Train In Data

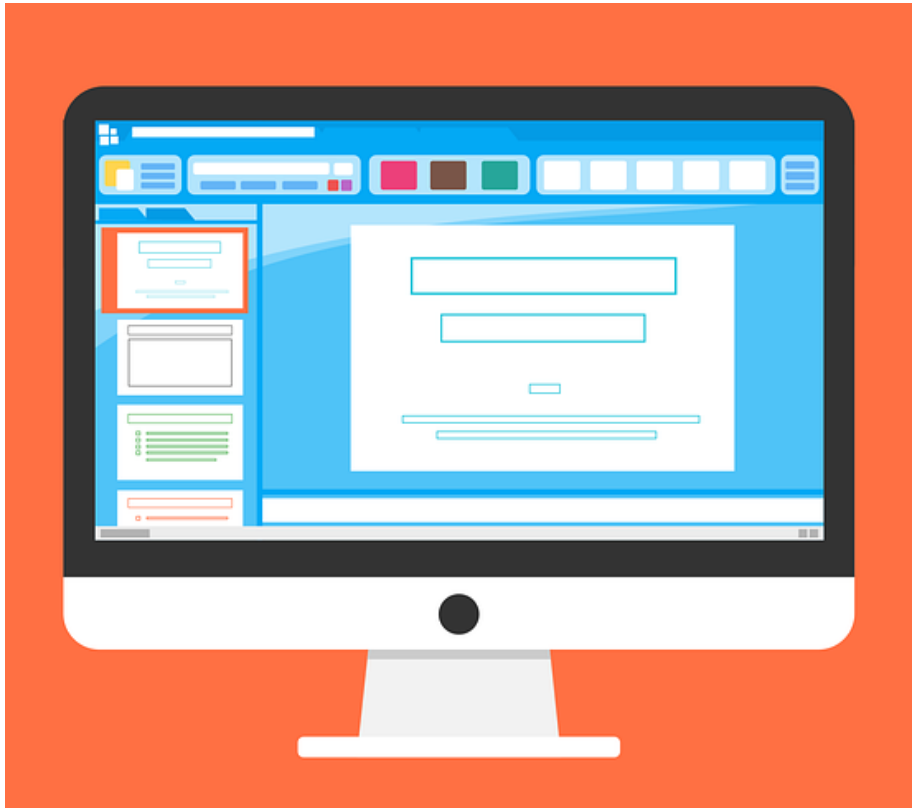
Moving Forward

Content

In each Section:

- Presentation
- Accompanying Jupyter notebook
  - Load, split (clean) the dataset
  - Select Features
    - Regression
    - Classification

Presentations



- Introduce the technique
- Describe how it works
- Advantages and shortcomings

Jupyter notebooks

- Implement technique in Python
- All notebooks have a similar structure

Jupyter notebook layout

- Imports

```
import pandas as pd
import numpy as np

from sklearn.model_selection import train_test_split

from sklearn.ensemble import RandomForestRegressor, RandomForestClassifier
from sklearn.metrics import roc_auc_score, r2_score

from mlxtend.feature_selection import SequentialFeatureSelector as SFS
```

Jupyter notebook layout

- Load a dataset

```
# load dataset

data = pd.read_csv('../dataset_2.csv')
data.shape
```

```
(50000, 109)
```

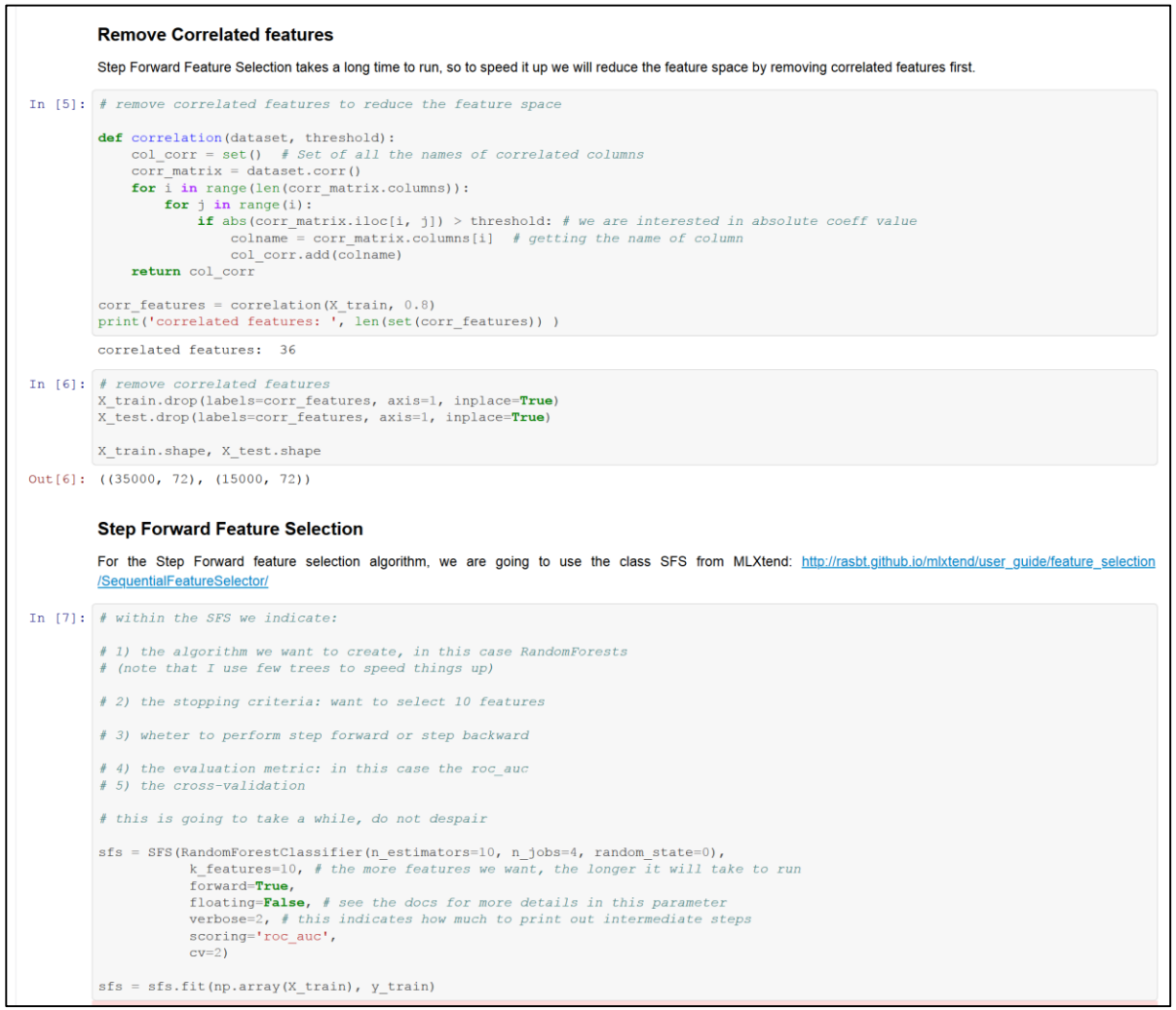
Jupyter notebook layout

- Split dataset into train and test set

```
]: # separate train and test sets

X_train, X_test, y_train, y_test = train_test_split(
    data.drop(labels=['target'], axis=1),
    data['target'],
    test_size=0.3,
    random_state=0)

X_train.shape, X_test.shape
```

```
]: ((35000, 108), (15000, 108))
```

Jupyter notebook layout



Remove Correlated features

Step Forward Feature Selection takes a long time to run, so to speed it up we will reduce the feature space by removing correlated features first.

In [5]:
```
# remove correlated features to reduce the feature space

def correlation(dataset, threshold):
    col_corr = set()  # Set of all the names of correlated columns
    corr_matrix = dataset.corr()
    for i in range(len(corr_matrix.columns)):
        for j in range(i):
            if abs(corr_matrix.iloc[i, j]) > threshold: # we are interested in absolute coeff value
                colname = corr_matrix.columns[i]  # getting the name of column
                col_corr.add(colname)
    return col_corr

corr_features = correlation(X_train, 0.8)
print('correlated features: ', len(set(corr_features)) )
```

```
correlated features:  36
```

In [6]:
```
# remove correlated features
X_train.drop(labels=corr_features, axis=1, inplace=True)
X_test.drop(labels=corr_features, axis=1, inplace=True)

X_train.shape, X_test.shape
```

Out[6]: ((35000, 72), (15000, 72))

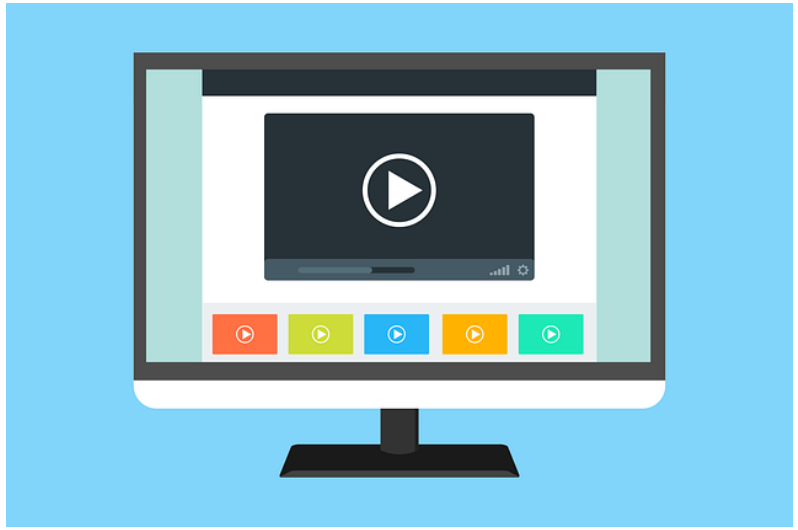
Step Forward Feature Selection

For the Step Forward feature selection algorithm, we are going to use the class SFS from MLXtend: http://rasbt.github.io/mlxtend/user_guide/feature_selection/SequentialFeatureSelector/

In [7]:
```
# within the SFS we indicate:

# 1) the algorithm we want to create, in this case RandomForests
# (note that I use few trees to speed things up)

# 2) the stopping criteria: want to select 10 features

# 3) wheter to perform step forward or step backward

# 4) the evaluation metric: in this case the roc_auc
# 5) the cross-validation

# this is going to take a while, do not despair

sfs = SFS(RandomForestClassifier(n_estimators=10, n_jobs=4, random_state=0),
          k_features=10, # the more features we want, the longer it will take to run
          forward=True,
          floating=False, # see the docs for more details in this parameter
          verbose=2, # this indicates how much to print out intermediate steps
          scoring='roc_auc',
          cv=2)

sfs = sfs.fit(np.array(X_train), y_train)
```

- Discuss the entire Jupyter notebook
- Discuss only the new and relevant code
  - To avoid being repetitive
  - Focus on the key learnings

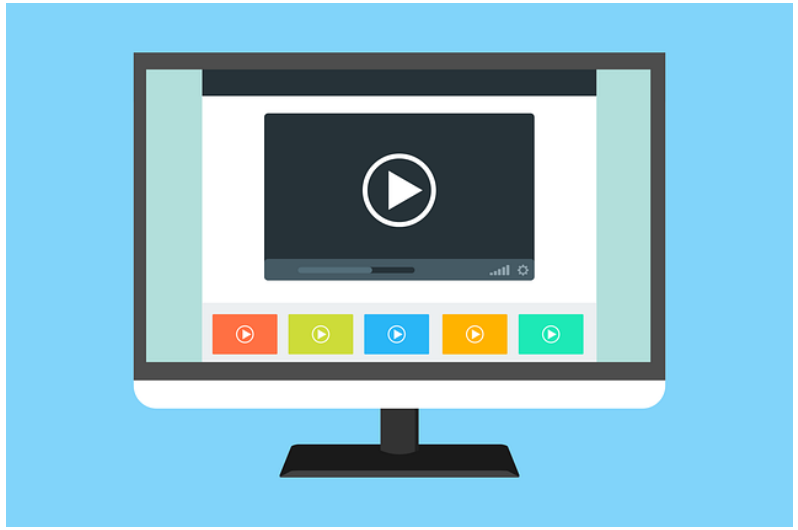
Code update



=

jupyter

Code update



≠

jupyter

Code in Github

Train In Data

THANK YOU

www.trainindata.com