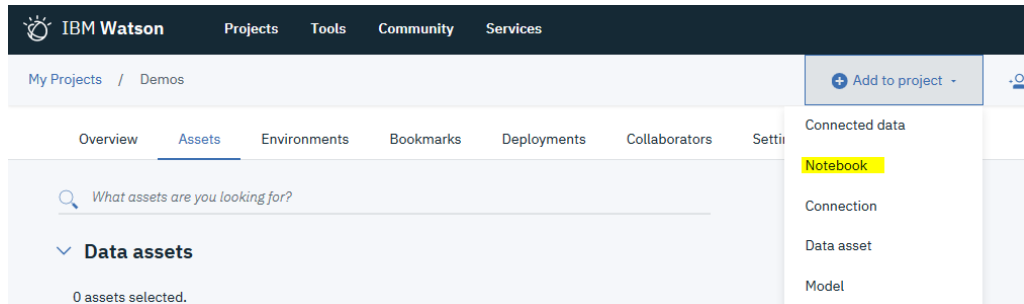


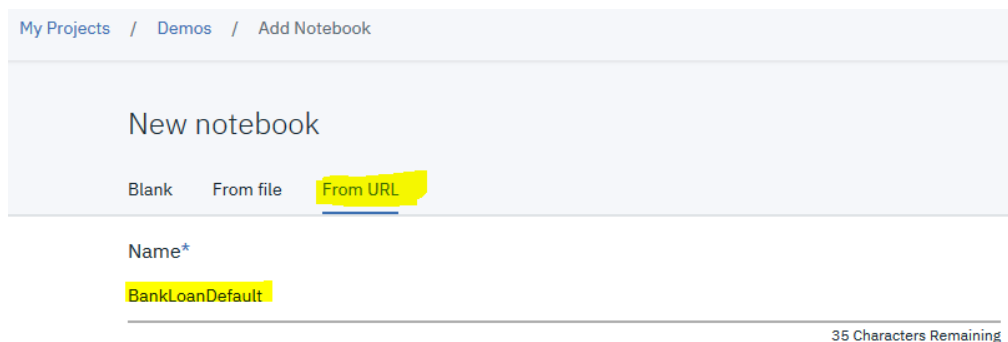
银行贷款违约预测 DSX 动手实践指导手册

1. 登录 DSX，新创建一个 Project，如 Demos

a) 点击右上方“Add to project”，选择“Notebook”



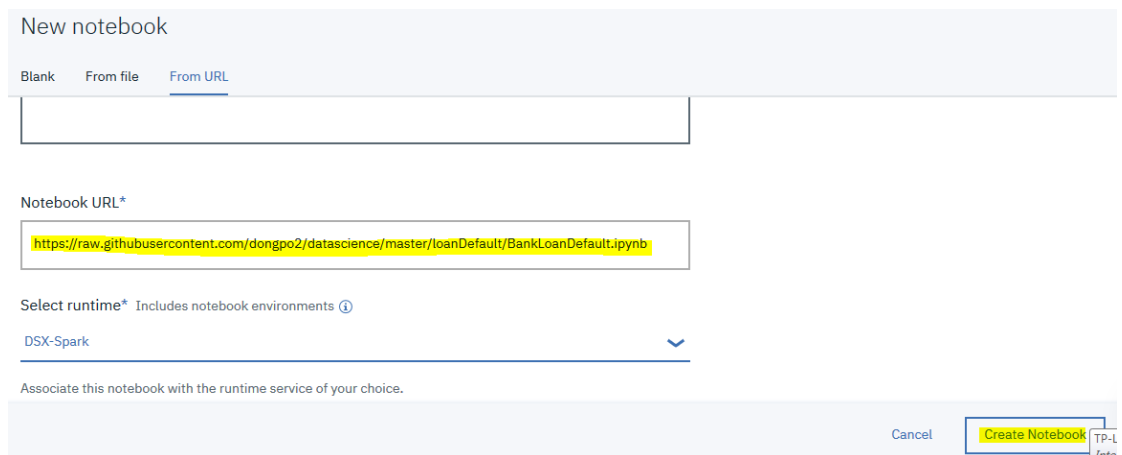
b) 选择“From URL”，在名字处输入如“BankLoanDefault”



c) 在下面的“Notebook URL”输入

<https://raw.githubusercontent.com/dongpo2/datascience/master/loanDefault/BankLoanDefault.ipynb>

然后，点击“Create Notebook”



d) 可以直接在 Notebook 中下载数据文件，如果直接展现，可以把内容保存为一个叫做 bankloan.csv 的文件

银行客户贷款违约预测

这里是利用IBM DSX来对银行客户贷款的数据集进行建模预测未来违约的可能性。

这个文档包括了从数据加载、数据理解、数据处理准备、建模、模型部署与API测试。

1. 加载数据

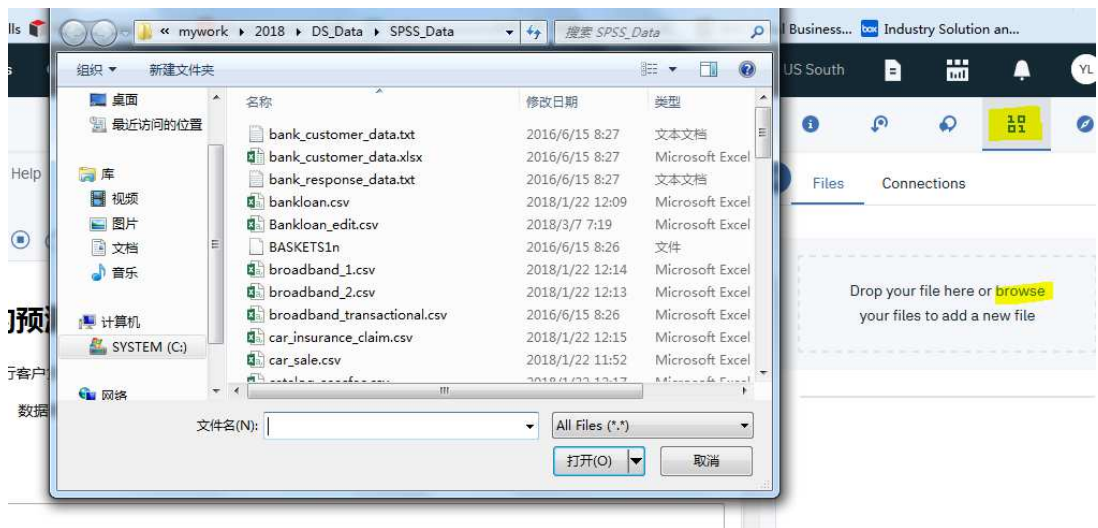
从github下载数据文件，可以采用2种方式(二选一，不同时运行)：

1. 下载数据文件到本地，然后直接从本地装载文件到DSX

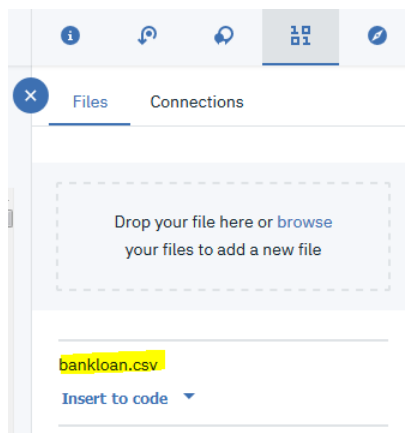
• URL: <https://raw.githubusercontent.com/dongpo2/datascience/master/loanDefault/bankloan.csv>

2. 或者，使用wget直接从github下载到DSX，然后保存到cloud-object-storage

- e) 点击右上角侧的“Data”按钮，出现右侧操作栏的“Files”页签下，点击“browse”选择下载的文件 bankloan.csv



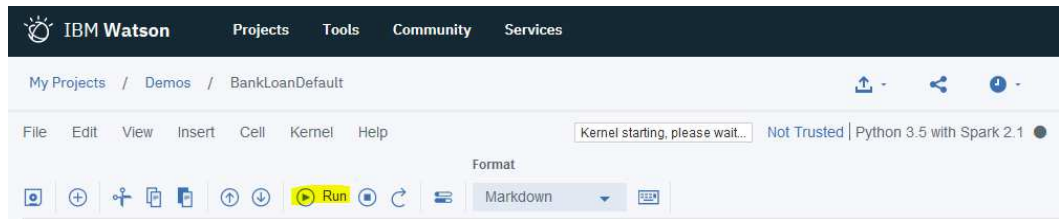
- f) 上传完成后，出现以下



2. 运行程序

- a) 运行程序可以使用鼠标点击工具栏上的“Run”图标，或者菜单 Cell 中各种运行命令。

一般交互式只运行当前焦点停留的 Cell。也可以使用快捷键 **Ctrl+Enter** 运行。



b) 导入数据集

焦点选择到 1.2 节第一个 Cell，并在右侧“Data”操作栏，选择 bankloan.csv Insert to code 下 Insert SparkSession DataFrame

1.2 导入数据集

把焦点放到下侧Cell，点击右上角“Data”图标，在右下侧选择bankloan.csv的“Insert to code”中“Insert SparkSession DataFrame”

```
In [1]: # The code was removed by DSI for sharing.

Out[1]: [Row(age=41, ed=3, employ=17, address=12, income=176.000, debtinc=9.300, creddebt=11.359, othdebt=5.009,
          default=1),
          Row(age=27, ed=1, employ=10, address=6, income=31.000, debtinc=17.300, creddebt=1.362, othdebt=4.001,
          default=0),
          Row(age=40, ed=1, employ=15, address=14, income=55.000, debtinc=5.500, creddebt=0.856, othdebt=2.169,
          default=0),
          Row(age=41, ed=1, employ=15, address=14, income=120.000, debtinc=2.900, creddebt=2.659, othdebt=0.821,
          default=0),
          Row(age=24, ed=2, employ=2, address=0, income=28.000, debtinc=17.300, creddebt=1.787, othdebt=3.057)
```

bankloan.csv

Insert to code

Insert pandas DataFrame

Insert SparkSession DataFrame

Insert Credentials

会自动插入导入数据集代码，然后运行，查看结果。

```
In [*]: import ibmos2spark
```

在运行状态 Cell 左侧会显示“*”，运行完毕会显示数字。

```
In [1]: import ibmos2spark
```

继续运行下一个 Cell 前，需要根据插入代码的变量名修改

修改DataFrame名字

```
In [3]: df_loan = df_data_8
```

c) 根据 Notebook 中的解释理解并运行，一直到 2.4 节完成。自己练习时，可以根据示例程序修改或者添加自己希望实现的内容

d) 保存数据集，在 2.5 节第一个 Cell，选中，然后在右侧“Data”操作栏，插入 pandas 代码 Insert pandas DataFrame

2.5 完成数据准备，保存为数据集，供建模人员使用

保存为一个数据文件，供下一阶段或者共享给其他人使用。
注意：接口操作函数，修改为相应的名称

鼠标停留在下侧Cell，在右侧选择bankloan.csv->“Insert to code”->“Insert pandas DataFrame”

复制读取csv文件一行中BUCKET名字符串到下面第二个Cell中BUCKET变量。

删除最下面几行读取数据代码，保留定义的函数，然后运行

```
In [22]: # The code was removed by DSI for sharing.
```

把下侧client_manual修改为上面生成的函数名。

```
In [24]: import io

BUCKET = ""
csvString = sparkDf.toPandas().to_csv(index=None)
body2 = client_manual.upload_fileobj(io.BytesIO(csvString.encode()), Bucket=BUCKET, Key='bankloan_edit.csv')
```

Drop your file here or browse
your files to add a new file

bankloan.csv

Insert to code

Insert pandas DataFrame

Insert SparkSession DataFrame

Insert Credentials

TP-LINK

然后根据 notebook 中的提示，分别修改再下一个 Cell 中的 BUCKET 变量值，以及写入 csv 文件的函数名(原来是 client_manual)。

根据 notebook 中的提示，删除不需要的代码，然后继续运行 Cell。

- e) 建模阶段，根据提示理解建模过程，逐个运行 Cell，一直到 3 章节结束
- f) 保存模型，运行完 4 章节的第一个 Cell，我们需要对 4 变量赋值

4. 保存模型

```
In [43]: from repository.mlrepositoryclient import MLRepositoryClient
         from repository.mlrepositoryartifact import MLRepositoryArtifact

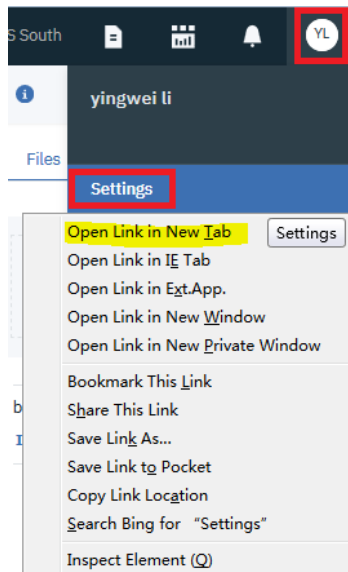
/usr/local/src/conda3_runtime/home/envs/DSX-Python35-Spark/lib/python3.5/site-packages/sklearn/cross_validation.py:41: DeprecationWarning: This module was deprecated in version 0.18 in favor of the model_selection module into which all the refactored classes and functions are moved. Also note that the interface of the new CV iterators are different from that of this module. This module will be removed in 0.20.
  "This module will be removed in 0.20.", DeprecationWarning)
```

赋值服务的相关变量：

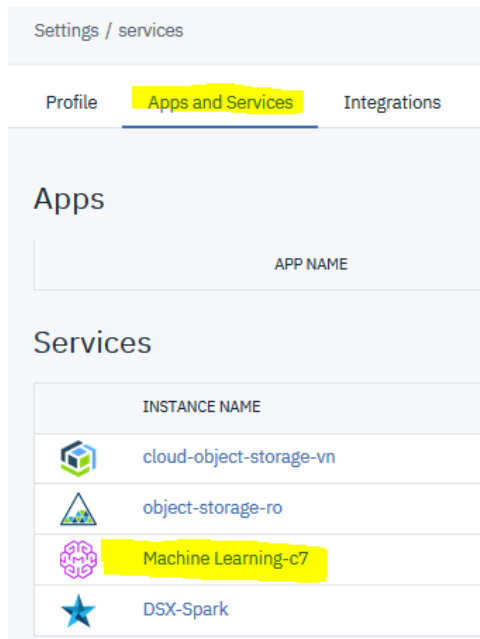
- url= "" # 右上角用户图标->setting, 点击"Apps and Services", 选择Machine Learning-xx, 选择服务凭证, 查看凭证, 复制url值到左侧
- username="" # 复制username值到左侧
- password="" # 复制"password"后面值到左侧
- instance_id="" # 复制"instance_id"后面值到左侧

```
In [44]: # 右上角用户图标->setting, 点击"Apps and Services", 选择Machine Learning-xx, 选择服务凭证, 查看凭证, 复制url值到左侧
url = ""
# 复制username值到左侧
username = ""
# 复制"password"后面值到左侧
password = ""
```

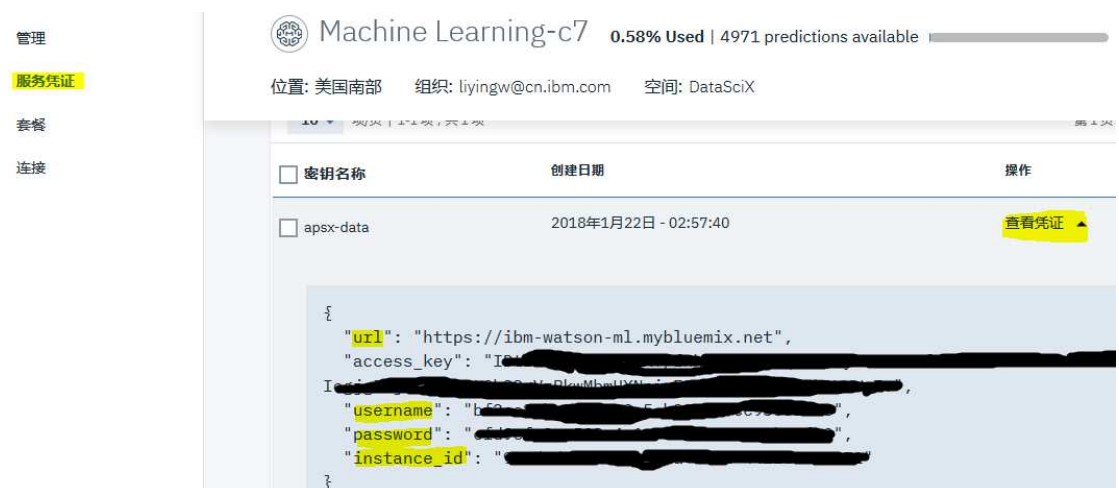
点击右上方用户图标，鼠标移动到下拉菜单的"Setting"，右键选择"Open Link in New Tab"，新打开一个标签页，管理



在新的浏览器页面上，选择"App and Services"，下面会列出使用的服务，选择点击"Machine-Learning-xx" 进入服务管理页。



点击左侧的“服务凭证”，然后再点击“查看凭证”展开内容。



分别复制四项值到 Notebook 中对应的变量值。

运行后续 Cell 完成模型保存。

可以在保存时，修改为自己希望的名字

可以修改保存的模型名字，如"Artifact Model" -> "<your name> Model"

```
In [47]: from repository.mlrepository.meta_props import MetaProps
from repository.mlrepository.meta_names import MetaNames
model_artifact = MLRepositoryArtifact(model_rf, training_data=train_data, name="Artifact Model", meta_props=MetaProps({
    MetaNames.LABEL_FIELD: "default" })))
```

g) 模型部署

运行模型部署相关 Cell，其中可以修改为自己希望的名字

可以修改部署模型名称

```
In [57]: payload_online = {"name": "Bank Loan Default Prediction", "description": "My Deployment", "type": "online"}
response_online = requests.post(endpoint_deployments, json=payload_online, headers=header)

print(response_online)
#print(response_online.text)
```

h) 测试，可以在 Notebook 中完成测试，

6. 测试模型API

```
[81]: payload_scoring = {"fields": ["age", "ed", "address", "employ", "income", "debtinc", "creddebt", "othdebt"], "values": [[33, 1, 5, 8,
response_scoring = requests.post(scoring_url, json=payload_scoring, headers=header)

print(response_scoring.text)
```

```
{
  "fields": ["age", "ed", "address", "employ", "income", "debtinc", "creddebt", "othdebt", "features", "rawPrediction",
"probability", "prediction", "predictedLabel"],
  "values": [[33, 1, 5, 8, 116.0, 15.0, 5.0, 10.0, [33.0, 1.0, 8.0, 5.0, 116.0, 15.0, 5.0, 10.0], [10.587353053077383, 9
.412646946922617], [0.5293676526538691, 0.47063234734613085], 0.0, "not default"], [31, 3, 3, 3, 52.0, 20.0, 15.0, 5.0,
[31.0, 3.0, 3.0, 3.0, 52.0, 20.0, 15.0, 5.0], [8.641658162473622, 11.358341837526378], [0.4320829081236811, 0.5679170918
763189], 1.0, "default"]]
```

修改数据，观察返回结果。

i) 也可以在 DSX 中进行测试

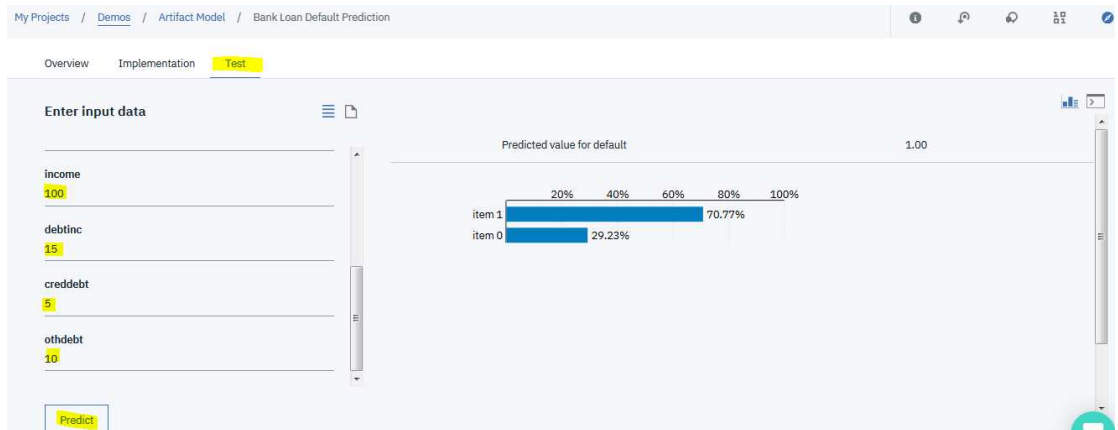
在项目首页，可以看到自己保存的模型

Models						+ New model
NAME	STATUS	TYPE	RUNTIME	LAST MODIFIED		ACTIONS
Artifact Model	trained	mlib-2.1	spark-2.1	16 Mar 2018		⋮
car_evaluation	trained	wml-1.1	spark-2.0	22 Jan 2018		⋮

点击自己的模型，进入管理页，选择“Deployment”页签，出现已经部署的模型。

Artifact Model				+ Add Deployment
Overview Evaluation Deployments				
NAME	STATUS	DEPLOYMENT TYPE	ACTIONS	
Bank Loan Default Prediction	DEPLOY_SUCCESS	Web Service	⋮	

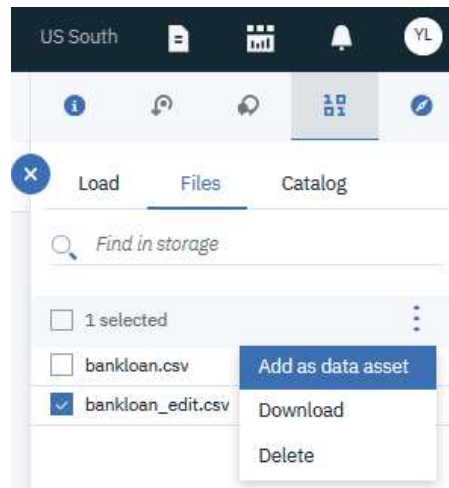
点击部署名，选择“Test”页签，输入数据，点击“Predict”，可以得到预测结果，可以显示为图形和文本两种方式。



3. 使用自动化建模工具进行建模

- a) DSX 支持不编程模式的建模方式，如果已经完成 Notebook，或至少完成其中 2.5 章节部分的保存数据集

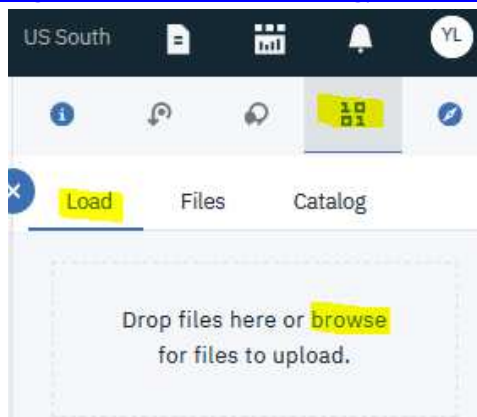
- i. 在项目首页，右侧 Data 操作栏中，选择“bankloan_edit.csv”点击右上角三个点的图标，选择“Add as asset”



- ii. 完成后，Data Assets 会出现该数据集

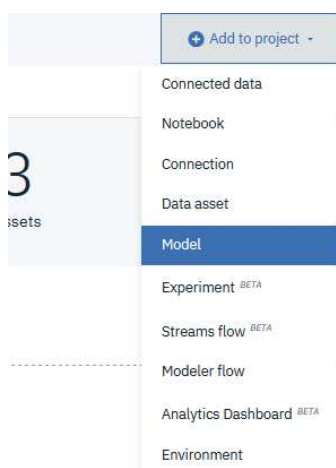
- b) 如果没有完成 Notebook 的 2.5 章节，可以直接下载数据集文件

https://raw.githubusercontent.com/dongpo2/datascience/master/loanDefault/bankloan_edit.csv



在项目首页，点击右侧“Data”图标，选择“Load”页签，点击“browse”，选择下载的文件，上传为数据集。

- c) 项目首页，点击“Add to project”，选择 Model



在新建 Model 页，输入名字，选择 Model Builder，选择 Manual，开始创建

New model BETA

Define model details

Name

AutoModel

91

Description

Model description

300

Machine Learning Service

Machine Learning-c7

Select model type

☒ Model builder

☐ From file

☐ From sample

Spark Service

DSX-Spark

Automatic

Prepare my data and create a model automatically

Manual

Let me prepare my data and select which models to train

Need something more flexible? Create a notebook or design an SPSS Modeler flow.

Cancel

CTP-1111

在”Select Data”，选择数据集，Next

My Projects / Demos / AutoModel

Select Data

Train

Evaluate

Select data asset

The model builder currently supports CSV files and IBM Db2 Warehouse on Cloud data assets.

What asset are you looking for?

NAME	TYPE	SERVICE
<input type="radio"/> bankloan.csv	Data Asset	Project
<input checked="" type="radio"/> bankloan_edit.csv	Data Asset	Project

Close

Next

在”Train”页，先选择标注列名称”default”

Train

Evaluate

Column value to predict (Label Col)

Select Label Col

income (Decimal)

debtinc (Decimal)

creddebt (Decimal)

othdebt (Decimal)

default (Integer)

Classification

Classification

DSX 将自动推荐任务类型(二分类)，在右上角的 “Add Estimators”，按 Ctrl+鼠标可以选择多项算法，这里选择随机森林和逻辑回归。

Select estimator(s)

What type of estimator are you looking for?

Logistic Regression

Analyzes a data set in which there are one or more independent variables that determine one of two outcomes. Only binary L...

Decision Tree Classifier

Maps observations about an item (represented in the branches) to conclusions about the item's target value (represented in...

Random Forest Classifier

Constructs multiple decision trees to produce the label that is a mode of each decision tree. It supports both binary and ...

Gradient Boosted Tree

Classifies...

Add Estimators

estimators

然后 Next 继续。在 “Evaluate” 页将给出不同算法的性能指标，选择合适的保存。

My Projects / Demos / AutoModel

Select Data

Train

Evaluate

Select model

	ESTIMATOR TYPE	STATUS	PERFORMANCE	AREA UNDER ROC CURVE	AREA UNDER PR CURVE	LAST EVALUATION	ACTIONS
<input type="radio"/>	LogisticRegression	Trained & Evaluated	Good	0.85211	0.67801	16 Mar 2018, 10:54 PM	⋮
<input checked="" type="radio"/>	RandomForestClassifier	Trained & Evaluated	Good	0.81534	0.48486	16 Mar 2018, 10:53 PM	⋮

Close

Previous

Save

保存成功后,进入模型页,选择”Deployments”页。点击右上角图标“Add Deployment”

AutoModel

Overview

Evaluation

Deployments

+

Add Deployment

NAME	STATUS	DEPLOYMENT TYPE	ACTIONS
Your model is not deployed.			

在新建部署页，输入名称，保存

Create Deployment

Web Service

Batch Prediction

Real-time Streaming Predictions

Name

Online-LoanDefault

Description

Web Service Deployment Description

300

Cancel

Save

d) 测试，参见 2.i 章节测试