Today's session

Exam Preparation

Business Data Analytics, Machine Learning

Mathias Havskov Artmann

mh

Agenda

1
Introduction

 Mathias

2
An example paper

 Mathias

3
The Oral Exam

 Mathias

4
Questions

 Mathias

5
Individual Projects

 Mathias

1 Introduction

Learning Objectives

- Understand and deploy techniques for exploring and analyzing structured data
- Understand and deploy basic machine learning techniques for classification and regression
- Understand and deploy techniques for visualizing and presenting results of data analytics
- Demonstrate an analytical understanding of business, societal, and ethical issues in the application of data analysis techniques

Materials for this course

- You can find the materials for this session on:
  https://go.dm-union.dk/bda

mh

What does the boxes mean?

- You will meet several types of info boxes on the slides that will each have an icon

This is referring to a Moodle resource

This is general info

This contains a Python code example

Important Definitions

- Estimator (model):
  - Algorithm that learns from data to make predictions or decisions.
- Hyperparameter:
  - Settings adjusted before training that influence the model's learning process.
- Transformers:
  - Tools that preprocess data to make it suitable for analysis.
- Pipeline:
  - Chains multiple estimators and transformers together
  - A row of actions. Do this then do this.
- Scaling
  - Standardizing values to remove big differences between scales in different columns
- Stratification
  - Ensuring data splits maintain proportional representation of categories.

2 An example paper

Be aware

- This paper shows one way to conduct the project – there are many different approaches, and you should take the one that you feel most comfortable with/makes the most sense for your case
- The machine learning results in this case were not good, but what matters most is the approach (not all machine learning projects in the “real world” succeed)
- Always think about what you are trying to achieve

Be aware

- Don’t plagiarize!
- Only use code that you understand

Finding a dataset

- You can check out the links on Moodle or the suggestions uploaded by Dan
- Remember to find a dataset that has business relevance and a machine learning task that make sense

Exploring a dataset

- If you have found your dataset on Kaggle you can see some basic information about the features on the page
- You can also use Ydata Profiling to generate a HTML-report showing basic information about your dataset

Why this dataset?

- The dataset was located at Kaggle
- Airline delays constitute a significant cost to the aviation industry each year (business relevance – remember it’s a business school)
- The features/columns seem relevant to be making predictions
- There is a column that can be used as the target variable

mh

Different target variables

Need a recap on levels of measurement?

Link: Levels of Measurement

- Supervised Machine Learning
  - Classification (coffee type, bank approval etc.)
    - Predicting a category
    - Binary or multiclass-classification
    - Nominal or ordinal (categorize)
  - Regression (prices, numeric rating, etc.)
    - Predicting a number
    - Interval and ratio (equal intervals)
- There are other methods – forecasting, NLP, etc.

In this case

- We could have chosen either classification or regression
  - Regression: Guessing the delay time in minutes
  - Classification: Is the flight more than 15 minutes delayed? Yes or No
- In this case, we chose classification because
  - Often flights are a few minutes late without any issues – this can be caught up in the air
  - FAA defines a flight delay as more than 15 minutes (remember sources)

Finding relevant features

- Avoid features that could spill or directly lead to the target (also features that you might not have at the time of prediction)
- Treat them probably
  - Should they be scaled? (depending on the estimator)
  - Should they be encoded? (**dummy** or label encoding)
  - Should they be converted? (to maintain ordinality, custom columns)
  - Are there NaN values
- Machine Learning works with numbers or math
- Descriptions, long texts, etc. are not good features (they need a lot of preprocessing using a multitude of methods)
  - Too many unique possibilities

What should be in the paper?

- Sections
  - Introduction
  - Business Relevance (remember ethical implications)
  - Data Description
  - Data Visualization
  - Data Preprocessing
  - Data Modelling and Results
    - Model 1…
    - Model 2…
  - Conclusion
  - References
  - Appendixes

Introduction – Funnel approach

In today’s world, the aviation industry holds a significant position in promoting global connectivity and driving economic growth. It can be considered one of the most competitive industries, contributing a sum of $2.7 trillion (3.6 per cent) to the world's GDP (Asquith, 2020). However, despite its importance, the industry faces challenges known to disrupt its operation. One such challenge is flight delays, which can not only impact the industry's efficiency and profitability but can also cause a significant inconvenience to many passengers. As one once said, “time is money” and therefore every minute of delay is a direct hit on an airline's profitability.

Subsequently, this paper will utilize a dataset that includes information regarding flight delays and non-delays, complemented by accompanying characteristics. The aim is to perform a range of machine learning models, improving each one of them in the “train” stage, thereby obtaining the best possible result when “testing.” Predicting flight delays successfully would allow firms to improve their operations, resulting in more satisfied customers and greater profits. This has the potential to revolutionize the aviation industry positively for many years to come.

Visualization

- Use this section to explore potential correlations and learn more about your data
- Find the most appropriate plots
- When you do visualize do it with purpose and thought
- You are simplifying data – it can be a strong method, but you can easily lose important distinctions
- No specific software is a requirement

Visualization

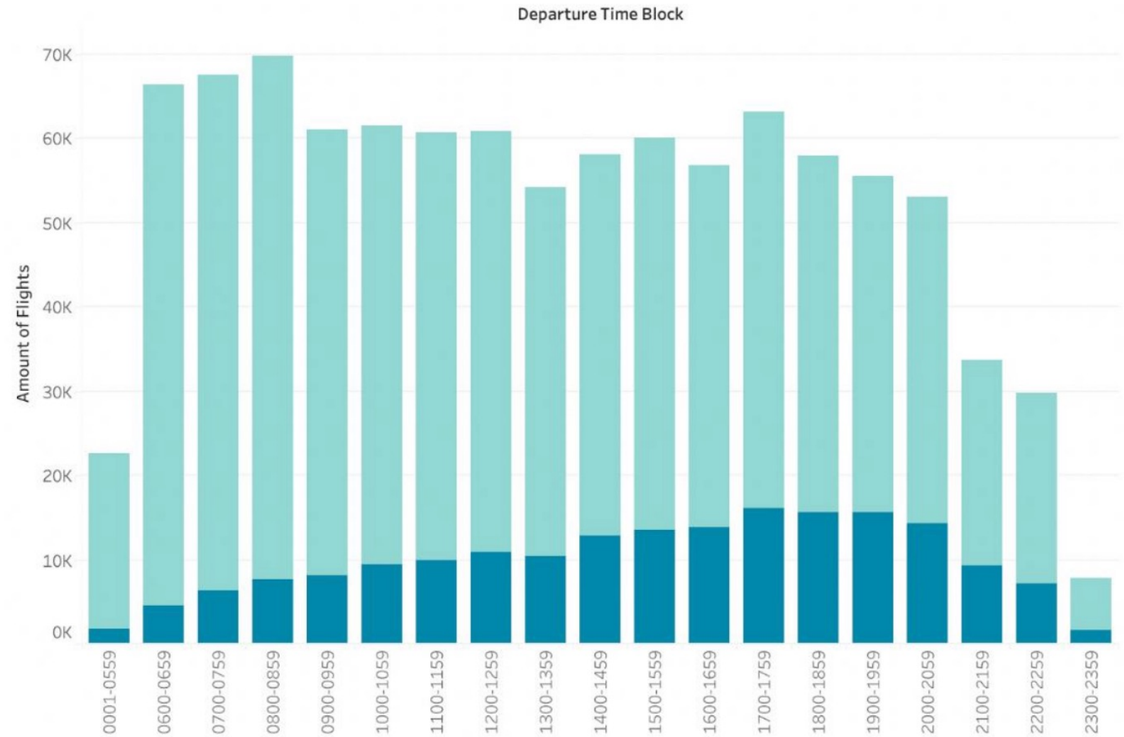


Figure 5 Distribution of delays during day

- This graph gives some insights, but be aware of what you can get from it – it is hard to compare delayed flights out of all since they are not in percent

Source: Example Exam Paper – Figure 5

Splitting the data

- So far you have worked with a single-split
  - Training Data
  - Testing Data
- This is not required in this course, but you can consider this a better approach
  - Training Data
  - Validation Data
  - Testing Data
- The same idea can be obtained by using cross-validation

Techniques to imbalanced data

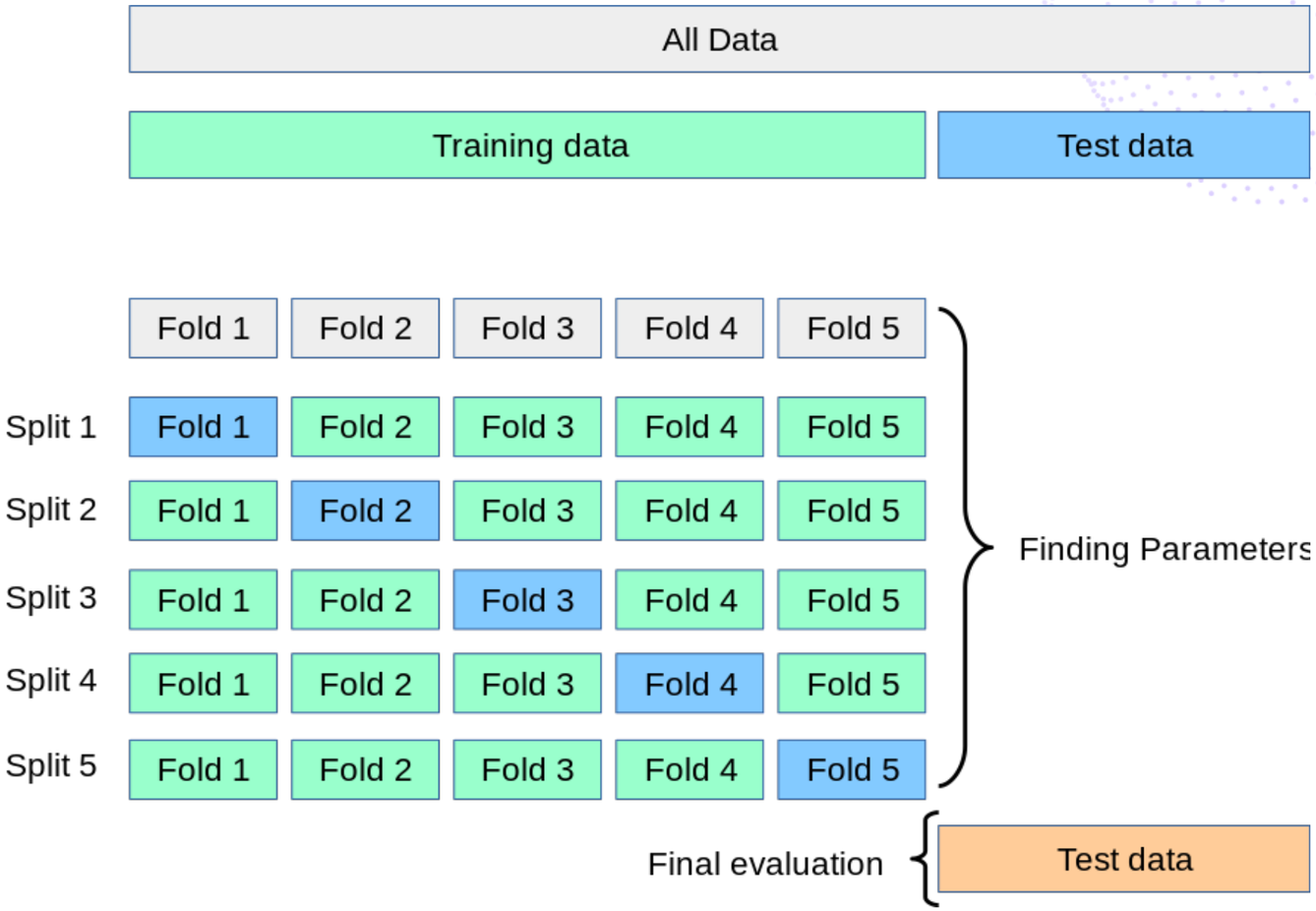
- Examples:
  - Random Undersampling
  - Synthetic Minority Over-sampling Technique (SMOTE)
- Be careful – normally you would keep the same proportions as in the original dataset

Grid Search

- A systematic walkthrough of some different settings for the hyperparameters of the estimator (model)
- Uses a technique called cross-validation
- This is the way to go

mh

Cross Validation



Source: Sci-Kit Learn: Cross-validation: evaluating estimator performance

mh

```python
# The scoring methods that needs to be calculated
scoring = {'accuracy': 'accuracy', 'precision': 'precision', 'recall': 'recall', 'f1': 'f1'}

# The different parameters that should be searched by GridSearchCV
param_grid = {
    'kneighborsclassifier__weights': ['uniform', 'distance'],
    'kneighborsclassifier__n_neighbors': list(range(1, 15, 2)),
    'kneighborsclassifier__metric': ['manhattan', 'euclidean', 'hamming', 'minkowski', 'chebyshev']
}

# Our pipeline that first scales the data without leakage and then fits the model
model = make_pipeline(StandardScaler(), KNeighborsClassifier())

# The grid is build. We are using f1 as our selection method
grid = GridSearchCV(model, param_grid, cv=3, refit='f1', scoring=scoring, return_train_score=True, verbose=3)

# We are fitting the training data
grid.fit(X_train_enc, y_train_enc)

# We are doing a prediction on our test data (used for confusion matrix)
y_pred = grid.predict(X_test_enc)

# Confusion matrix is generated based on the predictions and the actual labels
cm = confusion_matrix(y_test_enc, y_pred, labels=grid.classes_)

# Utilizing a customized function of previous work: https://github.com/DTrimarchi10/confusion_matrix
make_confusion_matrix(cm, group_names=labels, categories=categories, title='Tuned and Scaled', model='KNeighborsClassifier', model_type='neighbors', params=grid.best_params_)

# Generating Different Metrics
print(f'True Positives: {cm[1][1]}')
print(f'False Positives: {cm[0][1]}')
print(f'Training Accuracy: {grid.best_estimator_.score(X_train_enc, y_train_enc):.4f}')
print(f'Testing Accuracy: {grid.best_estimator_.score(X_test_enc, y_test_enc):.4f}')
print(f'Precision: {precision_score(y_test_enc, y_pred):.4f}')
print(f'Recall: {recall_score(y_test_enc, y_pred):.4f}')
print(f'F1: {f1_score(y_test_enc, y_pred):.4f}')
```

This shows the most essential code

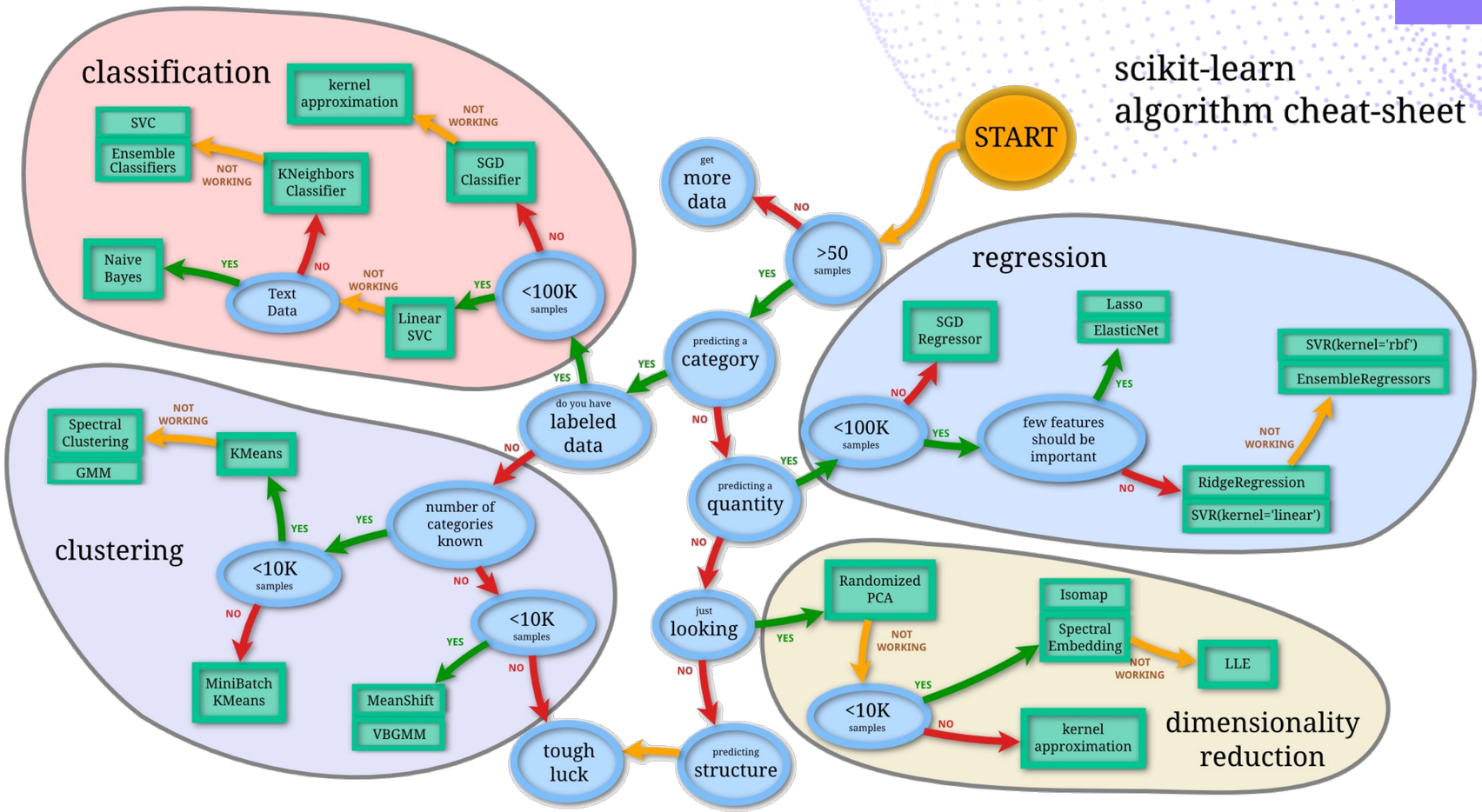
Source: Example Exam Paper – Appendix 3

mh

Choosing the right estimator

General rule

Rows should be at least 10x amount of columns



Source: Sci-Kit Learn: Choosing the Right Estimator

Choosing the right estimator

sklearn.tree.DecisionTreeClassifier

class sklearn.tree.**DecisionTreeClassifier**(*, *criterion='gini'*, *splitter='best'*, *max_depth=None*, *min_samples_split=2*, *min_samples_leaf=1*, *min_weight_fraction_leaf=0.0*, *max_features=None*, *random_state=None*, *max_leaf_nodes=None*, *min_impurity_decrease=0.0*, *class_weight=None*, *ccp_alpha=0.0*, *monotonic_cst=None*) [source]

A decision tree classifier.

Read more in the User Guide.

Parameters:

criterion : *{"gini", "entropy", "log_loss"}, default="gini"*
The function to measure the quality of a split. Supported criteria are "gini" for the Gini impurity and "log_loss" and "entropy" both for the Shannon information gain, see Mathematical formulation.

splitter : *{"best", "random"}, default="best"*
The strategy used to choose the split at each node. Supported strategies are "best" to choose the best split and "random" to choose the best random split.

max_depth : *int, default=None*
The maximum depth of the tree. If None, then nodes are expanded until all leaves are pure or until all leaves contain less than min_samples_split samples.

min_samples_split : *int or float, default=2*
The minimum number of samples required to split an internal node:

- If int, then consider `min_samples_split` as the minimum number.
- If float, then `min_samples_split` is a fraction and `ceil(min_samples_split * n_samples)` are the minimum number of samples for each split.

Changed in version 0.18: Added float values for fractions.

min_samples_leaf : *int or float, default=1*
The minimum number of samples required to be at a leaf node. A split point at any depth will only be considered if it leaves at least `min_samples_leaf` training samples in each of the left and right branches. This may have the effect of smoothing the model, especially in regression.

Source: Sci-Kit Learn: DecisionTreeClassifier

- Find information about hyperparameters on Sci-Kit Learn website

Scoring Methods

Positive has nothing to do with emotions!

Outcome where the event of interest is true

- Quantifying the quality of predictions
- Classification:
  - Accuracy - The proportion of all predictions that the model got right
  - Precision - The proportion of positive identifications that were actually correct
  - Recall
  - F1-score (harmonic mean of precision and recall)
    - Helps evaluate the trade-off between precision and recall
- Regression
  - Mean Squared Error (MSE) - Emphasizes larger errors more than smaller ones – outliers are penalized
  - Mean Absolute Error (MAE) – All are emphasized the same
  - R2-score - How much of the variation in target can be explained by the features

Essential trade-offs to be considered

- Complexity Trade off
  - Low
    - Underfitting
    - High Bias
    - Low variance
    - Simple
  - High
    - Overfitting
    - Low bias
    - High variance
    - Complex

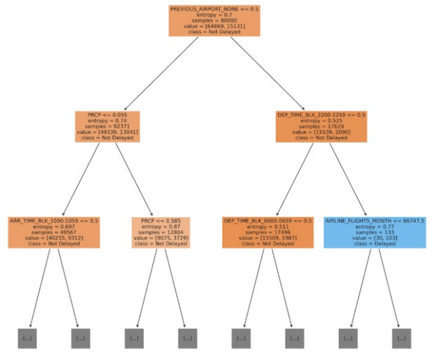
mh

Good estimators to learn more about features

- Linear/logistic regression
  - Coefficients
- Decision Tree/Random Forest
  - Feature importance
  - Tree visualization

Appendix 1. Feature Importance and tree from DecisionTreeClassifier

Features	Feature Importance
AWND	0.069016
TMAX	0.068040
DEST_AIRPORT_ID	0.062723
CONCURRENT_FLIGHTS	0.056959
AIRLINE_AIRPORT_FLIGHTS_MONTH	0.054148
PLANE_AGE	0.051108
AIRPORT_FLIGHTS_MONTH	0.039729
DAY_OF_WEEK	0.035605
PRCP	0.032706
AIRLINE_FLIGHTS_MONTH	0.032555
NUMBER_OF_SEATS	0.028292
DISTANCE_GROUP	0.026164
MONTH	0.024687
AVG_MONTHLY_PASS_AIRPORT	0.019591
PREVIOUS_AIRPORT_NONE	0.010299



Appendix 2. Coefficients from LogisticRegression

Features	Coefficients
CARRIER_NAME_SkyWest Airlines Inc.	0.477400
AIRPORT_FLIGHTS_MONTH	0.383424
CARRIER_NAME_Atlantic Southeast Airlines	0.226056
AIRLINE_FLIGHTS_MONTH	0.209623
PRCP	0.190826
PREVIOUS_AIRPORT_Ogdensburg International	0.115425
AWND	0.107515
DEP_TIME_BLK_1700-1759	0.091647
PREVIOUS_AIRPORT_NONE	0.090518
ARR_TIME_BLK_2300-2359	0.087438
TAIL_NUM_N837AE	0.087048
ARR_TIME_BLK_2100-2159	0.085994
DEPARTING_AIRPORT_Newark Liberty International	0.085138
TAIL_NUM_N8710M	0.085000
CARRIER_NAME_JetBlue Airways	0.083161

Features	Coefficients
CARRIER_NAME_Comair Inc.	-0.491876
CARRIER_NAME_American Eagle Airlines Inc.	-0.403183
DEP_TIME_BLK_0600-0659	-0.260649
FLT_ATTENDANTS_PER_PASS	-0.246566
PREVIOUS_AIRPORT_Ford	-0.203878
TAIL_NUM_N430WN	-0.198267
TAIL_NUM_N522LR	-0.186567
TAIL_NUM_N451WN	-0.179058
PREVIOUS_AIRPORT_Great Falls International	-0.178982
TAIL_NUM_N247SY	-0.178012
PREVIOUS_AIRPORT_Dothan Regional	-0.177071
TAIL_NUM_N8696E	-0.173494
TAIL_NUM_N909DE	-0.170993
TAIL_NUM_N12564	-0.170825
TAIL_NUM_N406YX	-0.169848

Source: Example Exam Paper – Appendix 1 & 2

mh

Results

Estimator	Train Accuracy	Test Accuracy	Precision	Recall	F1	Tuned	Scaled	One-Hot Encoding
RandomForestClassifier - RandomUnderSampler	0.7580	0.6196	0.2816	0.6521	0.3934	False	False	True
DecisionTreeClassifier - RandomUnderSampler	0.7273	0.5557	0.2290	0.5699	0.3267	True	False	True
DummyClassifier - Constant Delayed	0.1891	0.1892	0.1892	1.0000	0.3181	False	False	False
DecisionTreeClassifier	1.0000	0.7310	0.2839	0.2773	0.2806	True	False	True
DecisionTreeClassifier - SMOTE	1.0000	0.7230	0.2708	0.2744	0.2726	True	False	True
KNeighborsClassifier	1.0000	0.7276	0.2730	0.2649	0.2689	True	True	False
DecisionTreeClassifier	1.0000	0.7452	0.2870	0.2337	0.2576	False	False	True
RandomForestClassifier	0.8985	0.7238	0.2436	0.2186	0.2304	True	False	True
MLPClassifier	0.8285	0.7906	0.3650	0.1443	0.2069	False	True	False
RandomForestClassifier - SMOTE	1.0000	0.8043	0.4348	0.1155	0.1825	False	False	True
KNeighborsClassifier	0.8346	0.7872	0.3300	0.1213	0.1774	False	True	False
KNeighborsClassifier	0.8281	0.7802	0.2568	0.0854	0.1281	False	False	False
KNeighborsClassifier	0.8281	0.7802	0.2568	0.0854	0.1281	False	False	False
LogisticRegression	0.8192	0.7965	0.3375	0.0788	0.1277	True	True	True
LogisticRegression	0.8192	0.7965	0.3375	0.0785	0.1274	False	True	True
RandomForestClassifier	1.0000	0.8118	0.5294	0.0476	0.0873	False	False	True
DummyClassifier - Most Frequent	0.8110	0.8110	0.0000	0.0000	0.0000	False	False	False

Source: Example Exam Paper - Model Results

3 The Oral Exam

An oral exam presentation is not full summary

- It includes a brief summary, but focuses on your reflections about the process
- What could have been done differently?
- Bring keywords, avoid long sentences (you could easily get out of focus and you are not as prepared for unnecessary changes in the conversation)
- Who are your examiner and censor?

This was our structure

J: Introduction
J: Quick introduction of paper - dataset, source

M: Visualization (Exploratory Data Analysis) - Aim: explore data, find relationships and patterns
- •Box plot: showcase statistical measures and indicate outliers, which could be treated if any were found
- •Time-series data -> Line graph (does it indicate seasonality? Are there peaks with the year?)
- •J: In our geomap, incorporate size alongside color to indicate the average nr of people in an airport during the day to complement the existing visualization and find a correlation between these two features

Pre-processing - Aim: Make data ready for machine learning
- •J: Encoding - One-Hot vs Label
- •J: Undersampling should have been used from the start, then we could have used accuracy as an evaluation parameter. As it performed better than SMOTE.
- • No reason to generate synthetic samples when you have so much data.
- •A: Apply machine learning to bigger samples - especially MLPClassifier (as neural networks work better on larger amounts of data) - maybe tuning on a small sample and then training on a bigger sample
- •M: Scaling of data - prevent data leaks by pipelining
- •A: Validation method - Cross-validation

Machine learning - Aim: Try to create a well-performing model (Primary: F1-score)
- A: Scoring method - F1-method vs precision vs recall vs accuracy
- M: Other estimators

Future Work
1.Could other features be collected including - pricing level of carriers, baggage amount on flight, overbooking, connecting passengers
   1. Could not go into detail about passengers due to ethical issues
2.Extended time period (and maybe more locations)
3.Collecting data real time and training the model as we go to get a bigger understanding of current situations that could implicate delays.

Good things to bring

- A sheet that quickly shows what estimators were used and what parameters were tuned with which values
- Table that gives an overview of the different estimators, settings, and results
- An overview of your columns – also the ones that you decided to remove
- Keywords to your presentation
- Consider bringing at least 1 printed version of your paper (they could ask questions about specific pages)

The conversation

- Its very chill
- They can ask question about the entire curriculum
- You can direct the talk based on your presentation to some extent (make them interested in your reflections – then they will probably want to know more)

4 Questions

mh

5 Individual projects

What can you do when waiting?

- Feel free to try out the code examples.
- Check out UCloud
- Talk with your group about your project and how you can incorporate the reflections made in this session