

## Practical 10

### Jumping Rivers

#### Question 1 - Titanic

We're going to try and better the model prediction survival in the notes (shouldn't be hard!). The following code will load the data in and take a look at it

```
import pandas as pd
import jupyterml
titanic = jupyterml.datasets.load_titanic()
titanic.head()
```

- a) Set up your `X_train` and `y_train` objects such that your response variable is `Survived` and the one predictor variable is `Pclass`.
- b) `Pclass` represents the class of the persons room on the titanic. Should this be a categoric or a numeric variable? What data pre-processing should you therefore be using?
- c) Write a pipeline the preprocesses the data in the correct way, then fits a regression model and then fit the model to your data.
- d) For each class, what is the predicted category of survival and the corresponding probability for that category?
- e) Overall, how many predictions did we get correct?
- f) Of those that survived, what proportion were actually classified that way?
- g) The following code will perform 10-fold cross validation on the data and return the accuracy. Make it return the precision and recall

```
from sklearn.model_selection import cross_validate
from sklearn.metrics import make_scorer
import pandas as pd

acc = make_scorer(accuracy_score)

output = cross_validate(model,X_train,y_train,scoring={
    'acc' : acc
}, cv = 10, return_train_score=False)
```

What is the average test accuracy, precision and recall? What does this tell you about the model?

*Question 2 - Advancing titanic*

To attempt to improve the model, we want to include Age in the model.

- a) Set up your `X_train` model appropriately
- b) Using `ColumnTransformer()`, `StandardScaler()` and `OneHotEncoder()`, set up an appropriate preprocessing object, then include it in a model pipeline and fit the model to the data
- c) The following code will set up a DataFrame of peoples ages and pclasses. Use your model to predict whether these people would survive.
- d) We could plot the new persons like so.

```
import seaborn as sns
sns.scatterplot(x = "Age", y = "Pclass", hue = "pred", data = new_values)
```

What is this graph showing? What does this say about the relationship between Age, Pclass and Survived?

- e) Just like in part g) of the previous question, the following code will perform 10-fold criss validation on the new model.

```
from sklearn.model_selection import cross_validate
from sklearn.metrics import make_scorer
import pandas as pd

acc = make_scorer(accuracy_score)

def precision(y_true,y_pred):
    return precision_score(y_true,y_pred,pos_label = 1)

def recall(y_true,y_pred):
    return recall_score(y_true, y_pred, pos_label = 1)

prec = make_scorer(precision)
rec = make_scorer(recall)
output = cross_validate(model,X_train,y_train,scoring={
    'acc' : acc,
    'prec' : prec,
    'rec' : rec
}, cv = 10, return_train_score=False)
```

How does the test accuracy compare to the previous model? Have we improved results?