AUTOIT

Members: Willy Lee, Jonathan Rogers, Rushil Mojidra, Tiger Ruan, Jacob Pfeiffer

Background of AutoML

- What is AutoML
  - Automated Machine Learning
  - Focuses on automating preprocessing of data and model selection
  - Seeks to automate tedious parts of the machine learning process

- Benefits of AutoML
  - Give data scientist more time to work on the more technical aspects of ML
  - Makes the analytical power of ML available to smaller companies with less Data science expertise

AutoML Libraries

- AutoKeras
  - Based on Keras library
  - Supports image classification/regression, text classification/regression, structured data classification/regression

- Auto-PyTorch
  - Based on PyTorch library
  - Supports automl for neural architectures
  - Automated deep learning

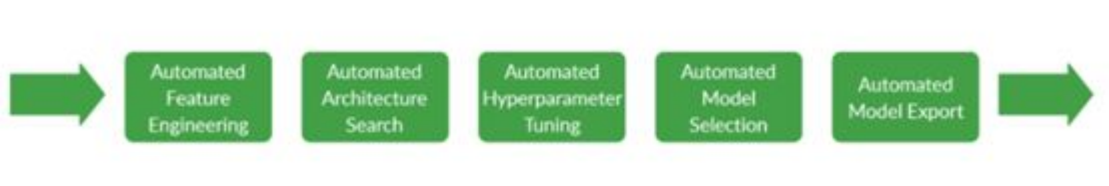
- Auto-sklearn
  - Based on scikit-learn library
  - Automated classification, regression, and clustering

- Many others

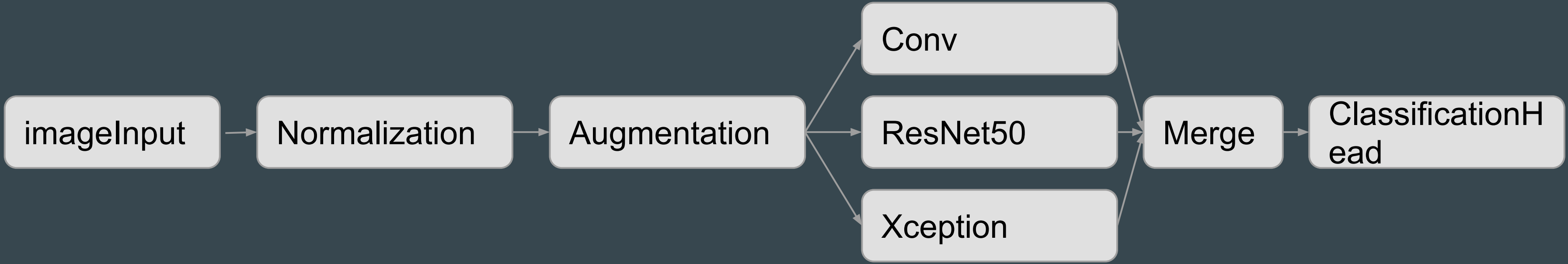
Project 1-1

-Image classification with Autokeras

Applying AutoML--pipeline


Automated Feature Engineering
Automated Architecture Search
Automated Hyperparameter Tuning
Automated Model Selection
Automated Model Export

ResNet and Xception are merged into the architecture search in pipeline



imageInput

Normalization

Augmentation

Conv

ResNet50

Xception

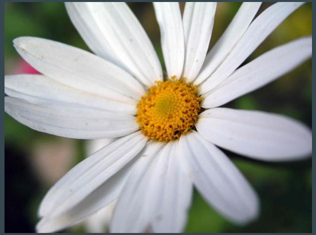
Merge

ClassificationHead

Data prep - Flower dataset

Classification - Flower Recognition

- 4232 images of flowers
- Five classes - about 800 photos each
- Photos are of different proportions



Daisy

Dandelion

Rose

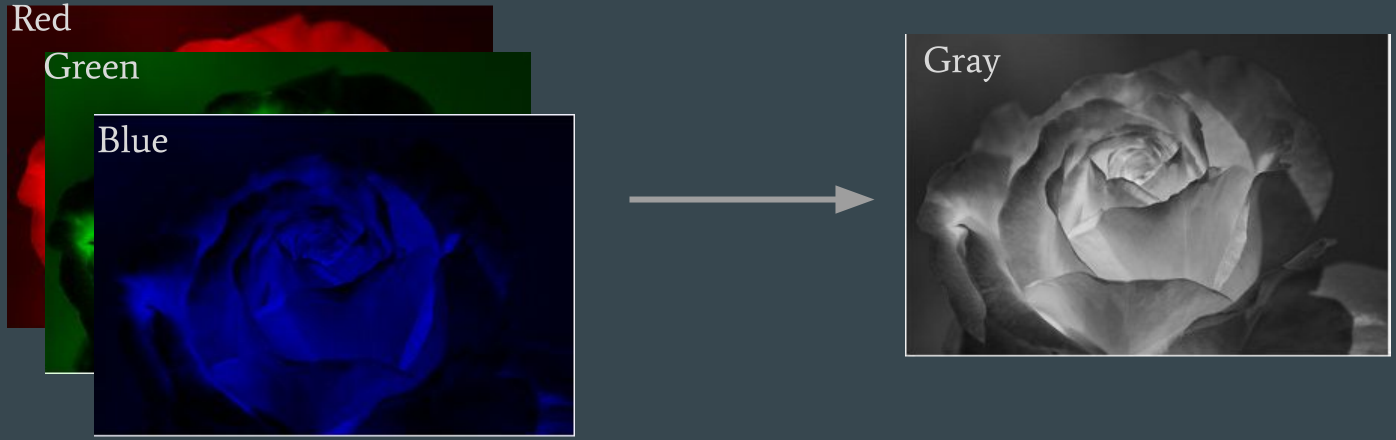
Sunflower

Tulip

Data prep - Flower dataset

Classification - Flower Recognition

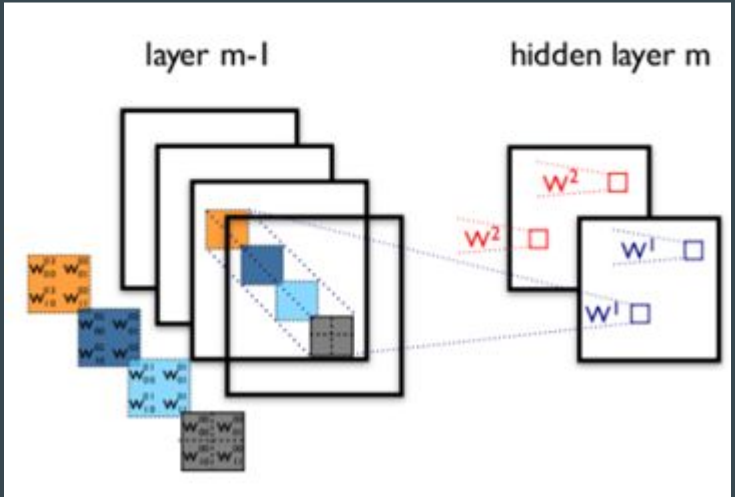
- Dataset split - 80% training & 20% testing (for each class)
- Reduced resolution - 128 x 128 pixels
- RGB channel and Gray channel



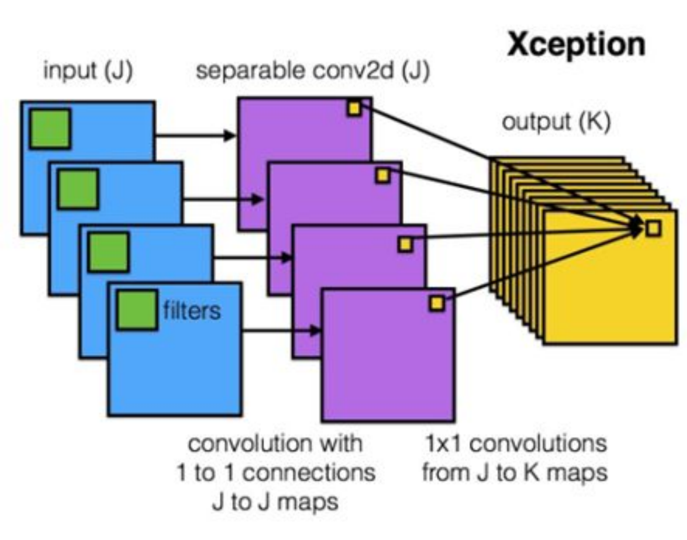
Deeper into the architecture

Xception(extreme inception)

- **fewer connections with lighter model**

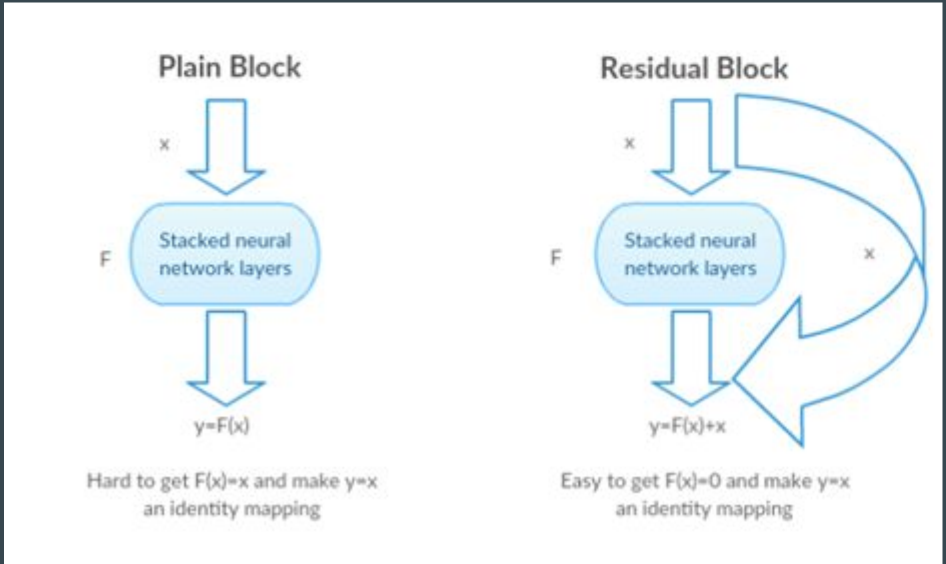


Standard Conv



Xception

ResNet(Residual Networks)

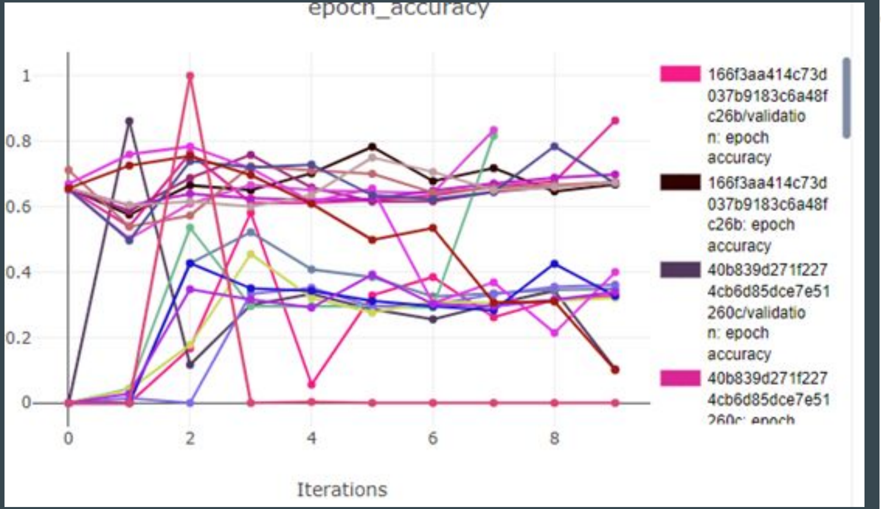

Plain Block
x
F
Stacked neural network layers
y=F(x)
Hard to get F(x)=x and make y=x an identity mapping
Residual Block
x
F
Stacked neural network layers
x
y=F(x)+x
Easy to get F(x)=0 and make y=x an identity mapping

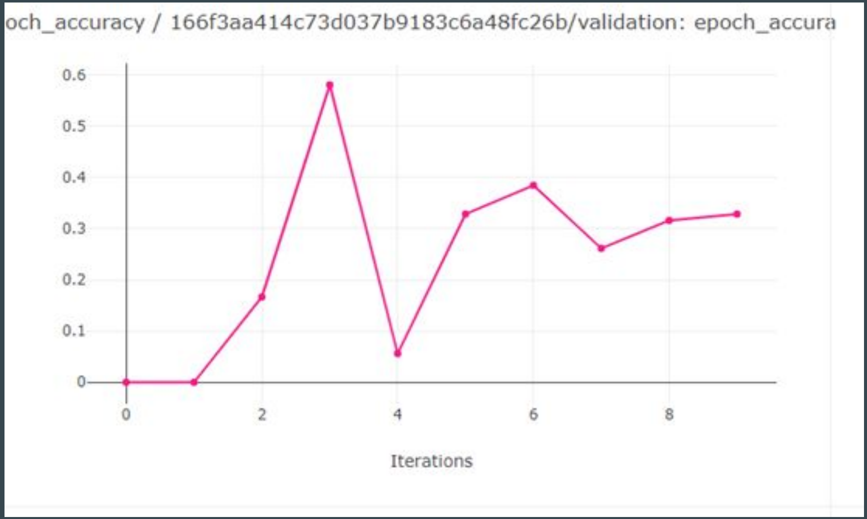
- Good way to solve degradation problem in deeper neural network.

Results

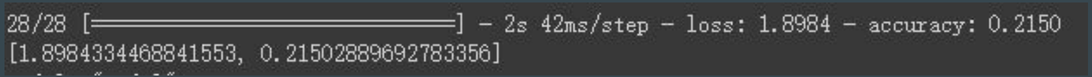
Gray scale

train





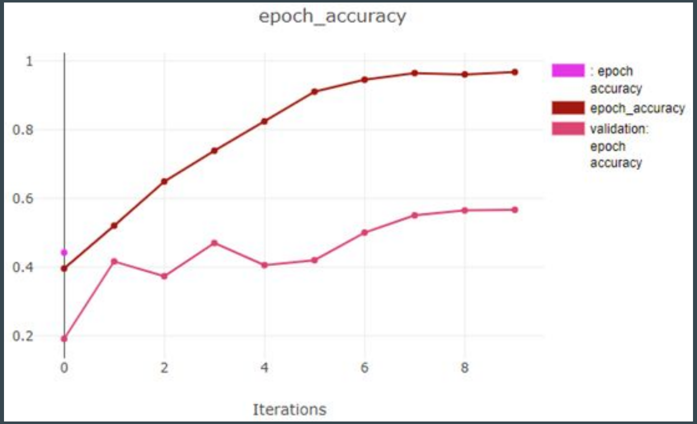
test

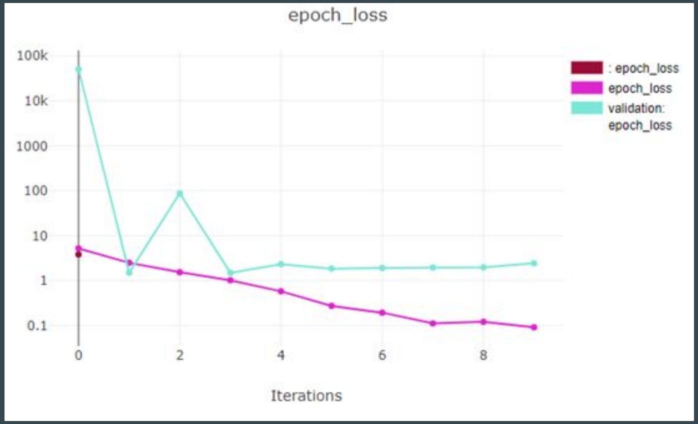


```
28/28 [==============================] - 2s 42ms/step - loss: 1.8984 - accuracy: 0.2150
[1.8984334468841553, 0.21502889692783356]
```

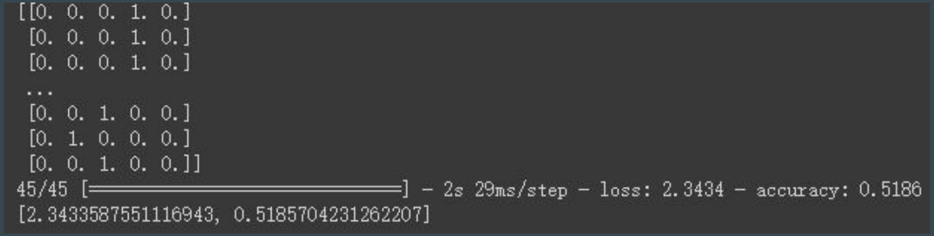
RGB scale

train





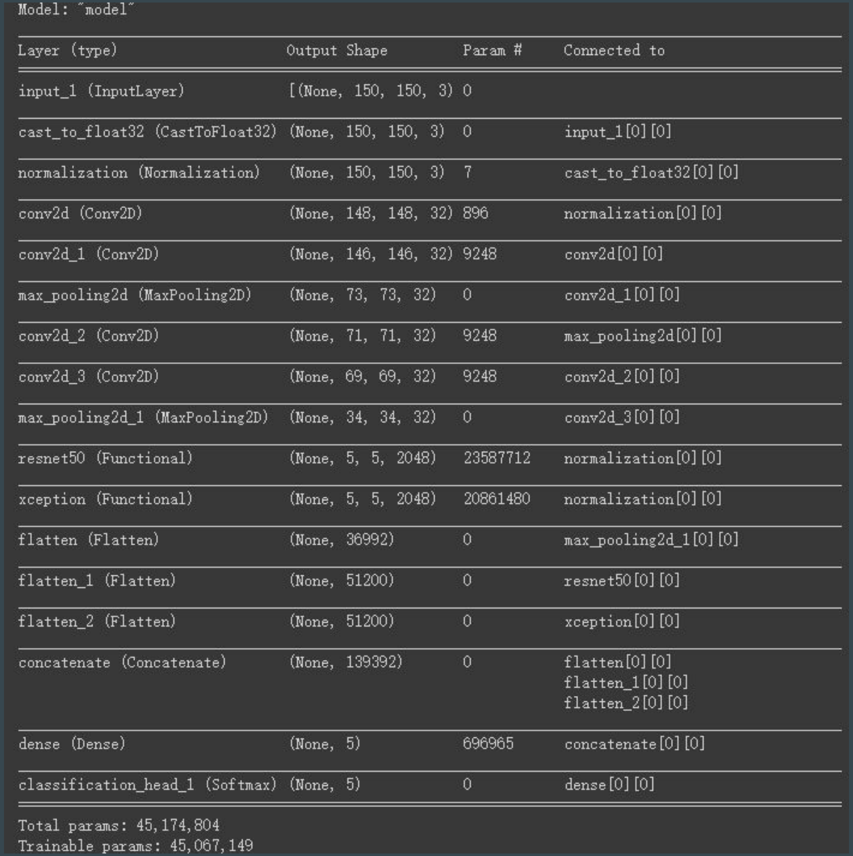
test



```
[[0. 0. 0. 1. 0.]
 [0. 0. 0. 1. 0.]
 [0. 0. 0. 1. 0.]
 ...
 [0. 0. 1. 0. 0.]
 [0. 1. 0. 0. 0.]
 [0. 0. 1. 0. 0.]]
45/45 [==============================] - 2s 29ms/step - loss: 2.3434 - accuracy: 0.5186
[2.3433587551116943, 0.5185704231262207]
```

Model summary

Best fit model



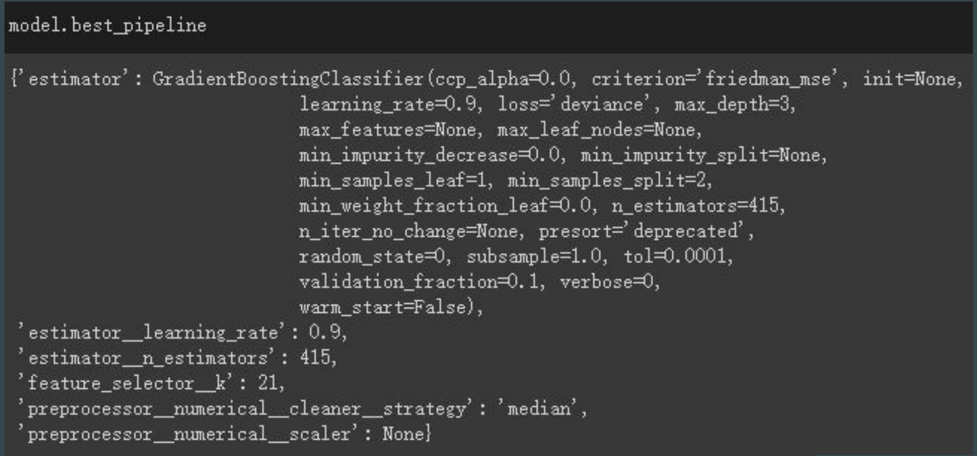
```
Model: "model"
________________________________________________________________________________________________________
Layer (type)                      Output Shape          Param #     Connected to
========================================================================================================
input_1 (InputLayer)              [(None, 150, 150, 3)  0
________________________________________________________________________________________________________
cast_to_float32 (CastToFloat32)   (None, 150, 150, 3)   0           input_1[0][0]
________________________________________________________________________________________________________
normalization (Normalization)     (None, 150, 150, 3)   7           cast_to_float32[0][0]
________________________________________________________________________________________________________
conv2d (Conv2D)                   (None, 148, 148, 32)  896         normalization[0][0]
________________________________________________________________________________________________________
conv2d_1 (Conv2D)                 (None, 146, 146, 32)  9248        conv2d[0][0]
________________________________________________________________________________________________________
max_pooling2d (MaxPooling2D)      (None, 73, 73, 32)    0           conv2d_1[0][0]
________________________________________________________________________________________________________
conv2d_2 (Conv2D)                 (None, 71, 71, 32)    9248        max_pooling2d[0][0]
________________________________________________________________________________________________________
conv2d_3 (Conv2D)                 (None, 69, 69, 32)    9248        conv2d_2[0][0]
________________________________________________________________________________________________________
max_pooling2d_1 (MaxPooling2D)    (None, 34, 34, 32)    0           conv2d_3[0][0]
________________________________________________________________________________________________________
resnet50 (Functional)             (None, 5, 5, 2048)    23587712    normalization[0][0]
________________________________________________________________________________________________________
xception (Functional)             (None, 5, 5, 2048)    20861480    normalization[0][0]
________________________________________________________________________________________________________
flatten (Flatten)                 (None, 36992)         0           max_pooling2d_1[0][0]
________________________________________________________________________________________________________
flatten_1 (Flatten)               (None, 51200)         0           resnet50[0][0]
________________________________________________________________________________________________________
flatten_2 (Flatten)               (None, 51200)         0           xception[0][0]
________________________________________________________________________________________________________
concatenate (Concatenate)         (None, 139392)        0           flatten[0][0]
                                                                    flatten_1[0][0]
                                                                    flatten_2[0][0]
________________________________________________________________________________________________________
dense (Dense)                     (None, 5)             696965      concatenate[0][0]
________________________________________________________________________________________________________
classification_head_1 (Softmax)   (None, 5)             0           dense[0][0]
========================================================================================================
Total params: 45,174,804
Trainable params: 45,067,149
```

Project 1-2

-.txt data classification/regression with self-built automl model

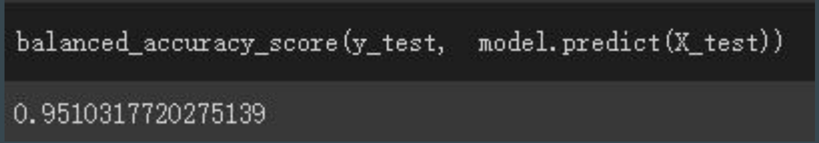
Tested on breast_cancer data/txt file

Overall pipeline is similar to autokeras, still in progress to refine parameters



```
model.best_pipeline
```

```
{'estimator': GradientBoostingClassifier(ccp_alpha=0.0, criterion='friedman_mse', init=None,
                            learning_rate=0.9, loss='deviance', max_depth=3,
                            max_features=None, max_leaf_nodes=None,
                            min_impurity_decrease=0.0, min_impurity_split=None,
                            min_samples_leaf=1, min_samples_split=2,
                            min_weight_fraction_leaf=0.0, n_estimators=415,
                            n_iter_no_change=None, presort='deprecated',
                            random_state=0, subsample=1.0, tol=0.0001,
                            validation_fraction=0.1, verbose=0,
                            warm_start=False),
 'estimator__learning_rate': 0.9,
 'estimator__n_estimators': 415,
 'feature_selector__k': 21,
 'preprocessor__numerical__cleaner__strategy': 'median',
 'preprocessor__numerical__scaler': None}
```



```
balanced_accuracy_score(y_test,  model.predict(X_test))
```

```
0.9510317720275139
```

Discussion

- Differences are observed in different trains. Result hard to reproduce.
- Still takes quite a bit computational power, parallel training with GPU would be helpful
- Need better understanding behind the hp tuning for better performance

Potential Optimizations

- Make our auto model more stable for training
- Could implement transfer training in next steps

Future of AutoML is bright

- Data scientist's productivity
- Deep Learning improvement
- Getting more and more exposed to business models