BaselineModels

January 28, 2022

[1]:

```
import pandas as pd
import numpy as np
import scipy.stats as stats
import ModelClass
import matplotlib.pyplot as plt
import seaborn as sns

#from sklearnex import patch_sklearn
#patch_sklearn(verbose=False)
from sklearn.preprocessing import StandardScaler
from sklearn.impute import SimpleImputer
from sklearn.pipeline import Pipeline
from sklearn.compose import ColumnTransformer, make_column_selector
from sklearn.metrics import plot_confusion_matrix, recall_score,␣
 ↪accuracy_score, precision_score, f1_score
from sklearn.neighbors import KNeighborsClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
```

Loading the Data

[2]:

```
X = pd.read_csv('data/Training-set-values.csv')
y = pd.read_csv('data/Training-set-labels.csv')

X['date_recorded'] = pd.to_datetime(X['date_recorded']).astype(np.int64)
```

Preprocessors

[3]:

```
# Super basic numeric transformer

numeric_transformer = Pipeline(
    steps=[('imputer', SimpleImputer(strategy='median'))]
)

numeric_preprocessor = ColumnTransformer(
    transformers=[
        ("numeric", numeric_transformer, make_column_selector(dtype_include=np.
 ↪number)),
    ]
)
```

0.0.1 Models

[4]:

```
# kNearestNeighbors
kNearestNeighbors = {'classifier': KNeighborsClassifier(n_jobs=3),␣
 ↪'preprocessor': None}

# Logistic Regressoion
LogisticRegressionModel = {'classifier': LogisticRegression(C=1e6, n_jobs=3),␣
 ↪'preprocessor': None}

# Decision Trees
DecisionTrees = {'classifier': DecisionTreeClassifier(),'preprocessor': None}
# Decision Trees - adjusted
DecisionTreesAdjusted = {'classifier':␣
 ↪DecisionTreeClassifier(criterion=['gini','entropy'], max_depth=[90,100],␣
 ↪min_samples_split=[2,3], class_weight='balanced'),'preprocessor':␣
 ↪numeric_preprocessor}

# Random Forest with numeric processor
RandomFM_1 = {'classifier': RandomForestClassifier(max_depth=20,␣
 ↪min_samples_split=4, n_jobs=3), 'preprocessor': numeric_preprocessor}
# Random Forest no processor
RandomFM_2 = {'classifier': RandomForestClassifier(max_depth=20,␣
 ↪min_samples_split=4, n_jobs=3), 'preprocessor': None}
# Random Forest default
# Included for RandomCVSearch later on
RandomFM_rs = {'classifier': RandomForestClassifier(n_jobs=3), 'preprocessor':␣
 ↪None}

models = {'kNearestNeighbors': kNearestNeighbors,
    'LogisticRegression': LogisticRegressionModel,
    'DecisionTrees': DecisionTrees,
    'DecisionTreesAdjusted': DecisionTreesAdjusted,
    'RandomFM_1': RandomFM_1,
    'RandomFM_2': RandomFM_2,
    'RandomFM_rs': RandomFM_rs}
```

0.0.2 Modeler

[5]:

```
model_run = ModelClass.Modeler(models, X=X, y=y)

# Adding in after the model_run object is created so we can add onto the␣
 ↪default preprocessor.
log_reg_regularized = {'classifier': LogisticRegression(n_jobs=3),␣
 ↪'preprocessor': model_run.create_default_prep(num_add=[('scaling',␣
 ↪StandardScaler())])}
model_run.add_model('log_reg_regularized', log_reg_regularized)
```

0.0.3 Search parameters and kwargs

[6]:

```
kNN_params = dict(leaf_size=[1,50],
                  n_neighbors=[1,30],
                  p=[1,2])

LogRegRCV_params = dict(penalty=['l1', 'l2', 'elasticnet'],
                        C=stats.uniform(loc=1, scale=10),
                        max_iter=list(range(100,400)))

DecisionTree_params = dict(criterion=['gini', 'entropy'],
                        max_depth = list(range(20,50)),
                        min_samples_split = list(range(2, 10)))

RandForestRCV_params = dict(n_estimators=list(range(100,300)),
                            criterion=['gini', 'entropy'],
                            max_depth = list(range(20,50)),
                            min_samples_split = list(range(2, 10)))

search_options = {'n_jobs': 3, 'random_state': 9280210, 'n_iter': 20}
```

0.1 Training LogisticRegression Model

[7]:

```
model_run.train_model('LogisticRegression')
```

```
root - INFO - Cross validate scores for LogisticRegression: [0.54242424
0.54242424 0.54242424 0.54253648 0.54242424]
root - INFO - LogisticRegression has been fit.
```

0.2 RandomizedSearchCV

[8]:

```
model_run.hyper_search('kNearestNeighbors', params=kNN_params,␣
 ↪searcher_kwargs=search_options, set_to_train=True)
```

```
/Users/valeriaviscarra/opt/anaconda3/envs/learn-env/lib/python3.8/site-
packages/sklearn/model_selection/_search.py:278: UserWarning: The total space of
```

```
parameters 8 is smaller than n_iter=20. Running 8 iterations. For exhaustive
searches, use GridSearchCV.
  warnings.warn(
/Users/valeriaviscarra/opt/anaconda3/envs/learn-env/lib/python3.8/site-
packages/joblib/externals/loky/process_executor.py:688: UserWarning: A worker
stopped while some jobs were given to the executor. This can be caused by a too
short worker timeout or by a memory leak.
  warnings.warn(
```

[9]:

```
model_run.hyper_search('log_reg_regularized', params=LogRegRCV_params,␣
 ↪searcher_kwargs=search_options, set_to_train=True)
```

```
/Users/valeriaviscarra/opt/anaconda3/envs/learn-env/lib/python3.8/site-
packages/joblib/externals/loky/process_executor.py:688: UserWarning: A worker
stopped while some jobs were given to the executor. This can be caused by a too
short worker timeout or by a memory leak.
  warnings.warn(
```

[10]:

```
model_run.hyper_search('DecisionTrees', params=DecisionTree_params,␣
 ↪searcher_kwargs=search_options, set_to_train=True)
```

[11]:

```
model_run.hyper_search('DecisionTreesAdjusted', params=DecisionTree_params,␣
 ↪searcher_kwargs=search_options, set_to_train=True)
```

[12]:

```
model_run.hyper_search('RandomFM_1', params=RandForestRCV_params,␣
 ↪searcher_kwargs=search_options, set_to_train=True)
```

```
/Users/valeriaviscarra/opt/anaconda3/envs/learn-env/lib/python3.8/site-
packages/joblib/externals/loky/process_executor.py:688: UserWarning: A worker
stopped while some jobs were given to the executor. This can be caused by a too
short worker timeout or by a memory leak.
  warnings.warn(
```

[13]:

```
model_run.hyper_search('RandomFM_2', params=RandForestRCV_params,␣
 ↪searcher_kwargs=search_options, set_to_train=True)
```

```
/Users/valeriaviscarra/opt/anaconda3/envs/learn-env/lib/python3.8/site-
packages/joblib/externals/loky/process_executor.py:688: UserWarning: A worker
stopped while some jobs were given to the executor. This can be caused by a too
short worker timeout or by a memory leak.
  warnings.warn(
```

[14]:

```
model_run.hyper_search('RandomFM_rs', params=RandForestRCV_params,␣
 ↪searcher_kwargs=search_options, set_to_train=True)
```

```
/Users/valeriaviscarra/opt/anaconda3/envs/learn-env/lib/python3.8/site-
packages/joblib/externals/loky/process_executor.py:688: UserWarning: A worker
stopped while some jobs were given to the executor. This can be caused by a too
```

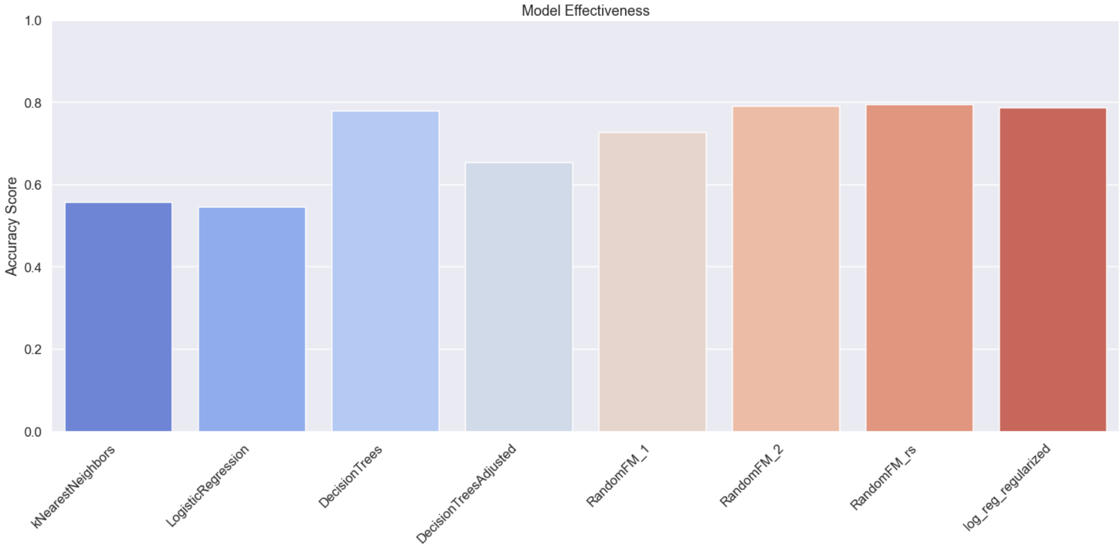
```
short worker timeout or by a memory leak.
  warnings.warn(
```

0.3 Test Models

```
[15]: model_run.test_all()
```

0.4 Plotting

```
[16]: model_run.plot_models(save='baseline_models_graph')
```



0.5 Modeler

0.5.1 Random Forests

```
[17]: model_run.model_evaluation('RandomFM_2')
```

```
root - INFO - Cross validate scores for RandomFM_2: [0.78787879 0.79169473
0.79416386 0.79203143 0.78731762]
root - INFO - RandomFM_2 has been fit.
root - INFO - RandomFM_2 test score: 0.7904377104377104
```

```
------------------------------------------------------------
[i] CLASSIFICATION REPORT
------------------------------------------------------------
```

```
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
<ipython-input-17-af797f7d324f> in <module>
----> 1 model_run.model_evaluation('RandomFM_2')
```

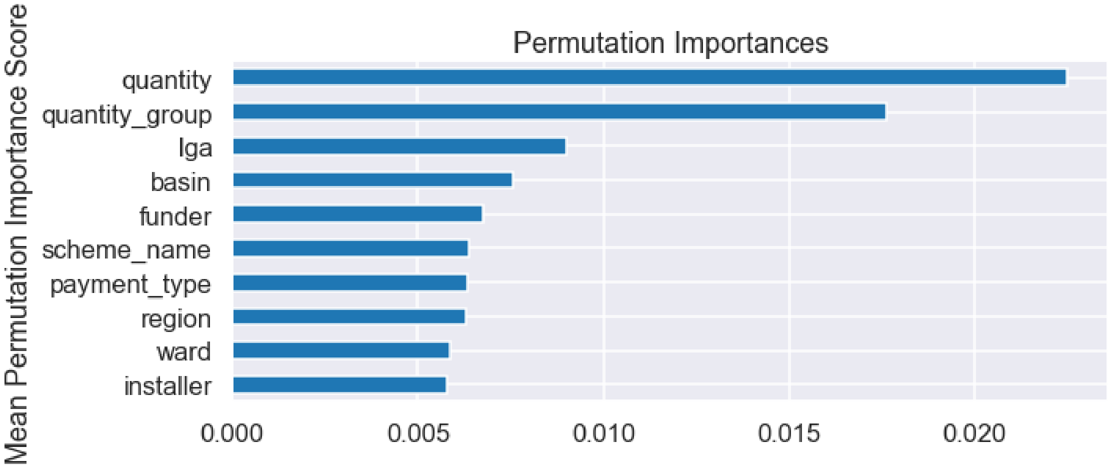
```
~/Desktop/Tanzania-Well-Project/ourfunctions.py in model_evaluation(self, name,
 ↪normalize, cmap, label)
    327         dashes = "---"*20
    328         print(dashes,table_header,dashes,sep="\n")
--> 329         print("Train Accuracy : ", round(self.
 ↪_models[name]['train_output'],4))
    330         print("Test Accuracy : ", round(self.
 ↪_models[name]['test_output'],4))
    331

KeyError: 'train_output'
```

[18]:

```
importance_kwargs = dict(n_repeats=10, n_jobs=3)
model_run.permutation_importance('RandomFM_2', perm_kwargs=importance_kwargs)
```

```
/Users/valeriaviscarra/opt/anaconda3/envs/learn-env/lib/python3.8/site-
packages/joblib/externals/loky/process_executor.py:688: UserWarning: A worker
stopped while some jobs were given to the executor. This can be caused by a too
short worker timeout or by a memory leak.
  warnings.warn(
root - INFO - Model RandomFM_2 has permutation importances of quantity
0.022465
quantity_group    0.017623
lga               0.008997
basin             0.007576
funder            0.006768
scheme_name       0.006384
payment_type      0.006343
region            0.006269
ward              0.005886
installer         0.005778
dtype: float64
```



[19]:

```
model_run.model_evaluation('RandomFM_rs')
```

```
root - INFO - Cross validate scores for RandomFM_rs: [0.78709315 0.79292929
0.79506173 0.79281706 0.78799102]
root - INFO - RandomFM_rs has been fit.
root - INFO - RandomFM_rs test score: 0.7915824915824916

------------------------------------------------------------
[i] CLASSIFICATION REPORT
------------------------------------------------------------
```

```
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
<ipython-input-19-cb8613376d25> in <module>
----> 1 model_run.model_evaluation('RandomFM_rs')

~/Desktop/Tanzania-Well-Project/ourfunctions.py in model_evaluation(self, name,␣
 ↪normalize, cmap, label)
    327         dashes = "---"*20
    328         print(dashes,table_header,dashes,sep="\n")
--> 329         print("Train Accuracy : ", round(self.
 ↪_models[name]['train_output'],4))
    330         print("Test Accuracy : ", round(self.
 ↪_models[name]['test_output'],4))
    331

KeyError: 'train_output'
```

[ ]:

```
importance_kwargs = dict(n_repeats=10, n_jobs=3)
model_run.permutation_importance('log_reg_regularized',␣
 ↪perm_kwargs=importance_kwargs)
```

[ ]:

```

```