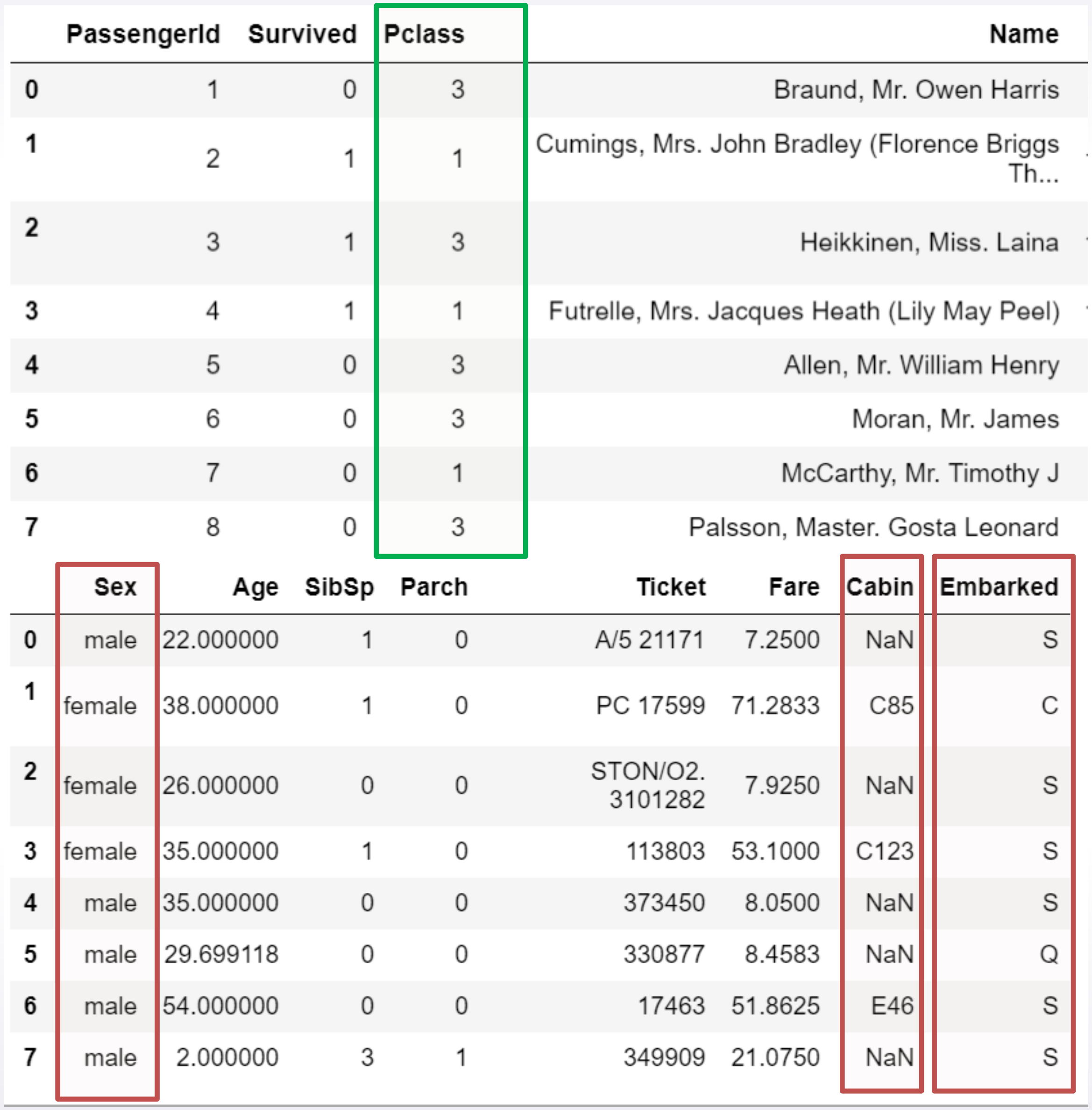
Categorical and ordinal features

Categorical

Titanic dataset



	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.000000	1	0	A/5 21171	7.2500	NaN	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.000000	1	0	PC 17599	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.000000	0	0	STON/O2. 3101282	7.9250	NaN	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.000000	1	0	113803	53.1000	C123	S
4	5	0	3	Allen, Mr. William Henry	male	35.000000	0	0	373450	8.0500	NaN	S
5	6	0	3	Moran, Mr. James	male	29.699118	0	0	330877	8.4583	NaN	Q
6	7	0	1	McCarthy, Mr. Timothy J	male	54.000000	0	0	17463	51.8625	E46	S
7	8	0	3	Palsson, Master. Gosta Leonard	male	2.000000	3	1	349909	21.0750	NaN	S

Ordinal features

Ticket class: 1,2,3

Driver's license: A, B, C, D

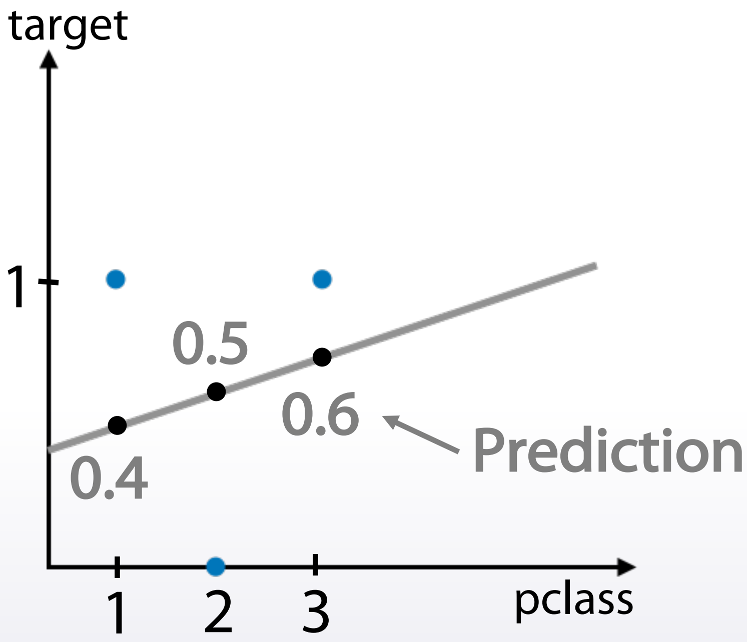
Education: kindergarden, school, undergraduate, bachelor, master, doctoral

Label encoding

pclass	1	2	3
target	1	0	1

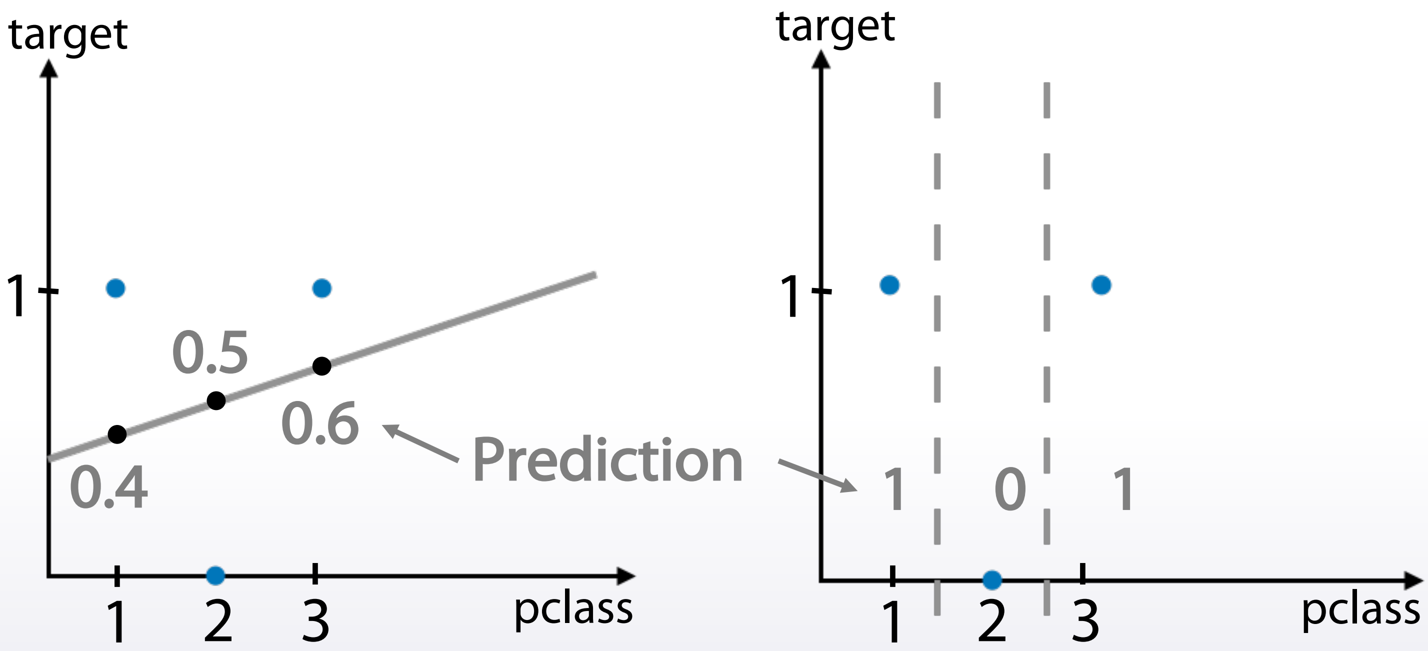
Label encoding

pclass	1	2	3
target	1	0	1

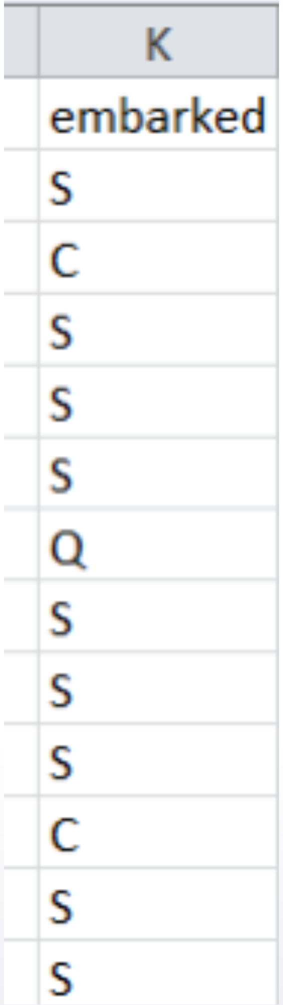


Label encoding

pclass	1	2	3
target	1	0	1



Label encoding



K
embarked
S
C
S
S
S
Q
S
S
S
C
S
S

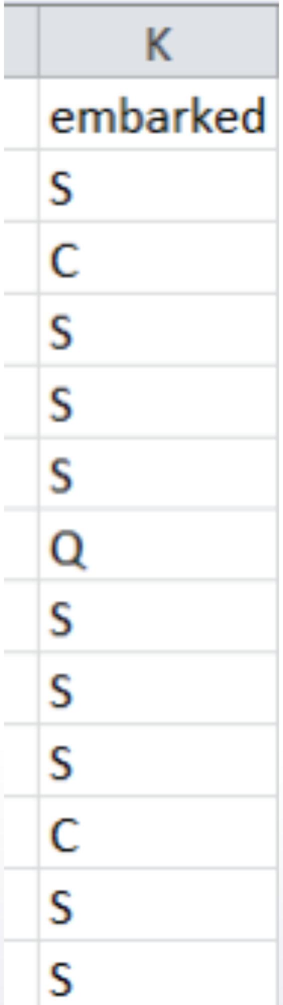
1. Alphabetical (sorted)
   [S,C,Q] -> [2, 1, 3]

   ```
   sklearn.preprocessing.LabelEncoder
   ```

2. Order of appearance
   [S,C,Q] -> [1, 2, 3]

   ```
   Pandas.factorize
   ```

Frequency encoding

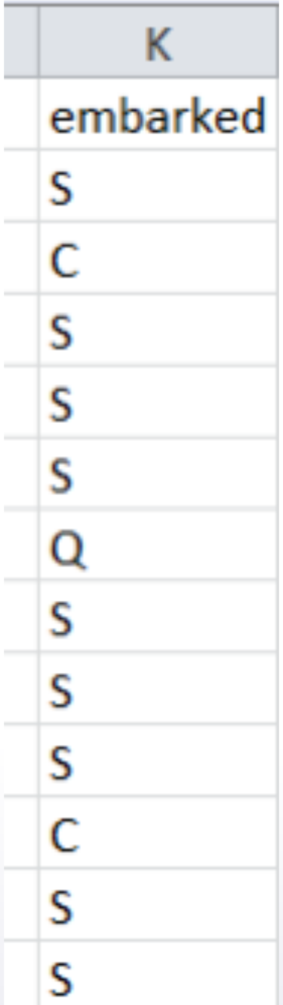


K
embarked
S
C
S
S
S
Q
S
S
S
C
S
S

[S,C,Q] -> [0.5, 0.3, 0.2]

```
encoding = titanic.groupby(‘Embarked’).size()
encoding = encoding/len(titanic)
titanic[‘enc’] = titanic.Embarked.map(encoding)
```

Frequency encoding



K
embarked
S
C
S
S
S
Q
S
S
S
C
S
S

[S,C,Q] -> [0.5, 0.3, 0.2]

```
encoding = titanic.groupby(‘Embarked’).size()
encoding = encoding/len(titanic)
titanic[‘enc’] = titanic.Embarked.map(encoding)
```

```
 from scipy.stats import rankdata
```

Categorical features

So here, for each unique value of Pclass feature, we just created a new column. As I said, this works well for linear methods, kNN, or neural networks. Furthermore, one -hot encoding feature is already scaled because minimum this feature is zero, and maximum is one. Note that if you care for a fewer important numeric features, and hundreds of binary features are used by one-hot encoding, it could become difficult for tree-methods they use first ones efficiently.

One-hot encoding

pclass
1
2
1
3

pclass==1	pclass==2	pclass==3
1		
	1	
1		
		1

`pandas.get_dummies, sklearn.preprocessing.OneHotEncoder`

Sparse matrices are often useful when they work with categorical features or text data. Most of the popular libraries can work with these sparse matrices directly namely, XGBoost, LightGBM, sklearn, and others.

Categorical features

Feature Generation

pclass	sex	pclass_sex
3	male	3male
1	female	1female
3	female	3female
1	female	1female

Pclass_sex==					
1male	1female	2male	2female	3male	3female
				1	
	1				
					1
	1				

Categorical features

1. Values in ordinal features are sorted in some meaningful order
2. Label encoding maps categories to numbers
3. Frequency encoding maps categories to their frequencies
4. Label and Frequency encodings are often used for tree-based models
5. One-hot encoding is often used for non-tree-based models
6. Interactions of categorical features can help linear models and KNN