In [1]:
```
#!pip install numpy pandas matplotlib scikit-learn seaborn
```

In [2]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

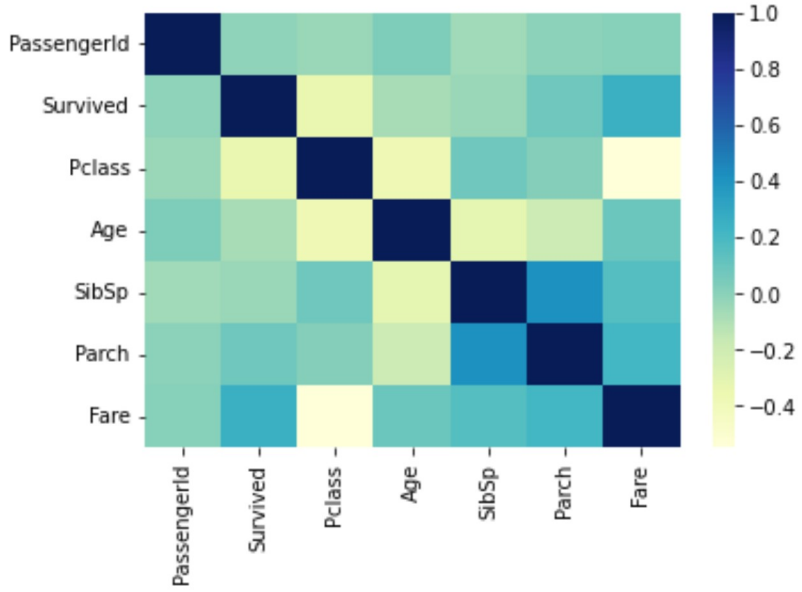
In [3]:
```
titanic_data = pd.read_csv('data/train.csv')
```

In [4]:
```
#titanic_data
```

In [5]:
```
#titanic_data.describe()
```

In [6]:
```
import seaborn as sns

sns.heatmap(titanic_data.corr(), cmap="YlGnBu")
plt.show()
```



In [7]:
```
from sklearn.model_selection import StratifiedShuffleSplit

split = StratifiedShuffleSplit(n_splits=1, test_size=0.2)
for train_indices, test_indices in split.split(titanic_data, titanic_data[["Survive
    strat_train_set = titanic_data.loc[train_indices]
    strat_test_set = titanic_data.loc[test_indices]
```

In [8]:
```
strat_test_set
```

Out[8]:

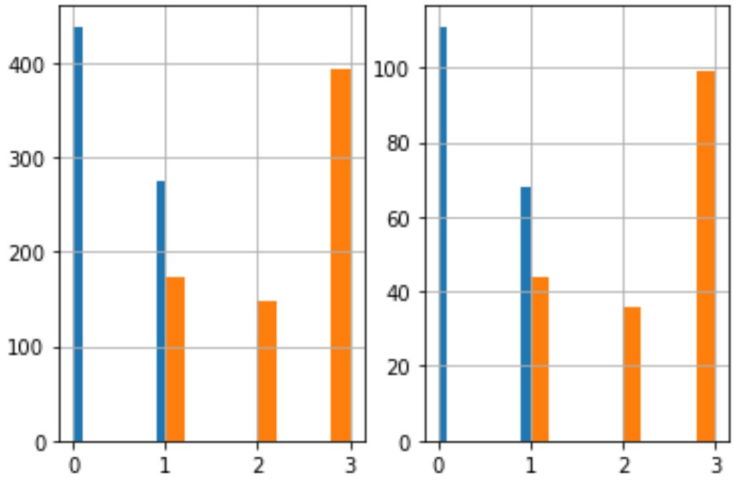
	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Ca
289	290	1	3	Connolly, Miss. Kate	female	22.0	0	0	370373	7.7500	N
68	69	1	3	Andersson, Miss. Erna Alexandra	female	17.0	4	2	3101281	7.9250	N
153	154	0	3	van Billiard, Mr. Austin Blyler	male	40.5	0	2	A/5. 851	14.5000	N
568	569	0	3	Doharr, Mr. Tannous	male	NaN	0	0	2686	7.2292	N
858	859	1	3	Baclini, Mrs. Solomon (Latifa Qurban)	female	24.0	0	3	2666	19.2583	N
...	...	...	...	...	...	...	...	...	...	...	
480	481	0	3	Goodwin, Master. Harold Victor	male	9.0	5	2	CA 2144	46.9000	N
595	596	0	3	Van Impe, Mr. Jean Baptiste	male	36.0	1	1	345773	24.1500	N
490	491	0	3	Hagland, Mr. Konrad Mathias Reiersen	male	NaN	1	0	65304	19.9667	N
527	528	0	1	Farthing, Mr. John	male	NaN	0	0	PC 17483	221.7792	
282	283	0	3	de Pelsmaeker, Mr. Alfons	male	16.0	0	0	345778	9.5000	N

179 rows × 12 columns

In [9]:
```
plt.subplot(1,2,1)
strat_train_set['Survived'].hist() #blue
strat_train_set['Pclass'].hist() #orange

plt.subplot(1,2,2)
strat_test_set['Survived'].hist() #blue
strat_test_set['Pclass'].hist() #orange

plt.show()
```



In [10]:
```
strat_train_set.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 712 entries, 7 to 110
Data columns (total 12 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   PassengerId  712 non-null    int64
 1   Survived     712 non-null    int64
 2   Pclass       712 non-null    int64
 3   Name         712 non-null    object
 4   Sex          712 non-null    object
 5   Age          574 non-null    float64
 6   SibSp        712 non-null    int64
 7   Parch        712 non-null    int64
 8   Ticket       712 non-null    object
 9   Fare         712 non-null    float64
 10  Cabin        159 non-null    object
 11  Embarked     712 non-null    object
dtypes: float64(2), int64(5), object(5)
memory usage: 72.3+ KB
```

In [11]:
```
from sklearn.base import BaseEstimator, TransformerMixin
from sklearn.impute import SimpleImputer

class AgeImputer(BaseEstimator, TransformerMixin):
    def fit(self, X, y=None):
        return self
    def transform(self, X):
        imputer = SimpleImputer(strategy="mean")
        X['Age'] = imputer.fit_transform(X[['Age']])
        return X
```

In [12]:
```
from sklearn.preprocessing import OneHotEncoder

class FeatureEncoder(BaseEstimator, TransformerMixin):
    def fit(self, X, y=None):
        return self

    def transform(self, X):
        encoder = OneHotEncoder()
        matrix = encoder.fit_transform(X[['Embarked']]).toarray()

        column_names = ["C", "S", "Q", "N"]

        for i in range(len(matrix.T)):
            X[column_names[i]] = matrix.T[i]

        matrix = encoder.fit_transform(X[['Sex']]).toarray()

        column_names = ["Female", "Male"]

        for i in range(len(matrix.T)):
            X[column_names[i]] = matrix.T[i]

        return X
```

In [13]:
```
class FeatureDropper(BaseEstimator, TransformerMixin):

    def fit(self, X, y=None):
        return self

    def transform(self, X):
        return X.drop(["Embarked", "Name", "Ticket", "Cabin", "Sex", "N"], axis=1,
```

In [14]:
```
from sklearn.pipeline import Pipeline

pipeline = Pipeline([("ageimputer", AgeImputer()),
                     ("featureencoder", FeatureEncoder()),
                     ("featuredropper", FeatureDropper())])
```

In [15]:
```
strat_train_set = pipeline.fit_transform(strat_train_set)
```

In [16]:
```
strat_train_set
```

Out[16]:

	PassengerId	Survived	Pclass	Age	SibSp	Parch	Fare	C	S	Q	Female	Male
7	8	0	3	2.000000	3	1	21.0750	0.0	0.0	1.0	0.0	1.0
851	852	0	3	74.000000	0	0	7.7750	0.0	0.0	1.0	0.0	1.0
45	46	0	3	29.764233	0	0	8.0500	0.0	0.0	1.0	0.0	1.0
498	499	0	1	25.000000	1	2	151.5500	0.0	0.0	1.0	1.0	0.0
730	731	1	1	29.000000	0	0	211.3375	0.0	0.0	1.0	1.0	0.0
...	...	...	...	...	...	...	...	...	...	...	...	..
415	416	0	3	29.764233	0	0	8.0500	0.0	0.0	1.0	1.0	0.0
732	733	0	2	29.764233	0	0	0.0000	0.0	0.0	1.0	0.0	1.0
346	347	1	2	40.000000	0	0	13.0000	0.0	0.0	1.0	1.0	0.0
150	151	0	2	51.000000	0	0	12.5250	0.0	0.0	1.0	0.0	1.0
110	111	0	1	47.000000	0	0	52.0000	0.0	0.0	1.0	0.0	1.0

712 rows × 12 columns

In [17]:
```
strat_train_set.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 712 entries, 7 to 110
Data columns (total 12 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   PassengerId  712 non-null    int64  
 1   Survived     712 non-null    int64  
 2   Pclass       712 non-null    int64  
 3   Age          712 non-null    float64
 4   SibSp        712 non-null    int64  
 5   Parch        712 non-null    int64  
 6   Fare         712 non-null    float64
 7   C            712 non-null    float64
 8   S            712 non-null    float64
 9   Q            712 non-null    float64
 10  Female       712 non-null    float64
 11  Male         712 non-null    float64
dtypes: float64(7), int64(5)
memory usage: 72.3 KB
```

In [18]:
```
strat_train_set
```

Out[18]:

	PassengerId	Survived	Pclass	Age	SibSp	Parch	Fare	C	S	Q	Female	Male
7	8	0	3	2.000000	3	1	21.0750	0.0	0.0	1.0	0.0	1.0
851	852	0	3	74.000000	0	0	7.7750	0.0	0.0	1.0	0.0	1.0
45	46	0	3	29.764233	0	0	8.0500	0.0	0.0	1.0	0.0	1.0
498	499	0	1	25.000000	1	2	151.5500	0.0	0.0	1.0	1.0	0.0
730	731	1	1	29.000000	0	0	211.3375	0.0	0.0	1.0	1.0	0.0
...	...	...	...	...	...	...	...	...	...	...	...	..
415	416	0	3	29.764233	0	0	8.0500	0.0	0.0	1.0	1.0	0.0
732	733	0	2	29.764233	0	0	0.0000	0.0	0.0	1.0	0.0	1.0
346	347	1	2	40.000000	0	0	13.0000	0.0	0.0	1.0	1.0	0.0
150	151	0	2	51.000000	0	0	12.5250	0.0	0.0	1.0	0.0	1.0
110	111	0	1	47.000000	0	0	52.0000	0.0	0.0	1.0	0.0	1.0

712 rows × 12 columns

In [19]:
```
from sklearn.preprocessing import StandardScaler

X = strat_train_set.drop(['Survived'], axis=1)
y = strat_train_set['Survived']

scaler = StandardScaler()
X_data = scaler.fit_transform(X)
y_data = y.to_numpy()
```

In [20]:
```
#X_data
```

In [21]:
```
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import GridSearchCV

clf = RandomForestClassifier()

param_grid = [
    {"n_estimators": [10, 100, 200, 500], "max_depth": [None, 5, 10], "min_samples_
]

grid_search = GridSearchCV(clf, param_grid, cv=3, scoring="accuracy", return_train_
grid_search.fit(X_data, y_data)
```

Out[21]:
```
GridSearchCV(cv=3, estimator=RandomForestClassifier(),
             param_grid=[{'max_depth': [None, 5, 10],
                          'min_samples_split': [2, 3, 4],
                          'n_estimators': [10, 100, 200, 500]}],
             return_train_score=True, scoring='accuracy')
```

In [22]:
```
final_clf = grid_search.best_estimator_
```

In [23]:
```
final_clf
```

Out[23]:
```
RandomForestClassifier(max_depth=10, min_samples_split=4)
```

In [24]:
```
strat_test_set = pipeline.fit_transform(strat_test_set)
```

In [25]:
```
#strat_test_set
```

In [26]:
```
X_test = strat_test_set.drop(['Survived'], axis=1)
y_test = strat_test_set['Survived']

scaler = StandardScaler()
X_data_test = scaler.fit_transform(X_test)
y_data_test = y_test.to_numpy()
```

In [27]:
```
final_clf.score(X_data_test, y_data_test)
```

Out[27]:
```
0.8212290502793296
```

In [28]:
```
final_data = pipeline.fit_transform(titanic_data)
```

In [29]:
```
final_data
```

Out[29]:

	PassengerId	Survived	Pclass	Age	SibSp	Parch	Fare	C	S	Q	Female	Male
0	1	0	3	22.000000	1	0	7.2500	0.0	0.0	1.0	0.0	1.0
1	2	1	1	38.000000	1	0	71.2833	1.0	0.0	0.0	1.0	0.0
2	3	1	3	26.000000	0	0	7.9250	0.0	0.0	1.0	1.0	0.0
3	4	1	1	35.000000	1	0	53.1000	0.0	0.0	1.0	1.0	0.0
4	5	0	3	35.000000	0	0	8.0500	0.0	0.0	1.0	0.0	1.0
...	...	...	...	...	...	...	...	...	...	...	...	...
886	887	0	2	27.000000	0	0	13.0000	0.0	0.0	1.0	0.0	1.0
887	888	1	1	19.000000	0	0	30.0000	0.0	0.0	1.0	1.0	0.0
888	889	0	3	29.699118	1	2	23.4500	0.0	0.0	1.0	1.0	0.0
889	890	1	1	26.000000	0	0	30.0000	1.0	0.0	0.0	0.0	1.0
890	891	0	3	32.000000	0	0	7.7500	0.0	1.0	0.0	0.0	1.0

891 rows × 12 columns

In [30]:
```
X_final = final_data.drop(['Survived'], axis=1)
y_final = final_data['Survived']

scaler = StandardScaler()
X_data_final = scaler.fit_transform(X_final)
y_data_final = y_final.to_numpy()
```

In [31]:
```
production_clf = RandomForestClassifier()

param_grid = [
    {"n_estimators": [10, 100, 200, 500], "max_depth": [None, 5, 10], "min_samples_
]

grid_search = GridSearchCV(production_clf, param_grid, cv=3, scoring="accuracy", re
grid_search.fit(X_data_final, y_data_final)
```

Out[31]:
```
GridSearchCV(cv=3, estimator=RandomForestClassifier(),
             param_grid=[{'max_depth': [None, 5, 10],
                          'min_samples_split': [2, 3, 4],
                          'n_estimators': [10, 100, 200, 500]}],
             return_train_score=True, scoring='accuracy')
```

In [32]:
```
prod_final_clf = grid_search.best_estimator_
```

In [33]:
```
prod_final_clf
```

Out[33]:
```
RandomForestClassifier(max_depth=5, n_estimators=200)
```

In [34]:
```
titanic_test_data = pd.read_csv("data/test.csv")
```

In [35]:
```
titanic_test_data
```

Out[35]:

	PassengerId	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Em
0	892	3	Kelly, Mr. James	male	34.5	0	0	330911	7.8292	NaN	
1	893	3	Wilkes, Mrs. James (Ellen Needs)	female	47.0	1	0	363272	7.0000	NaN	
2	894	2	Myles, Mr. Thomas Francis	male	62.0	0	0	240276	9.6875	NaN	
3	895	3	Wirz, Mr. Albert	male	27.0	0	0	315154	8.6625	NaN	
4	896	3	Hirvonen, Mrs. Alexander (Helga E Lindqvist)	female	22.0	1	1	3101298	12.2875	NaN	
...	...	...	...	...	...	...	...	...	...	...	...
413	1305	3	Spector, Mr. Woolf	male	NaN	0	0	A.5. 3236	8.0500	NaN	
414	1306	1	Oliva y Ocana, Dona. Fermina	female	39.0	0	0	PC 17758	108.9000	C105	
415	1307	3	Saether, Mr. Simon Sivertsen	male	38.5	0	0	SOTON/O.Q. 3101262	7.2500	NaN	
416	1308	3	Ware, Mr. Frederick	male	NaN	0	0	359309	8.0500	NaN	
417	1309	3	Peter, Master. Michael J	male	NaN	1	1	2668	22.3583	NaN	

418 rows × 11 columns

In [36]:
```
final_test_data = pipeline.fit_transform(titanic_test_data)
```

In [37]:
```
final_test_data
```

Out[37]:

	PassengerId	Pclass	Age	SibSp	Parch	Fare	C	S	Q	Female	Male
0	892	3	34.50000	0	0	7.8292	0.0	1.0	0.0	0.0	1.0
1	893	3	47.00000	1	0	7.0000	0.0	0.0	1.0	1.0	0.0
2	894	2	62.00000	0	0	9.6875	0.0	1.0	0.0	0.0	1.0
3	895	3	27.00000	0	0	8.6625	0.0	0.0	1.0	0.0	1.0
4	896	3	22.00000	1	1	12.2875	0.0	0.0	1.0	1.0	0.0
...	...	...	...	...	...	...	...	...	...	...	...
413	1305	3	30.27259	0	0	8.0500	0.0	0.0	1.0	0.0	1.0
414	1306	1	39.00000	0	0	108.9000	1.0	0.0	0.0	1.0	0.0
415	1307	3	38.50000	0	0	7.2500	0.0	0.0	1.0	0.0	1.0
416	1308	3	30.27259	0	0	8.0500	0.0	0.0	1.0	0.0	1.0
417	1309	3	30.27259	1	1	22.3583	1.0	0.0	0.0	0.0	1.0

418 rows × 11 columns

In [38]:
```
final_test_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 418 entries, 0 to 417
Data columns (total 11 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   PassengerId  418 non-null    int64  
 1   Pclass       418 non-null    int64  
 2   Age          418 non-null    float64
 3   SibSp        418 non-null    int64  
 4   Parch        418 non-null    int64  
 5   Fare         417 non-null    float64
 6   C            418 non-null    float64
 7   S            418 non-null    float64
 8   Q            418 non-null    float64
 9   Female       418 non-null    float64
 10  Male         418 non-null    float64
dtypes: float64(7), int64(4)
memory usage: 36.0 KB
```

In [39]:
```
X_final_test = final_test_data
X_final_test = X_final_test.fillna(method="ffill")

scaler = StandardScaler()
X_data_final_test = scaler.fit_transform(X_final_test)
```

In [40]:
```
predictions = prod_final_clf.predict(X_data_final_test)
```

In [41]:
```
predictions
```

Out[41]:
```
array([0, 0, 0, 0, 1, 0, 1, 0, 1, 0, 0, 0, 1, 0, 1, 1, 0, 0, 1, 1, 0, 0,
       1, 0, 1, 0, 1, 0, 0, 0, 0, 0, 1, 1, 0, 0, 1, 1, 0, 0, 0, 0, 0, 1,
       1, 0, 0, 0, 1, 1, 0, 0, 1, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 1,
       1, 0, 0, 1, 1, 0, 1, 0, 1, 0, 0, 1, 0, 1, 1, 0, 0, 0, 0, 0, 1, 1,
       1, 1, 1, 0, 1, 0, 0, 0, 1, 0, 1, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0,
       0, 1, 1, 1, 1, 0, 0, 1, 0, 1, 1, 0, 1, 0, 0, 1, 0, 1, 0, 0, 0, 0,
       1, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1,
       0, 0, 1, 1, 0, 1, 1, 1, 1, 0, 0, 1, 0, 0, 1, 1, 0, 0, 0, 0, 0, 1,
       1, 0, 1, 1, 0, 0, 1, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 1,
       0, 1, 1, 1, 0, 1, 0, 0, 1, 0, 1, 0, 0, 0, 0, 1, 0, 0, 1, 0, 1, 0,
       1, 0, 1, 0, 1, 1, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1,
       0, 0, 0, 0, 1, 0, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 1,
       0, 0, 0, 0, 1, 0, 0, 0, 1, 1, 0, 1, 0, 0, 0, 0, 0, 1, 1, 1, 1, 0,
       0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 1,
       0, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0,
       1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 1, 0, 1, 1, 0,
       0, 0, 1, 0, 1, 0, 0, 1, 0, 1, 1, 0, 1, 0, 0, 0, 1, 0, 0, 1, 0, 0,
       1, 1, 0, 0, 0, 0, 0, 0, 1, 1, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1,
       0, 1, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 0, 1, 0, 0, 0],
      dtype=int64)
```

In [42]:
```
final_df = pd.DataFrame(titanic_test_data['PassengerId'])
final_df['Survived'] = predictions
final_df.to_csv("Data/predictions.csv", index=False)
```

In [43]:
```
final_df
```

Out[43]:

	PassengerId	Survived
0	892	0
1	893	0
2	894	0
3	895	0
4	896	1
...	...	...
413	1305	0
414	1306	1
415	1307	0
416	1308	0
417	1309	0

418 rows × 2 columns