Bootcamp Curriculum

Table of Contents

1 Curriculum Overview

SLU	Name	Core Competency	Status	Content
	Notation & Definitions		New	
SLU01	Pandas 101	Data Skills & Analysis	To Adapt	Content
SLU02	Subsetting Data in Pandas	Data Skills & Analysis	To Adapt	Content
SLU03	Visualization with Pandas & Matplotlib	Data Skills & Analysis	Existing	Content
SLU04	Basic Stats with Pandas	Data Skills & Analysis	To Adapt	Content
SLU05	Covariance & Correlation	Data Skills & Analysis	Existing	Content
SLU06	Dealing with Data Problems	Data Skills & Analysis	Existing	Content
SLU07	Regression with Linear Regression	Learning Algorithms	Existing	Content
SLU08	Classification with Logistic Regression	Learning Algorithms	Existing	Content
SLU09	Model Selection & Overfitting	ML Fundamentals	Existing	Content
SLU10	Metrics for Regression	ML Fundamentals	Existing	Content
SLU11	Metrics for Classification	ML Fundamentals	Existing	Content
SLU12	Support Vector Machines (SVM)	Learning Algorithms	New	Content
SLU13	Tree-Based Models	Learning Algorithms	New	Content
SLU14	k-Nearest Neighbors (kNN)	Learning Algorithms	New	Content
SLU15	Feature Engineering (aka Real Wold Data)	Data Skills & Analysis	Existing	Content
SLU16	Data Sufficiency & Selection	Data Skills & Analysis	Existing	Content
SLU17	Ethics & Fairness	Data Skills & Analysis	New	Content
SLU18	Hyperparameter Tuning	ML Fundamentals	Existing	Content
SLU19	Workflow		Existing	Content

2 Competencies

2.1 Data Skills & Analysis

Introduction to the Pandas library and its applications for:

- Data preparation
- Analysis
- Visualization.

2.2 Learning Algorithms

A practical overview of fundamental supervised learning techniques:

- Linear Regression
- Logistic Regression
- SVM
- Tree-Based Models
- k-Nearest Neighbors.

2.3 ML Fundamentals

Key concepts for the basic practice of Data Science and Machine Learning:

- Over- and under-fitting

- Evaluation Metrics
- Hyperparameter tuning.

3 Contents

SLU01

Pandas 101

Main topics

1. Object Creation
2. Basic Functionality
3. Pandas' IO Tools.

Detailed curriculum

1. Importing Pandas (i.e., `import pandas as pd`)
2. Object Creation
    1. `pd.Series`
    2. `pd.DataFrame`
3. Basic Functionality
    1. `.head()`, `.tail()`
    2. Attributes
        1. `.shape`
        2. Axis Labels
            1. Series: `.index`
            2. DataFrame: `.index`, `.columns`
        3. Underlying Data
            1. `.values`
            2. `.array`, `to_numpy`
        4. Data Types
            1. Series: `dtype`
            2. DataFrame: `dtypes`
    3. Summarizing Data
        1. `describe`
        2. `info`
4. Pandas' IO Tools
    1. Read a CSV file into a DataFrame
        1. `pd.read_csv`
    2. Store the contents of a DataFrame as a CSV file
        1. `pd.to_csv`
    3. Reference to other similar IO tools to read and write data (e.g., JSON, Excel).

SLU02

Subsetting Data in Pandas

Main topics

1. Basic Indexing
2. Adding Rows & Columns
3. Removing Rows & Columns

Detailed curriculum

1. Basic Indexing
    1. Set the DataFrame index
        1. Using `index=` on object creation with `pd.Series` or `pd.DataFrame`
        2. Using `index_col=` when reading with `pd.read_csv`
        3. Using existing columns
            1. `reset_index()`
            2. `set_index()`, `sort_index()`
    2. Indexers: `[]`, `.loc[]`, and `.iloc[]`

        1. Selecting Rows with .loc[] and .iloc[]
            1. Selection by Label, i.e., .loc[indexer]
                1. Single row by passing a single Label
                2. Multiple rows
                    1. By passing a list or array of labels
                    2. A slice object, i.e., slice notation
                    3. A boolean array or Series
            2. Selection by Position, i.e., .iloc[position]
                1. Single row by passing an integer
                2. Multiple rows
                    1. By passing a list or array of integers
                    2. A slice object with integers
                    3. A boolean array
        2. Selecting Columns with []
            1. Using the indexing operator [], i.e., square brackets notation df[col]
                1. Single column
                2. Multiple columns
                    1. By passing a list or array of columns, in any order
            2. Attribute access (not indexing), i.e., dot notation df.col
        3. Multi-Axis Indexing with .loc[] and .iloc[]
            1. Multi-Axis Selection by Label
                1. .loc[row_indexer, col_indexer]
            2. Multi-Axis Selection by Position
                1. .iloc[row_position, col_position]
2. Adding Rows & Columns
    1. Using the indexing operator df[new_col]=
    2. Assigning new columns to a DataFrame with .assign()
3. Removing Rows & Columns
    1. Using drop() to remove specified labels
        1. From Rows
            1. .drop(axis=1), .drop(columns=)
        2. From Columns
            1. drop()

SLU03

Visualization with Pandas & Matplotlib

Main topics

1. Basic Plotting in Pandas
2. Different Types of Plots
3. Formatting & Styling

Detailed curriculum

1. Importing Matplotlib
    1. import matplotlib.pyplot as plt
    2. %matplotlib inline
2. Basic Plotting in Pandas
    1. Line plots
        1. Series: .plot()
        2. DataFrame: .plot(), .plot(x=, y=)
3. Different Types of Plots
    1. How to choose the right type
    2. Bar plots
        1. Using .plot(kind='bar'), .plot(kind='barh')
        2. Using .plot.bar(), .plot.barh()
        3. Stacked bar plots with stacked=True
    3. Histograms
        1. Using .plot(kind='hist')
        2. Using .plot.hist(), .plot.hist(xlim=)
    4. Boxplot
        1. Using .plot(kind='box')

        2. Using .plot.box(), .plot.box(vert=False)
    5. Scatterplot
        1. Using .plot(kind='scatter')
        2. Using .plot.scatter()
4. Formatting & Styling
    1. Using the Matplotlib API (some can be used on Pandas directly)
        1. Size: matplotlib.rcParams["figure.figsize"]
        2. Styles: plt.style.use()
        3. Legend: .plot(legend=), plt.legend()
        4. Labels: plt.xlabel(), plt.ylabel()
        5. Title: plt.title()
        6. Text: plt.figtext()
    2. Avoiding chartjunk
        1. Don't use unnecessary and/or confusing elements
        2. Use the minimum set of visuals necessary to be informative.

SLU04

Basic Stats with Pandas

Main topics

1. Descriptive Statistics
2. Inspecting Distributions
3. Outlier Detection

Detailed curriculum

1. Descriptive Statistics in Pandas
    1. .count()
    2. .sum(), .cumsum()
    3. .nunique()
    4. .mean(), .median()
    5. .mode()
    6. .min(), .max()
    7. .idxmin(), .idxmax()
    8. .var(), .std()
    9. .quantile()
    10. .rank()
2. Inspecting the Distribution of the Data
    1. Skew with .skew()
    2. Kurtosis with .kurt()
    3. Probability Density Function (PDF)
        1. What is it and why is it useful
        2. Using .plot(kind='density')
        3. Using .plot.density()
    4. Cumulative Density Function (CDF)
        1. What is it and why is it useful
        2. Using .plot(kind='hist', histtype='step', density=True, bins=100, cumulative=True)
3. Outlier Detection
    1. Visually
    2. Using standard deviations from the mean
    3. Simple techniques to deal with outliers
        1. Delete observations
        2. Log transformation
            1. Visualize the transformation with plot.hist(logx=True)
            2. Log transform a column in Pandas
                1. df[col] = np.log(df[col] or .assign(col=np.log(df[col])
                2. Note for the Teacher: do not use .apply(), because vectorization

---

SLU05

Covariance & Correlation

Main topics

1. Covariance
2. Correlation
3. Causality

Detailed curriculum

1. Covariance
    1. What is it and why is it useful
    2. Using `.cov`
2. Correlation
    1. What is it and why is it useful
    2. Spearman correlation with `.corr()`
    3. Pearson correlation with `.corr(method='spearman')`
    4. Correlation matrix with `.corr()`
3. Causality (or lack thereof)
    1. Observational vs. experimental data
    2. Spurious correlation
    3. Examples of can't we infer that A causes B, despite the correlation
        1. Third factor C causes A and B, i.e., common-causal or confounding variable
        2. B causes A
        3. Bidirectional causation
        4. Coincidental relationships

SLU06

Dealing with Data Problems

Main topics

1. Tidy Data
2. Data Entry Problems
3. Imputation of Missing Values

Detailed curriculum

1. Tidy Data (see this and this for reference)
    1. Each variable forms a column
    2. Each observation forms a row
    3. Each type of observational unit forms a table
2. Data Entry Problems
    1. Unstructured Data
        1. Counts of unique elements
            1. `.value_counts()`
        2. String methods in Pandas with `.str`
            1. `.str.lower()`, `str.upper()`
            2. `.str.strip()`
            3. `.str.replace()`
            4. `.str.split()`
            5. `.str.cat()`
    2. Duplicated Entries
        1. Finding duplicated entries with `.duplicated()`
        2. Remove duplicates with `drop_duplicates()`
3. Imputation of Missing Values
    1. `.value_counts(dropna=False)`
    2. `.isnull()`
    3. Remove missing values with `.dropna()`
    4. Simple imputation techniques
        1. Fill missing values with `.fillna()`
        2. Replacing missing numerical values with the mean
        3. Replacing categorical values with a new category

SLU07

Regression with Linear Regression

Main topics

1. Practical Introduction to Linear Regression
2. Training a Linear Regression
3. Using Linear Regression

Detailed curriculum

1. Practical Introduction to Linear Regression
    1. Regression as the problem of predicting real-valued labels
    2. Linear Regression as a hyperplane, defined as a linear combination of features plus an intercept term
    3. The hyperplane needs to be as close to all examples as possible
2. Training a Linear Regression
    1. Components of all learning algorithms
        1. A loss function
        2. An optimization objective based on the loss function, e.g., a cost function
        3. An optimizer or optimization routine, also known as the solver
    2. Learning the parameters in Linear Regression
        1. Loss function: Square Loss
        2. Objective: Mean Squared Error (MSE)
        3. Solver: Ordinary Least Squares (OLS)
3. Using Linear Regression
    1. Practical strengths
        1. Simplicity
        2. The generalization to unseen examples
    2. Using the Linear Regression from scikit-learn
        1. Importing from scikit (i.e., `from sklearn.linear_model import LinearRegression`)
        2. `sklearn.linear_model.LinearRegression`
            1. `.fit(X, y)`
            2. `.coef_`, `.intercept_`
            3. `.predict()`

SLU08

Classification with Logistic Regression

Main topics

1. Practical Introduction to Classification
2. Classification with Logistic Regression
3. Using Logistic Regression

Detailed curriculum

1. Practical Introduction to Classification
    1. Classification is the problem of assigning one of a finite set of classes
    2. Binary classification
    3. Multiclass classification
2. Classification with Logistic Regression
    1. Applying the standard logistic or sigmoid function
        1. Output can be interpreted as the probability of the positive label
        2. Defining a threshold
    2. Learning the parameters in Logistic Regression
        1. Objective: maximum likelihood, adapted to log-likelihood
        2. Solver: gradient descent
            1. Intro to batch gradient descent
            2. Point to external resources as optional, advanced material
3. Using Logistic Regression
    1. Practical strengths
    2. `sklearn.linear_model.LogisticRegression`

1. .fit(X, y)
2. .predict()
3. .predict_proba()

SLU09

Model Selection & Overfitting

Main topics

1. Generalization Error
2. Model Selection
3. Regularized Linear Regression

Detailed curriculum

1. Generalization Error
    1. Decomposition
        1. Bias
        2. Variance
        3. Irreducible error
    2. Bias-variance trade-off
    3. Sources of complexity
2. Model Selection
    1. Offline evaluation
        1. Leave-one-out or hold-out method
            1. sklearn.model_selection.train_test_split
            2. In-sample or training error
            3. Out-of-sample or testing error
            4. Validation dataset
            5. Evaluating overfitting and underfitting
        2. K-Fold cross-validation
            1. sklearn.model_selection.cross_val_score
        3. Data leakage
    2. Practical considerations
        1. Training time
        2. Prediction time
3. Regularized Linear Models
    1. Intuition and use-cases
    2. Lasso, or L1
        1. sklearn.linear_model.Lasso
    3. Ridge, or L2
        1. sklearn.linear_model.Ridge
    4. Elastic Net
        1. sklearn.linear_model.ElasticNet

SLU10

Metrics for Regression

Main topics

1. Loss Function vs. Evaluation Metric
2. Evaluation Metrics for Regression
3. Using the Metrics

Detailed curriculum

1. Loss Function vs. Evaluation Metric
2. Evaluation Metrics for Regression
    1. Mean Absolute Error (MAE)
        1. sklearn.metrics.mean_absolute_error
    2. Mean Squared Error (MSE)

        1. sklearn.metrics.mean_squared_error
    3. Root Mean Squared Error (RMSE)
        1. From MSE
    4. Coefficient of Determination or R^2
        1. sklearn.metrics.r2_score
        2. sklearn.linear_model.LinearRegression.score
    5. Adjusted R^2
        1. From R^2
3. Using the Metrics
    1. Hold-out method
        1. Training error
        2. Testing error
    2. K-Fold
        1. sklearn.model_selection.cross_val_score(scoring=)

SLU11

Metrics for Classification

Main topics

1. Limitations of Accuracy
2. Precision & Recall
3. AUC-ROC Curve

Detailed curriculum

1. Limitations of Accuracy
    1. Accuracy Score
        1. sklearn.metrics.accuracy_score
    2. Class Imbalance
    3. Implications for the Accuracy Score
2. Precision & Recall
    1. Confusion Matrix
        1. sklearn.metrics.confusion_matrix
    2. Precision
        1. sklearn.metrics.precision_score
    3. Recall
        1. sklearn.metrics.recall_score
    4. F1-Score
        1. sklearn.metrics.f1_score
3. AUC-ROC Curve
    1. ROC Curve
        1. Intuition
            1. False Positive Rate (FPR)
            2. True Positive Rate (TPR)
            3. Thresholds
        2. sklearn.metrics.roc_curve
    2. Area Under the ROC Curve (AUC)
        1. sklearn.metrics.roc_auc_score

SLU12

Support Vector Machines (SVM)

Main topics

1. Understanding the Decision Boundary
2. Implementing SVMs
3. Using SVMs

Detailed curriculum

1. Understanding the Decision Boundary
    1. Margin & Generalization
    2. Linearly Separable Classes
    3. Non-Linear Decision Boundaries
2. Implementing SVMs
    1. Linear Model
        1. Hard-margin SVM
        2. Dealing with Noise
            1. Soft-margin SVM: misclassification and the `C` penalty
    2. Inherent Non-Linearity
        1. Kernel trick
        2. Kernels or kernel functions
3. Using SVMs
    1. Practical strengths
    2. `sklearn.svm.SVC`
    3. `sklearn.svm.SVR`

SLU13

Tree-Based Models

Main topics

1. Decision Trees
2. Ensemble Learning: Bagging & Boosting
3. Using Tree-Based Models

Detailed curriculum

1. Decision Trees
    1. Debunking Decision Trees
    2. Flexibility & Overfitting
2. Ensemble Learning: Bagging & Boosting
    1. Bootstrap Aggregation, i.e., Bagging
        1. Bootstrapping
        2. Random Forests
    2. Boosting
        1. Gradient Boosting
3. Using Tree-Based Models
    1. Random Forests
        1. Practical strengths
        2. `sklearn.ensemble.RandomForestClassifier`
        3. `sklearn.ensemble.RandomForestRegressor`
    2. Grandient Boosting
        1. Practical strengths
        2. `sklearn.ensemble.GradientBoostingClassifier`
        3. `sklearn.ensemble.GradientBoostingRegressor`

SLU14

k-Nearest Neighbors (kNN)

Main topics

1. Key Differentiators of kNN
2. A Primer on Distance
3. Using kNN

Detailed curriculum

1. Key Differentiators of kNN
    1. Non-Parametric
    2. Lazy
    3. Reliance on similarity as defined by closeness, i.e., distance

4. kNN in a nutshell
    1. Doesn't learn or build a model
        1. No hypothesis function
        2. No learned weights
        3. No discarding the training data after training
        4. No training, one could say
    2. Instead, keeps all the training data in memory
    3. Predictions use the k closest examples to return a majority label
        1. Implications of k
2. A Primer on Distance
    1. Distance metrics
        1. Euclidean distance
        2. Dot-product
        3. Cosine distances (the most used, in practice)
    2. Unexpected behaviour in high dimensions, i.e., course of dimensionality
3. Using kNN
    1. Practical strenghts
        1. Despite slow prediction time
    2. `sklearn.neighbors.KNeighborsClassifier`
    3. `sklearn.neighbors.KNeighborsRegressor`

SLU15

Feature Engineering (aka Real World Data)

Main topics

1. Types of Statistical Data
2. Dealing with Categorical Features
3. Dealing with Numerical Features

Detailed curriculum

1. Types of Statistical Data
    1. Numerical
        1. Discrete
        2. Continuous
    2. Categorical
        1. Binary
        2. Ordinal
        3. Nominal
    3. `.select_dtypes()`
2. Dealing with Categorical Features
    1. Introducing sklearn-like transformers
        1. `.fit()`
        2. `.transform()`
    2. Encoding categories
        1. Binary
            1. `pd.map()`
        2. Label or ordinal encoding
            1. Importing category-encoders, i.e., `import category-encoders as ce`
            2. `ce.ordinal.OrdinalEncoder`
        3. One-hot or dummy encoding
            1. `pd.get_dummies(drop_first=True)`
            2. `ce.one_hot.OneHotEncoder`
        4. Target encoding
            1. `ce.target_encoder.TargetEncoder`
3. Dealing with Numerical Features
    1. Discretization
        1. Binning
            1. `pd.cut()`
            2. `sklearn.preprocessing.KBinsDiscretizer`
        2. Feature binarization, according to a threshold
            1. `df[col] > threshold`

        2. sklearn.preprocessing.Binarizer
    2. Scaling
        1. Normalization
            1. sklearn.preprocessing.Normalizer
        2. Standardization or z-score normalization
            1. sklearn.preprocessing.StandardScaler
        3. Robust scaling in the presence of outliers
            1. sklearn.preprocessing.RobustScaler

SLU16

Data Sufficiency & Selection

Main topics

1. Why Feature Selection
2. Techniques for Feature Selection
3. Learning Curves

Detailed curriculum

1. Why Feature Selection
    1. Why having more features is not necessarily better
        1. Lack of interpretability
        2. Unexpected behaviour
        3. Overfitting, i.e., poor performance
        4. Training and prediction times
    2. Heuristic: number of features shouldn't exceed 20% of the number of observations
2. Techniques for Feature Selection
    1. Univariate selection
        1. Intuition, using Pandas
        2. sklearn.feature_selection.SelectKBest
    2. Correlation
        1. Identify the features that correlated the most with the target variable
        2. Correlated features
    3. Feature importances
        1. .feature_importances_ for tree-based models
        2. .coef_ for linear models using previously scaled features
        3. sklearn.feature_selection.SelectFromModel
3. Learning Curves
    1. Debugging bias and variance, i.e., under- and overfitting
    2. Would the model benefit from additional data
    3. sklearn.model_selection.learning_curve
    4. Plotting and interpreting the learning curves

SLU17

Ethics & Fairness

Main topics

1. Informed Consent, Privacy & Security
2. Bias & Discrimination
3. Trustworthiness

Detailed curriculum

1. Informed Consent, Privacy & Security
    1. Informed consent
        1. Right to be forgotten
    2. Privacy and security
        1. Limit the exposure of private information
        2. Data security & retention plan

    3. Reassess and roll-back
    4. Unintended use
2. Bias & Discrimination
    1. Sources of bias
        1. Reporting bias
        2. Automation bias
        3. Selection bias
        4. Group attribution bias
    2. Honest representation
    3. Fairness across groups
        1. Proxy discrimination
        2. Fairness criteria
3. Trustworthiness
    1. Communicate bias and limitations
    2. Concept drift
    3. Explainability
    4. Reproducibility

SLU18

Hyperparameter Tuning

Main topics

1. Understanding Hyperparameters
2. Hyperparameters Cheatsheet
3. Hyperparameter Optimization

Detailed curriculum

1. Understanding Hyperparameters
    1. How are they different from model parameters
    2. Why are they important
2. Hyperparameters Cheatsheet
    1. Linear & Logistic Regression
    2. SVM
    3. Tree-Based Models
        1. Random Forests
        2. Grandient Boosting
    4. k-NN
3. Hyperparameter Optimization
    1. Grid search
        1. `sklearn.model_selection.GridSearchCV`
    2. Random search
        1. `sklearn.model_selection.RandomizedSearchCV`

SLU19

Basic Workflow

Main topics

1. Workflow
2. Workflow Tips
3. Pipelines & Custom Estimators

Detailed curriculum

1. Workflow
    1. Get the data
    2. Data analysis and preparation
        1. Data analysis
        2. Dealing with data problems

        3. Feature engineering
        4. Feature selection
    3. Train model
    4. Evaluate results
    5. Iterate
2. Workflow Tips
    1. Inspect the data
    2. Establish a simple baseline as fast as possible
    3. Increase complexity incrementally
    4. Don't overuse the test set, or you risk overfitting to it
    5. Keep training and test pipelines consistency at all times
        1. Learn parameters on training data, e.g., mean and variance for standardization
            1. e.g., `.fit()`
        2. Apply the exact same transformations, with the same parameters, on train and test data
            1. e.g., `.fit_transform()` for training data
            2. e.g., `.transform()` for test data
3. Pipelines & Custom Estimators
    1. Pipelines
        1. Pipelines ensure train and test pipeline consistency with minimum effort
            1. `sklearn.pipeline.Pipeline`
        2. Compatible with all `sklearn`-like estimators, i.e., transformers, models
        3. Pipeline object can be used with cross-validation and hyperparameter tuning
        4. Trained object can be used to make predictions on unseen data, i.e., `.predict()`
            1. This will run all the data preparation
            2. And use the trained model to make predictions
    2. Custom Estimators
        1. Make all transformations compatible with a Pipeline object, enforcing good standards
            1. `sklearn.base.BaseEstimator`
            2. `sklearn.base.TransformerMixin`

4 Topics Not Covered

1. Pandas
    1. Method chaining
    2. Multi-Index and Advanced Indexing
    3. Merging, joining, and concatenating
    4. Split-apply-combine
    5. `.pop()`
2. ~~Regularization~~
3. ~~Gradient Descent~~
4. ~~Leakage~~
5. ~~Bootstrapping~~
6. Bayesian Optimization for Hyperparameter Tuning
7. Feature Unions