In [153…
```
import boto3, re, sys, math, json, os, sagemaker, urllib.request
from sagemaker import get_execution_role
from sagemaker.sklearn.processing import SKLearnProcessor
from sagemaker.predictor import csv_serializer
import numpy as np
from sklearn.model_selection import train_test_split

region = boto3.session.Session().region_name
m_boto3 = boto3.client('sagemaker')

role = get_execution_role()
sklearn_processor = SKLearnProcessor(
    framework_version="0.20.0", role=role, instance_type="ml.m5.xlarge", instance_c
)
```

In [154…
```
import pandas as pd

input_data = "s3://bda-loans-project/loans.csv".format(region)
df = pd.read_csv(input_data)
df.head(n=10)
```

Out[154]:

	id	loan_status	loan_amount	funded_amount_by_investors	loan_term	interest_rate	instal
0	1077501	fully paid	5000	4975.0	36	10.65	
1	1077430	charged off	2500	2500.0	60	15.27	
2	1077175	fully paid	2400	2400.0	36	15.96	
3	1076863	fully paid	10000	10000.0	36	13.49	
4	1075358	current	3000	3000.0	60	12.69	
5	1075269	fully paid	5000	5000.0	36	7.90	
6	1069639	fully paid	7000	7000.0	60	15.96	
7	1072053	fully paid	3000	3000.0	36	18.64	
8	1071795	charged off	5600	5600.0	60	21.28	
9	1071570	charged off	5375	5350.0	60	12.69	

10 rows × 23 columns

In [155…
```
df = df.drop(columns=['id','grade', 'sub_grade', 'issued_on', 'employer_title', 'ea
```

In [156…
```
df = df.dropna()
```

In [157…
```
df.drop_duplicates(inplace=True)
```

In [158…
```
df.head(n=10)
```

Out[158]:

	loan_status	loan_amount	funded_amount_by_investors	loan_term	interest_rate	installment	ve
0	fully paid	5000	4975.0	36	10.65	162.87	
1	charged off	2500	2500.0	60	15.27	59.83	
2	fully paid	2400	2400.0	36	15.96	84.33	
3	fully paid	10000	10000.0	36	13.49	339.31	
4	current	3000	3000.0	60	12.69	67.79	
5	fully paid	5000	5000.0	36	7.90	156.46	
6	fully paid	7000	7000.0	60	15.96	170.08	
7	fully paid	3000	3000.0	36	18.64	109.43	
8	charged off	5600	5600.0	60	21.28	152.39	
9	charged off	5375	5350.0	60	12.69	121.45	

In [164…
```
columns = [
    "loan_amount",
    "funded_amount_by_investors",
    "loan_term",
    "interest_rate",
    "installment",
    "purpose",
    "dti",
    "inquiries_last_6_months",
    "open_credit_lines",
    "derogatory_public_records",
    "revolving_line_utilization_rate",
    "total_credit_lines",
    "employment_length",
    "home_ownership",
    "annual_income",
    "loan_status"
]
class_labels = ["fully paid", "charged off", "current"]
len(columns)
```

Out[164]: 16

In [165…
```
df.replace(class_labels, [0, 1, 2], inplace=True)
df.replace(['verified', 'source verified', 'not verified'], [0, 1, 2], inplace=True
df.replace(['car', 'credit_card','debt_consolidation', 'educational','home_improvem
df.replace(['mortgage', 'none', 'other', 'own', 'rent'], [0, 1, 2,3,4], inplace=Tru
```

In [166…
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 38595 entries, 0 to 39716
Data columns (total 17 columns):
 #   Column                          Non-Null Count  Dtype  
---  ------                          --------------  -----  
 0   loan_status                     38595 non-null  int64  
 1   loan_amount                     38595 non-null  int64  
 2   funded_amount_by_investors      38595 non-null  float64
 3   loan_term                       38595 non-null  int64  
 4   interest_rate                   38595 non-null  float64
 5   installment                     38595 non-null  float64
 6   verification_status             38595 non-null  int64  
 7   purpose                         38595 non-null  int64  
 8   dti                             38595 non-null  float64
 9   inquiries_last_6_months         38595 non-null  int64  
 10  open_credit_lines               38595 non-null  int64  
 11  derogatory_public_records       38595 non-null  int64  
 12  revolving_line_utilization_rate 38595 non-null  float64
 13  total_credit_lines              38595 non-null  int64  
 14  employment_length               38595 non-null  float64
 15  home_ownership                  38595 non-null  int64  
 16  annual_income                   38595 non-null  float64
dtypes: float64(7), int64(10)
memory usage: 5.3 MB
```

In [167…

```
train_data, test_data = np.split(df.sample(frac=1, random_state=1729), [int(0.7 * l
print(train_data.shape, test_data.shape)

train_data.to_csv('train.csv', index=False, header=False)

boto3.Session().resource('s3').Bucket(bucket_name).Object(os.path.join(prefix, 'tra

s3_input_train = sagemaker.TrainingInput(s3_data='s3://{}/train'.format(bucket_name
```

```
(27016, 17) (11579, 17)
```

In [168…

```
from sklearn.model_selection import train_test_split
y_data = df['loan_status']
x_data = df [columns]
x_train, x_test, y_train, y_test = train_test_split(x_data, y_data, test_size=0.20,
x_train.shape, x_test.shape, y_train.shape, y_test.shape
```

Out[168]:

```
((30876, 16), (7719, 16), (30876,), (7719,))
```

In [169…

```
trainX = pd.DataFrame(x_train, columns=columns)
trainX['loan_status'] = y_train

testX = pd.DataFrame(x_test, columns=columns)
testX['loan_status'] = y_test
```

In [170…

```
trainX.to_csv('train.csv')
testX.to_csv('test.csv')
```

In [171…

```
# send data to S3. SageMaker will take training data from s3
trainpath = sess.upload_data(
    path='train.csv', bucket=bucket_name,
```

```
    key_prefix='train')

testpath = sess.upload_data(
    path='test.csv', bucket=bucket_name,
    key_prefix='test')
```

In [172…
```
# We use the Estimator from the SageMaker Python SDK
from sagemaker.sklearn.estimator import SKLearn

sklearn_estimator = SKLearn(
    entry_point='script.py',
    role = get_execution_role(),
    train_instance_count=1,
    train_instance_type='ml.m5.xlarge',
    framework_version='0.20.0',
    base_job_name='rf-scikit',
    hyperparameters = {'n-estimators': 500,
                       'max_leaf_nodes': 16
                       })
```

```
train_instance_type has been renamed in sagemaker>=2.
See: https://sagemaker.readthedocs.io/en/stable/v2.html for details.
train_instance_count has been renamed in sagemaker>=2.
See: https://sagemaker.readthedocs.io/en/stable/v2.html for details.
train_instance_count has been renamed in sagemaker>=2.
See: https://sagemaker.readthedocs.io/en/stable/v2.html for details.
train_instance_type has been renamed in sagemaker>=2.
See: https://sagemaker.readthedocs.io/en/stable/v2.html for details.
```

In [173…
```
sklearn_estimator.fit({'train':trainpath, 'test': testpath}, wait=False)
```

In [174…
```
from sklearn.model_selection import cross_val_score
from sklearn.tree import DecisionTreeClassifier
clf = DecisionTreeClassifier(random_state=0)
```

In [175…
```
clf.fit(x_train,y_train)
```

Out[175]:
```
DecisionTreeClassifier(random_state=0)
```

In [177…
```
predictions = clf.predict(x_test)
print(predictions[:5])
```

```
[0 1 0 0 0]
```

In [178…
```
from sklearn.metrics import accuracy_score
print(accuracy_score(y_test, predictions))
```

```
1.0
```