Computational efficiency Reading csv file of +2GB

I tried with 4 different approaches for data ingestion: pandas, modin, dask, and vaex. Dask showed the better performance when loading the csv file that was chosen for this task. It is possible that converting the file to other format before working with it would result in a better performance of other of the approached tested. But since we were asked to work with an csv file the better option for loading it is to use dask. Below I added some snapshots of the performance of each approach.

1. Pandas

```
In [1]: import pandas as pd
```

```
In [2]: %%time

        df = pd.read_csv('arboles.csv')
        df.head()
```

```
Wall time: 20.7 s
```

Out[2]:

	Unnamed: 0	diametro	longitud	especie
0	1	0.270126	6.358027	abedul
1	2	0.268801	7.385486	abedul
2	3	0.299461	6.195804	abedul
3	4	0.296035	9.254137	abedul
4	5	0.223520	10.240710	abedul

2. Dask

```
In [1]: import dask.dataframe as dd
```

```
In [2]: %%time

        df = dd.read_csv('arboles.csv')
        df.head()
```

```
Wall time: 742 ms
```

Out[2]:

	Unnamed: 0	diametro	longitud	especie
0	1	0.270126	6.358027	abedul
1	2	0.268801	7.385486	abedul
2	3	0.299461	6.195804	abedul
3	4	0.296035	9.254137	abedul
4	5	0.223520	10.240710	abedul

3. Modin

```
In [1]: import modin.pandas as pd
```

```
In [2]: from distributed import Client
        client = Client()
```

```
In [3]: %%time

        df = pd.read_csv('arboles.csv')
        df.head()
```

```
Wall time: 11.3 s
```

Out[3]:

	Unnamed: 0	diametro	longitud	especie
0	1	0.270126	6.358027	abedul
1	2	0.268801	7.385486	abedul
2	3	0.299461	6.195804	abedul
3	4	0.296035	9.254137	abedul
4	5	0.223520	10.240710	abedul

4. Vaex

In [1]:
```
import vaex
```

In [2]:
```
%%time

df = vaex.from_csv('arboles.csv')
df.head()
```

```
Wall time: 30.8 s
```

Out[2]:

#	Unnamed: 0	diametro	longitud	especie
0	1	0.270126	6.35803	'abedul'
1	2	0.268801	7.38549	'abedul'
2	3	0.299461	6.1958	'abedul'
3	4	0.296035	9.25414	'abedul'
4	5	0.22352	10.2407	'abedul'
5	6	0.331217	12.2544	'abedul'
6	7	0.307492	8.73223	'abedul'
7	8	0.372486	6.25044	'abedul'
8	9	0.320556	10.0489	'abedul'
9	10	0.286591	9.80952	'abedul'