Data Scientist – Home Assignment

Aviad Baram

https://github.com/aviadba/Aviad_Baram_home_assignment.git

Problem outline

Build a binary classification model to predict the outcome of test **TLJYWBE** using a dataset that contains a history of testing results.

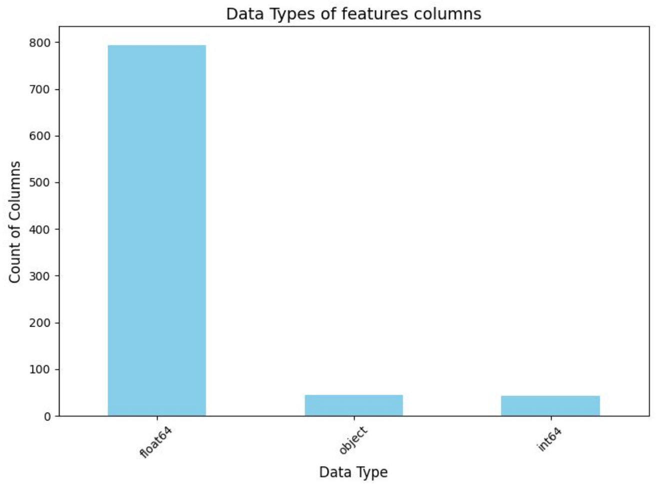
Source data - home_assignment.feather

Data - exploratory

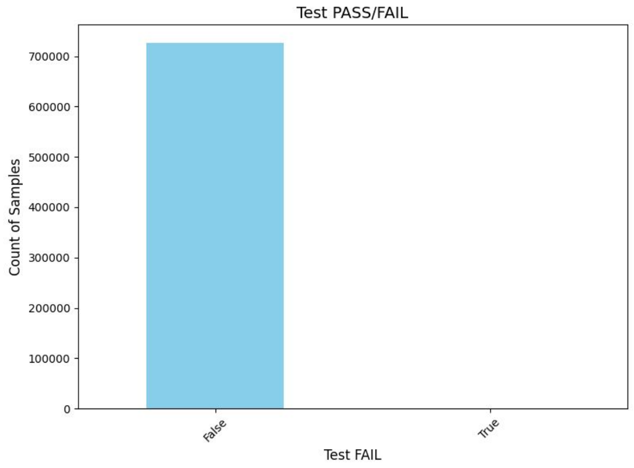
Raw data:
Samples: 726288
Features: 881

NaN ratio: 12.47%

Data dtypes
Float: 794
Categorical: 44
Integer: 43



Total test FAIL: 64
Total test PASS: 726224



Data - exploratory

Observations:

- Data set in highly imbalanced with PASS/FAIL ratio > 10^4. Classifications based on a lot of features with few samples are prone to overfitting ('curse of dimensionality').
- Number of FAIL samples is small. **Challenge is high recall of test FAIL.**
- Float64 is the majority dtype class

Decision:

- Drop all samples with a NaN response
- Drop all columns with NaN's (no need for imputation - reduces model bias)
- Build model based on numerical features only.
  - Majority class
  - Easier to manipulate
  - Categorical features often increase dimensionality
- Under sample majority class (PASS) and over-sample minority class (FAIL) to create a balanced dataset
- Split into Train/Test sets with at a 0.2 ratio

Implemented in: Pipe.imbalance_split_train_test followed by `Pipe.rawDataProcess` methods
Feature dimensions reduced to: **68.** **Cons** - may influence model generalization

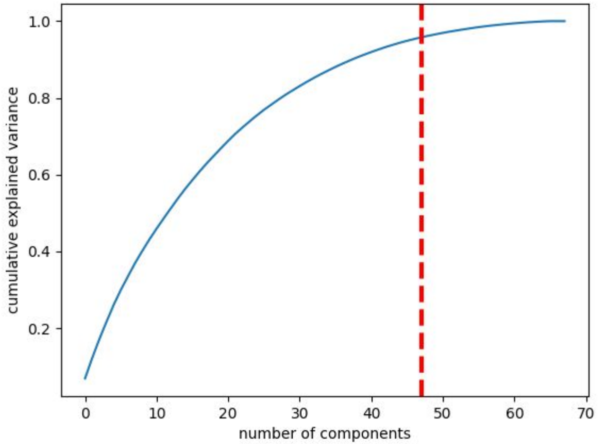
Pipeline

- Standardise data
- Reduce dimensionality by PCA - maintain 95% variance
- Standardise PCA features

Implemented in `Pipe.pca_fit_transform` method

- Train on Random Forest Classifier
  - Set hyperparameters: n_trees, max_depth and max_samples_split by cross-validation (3-fold)
  - **Hyperparameters tuned to max FAIL-recall**
- Predict Test data

Implemented in `Pipe.predict` method



RFC method was selected because it works well on small imbalanced datasets.

Cons - limited interpretability (especially after PCA), does not support RoC AUC tuning (i.e. does not generate a classification 'probability'. Other alternatives are One-class SVM, logistic regression

Results

Train

	Predicted FAIL	Predicted PASS
Actual FAIL	**101**	**2**
Actual PASS	**3**	**92**

Precision: 0.97, **Recall: 0.98**, F1:0.98

Test

	Predicted FAIL	Predicted PASS
Actual FAIL	**23**	**2**
Actual PASS	**9**	**18**

Precision: 0.72, **Recall: 0.92**, F1:0.81

Summary - the model generalises in identifying FAIL tests. Overfits PASS test as evident by PASS tests miss-classification.