Modul 3

Deployment and Feedback

Data Science Program

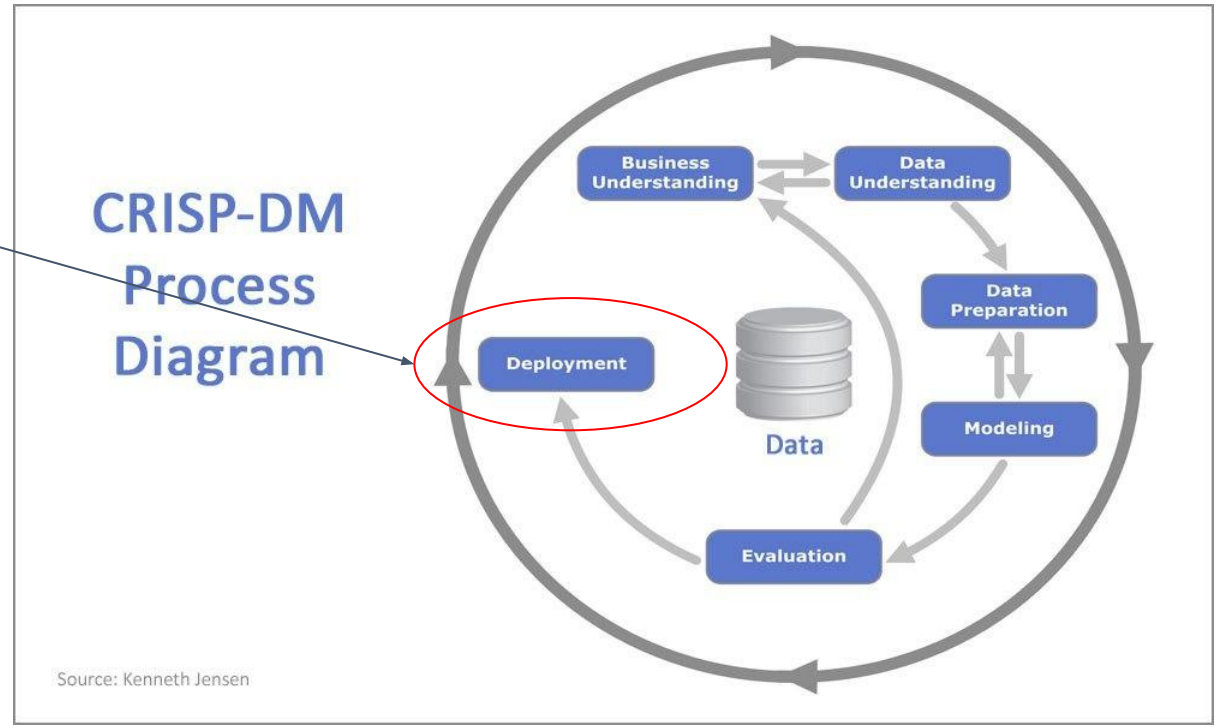
Purwadhika
Startup and Coding School

Outline

Introduction to Deployment

Training and Saving Model

Model Monitoring and Maintenance Plan



Purwadhika
Startup and Coding School

Introduction to Deployment

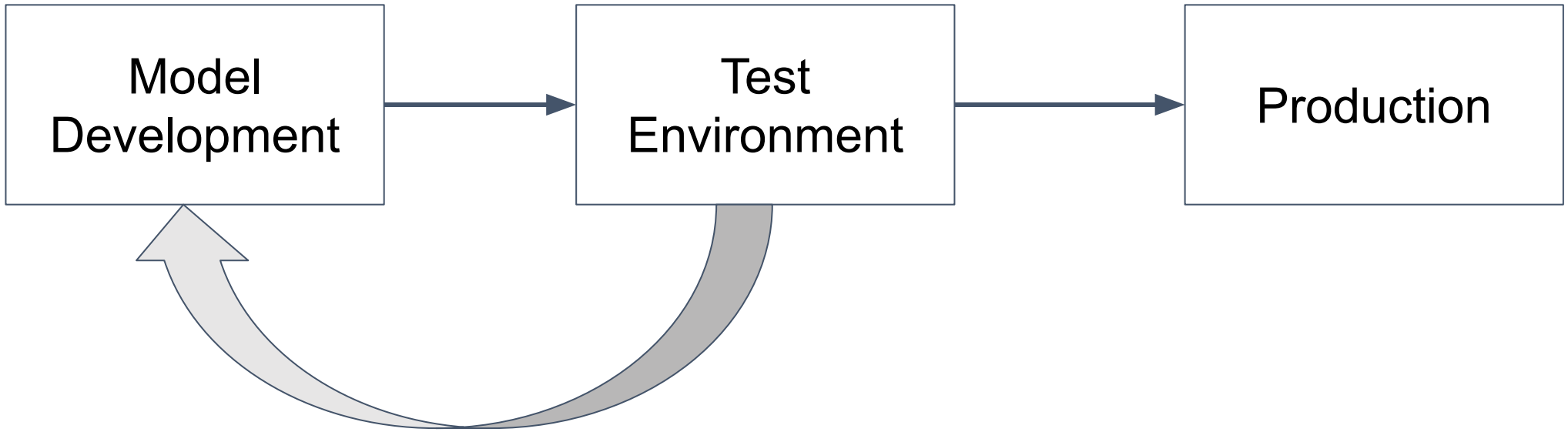
Purwadhika
Startup and Coding School

What is Deployment ?

Put your machine learning model into use.

Purwadhika
Startup and Coding School

Deployment Process



Model Development

- Approach (regression, classification, unsupervised learning etc)
- Data Preprocessing
- Model Training
- Model Evaluation

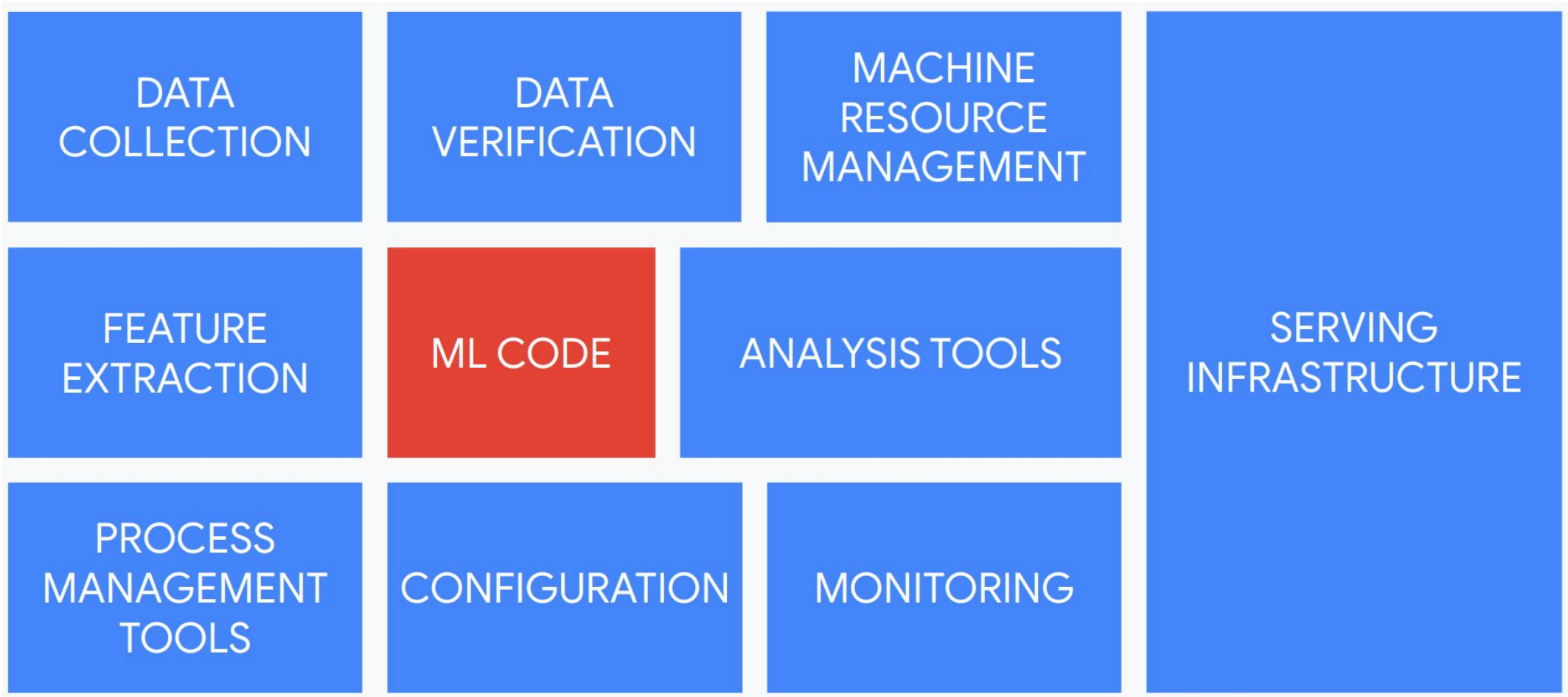
Deployment Environment

	Pattern 1 (REST API)	Pattern 2 (Shared DB)	Pattern 3 (Streaming)	Pattern 4 (Mobile App)
Training	Batch	Batch	Streaming	Streaming
Prediction	On the fly	Batch	Streaming	On the fly
Prediction result delivery	Via REST API	Through the shared DB	Streaming via Message Queue	Via in-process API on mobile
Latency for prediction	So so	High	Very Low	Low
System Management Difficulty	So so	Easy	Very Hard	So so

Source: https://www.udemy.com/course/deployment-of-machine-learning-models/

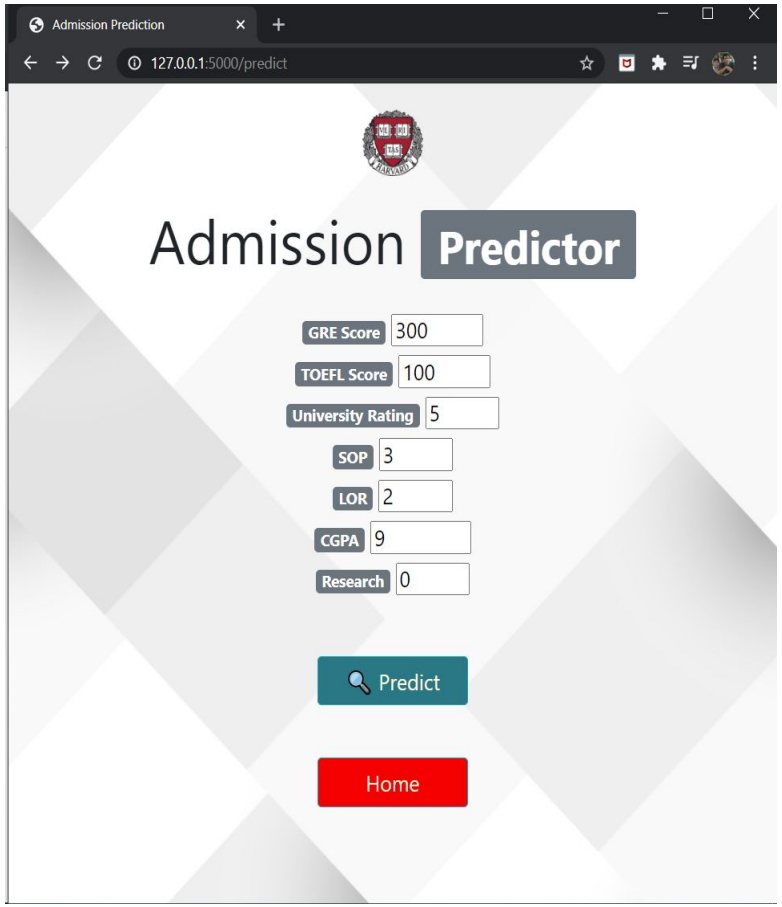
Things to Consider in Deployment

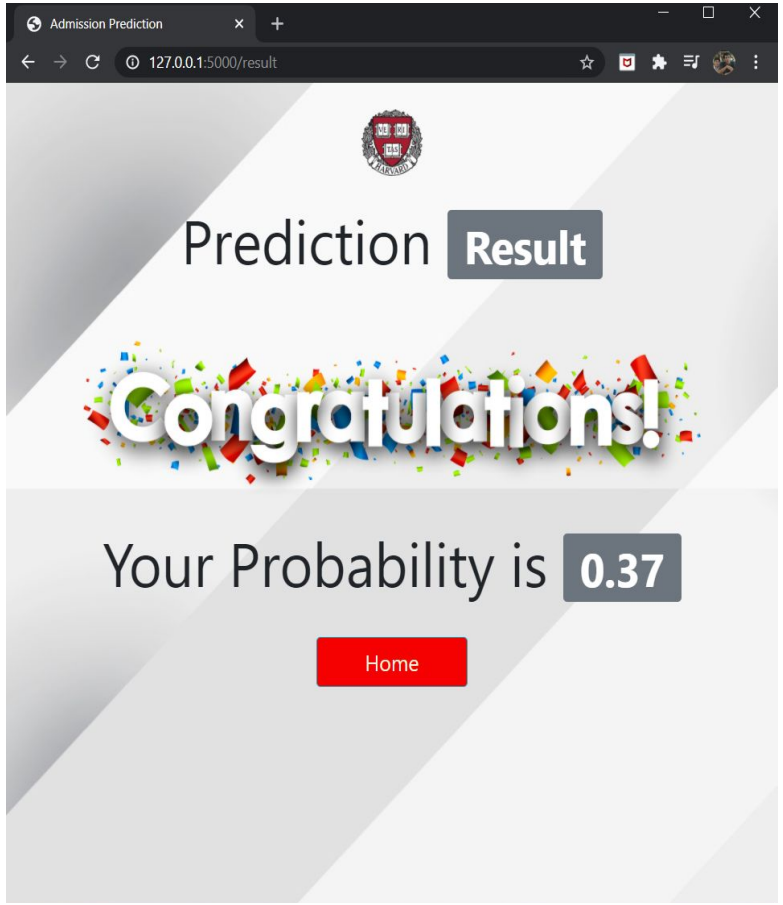
- Modularity
- Reproducibility
- Scalability
- Extensibility
- Testing
- Automation



Purwadhika
Startup and Coding School

Deployment Example : Web App





Purwadhika
Startup and Coding School

Training and Saving Model

Purwadhika
Startup and Coding School

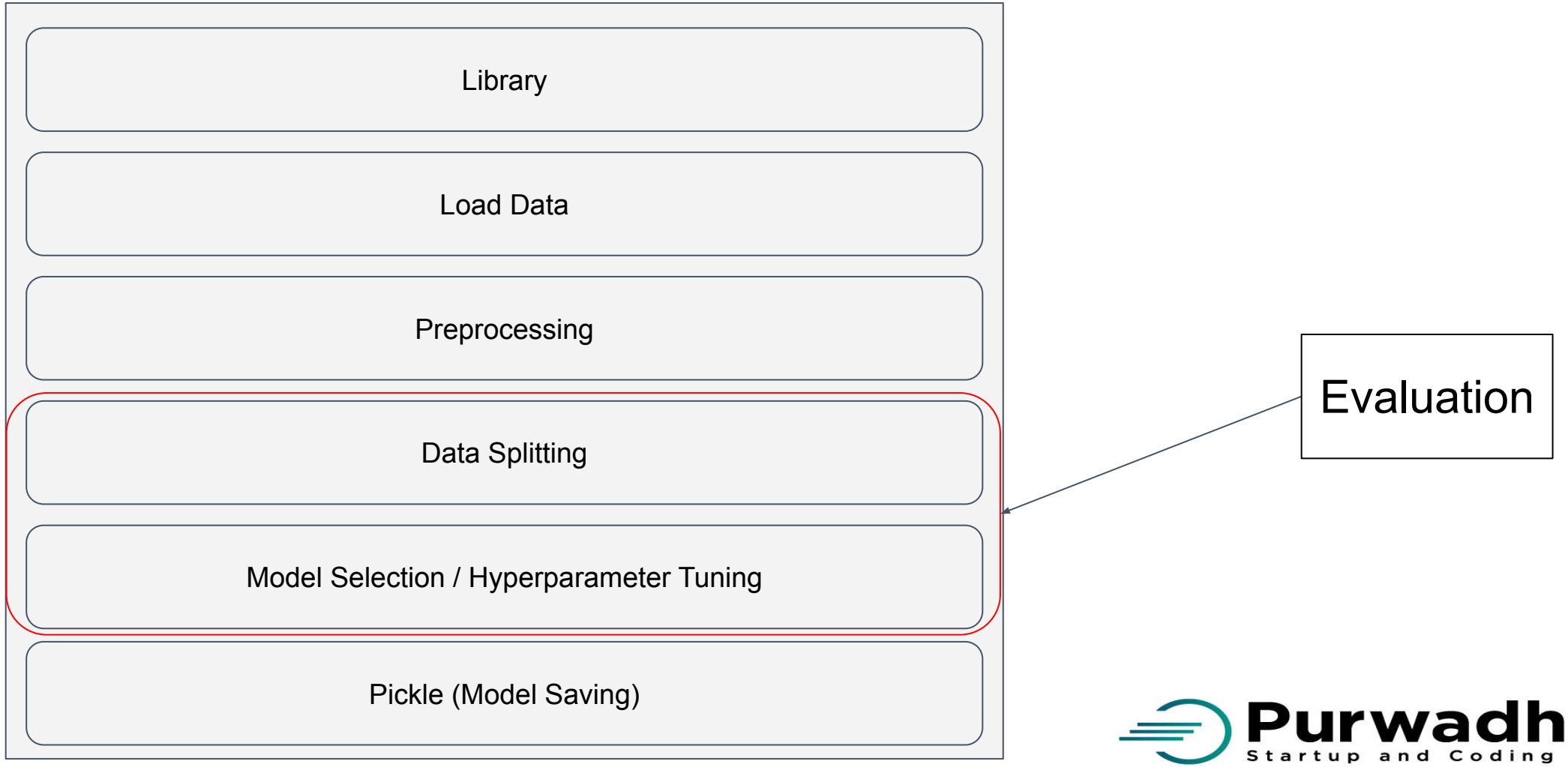
Training and Saving Model - Pickle

- Trained model can be saved to be used in the future
- Pickle library in python can save any model
- Model is saved with .sav extension

Training and Saving Model

- Training Script
- Prediction Script

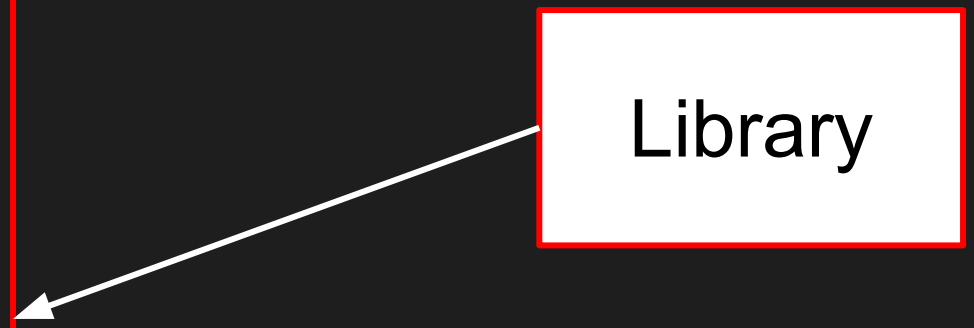
Training Script Structure



Purwadhika
Startup and Coding School

Library

```python
# Basic Operations
import pandas as pd
import numpy as np

# ML Models
from sklearn.ensemble import RandomForestClassifier

# Feature Engineering
from sklearn.pipeline import Pipeline
from sklearn.impute import SimpleImputer
from sklearn.preprocessing import OneHotEncoder
from sklearn.compose import ColumnTransformer
import category_encoders as ce

# Evaluation
from sklearn.model_selection import GridSearchCV
from sklearn.model_selection import cross_val_score, StratifiedKFold
from sklearn.model_selection import train_test_split
from sklearn.metrics import f1_score

# Model
import pickle

# warnings
import warnings
warnings.filterwarnings('ignore')
```



Purwadhika
Startup and Coding School

Data, Preprocessing

```
# data
df_adult = pd.read_csv('adult_training.csv')
```

```
# preprocess method
df_adult.replace('?',np.nan,inplace = True)
binary_encoder_pipeline = Pipeline([
                                    ('imputer',SimpleImputer(strategy = 'constant',fill_value = 'NC')),
                                    ('binary encoder',ce.BinaryEncoder())
])
transformer = ColumnTransformer([
                                  ('one hot encoder',OneHotEncoder(drop = 'first'),['relationship','race','sex']),
                                  ('binary encoder',binary_encoder_pipeline,['workclass','marital.status','occupation','native.country'])
                                 ], remainder = 'passthrough')

# ready dataset
X = df_adult.drop(columns = ['income'])
y = np.where(df_adult['income']=='>50K',1,0)
```

Library

Preprocessing

Purwadhika
Startup and Coding School

Evaluation, Pickle

```python
# Model Selection
model = RandomForestClassifier()

estimator = Pipeline([
                    ('preprocess',transformer),
                    ('clf',model)
])

hyperparam_space = {
    'clf__max_depth':[3,4,5,6,7],
    'clf__n_estimators':[100,200,300]
}

skfold = StratifiedKFold(n_splits = 5)

grid_search = GridSearchCV(
    estimator, # model to tune
    param_grid = hyperparam_space, # hyperparameter space
    cv = skfold, # evaluation method
    scoring = 'f1', # metrics
    verbose = True, # show progres
    n_jobs = -1 # use all cores
)

grid_search.fit(X, y)
```

```python
# Model Pickling
grid_search.best_estimator_.fit(X,y) # FINAL MODEL
filename = 'Model Final.sav'
pickle.dump(grid_search.best_estimator_,open(filename,'wb'))
```

Cross Validation

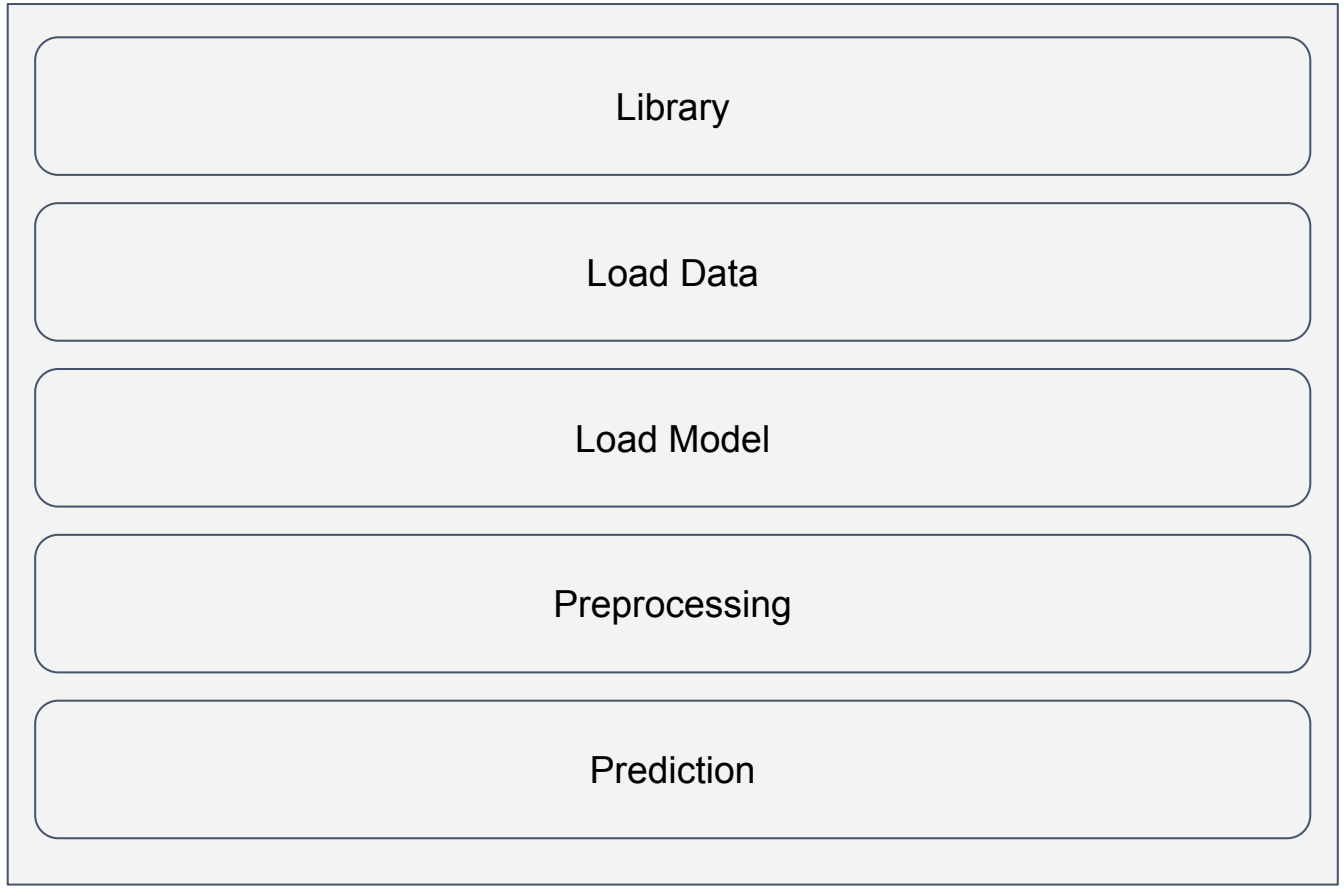
Evaluation

Metrics

Pickle

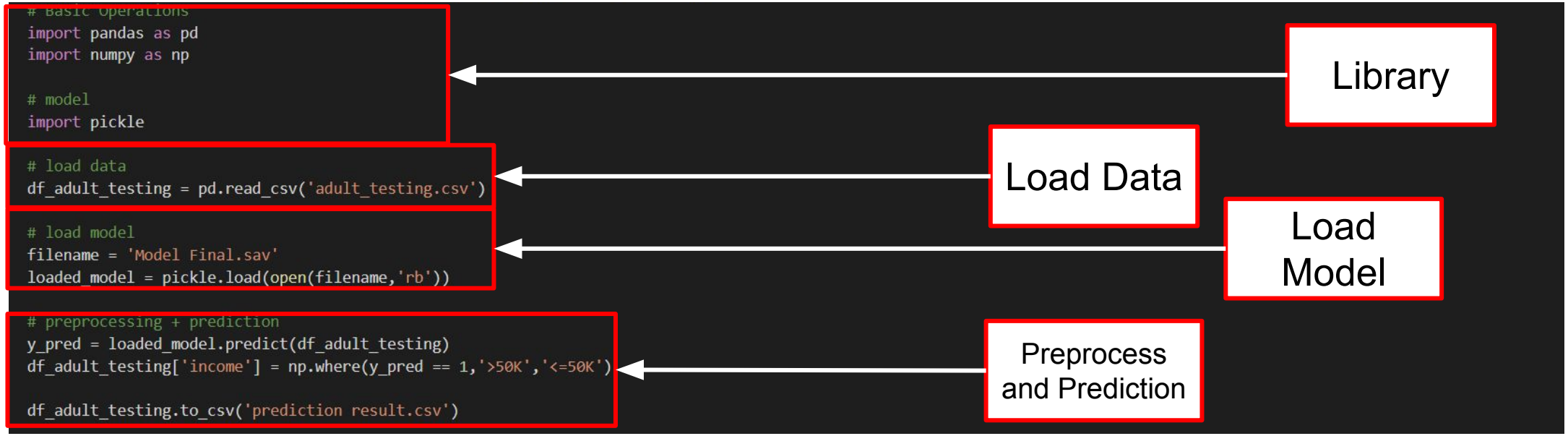
Purwadhika
Startup and Coding School

Prediction Script



Purwadhika
Startup and Coding School

Prediction Script Structure




```
# Basic Operations
import pandas as pd
import numpy as np

# model
import pickle
```

```
# load data
df_adult_testing = pd.read_csv('adult_testing.csv')
```

```
# load model
filename = 'Model Final.sav'
loaded_model = pickle.load(open(filename,'rb'))
```

```
# preprocessing + prediction
y_pred = loaded_model.predict(df_adult_testing)
df_adult_testing['income'] = np.where(y_pred == 1,'>50K','<=50K')

df_adult_testing.to_csv('prediction result.csv')
```

Purwadhika
Startup and Coding School

Model Monitoring and Maintenance Plan

Purwadhika
Startup and Coding School

Model Monitoring and Maintenance Plan

Deployment is an iterative process

We should monitor machine learning model after we deploy it

- Data Versioning
- Dictionary Versioning
- Feature Versioning
- Algorithm Versioning
- Transfer Learning (deep learning only)
- Interpretation Versioning

Data Versioning

Column and row can be growing

Cust ID	Age	...	Edu	Balance	...	Income	...	...	Default
C1									BAD
C2									BAD
C3									GOOD
C4									BAD
….									...
C2000									GOOD

Cust ID	...	Default
C1		BAD
C2		BAD
C3		GOOD
C4		BAD
….		...
C2000		GOOD
...		...
c2300		BAD

Purwadhika
Startup and Coding School

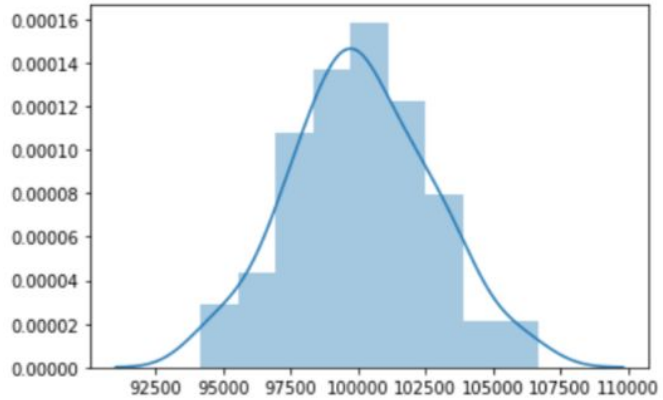
Dictionary Versioning

Data dictionary can be expanding and changing

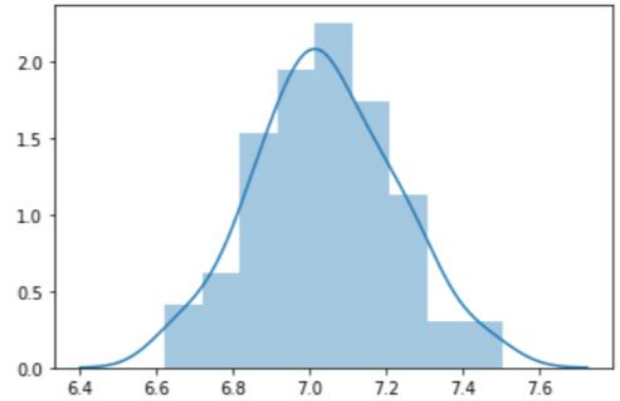
Feature	Details
Education	SD, SMP, SMA, S1, S2, S3
Occupation	Data Scientist, Data Engineer ... Data Analyst
...	

Feature	Details
Education	SD, SMP, SMA, S1, S2, S3, **Post-Doctoral**
Occupation	Data Scientist, Data Engineer ... Data Analyst, **Business Intelligence**
...	

Rupiah



Dollar



Purwadhika
Startup and Coding School

Feature Versioning

New feature also mean new possible feature engineering

Irrelevant feature can occur

Cust ID	...	Balance	...	Default
C1				BAD
C2				BAD
C3				GOOD
C4				BAD
….				...
C2000				GOOD

Cust ID	...	Balance	...	Income	Default
C1					BAD
C2					BAD
C3					GOOD
C4					BAD
….					...
C2000					GOOD

Balance to income = Balance / Income

Purwadhika
Startup and Coding School

Algorithm Versioning

Model winner can change

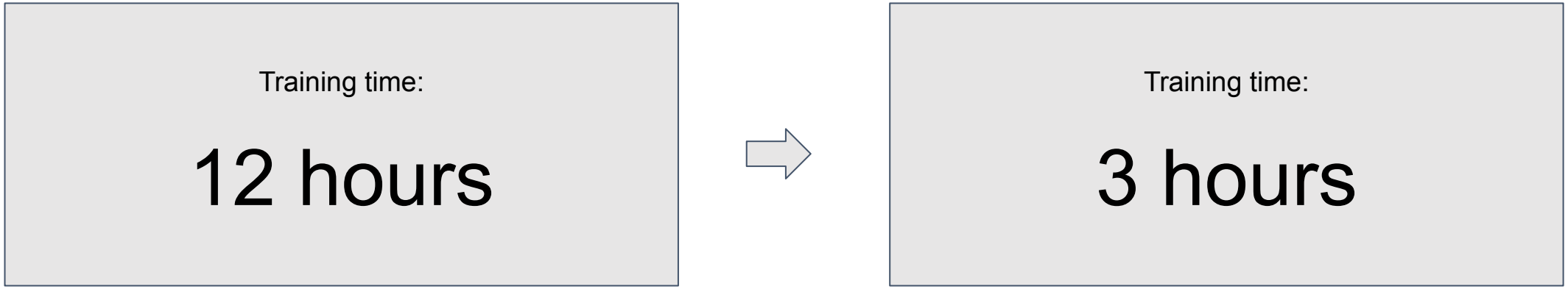
Model	Score
Logistic Regression	0.89
Random Forest	0.91
Gradient Boosting	0.87
Decision Tree	0.85

Model	Score
Logistic Regression	0.88
Random Forest	0.90
Gradient Boosting	0.92
Decision Tree	0.86

Purwadhika
Startup and Coding School

Transfer Learning

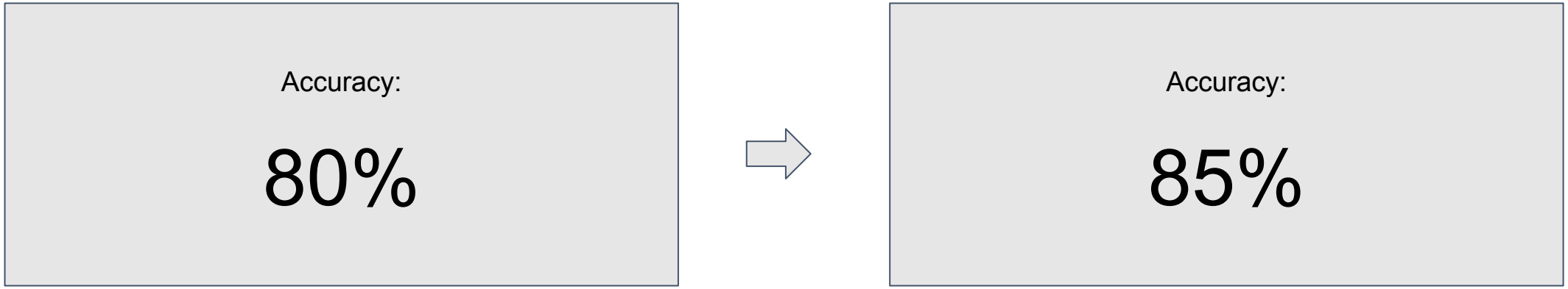
Continue training process from the current model (deep learning only)



Purwadhika
Startup and Coding School

Interpretation Versioning

Model insight can change overtime



Purwadhika
Startup and Coding School

References

https://towardsdatascience.com/deployment-of-machine-learning-model-demystified-part-1-1181d91815d2

https://www.udemy.com/course/deployment-of-machine-learning-models/

Purwadhika
Startup and Coding School