Flight delay Prediction

Presented by:
Đinh Gia Quang
Trần Thiện Nhân

Date Presented:
April 6th, 2023

Members

Đinh Gia Quang
1911900

Trần Thiện Nhân
2010481

Table of Context

01

Azure synapse

02

Dataset and solving problem

03

Demo code

Azure Synapse

Azure Synapse Analytics

Azure Synapse is a Microsoft Azure cloud-based integrated data analytics and big data processing service. It provides tools and capabilities to collect, store, process, and access data from various sources, including distributed data sources..

Azure Synapse is designed to support data warehousing, big data processing, and artificial intelligence (AI) applications to analyze data and make business decisions. It integrates the capabilities of Azure Data Factory, Azure Stream Analytics, Azure SQL Data Warehouse, and Power BI to provide a comprehensive data analytics solution for businesses

Azure Spark Pool

- Azure Spark pool is a Microsoft Azure integrated service designed to provide a distributed environment for processing big data and parallel computing on the Apache Spark platform.

- With Azure Spark pool, users can easily create and manage Spark clusters on Azure, while providing flexible scaling and automated resource management to meet businesses' computing and data processing needs.

- Azure Spark pool also provides a range of integrated tools for data processing, including Spark SQL, Spark Streaming, and Spark MLlib, making it easy for users to process and analyze big data on Azure quickly and efficiently.

Dataset and Solution

Source: Airlines Dataset to predict a delay | Kaggle.

About Dataset

Airlines dataset has 539383 instances and 8 different features. It will be used to predict that a flight given the information of the scheduled departure is delayed or not.

Different Feature Names:

Airline, Flight, AirportFrom, AirportTo, DayOfWeek, Time, Length, Delay

Solution

There are 5 steps to do:

- Preparing data
- Visualize some features of data
- Feature extraction
- Correlation among features
- Some ML models

Outline

Preparing data:

- Read data
- Count null value in dataset

Visualize some feature of data

Transform data

Count value

Remove outliers

Corelation among features

Some ML Models:

Transform data for ML models

Logic Regression

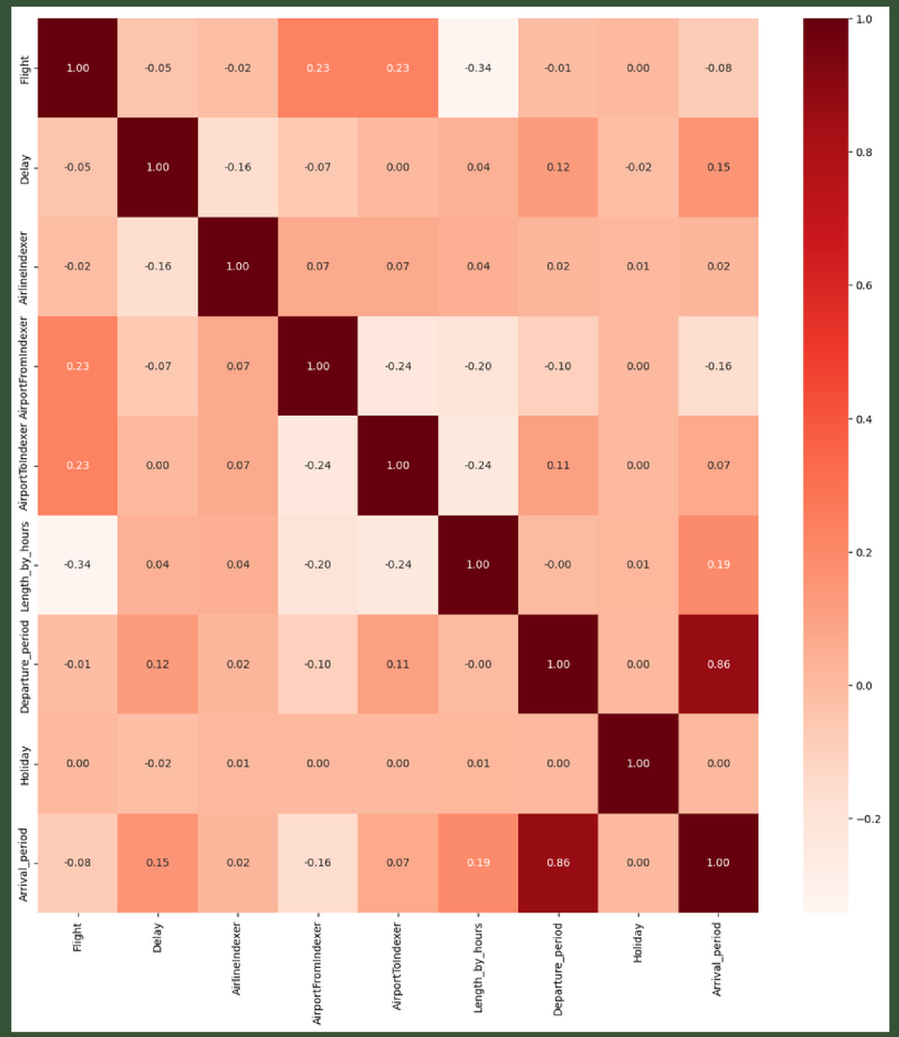
Auto tuning Logistic regression

Decition tree

Gradient Boosted tree classifier

Naive Bayes

Corelation among feature



- Correlation coefficients are numerical measures that indicate the strength and direction of the relationship between two or more variables.
- The diagonal of the table displays the correlation coefficient of each variable with itself, which is always 1. The off-diagonal elements of the matrix show the correlation coefficients between the pairs of variables.
- It can also be used to identify any multicollinearity or redundant variables in a regression analysis.

Machine Learning Algorithms

1

Basic Statistics

Classification and Regression

2

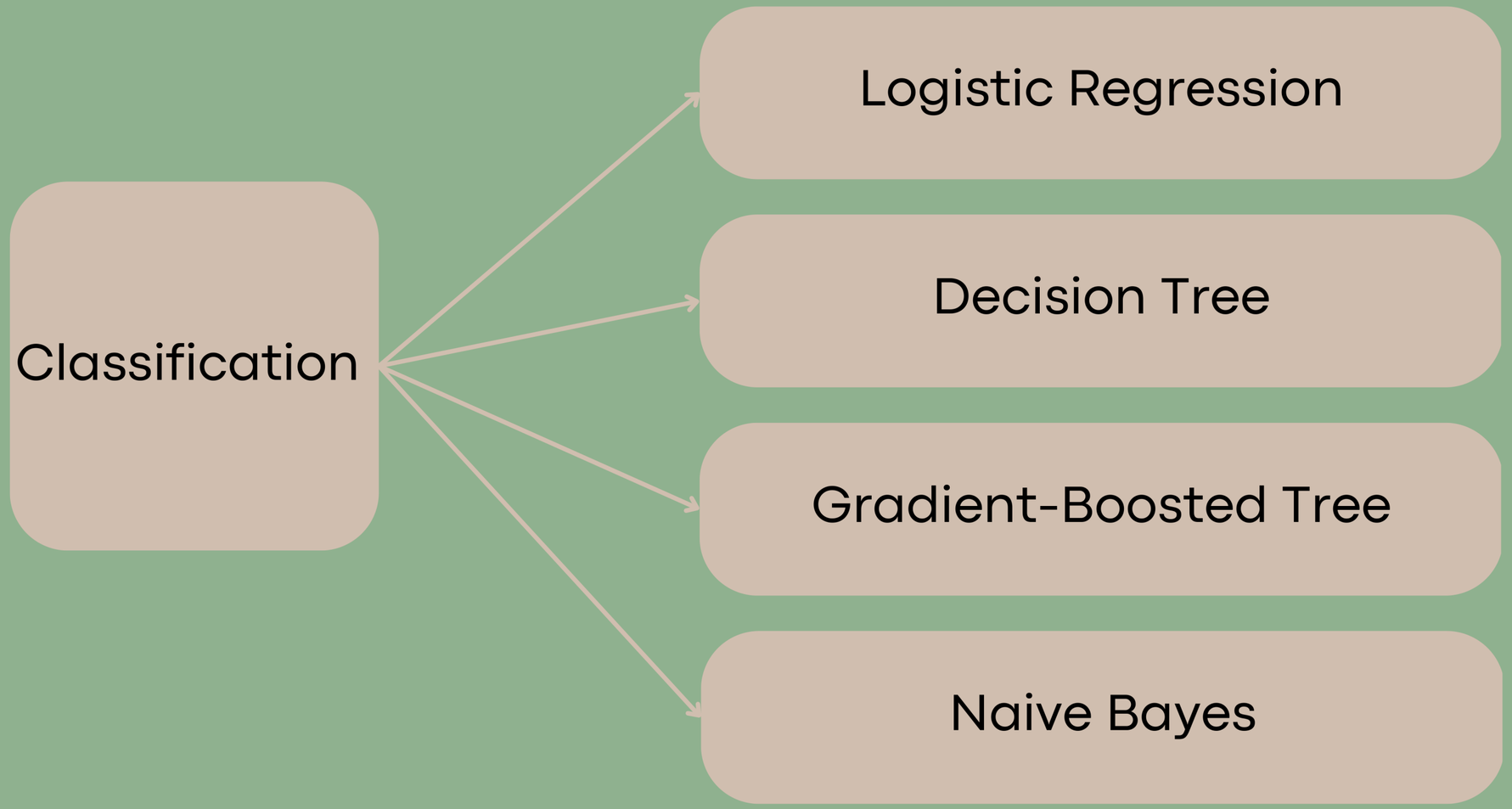
3

Clustering

Collaborative filtering

4

Machine Learning Algorithms



Auto Tuning ML Models

Why do you need to tune machine learning hyperparameters automatically?

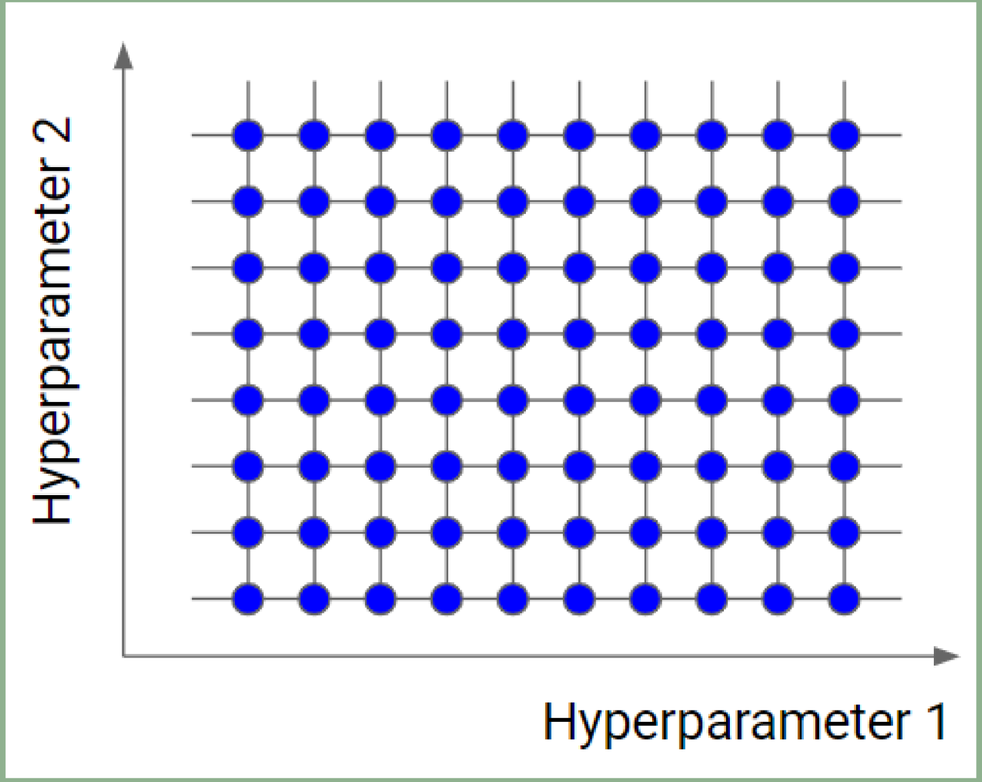
- To get the highest performance of your model
- Different types and numbers of hyperparameters depended on each algorithm.
- To save time when tuning model

Different approaches:

- Grid Search
- Random Search

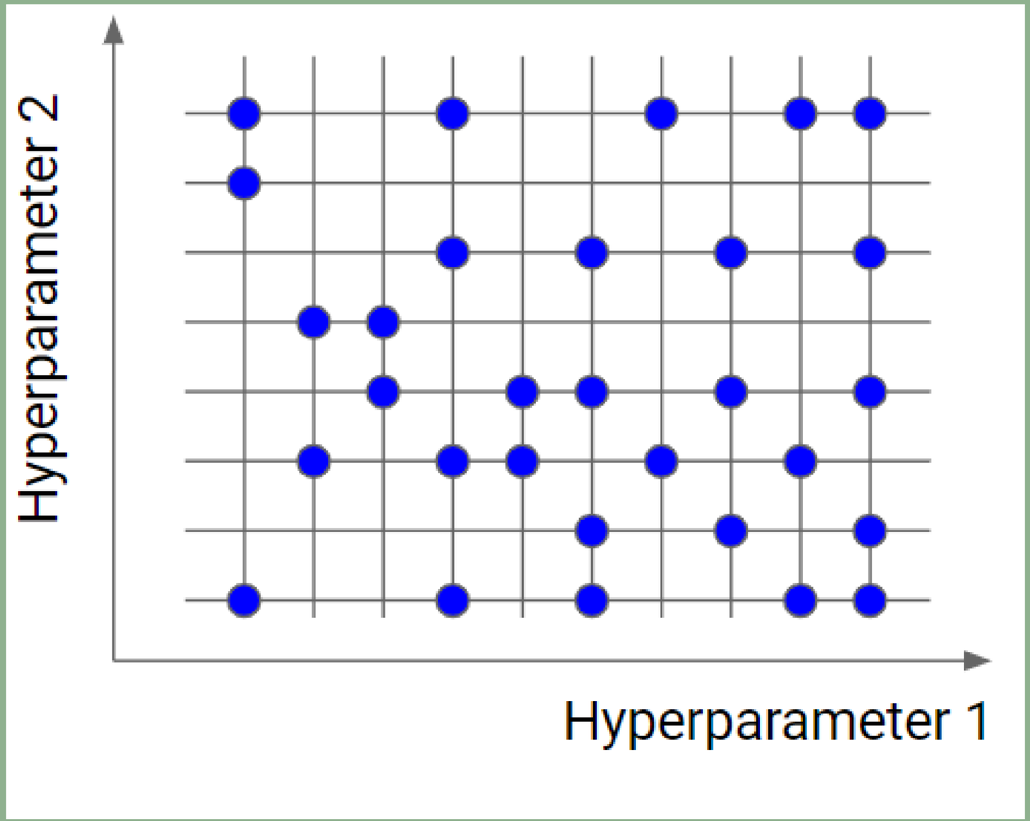
Grid Search

- The simplest algorithm for hyperparameter tuning
- Divide the domain of the hyperparameters into a discrete grid.
- Try every combination of values of the grid with some metrics and cross-validation.
- Choose the point which maximizes the average value in cross-validation



Random Search

- Similar to grid search
- Instead of using all the points in the grid, it tests only a randomly selected subset of these points.
- The smaller subset, the faster but less accurate the optimization
- The larger subset, the more accurate the optimization but the closer to a grid search.



Results

Logistic Regression

```
lr = LogisticRegression(featuresCol="features", labelCol="Delay", regParam=0.3, elasticNetParam=0.8)
pipeline = Pipeline(stages=[featureIndexer, lr])
model = pipeline.fit(training_data)
```

```
StatementMeta(spark, 1, 73, Finished, Available)
```

```
predictions = model.transform(test_data)
evaluator = MulticlassClassificationEvaluator(labelCol="Delay", predictionCol="prediction", metricName="accuracy")
accuracy = evaluator.evaluate(predictions)
print("Accuracy on test data = %g" % accuracy)
```

```
StatementMeta(spark, 1, 74, Finished, Available)

Accuracy on test data = 0.555594
```

Results

Auto Tuning Logistic regression

```
lr = LogisticRegression()
evaluator = MulticlassClassificationEvaluator(labelCol="label", predictionCol="prediction", metricName="accuracy")
grid = ParamGridBuilder().addGrid(lr.regParam, np.arange(0, 1, .05)).addGrid(lr.elasticNetParam, np.arange(0, 1, 0.2)).build()
tvs = TrainValidationSplit(estimator=lr, estimatorParamMaps=grid, evaluator=evaluator, parallelism=1, seed=42)
tvsModel = tvs.fit(dataset=training_data.select("features", fn.col("Delay").alias("label")))
```

```
StatementMeta(spark, 1, 43, Finished, Available)
```

```
accuracy = evaluator.evaluate(tvsModel.transform(test_data.select("features", fn.col("Delay").alias("label"))))
print(tvsModel.validationMetrics)
print("Accuracy on test data = %g" % accuracy)
```

```
StatementMeta(spark, 1, 44, Finished, Available)

[0.6115845347458473, 0.6115845347458473, 0.6115845347458473, 0.6115845347458473, 0.6115845347458473, 0.6118283263704968, 0.612182932369987,
Accuracy on test data = 0.616134
```

Demo code

Email

quang.dinh.leak_tb@hcmut.edu.vn
nhan.trankhmtbku@hcmut.edu,vn