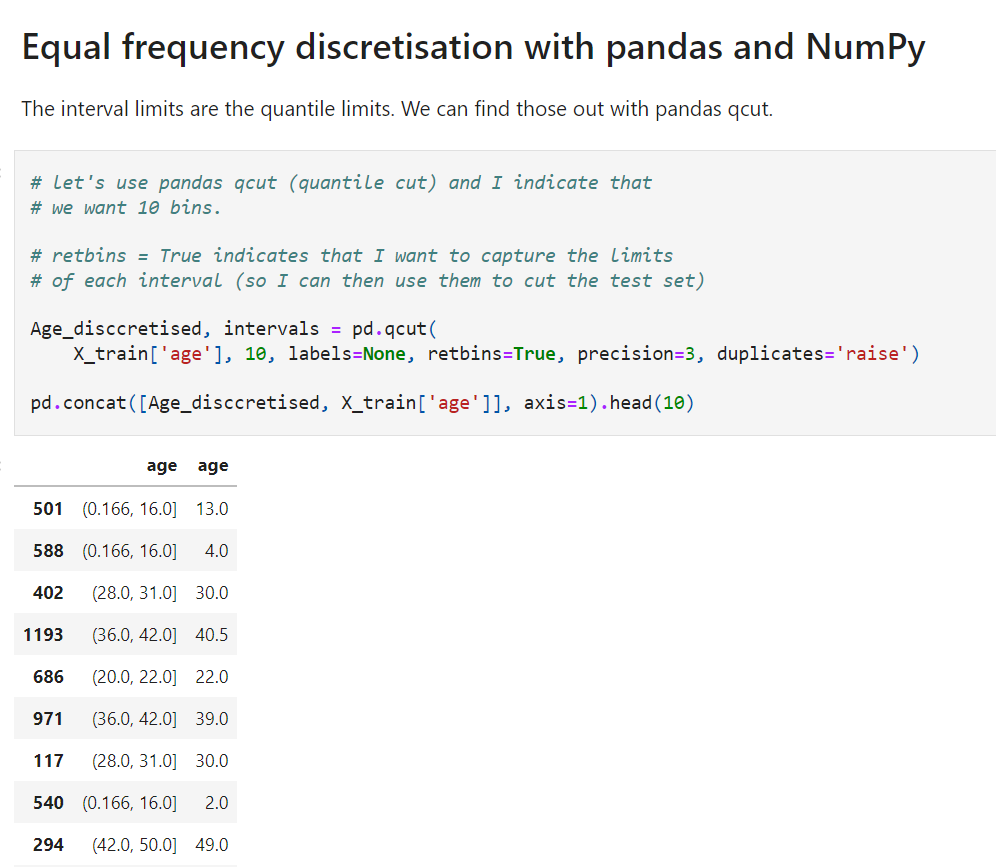
Equal frequency discretisation using **quantiles** consists of dividing the continuous variable into **N quantiles**, N to be defined by the user.

Equal frequency binning is straightforward to implement and by spreading the **values of the observations more evenly** it may help boost the algorithm's performance. This arbitrary binning may also disrupt the relationship with the target.





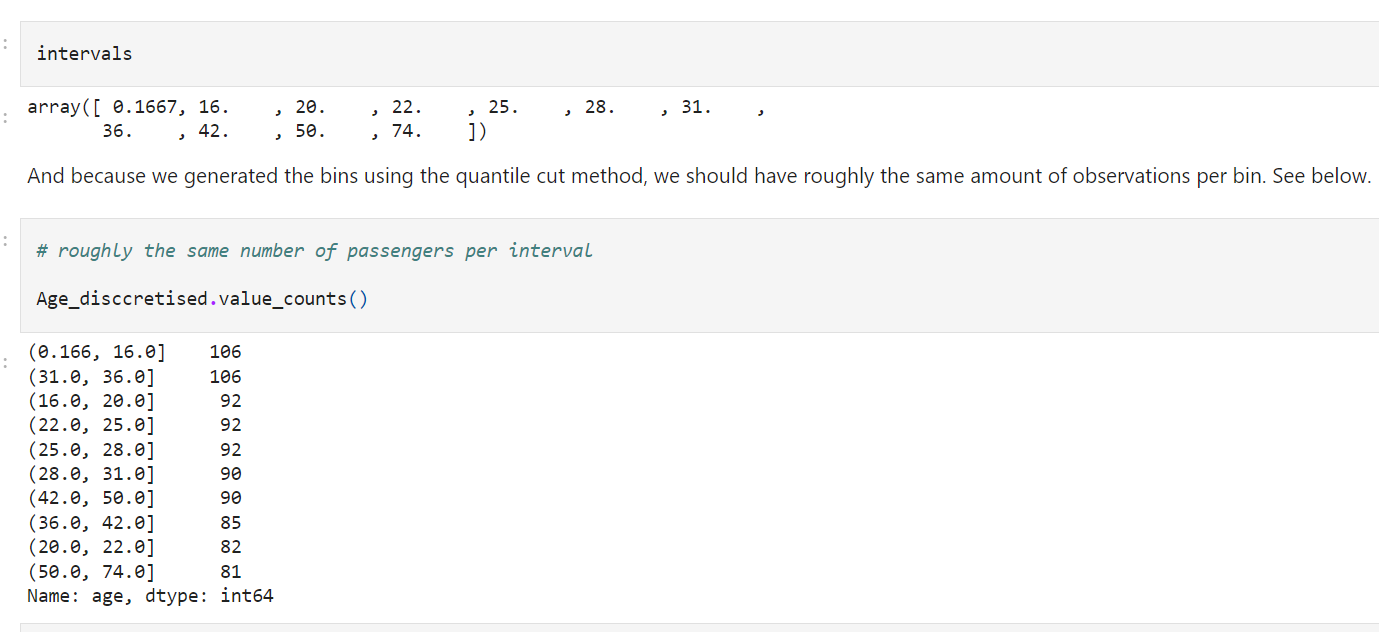


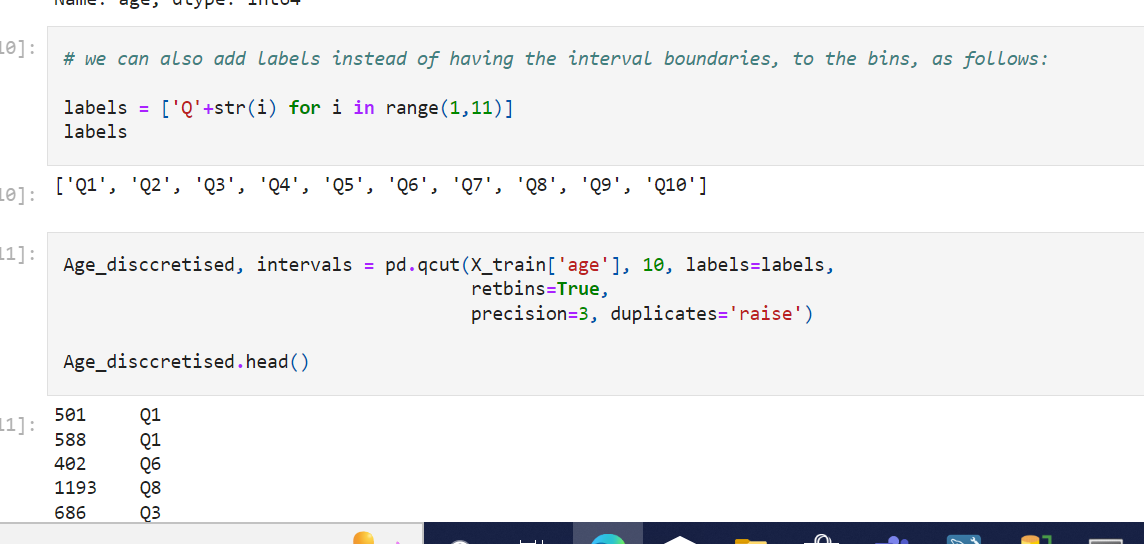


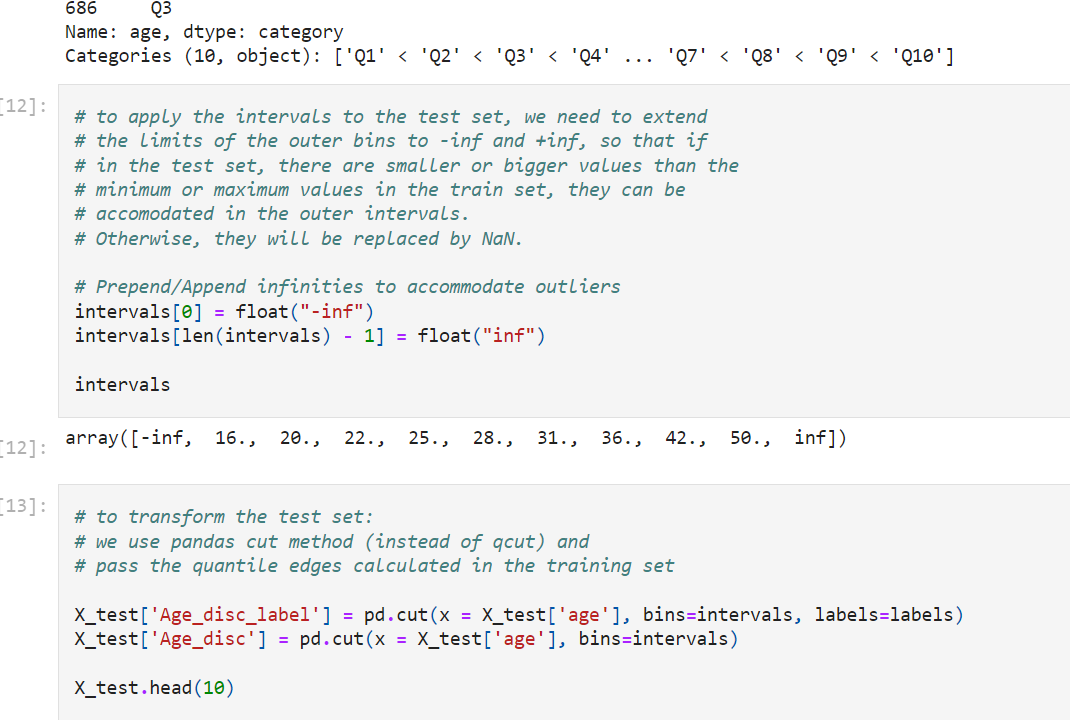
We can see in the above output how by discretising **using quantiles**, we placed each Age observation within one interval. For example, age 30 was placed in the **28-31 interval**, whereas age 49 was placed into the **42-50 interval.**

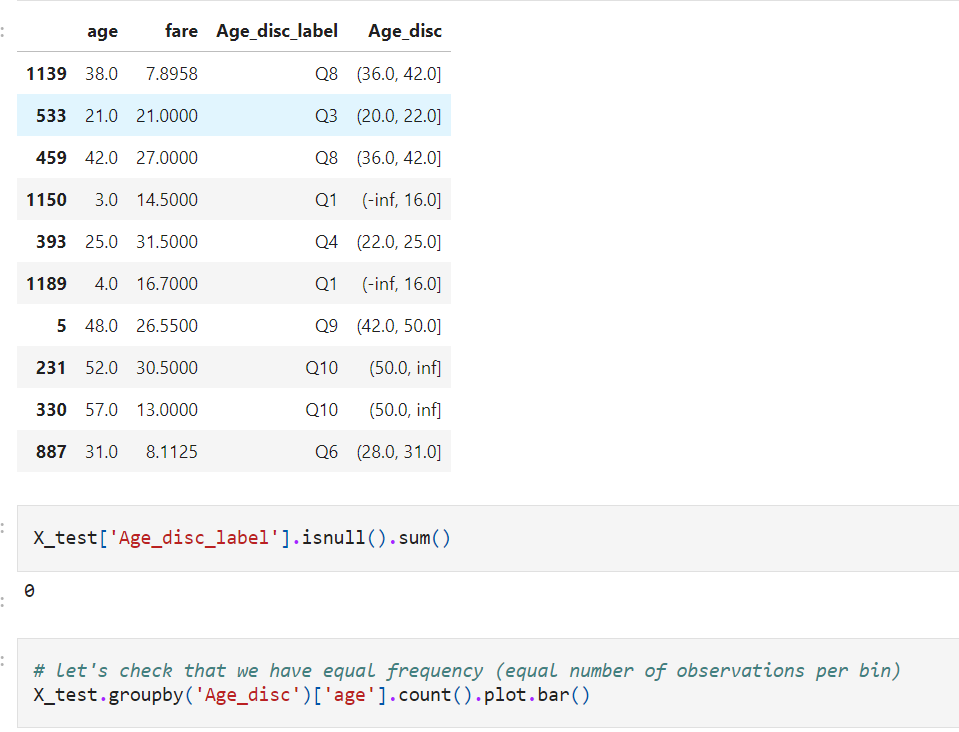
Note how the interval widths are different.

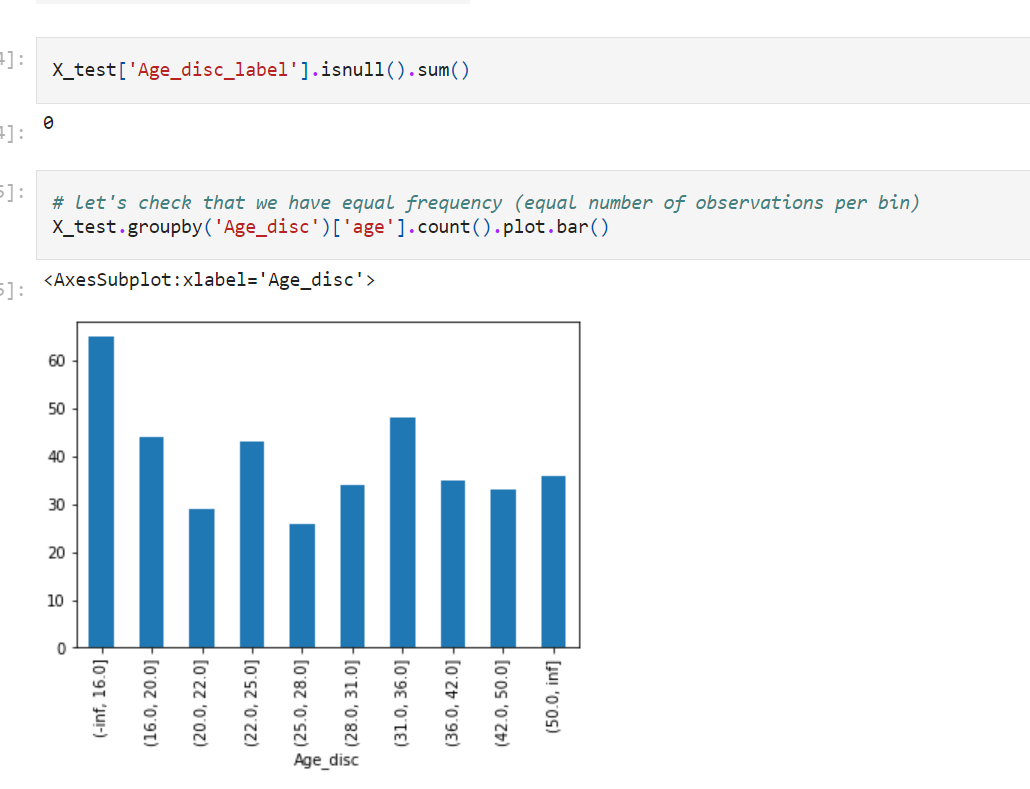
We can visualise the interval cut points below:



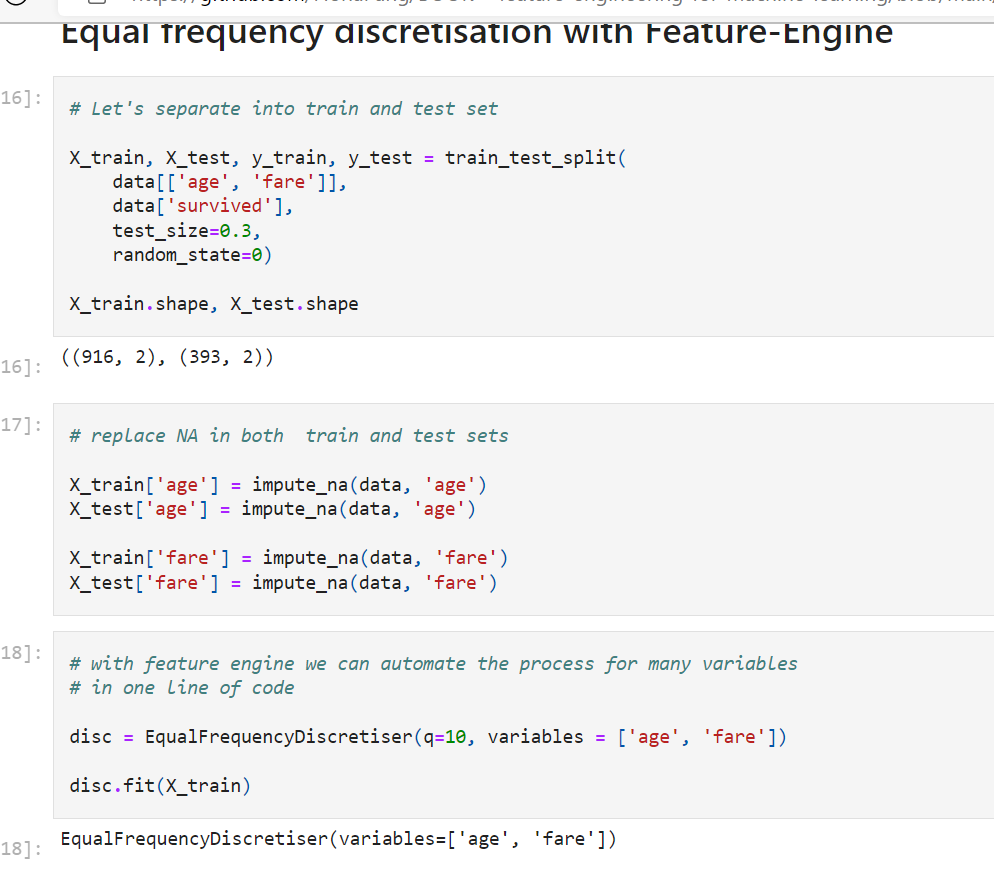


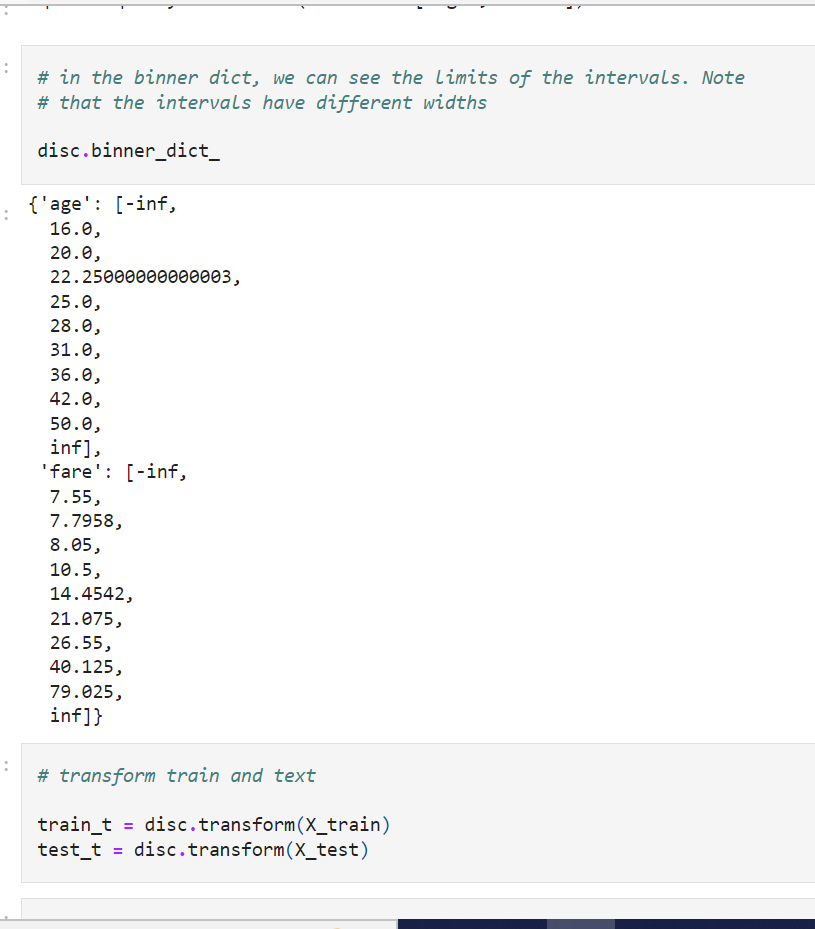


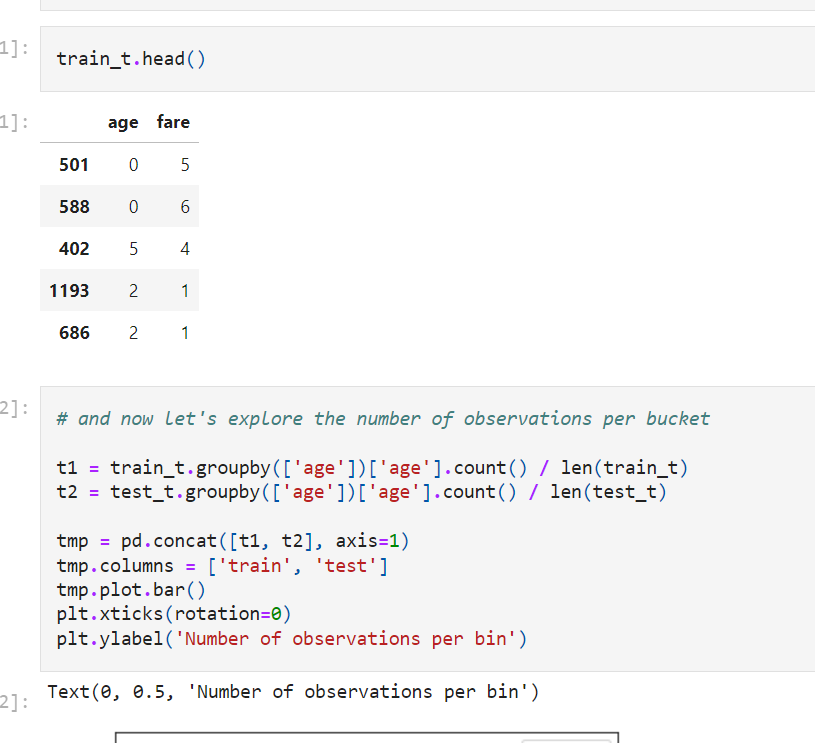


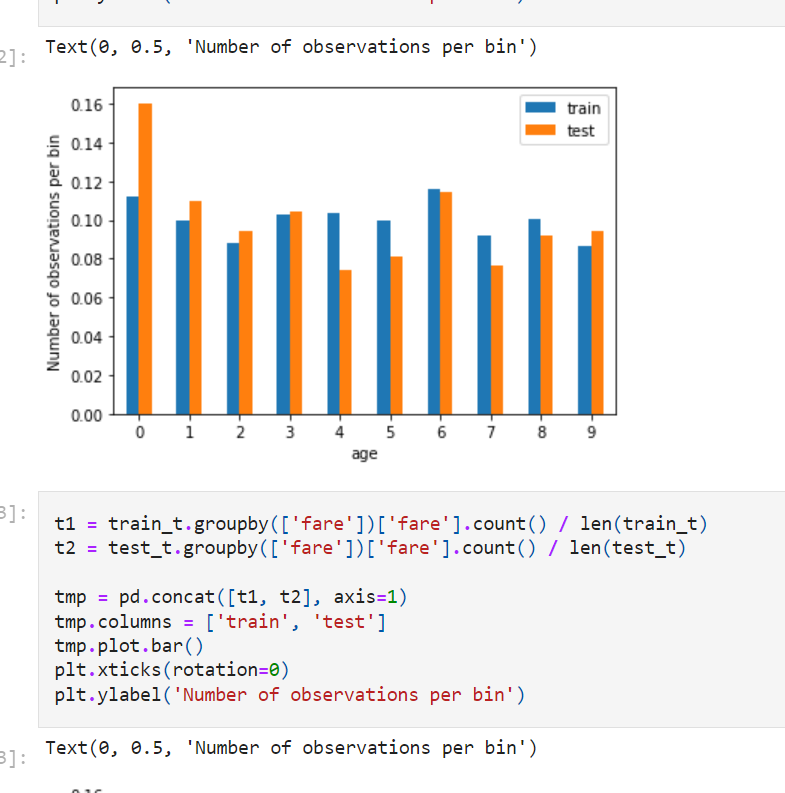


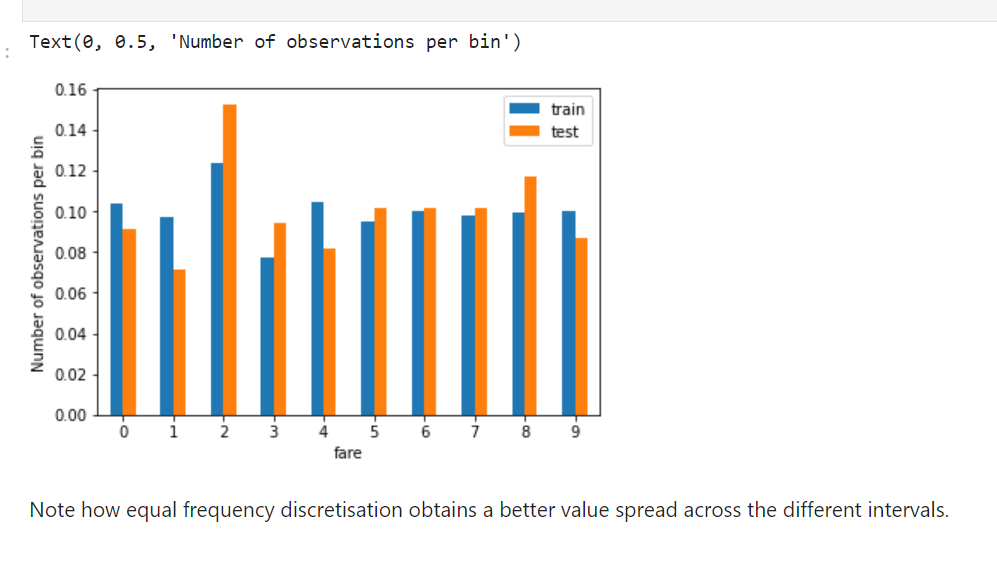
We can see that the top intervals have less observations. This may happen with skewed distributions if we try to divide in a high number of intervals. To make the value spread more homogeneous, we should discretise in less intervals.



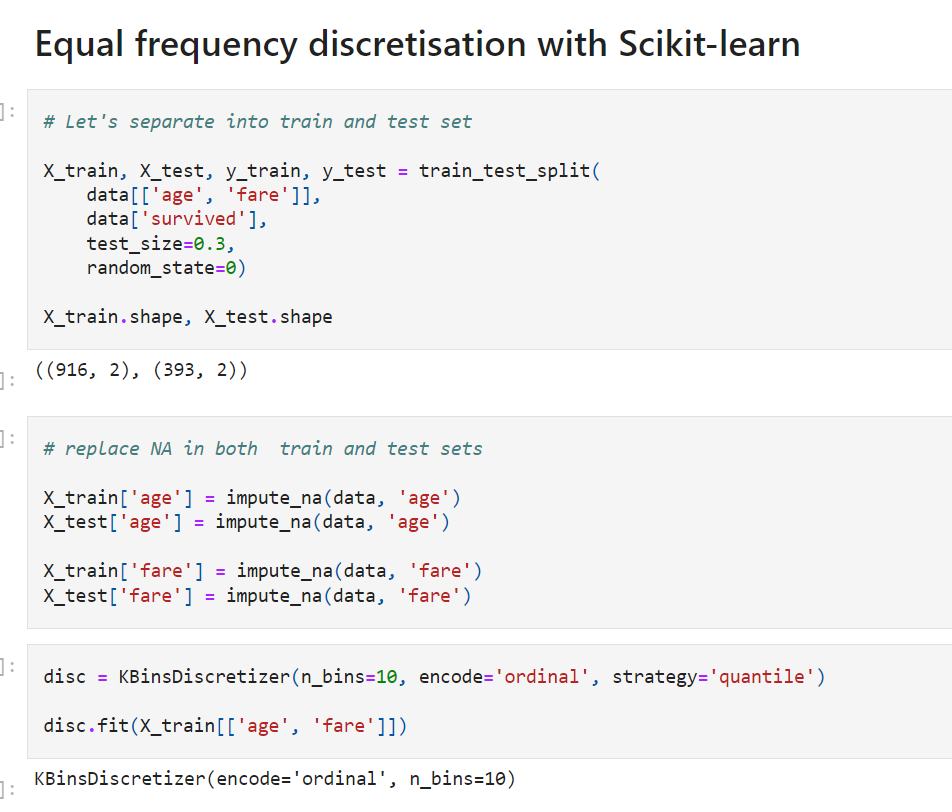


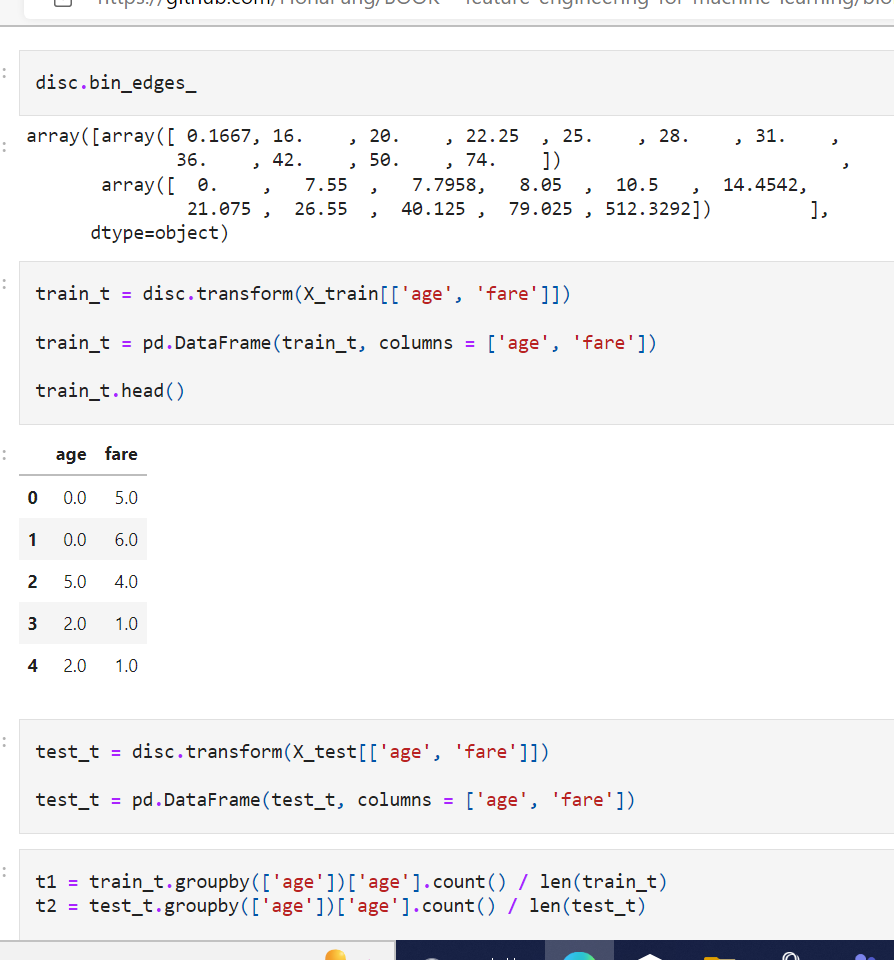


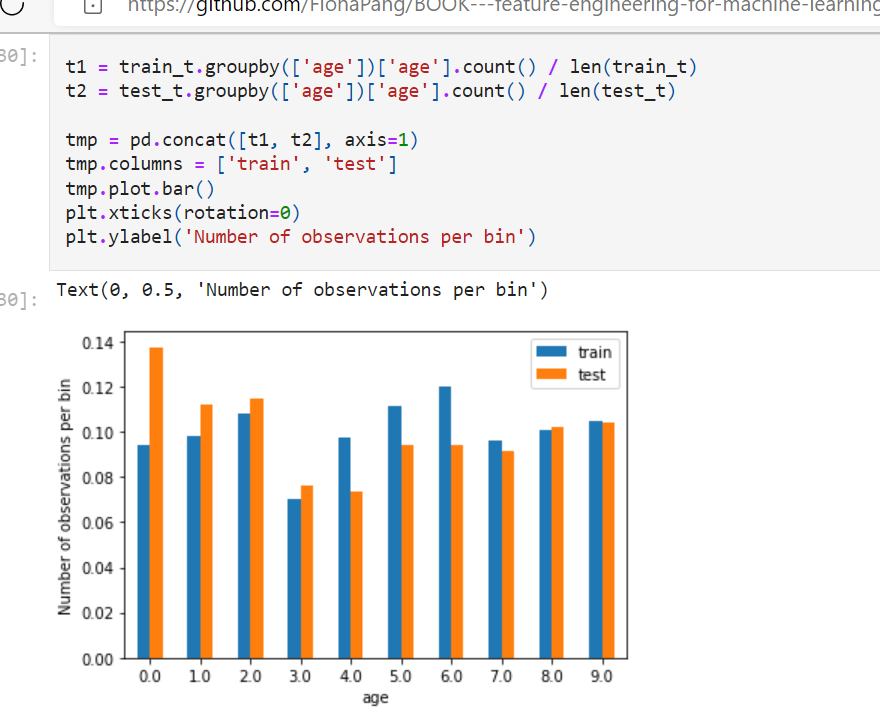




**Better value spread**







**Discretisation with k-means clustering**

This discretisation method consists in **applying k-means clustering to the continuous variable.**

Briefly, the algorithm works as follows:

* 1) Initialization: random creation of **K centers**
* 2) Each data point is associated with the closest center
* 3) Each center position is re-computed as the center of its associated points

Steps 2 and 3 are repeated until convergence is reached. The algorithm minimises the pairwise squared deviations of points within the same cluster.

More details about k-means [here](https://en.wikipedia.org/wiki/K-means_clustering)

Nice blog with graphical explanation of k-means [here](https://towardsdatascience.com/how-does-k-means-clustering-in-machine-learning-work-fdaaaf5acfa0)

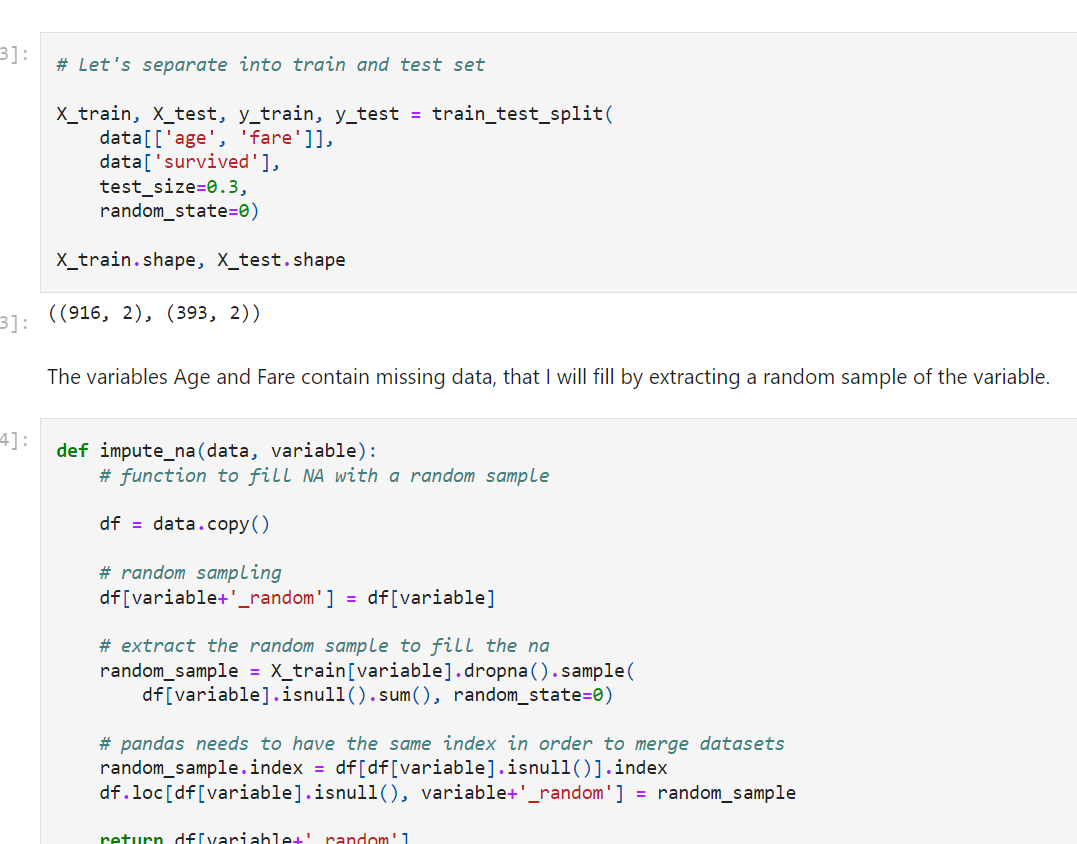
Note that the user, **needs to define the number of clusters**, as with **equal width and equal frequency discretisation.**

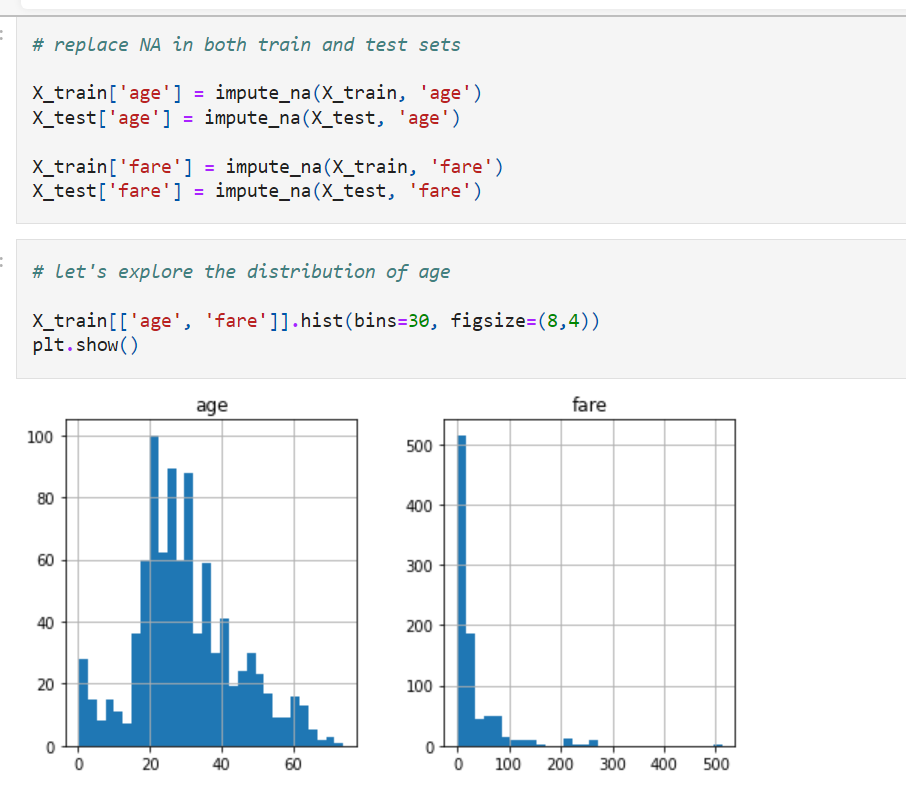
**Opinion of the instructor**

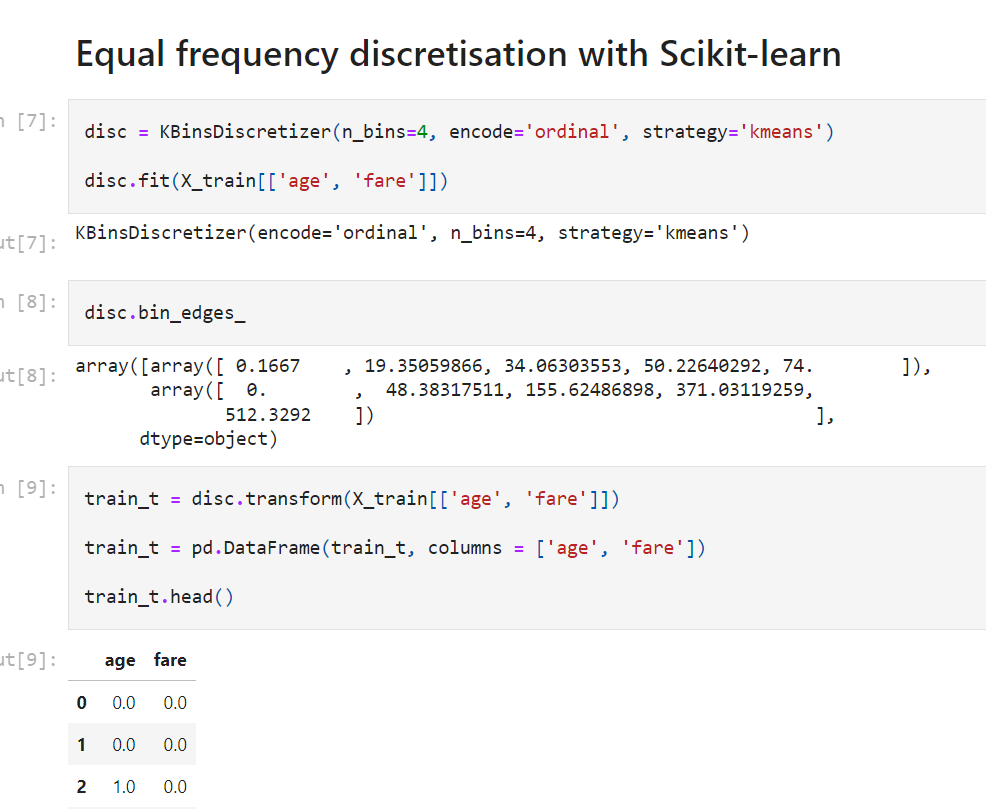
I personally don't see how this technique is different from equal width discretisation, when the variables are continuous throughout the value range. Potentially it would make a different if the values were arranged in real clusters.

So my recommendation is, unless you have reasons to believe that the values of the variable are organised in clusters, then use equal width discretisation as an alternative to this method.

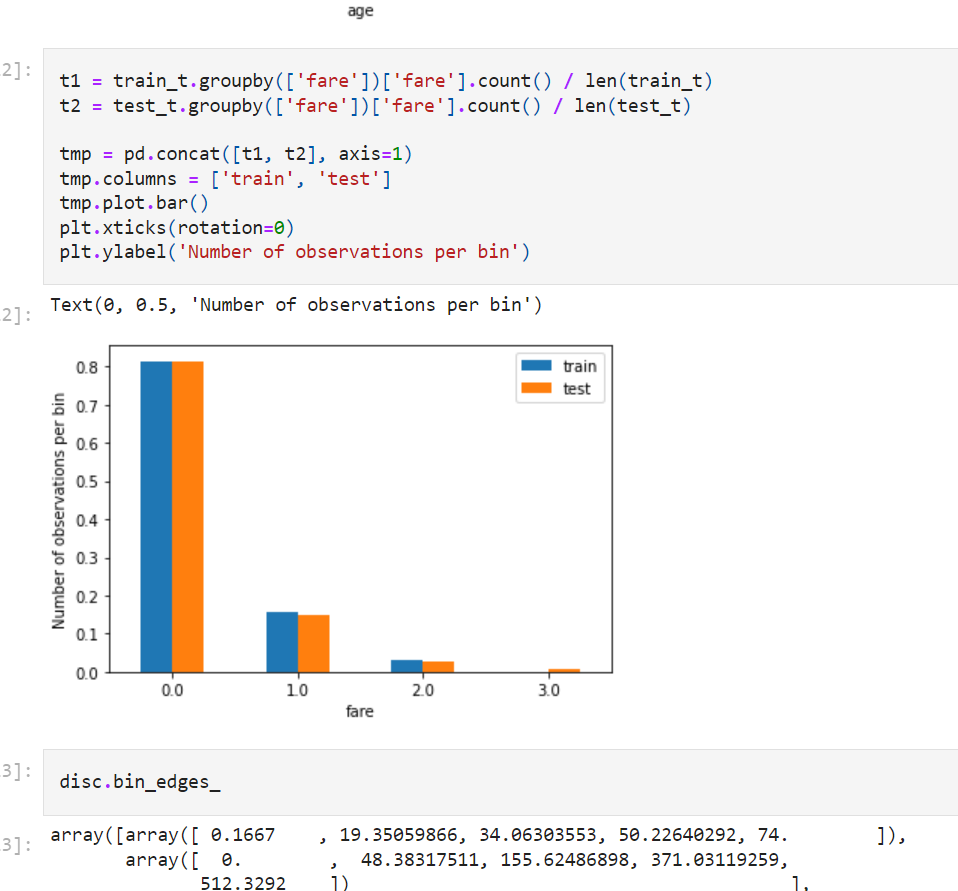


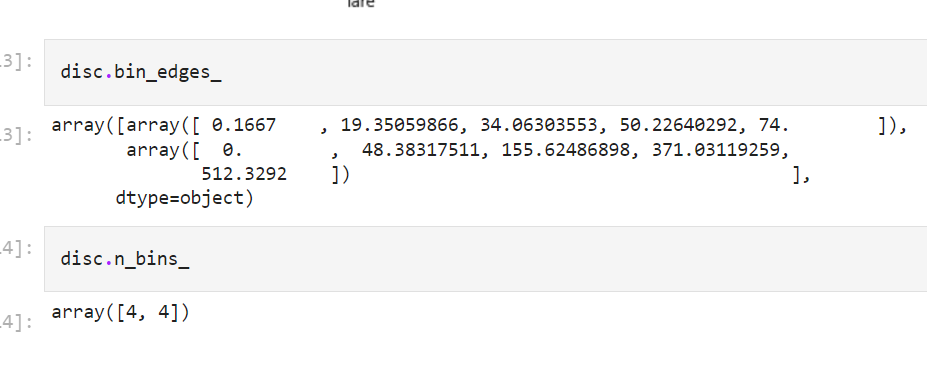












**Discretisation plus Encoding**

What shall we do with the variable after discretisation? should we use the buckets as a numerical variable? or **should we use the intervals as categorical variable?**

The answer is**, you can do either.**

If you are building

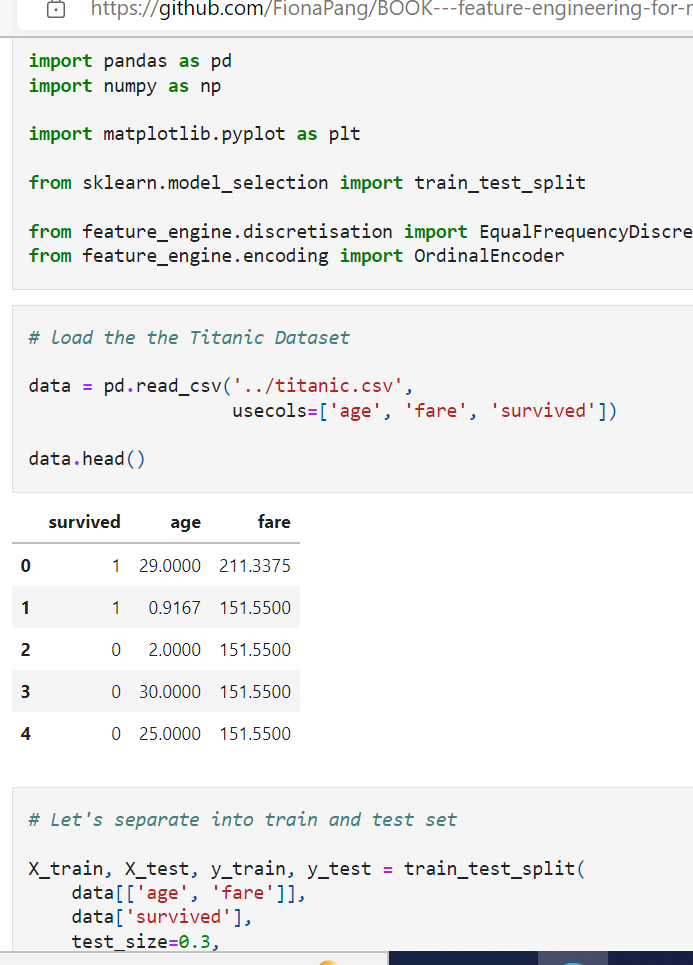
decision tree based algorithms and the **output of the discretisation are integers** (each integer referring to a bin), then you can use those directly, as **decision trees will pick up non-linear relationships** between the discretised variable and the target.

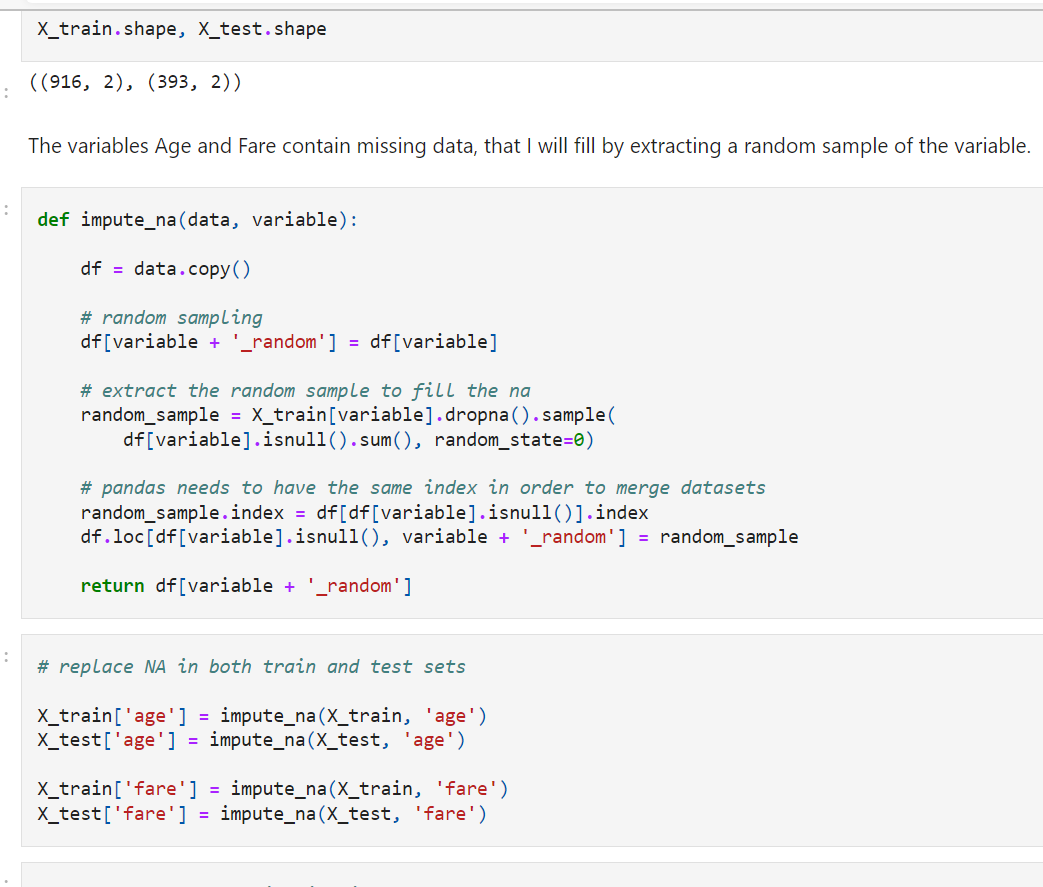
Use normal bins

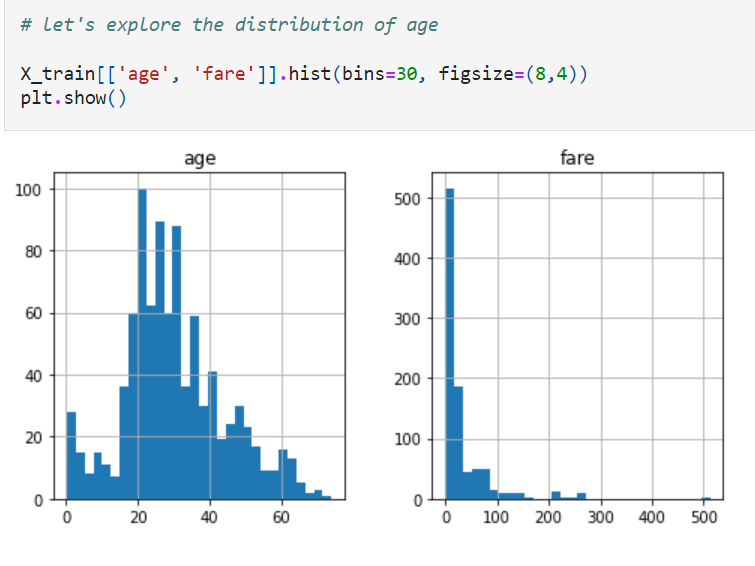
If you are **building linear models instead**, the bins may not necessarily hold a linear relationship with the target. In this case, it may help improve **model performance to treat the bins as categories** and to one hot encoding, or target guided encodings like mean encoding, weight of evidence, or target guided ordinal encoding.

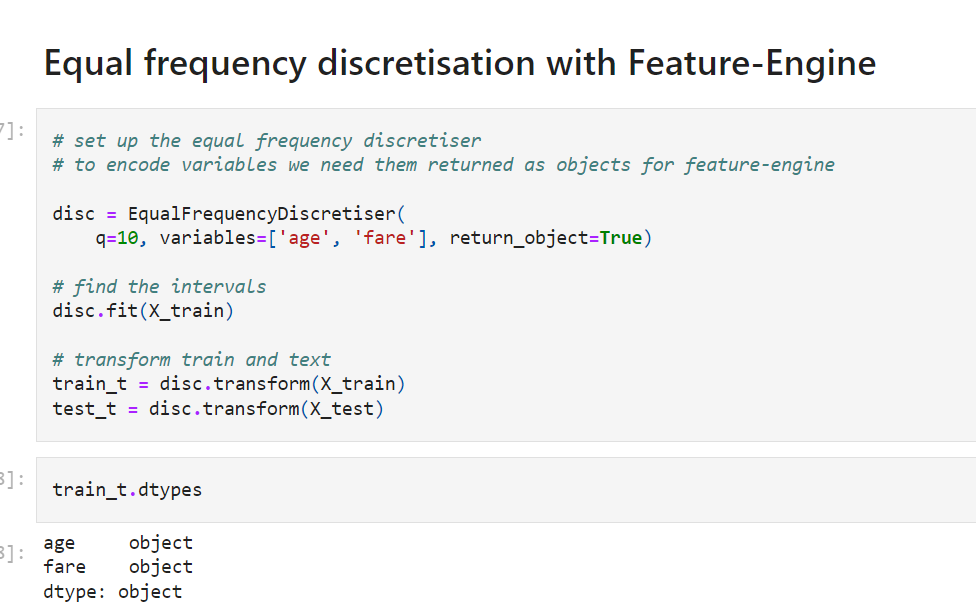
Linear models: do one hot

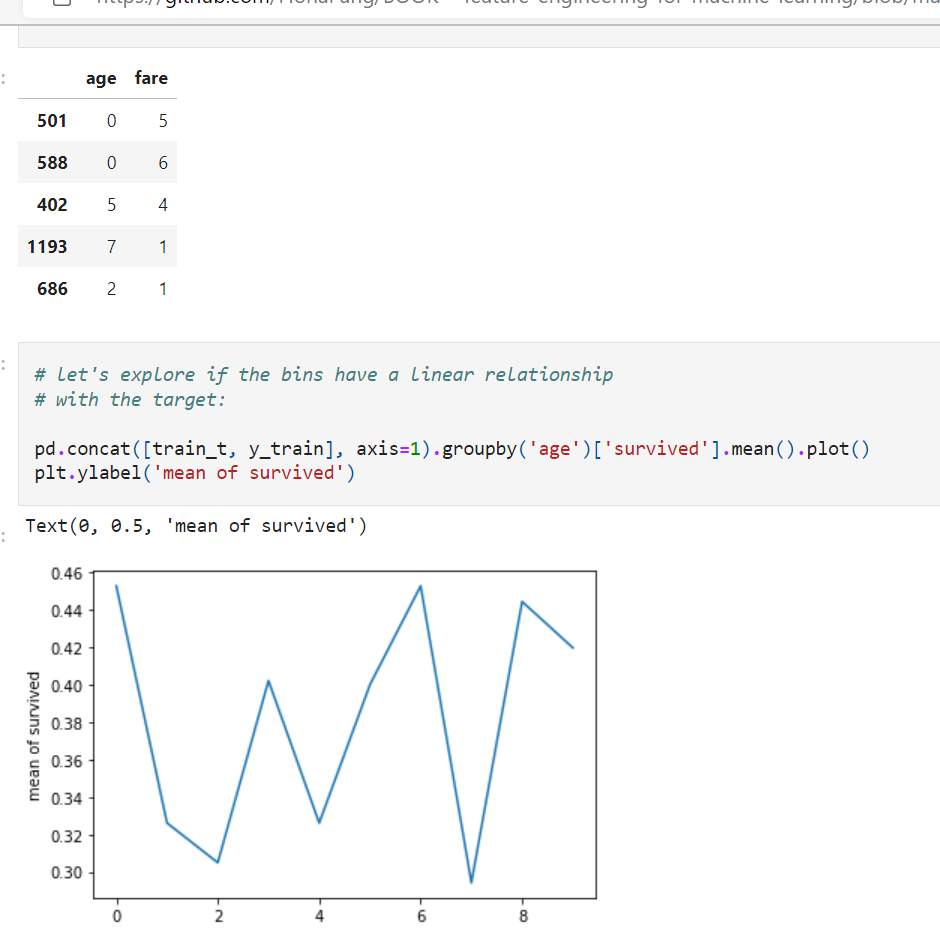
We can easily do so by combining feature-engine's discretisers and encoders.

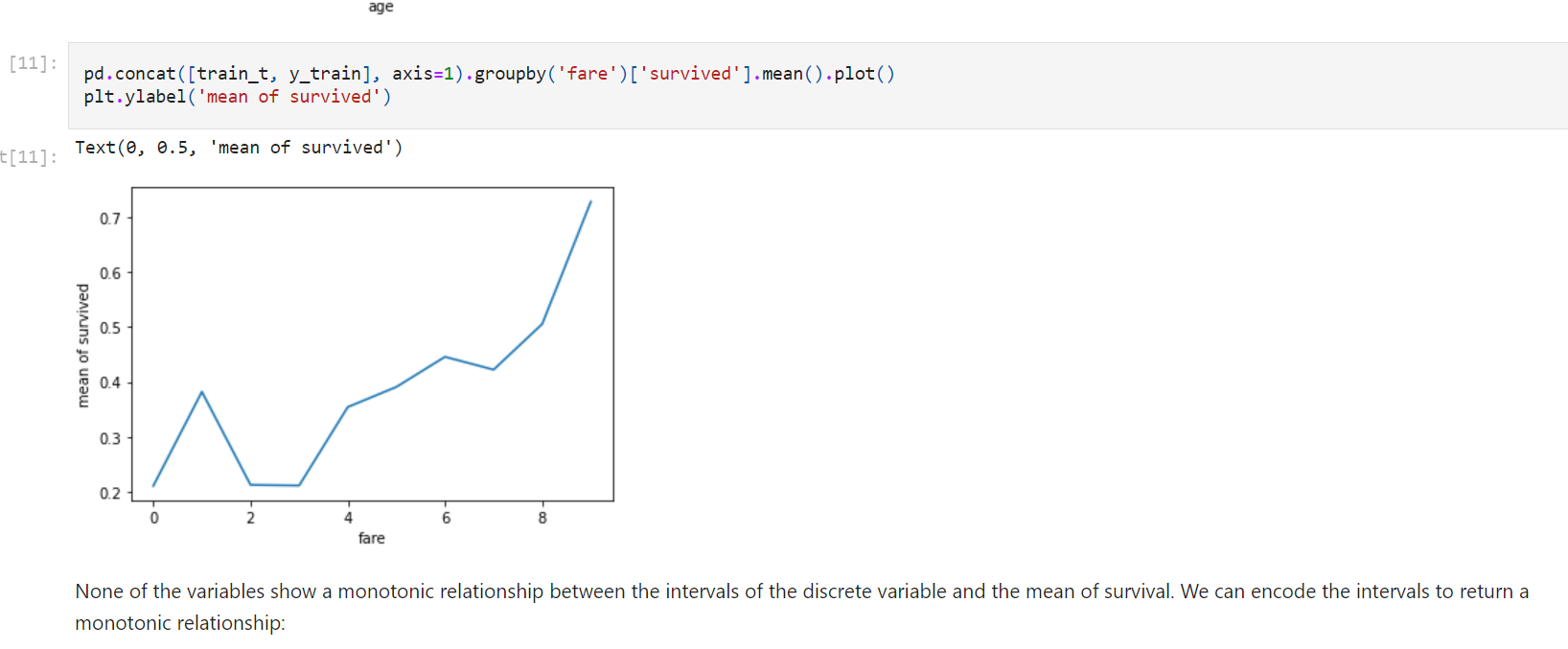


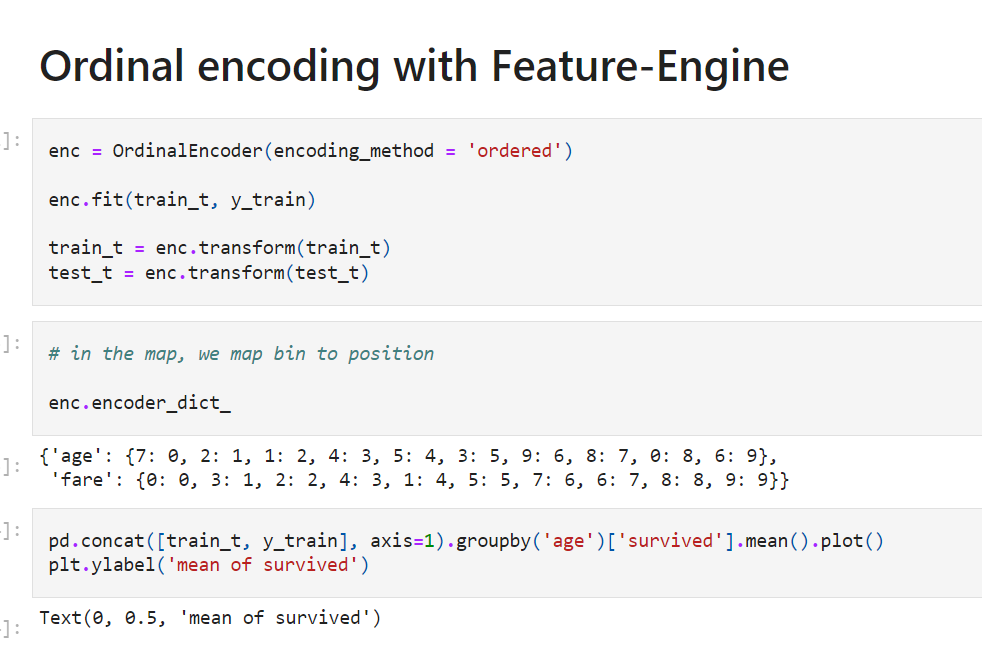


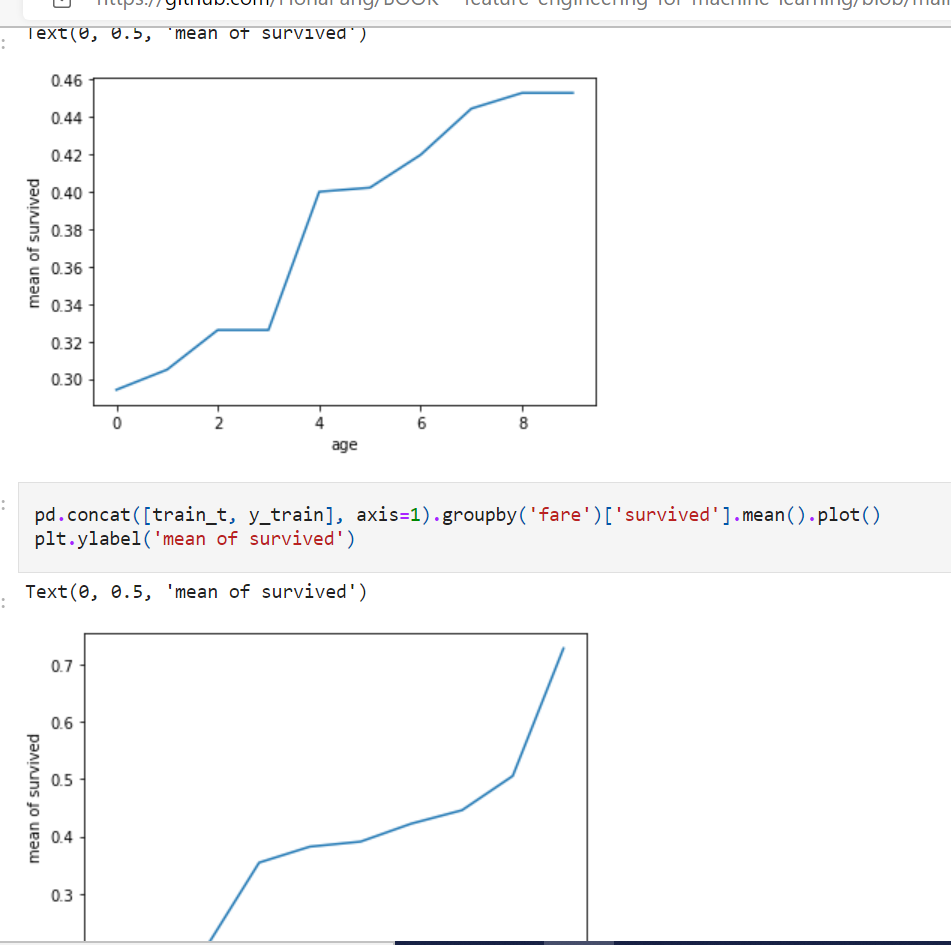


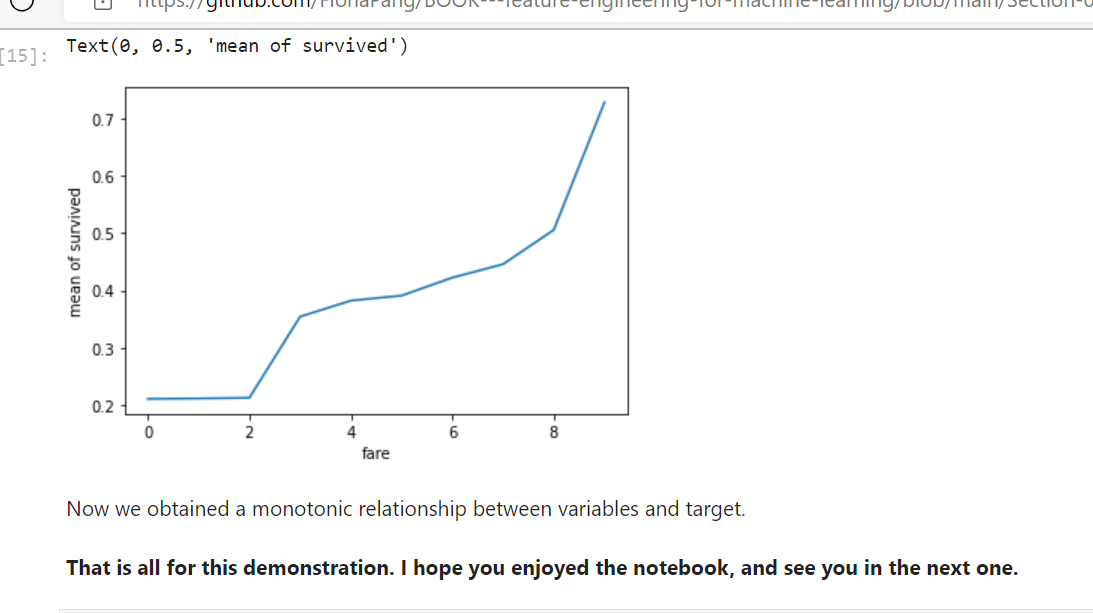












**Discretisation with Decision Trees**

Discretisation with Decision Trees consists in using a **decision tree to identify the optimal bins**.

**When a decision tree makes a decision,** it assigns an **observation to one of n end leaves**.

Therefore, any decision tree will generate a **discrete output,** which values are the predictions at each of its n leaves.

How to do discretisation with trees?

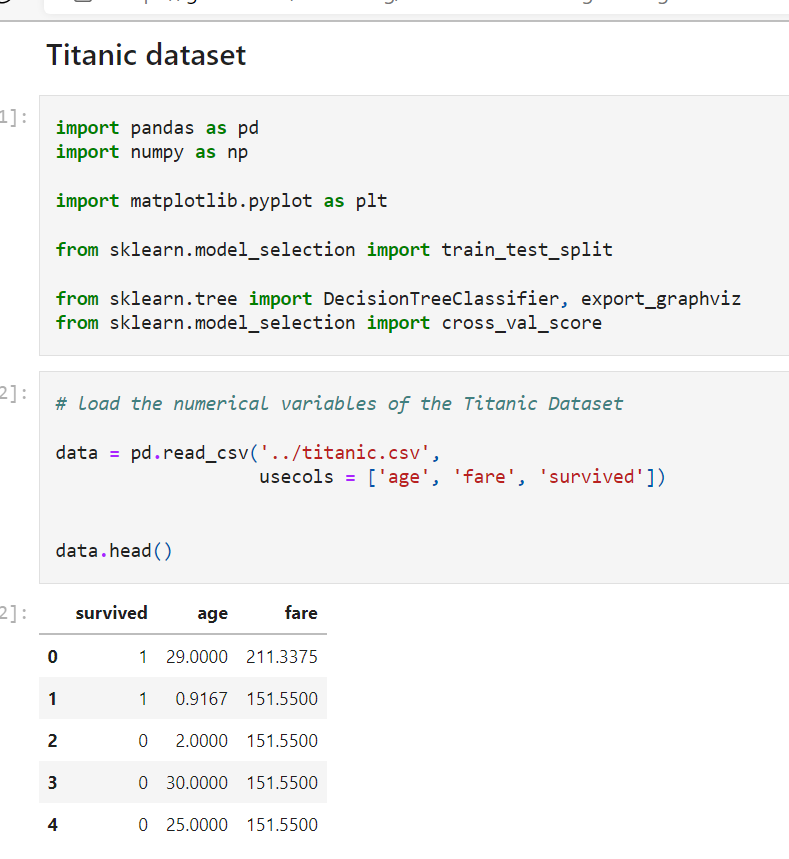
* 1) Train a **decision tree of limited depth (2, 3 or 4)** using the variable we want to discretise and the target.
* 2) **Replace the values by the output returned by the tree.**

**Advantages**

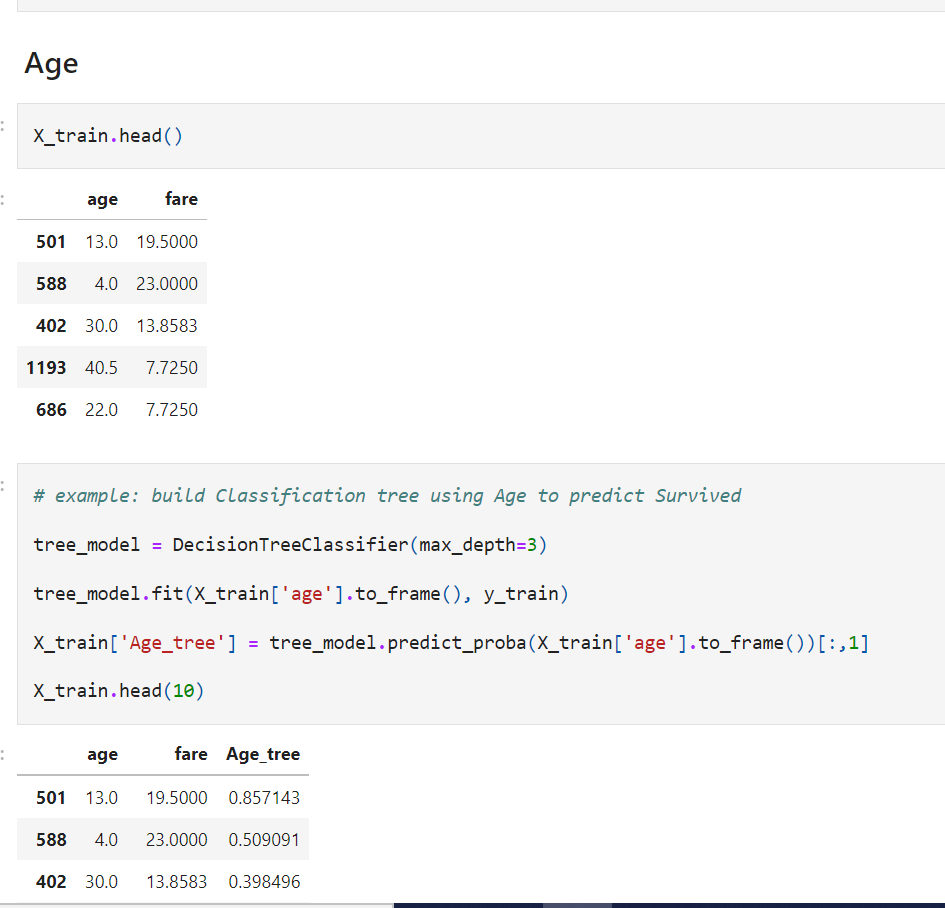
* The output returned by the decision tree is **monotonically related to the target**.
* The **tree end nodes, or bins** in the discretised variable show **decreased entropy**: that is, the **observations within each bin are more similar among themselves** than to those of other bins.

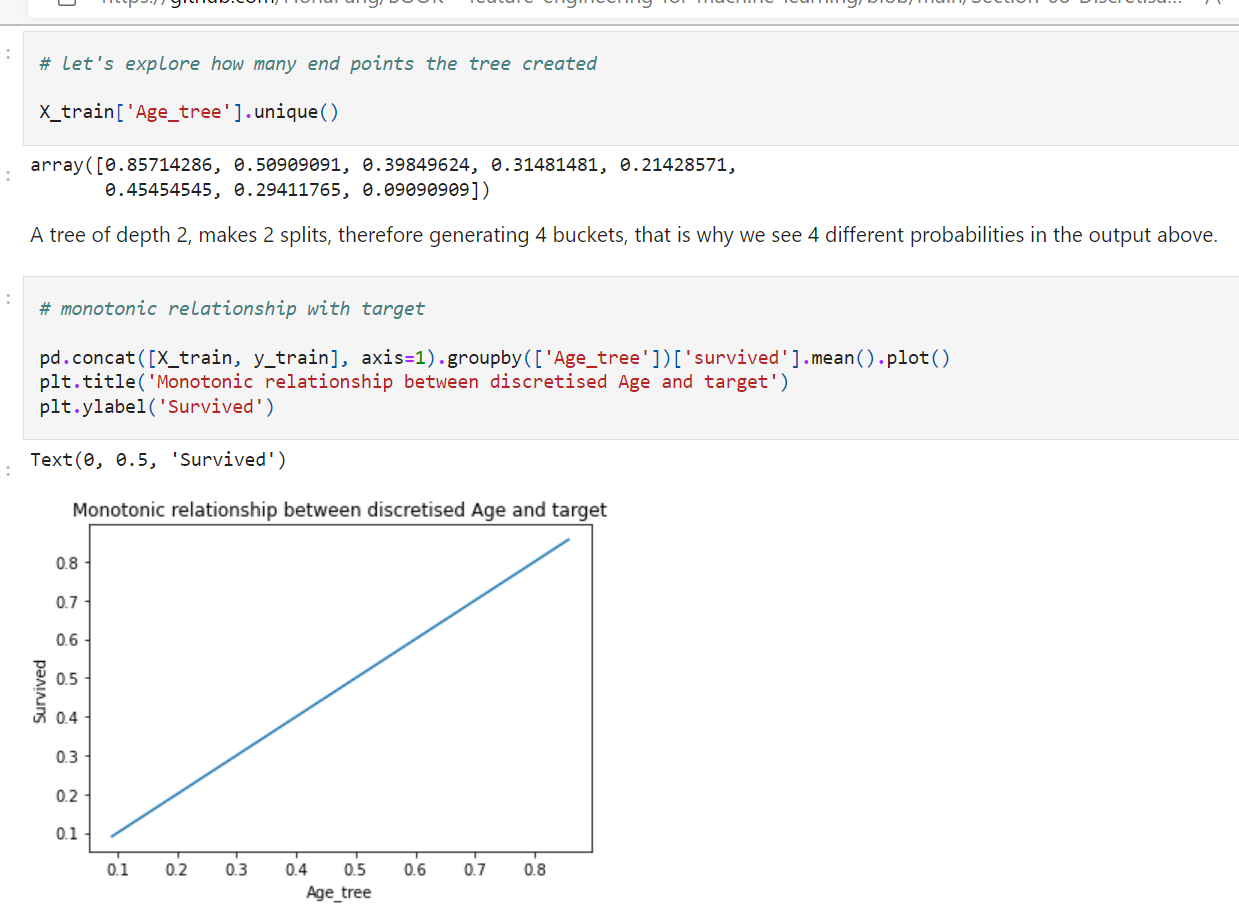
**Limitations**

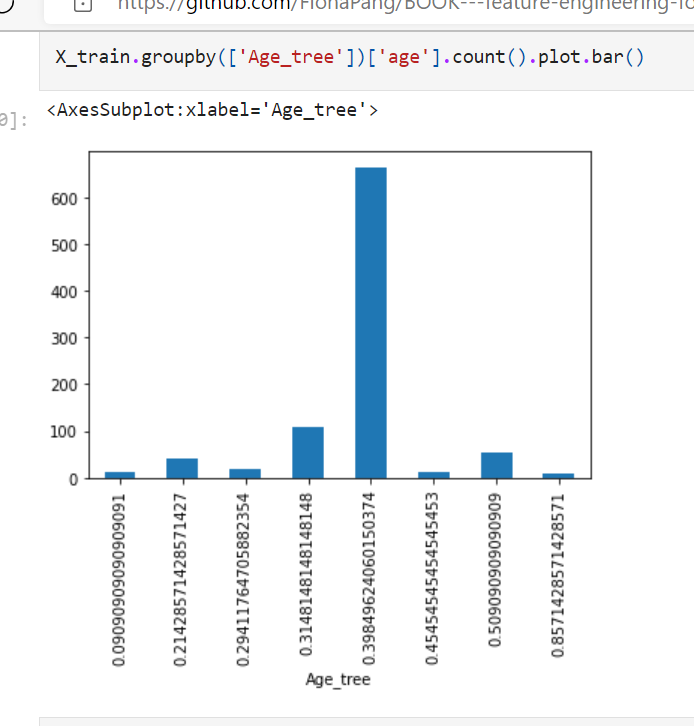
* **Prone over-fitting**
* More importantly, some **tuning of the tree parameters needed** to obtain the **optimal number of splits** (e.g., tree depth, minimum number of samples in one partition, maximum number of partitions, and a minimum information gain). This it can be time consuming.

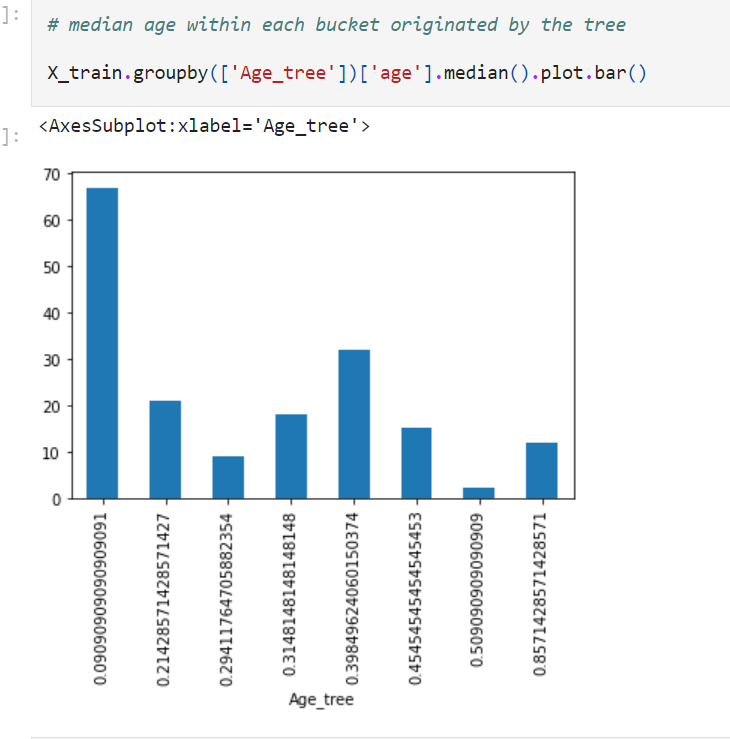


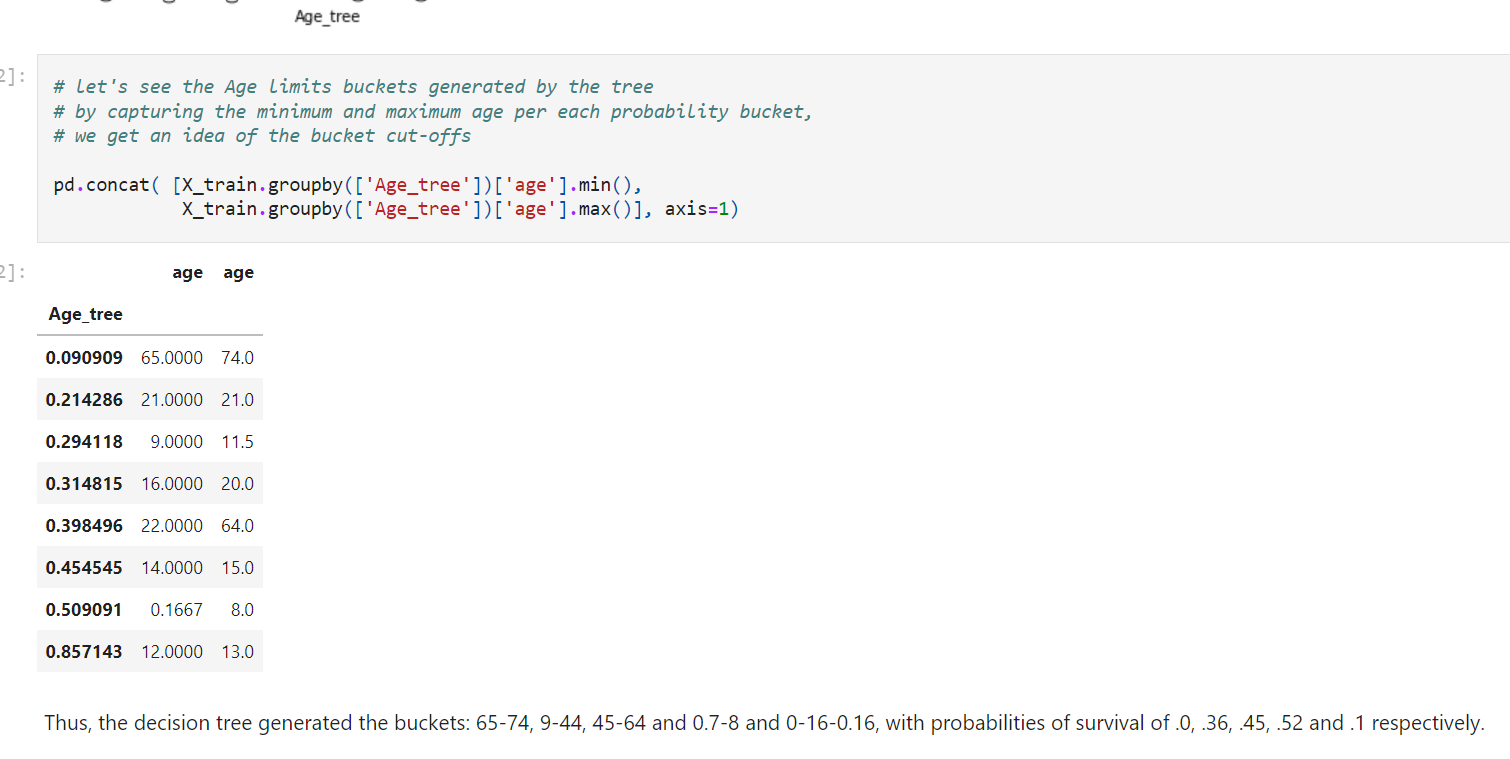


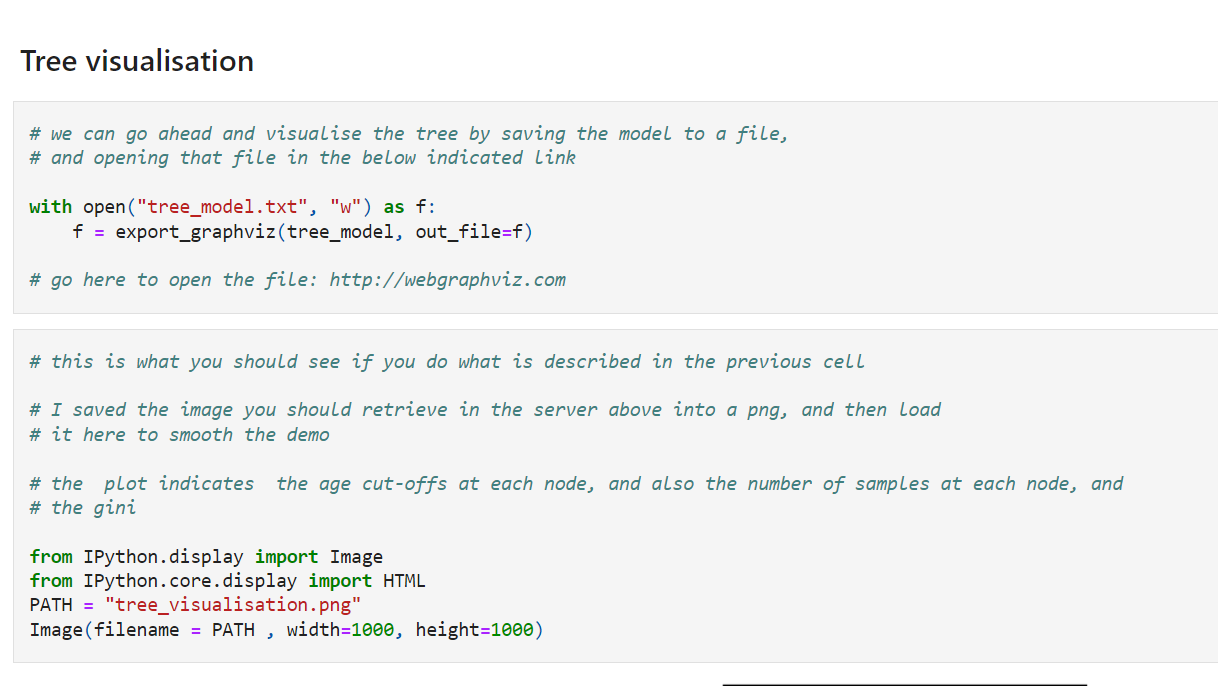


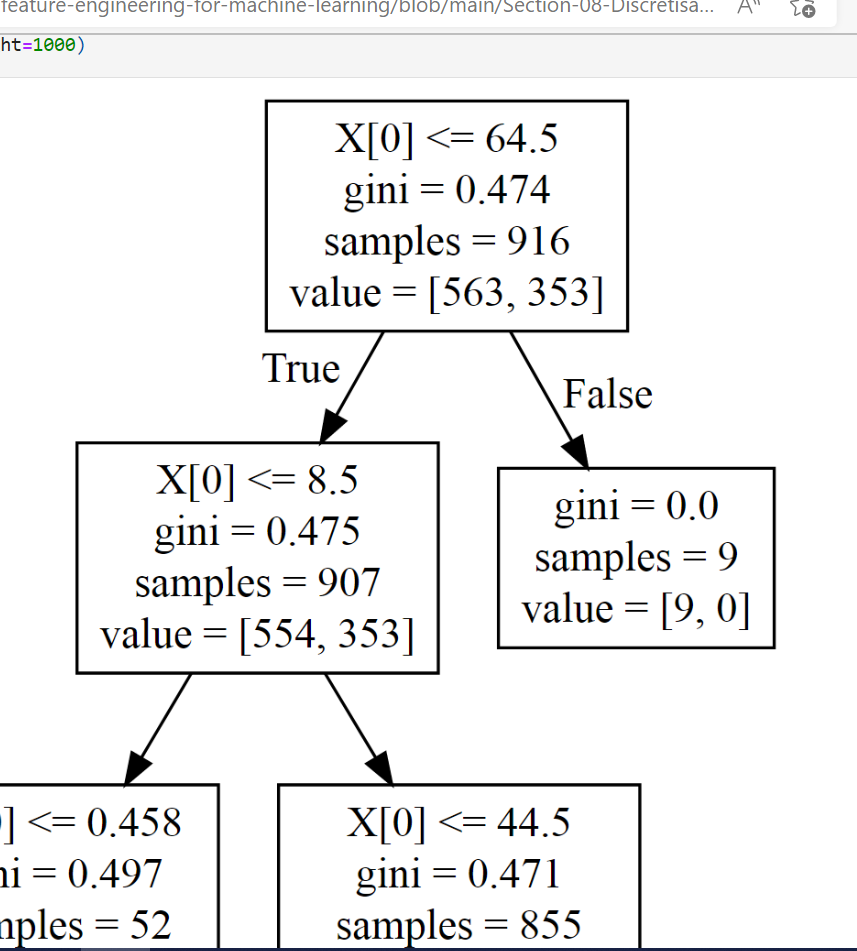


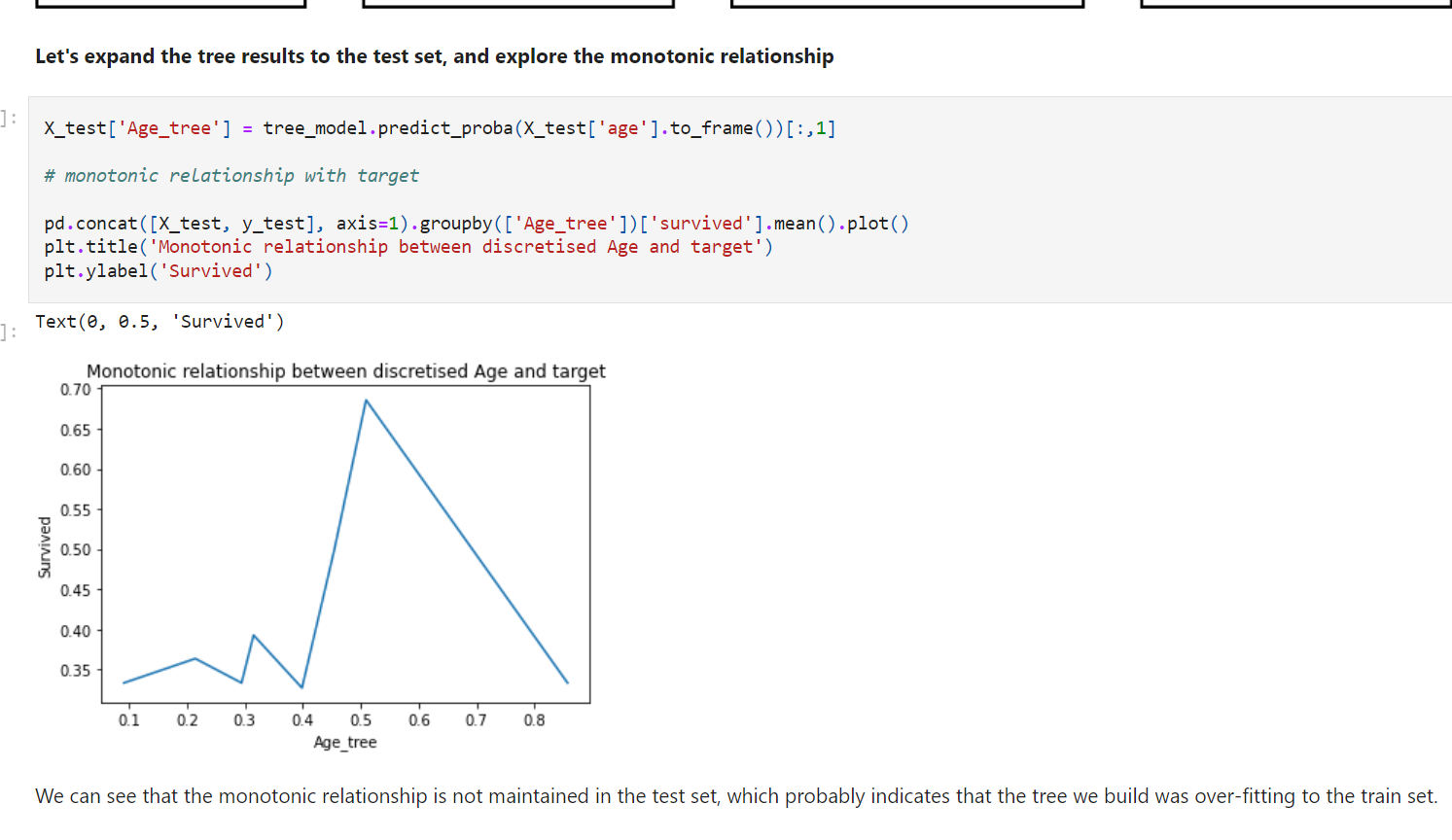












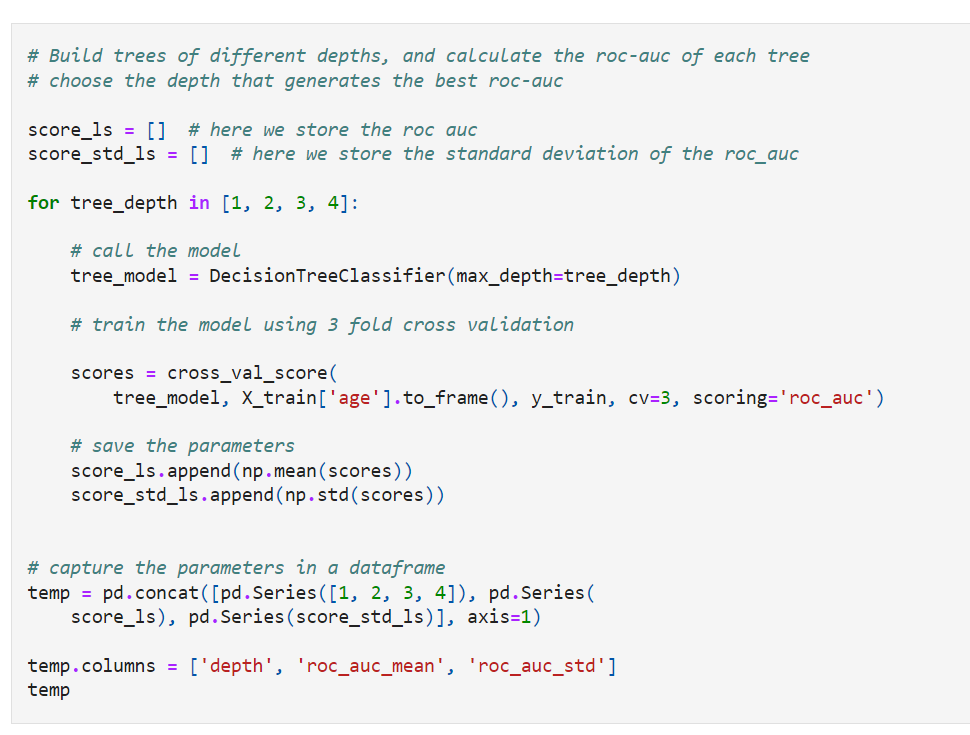
We can see that the monotonic relationship is not maintained in the test set, which probably indicates that the tree we build was over-fitting to the train set.

**Building the optimal decision tree**

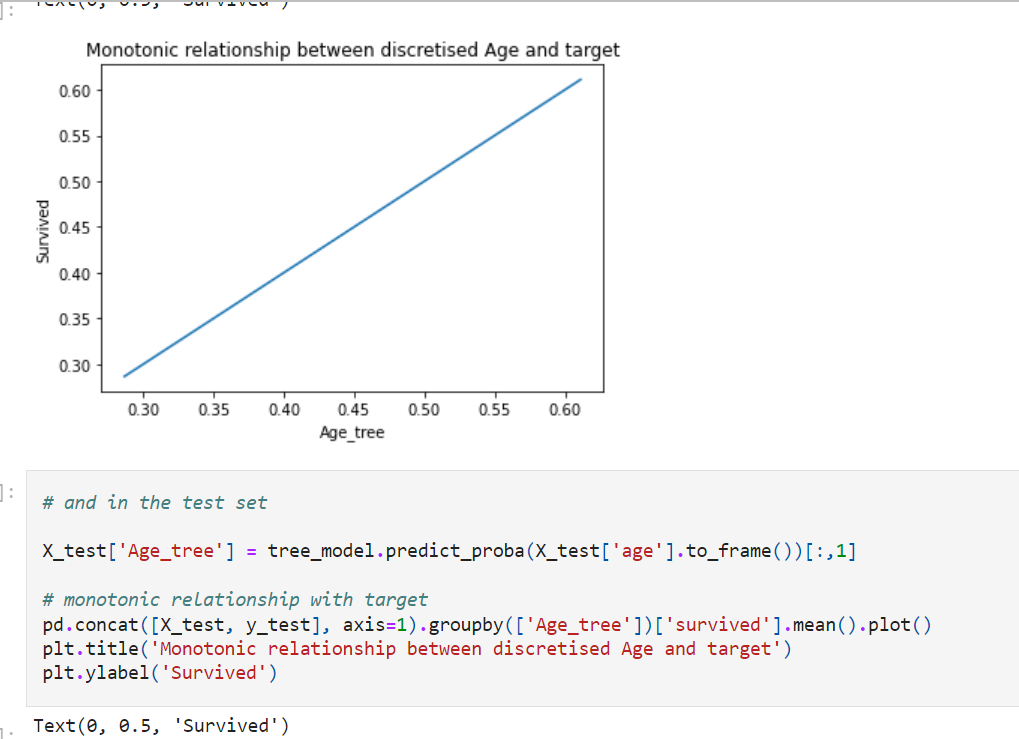
There are a number of parameters that we could optimise to obtain the best bin split using decision trees.

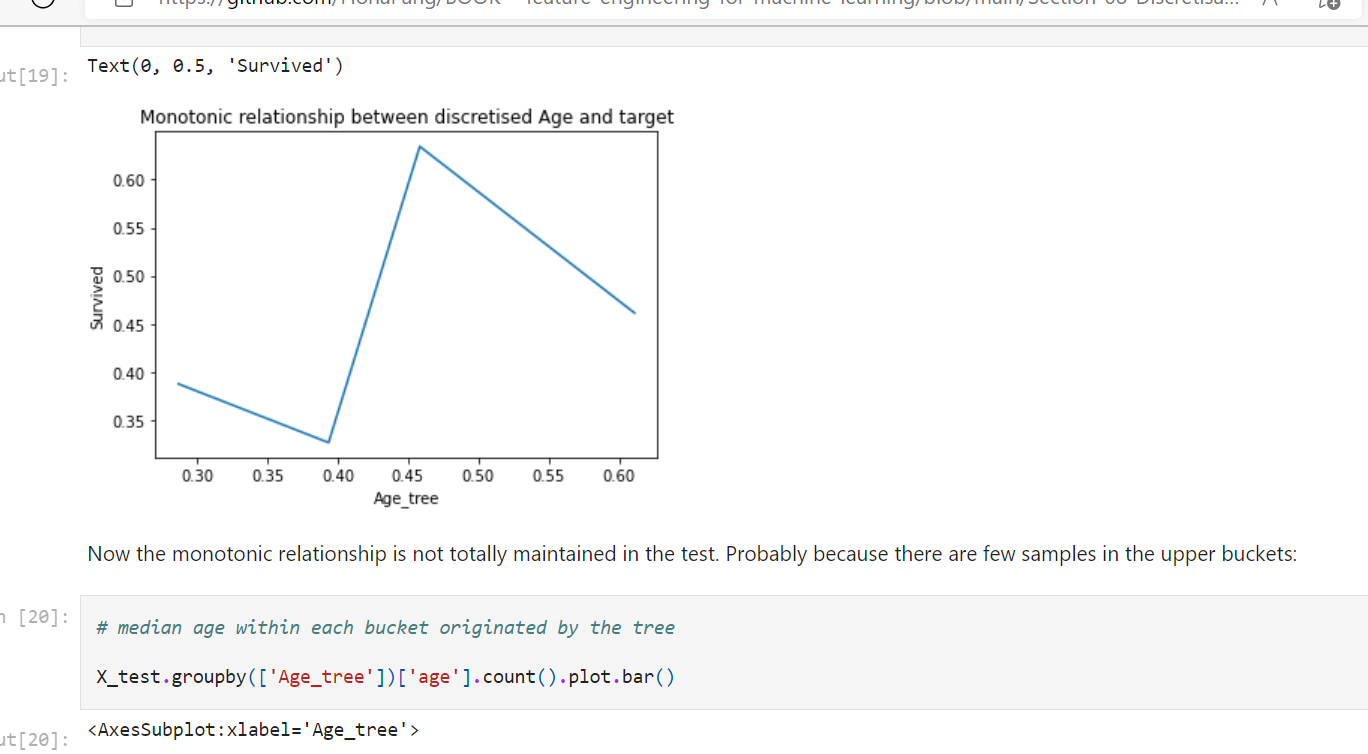
I will optimise the tree depth for this demonstration. But remember that we could also optimise the remaining parameters of the decision tree.

Visit [sklearn website](http://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html" \l "sklearn.tree.DecisionTreeClassifier) to see which other parameters can be optimised.











**Discretisation with Decision Trees using Feature-Engine**

feature Engine allows you to implement decision tree discretisation on all your numerical variables very easily, including search over the multiple parameters of the decision tree, to find the best one.

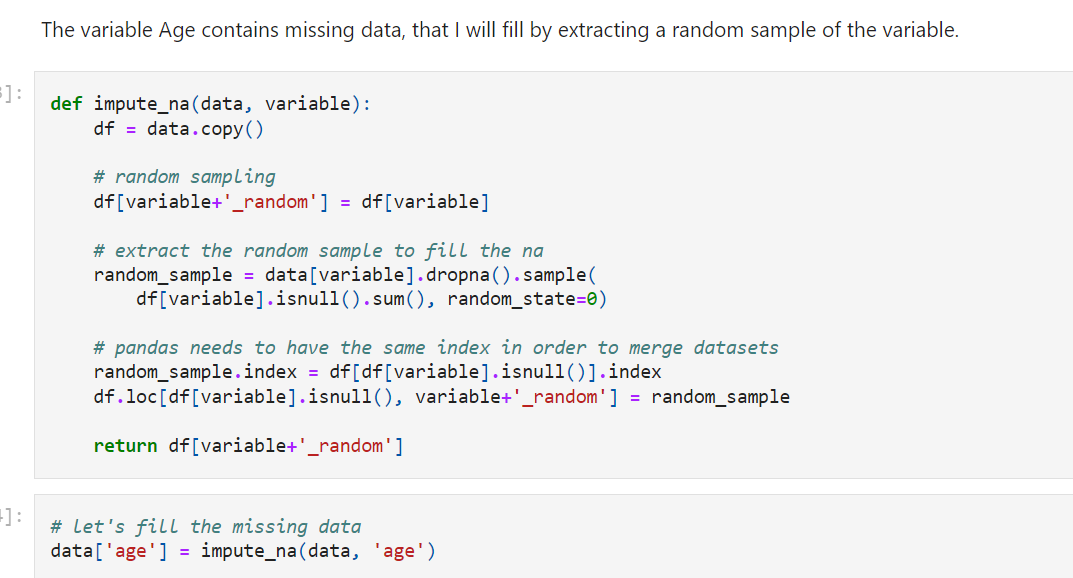
[BOOK---feature-engineering-for-machine-learning/08.06-Discretisation-using-Decision-Trees-and-Feature-Engine.ipynb at main · FionaPang/BOOK---feature-engineering-for-machine-learning · GitHub](https://github.com/FionaPang/BOOK---feature-engineering-for-machine-learning/blob/main/Section-08-Discretisation/08.06-Discretisation-using-Decision-Trees-and-Feature-Engine.ipynb)

**Domain knowledge discretisation**

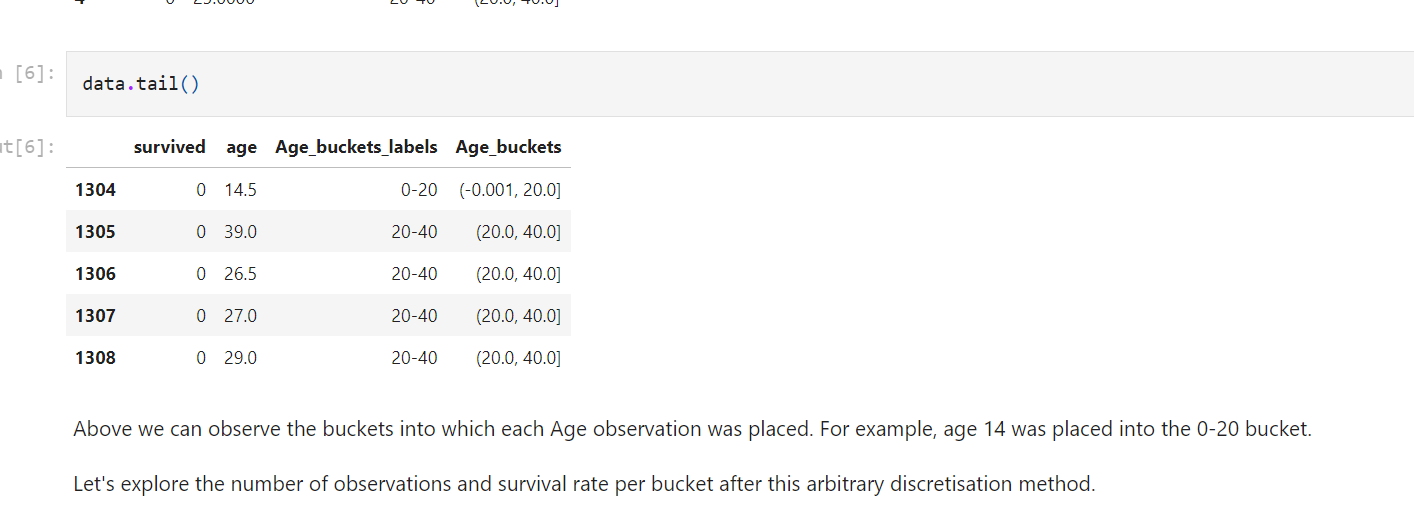
Frequently, when engineering variables in a business setting, the business experts determine the **intervals in which they think the variable should be divided** so that it **makes sense for the business.** Typical examples are the discretisation of variables like **Age and Income.**

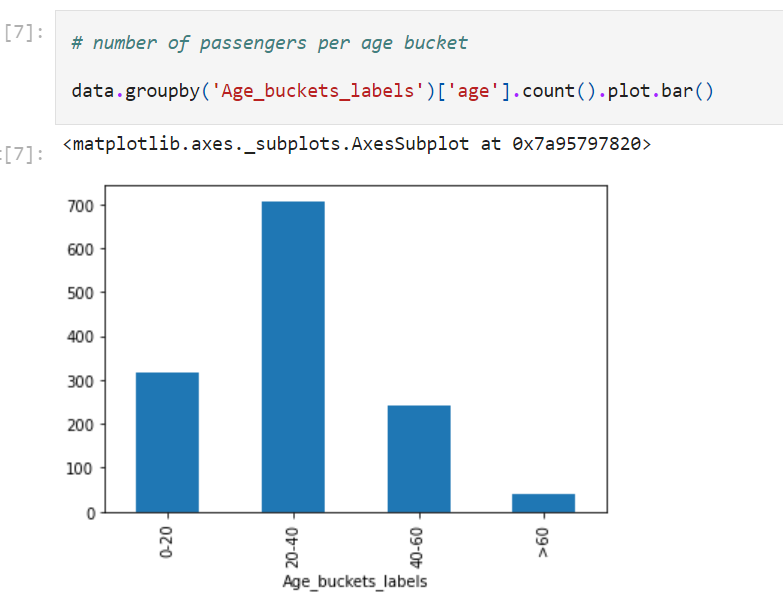
Income for example is usually capped at **a certain maximum value**, and **all incomes above that value fall into the last bucket**. As per Age, it is usually divided in certain groups according to the business need, for example division **into 0-21 (for under-aged), 20-30 (for young adults), 30-40, 40-60, and > 60 (for retired or close to) are frequent.**







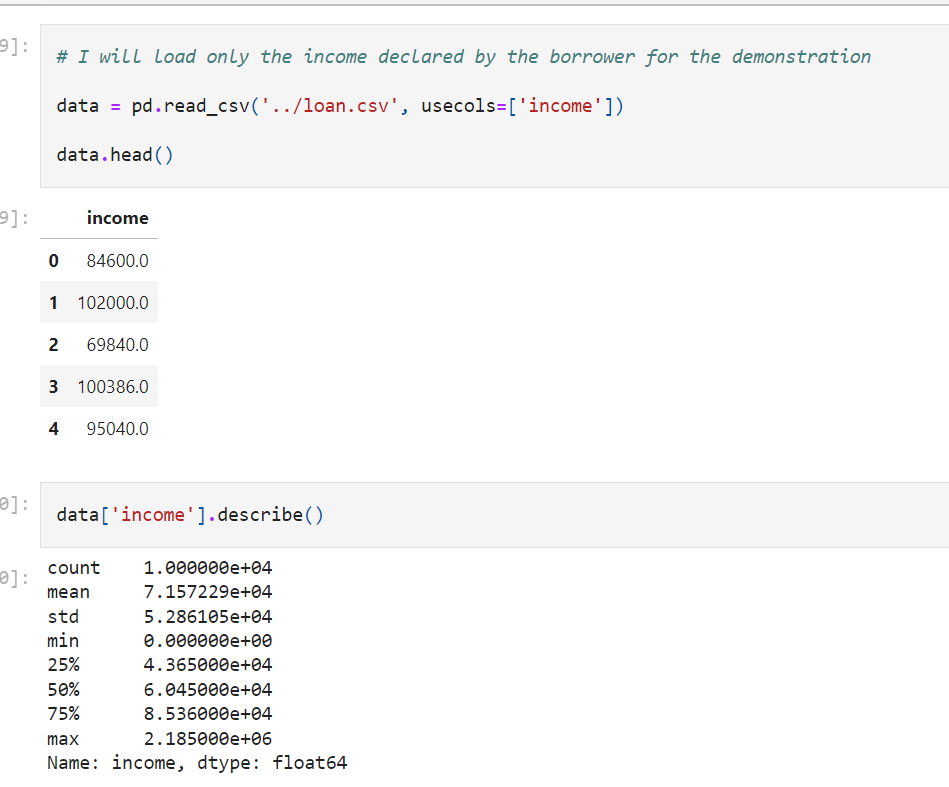


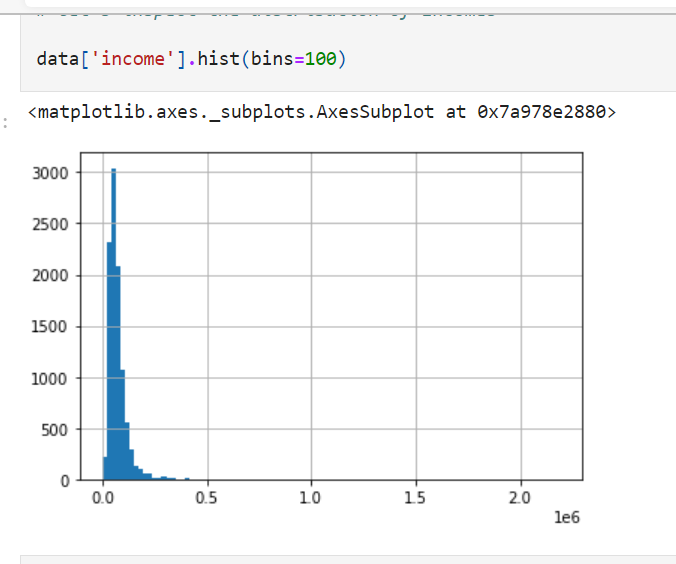


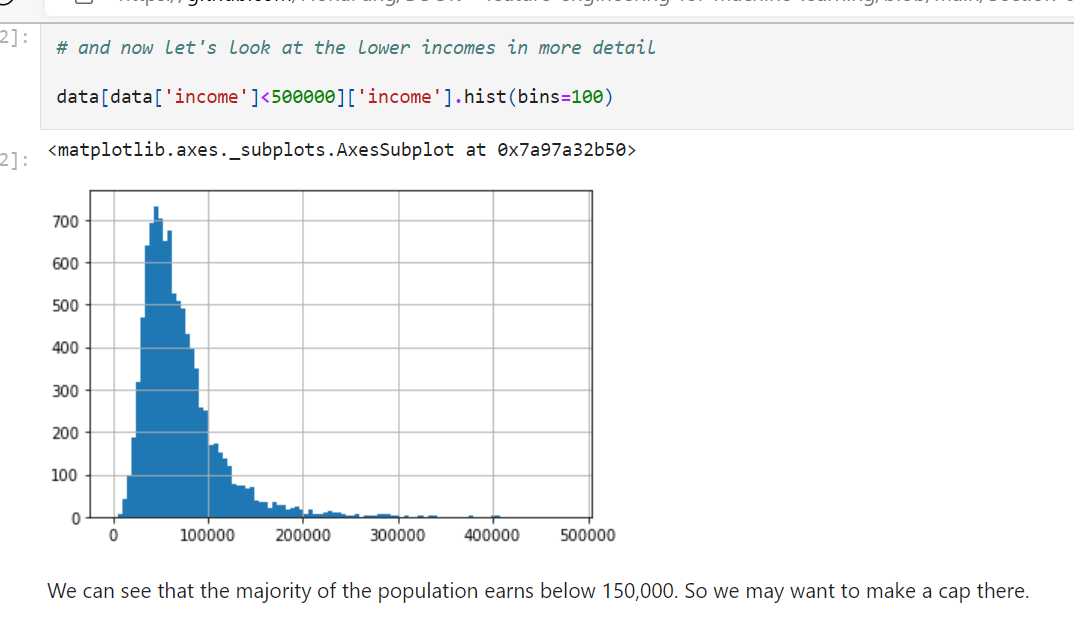


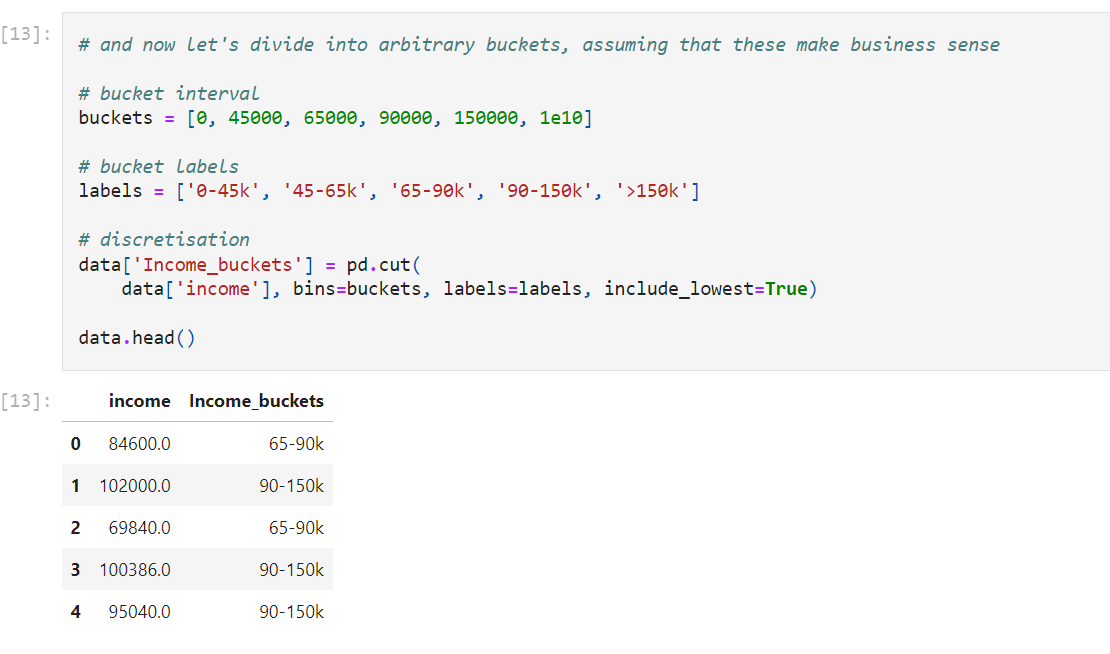
**Peer to peer**

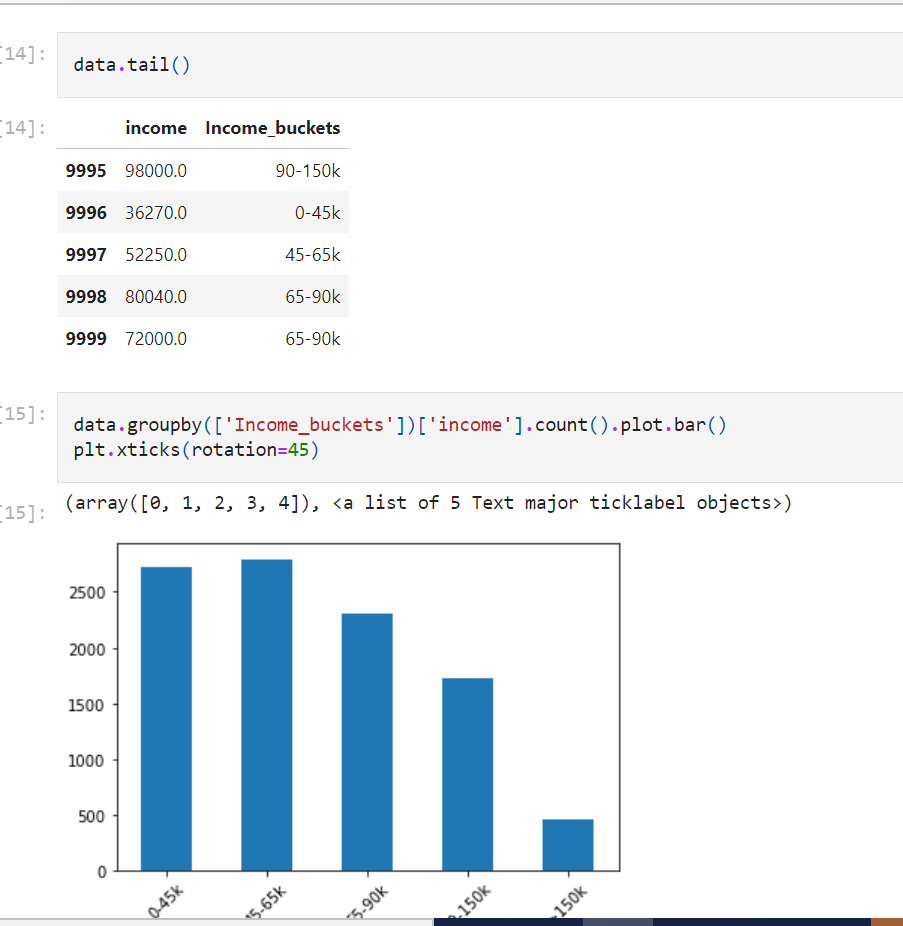
Let's explore discretisation using domain knowledge in a different business scenario. I will use the loan book from the peer to peer lending company. This dataset contains information on loans given to people, and the financial characteristics of those people as well as the loan performance.

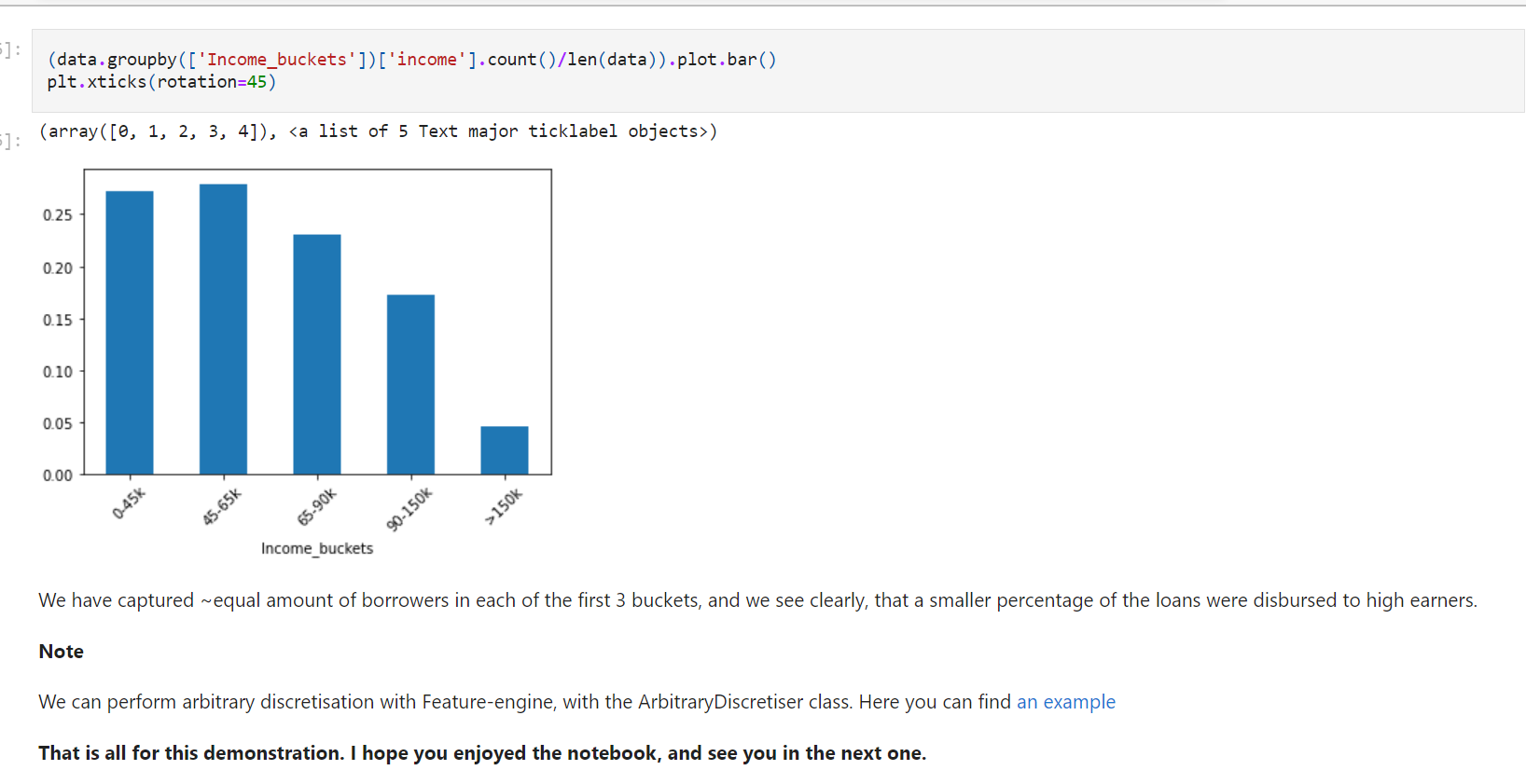












**K means discretization summary:**

**Doent improve value spread**

**Handles outlier**

**Create discrete variables**

**Good to combine with categorical encoding**

**Discretization with decision tree:**

**Doent improve values spread**

**Handle outliers**

**Create discrete variable**

**Creats monotonic relationship**

**Additional reading resources**

Additional reading resources on Discretisation

Articles

* [Discretisation: An Enabling Technique](http://www.public.asu.edu/~huanliu/papers/dmkd02.pdf)
* [Supervised and unsupervised discretisation of continuous features](http://ai.stanford.edu/~ronnyk/disc.pdf)
* [ChiMerge: Discretisation of Numeric Attributes](https://www.aaai.org/Papers/AAAI/1992/AAAI92-019.pdf)

Master thesis

* [Beating Kaggle the easy way](https://www.ke.tu-darmstadt.de/lehre/arbeiten/studien/2015/Dong_Ying.pdf)

Blog

* [Tips for honing logistic regression models](https://blog.zopa.com/2017/07/20/tips-honing-logistic-regression-models/)
* [ChiMerge discretisation algorithm](https://alitarhini.wordpress.com/2010/11/02/chimerge-discretization-algorithm/)

Other

* [Score card development stages: Binning](https://plug-n-score.com/learning/scorecard-development-stages.htm#Binning)