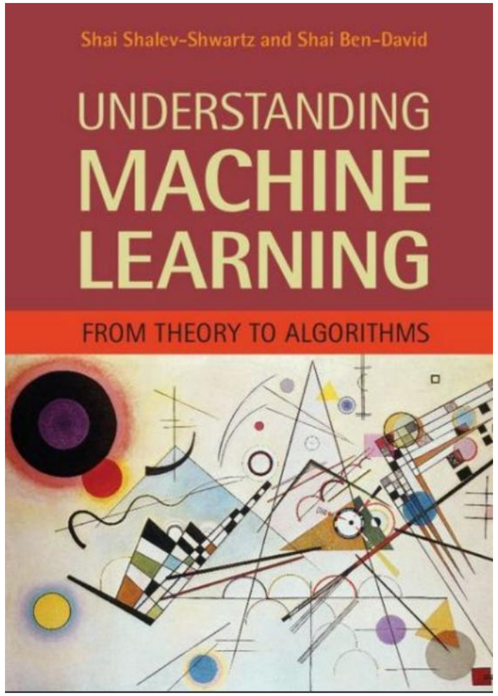
Cloud Computing Linear Regression-AWS sagemaker

Farid Afzali, Ph.D., P.Eng.

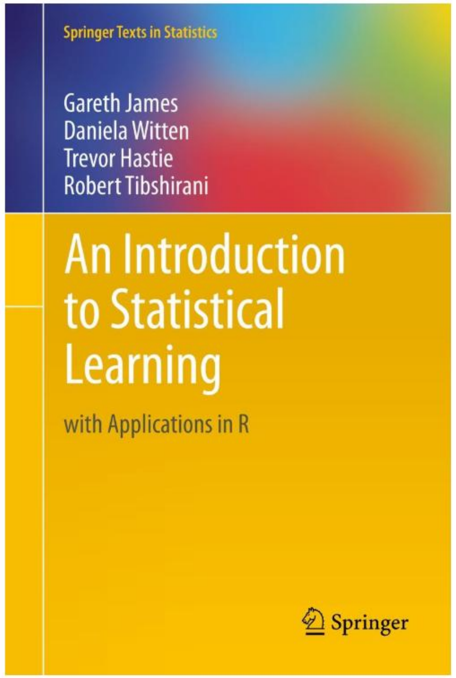
Exploratory Data Analysis

Additional Resources, Page #123:
http://www.cs.huji.ac.il/~shais/UnderstandingMachineLearning/understanding-machine-learning-theory-algorithms.pdf


Shai Shalev-Shwartz and Shai Ben-David
UNDERSTANDING
MACHINE
LEARNING
FROM THEORY TO ALGORITHMS

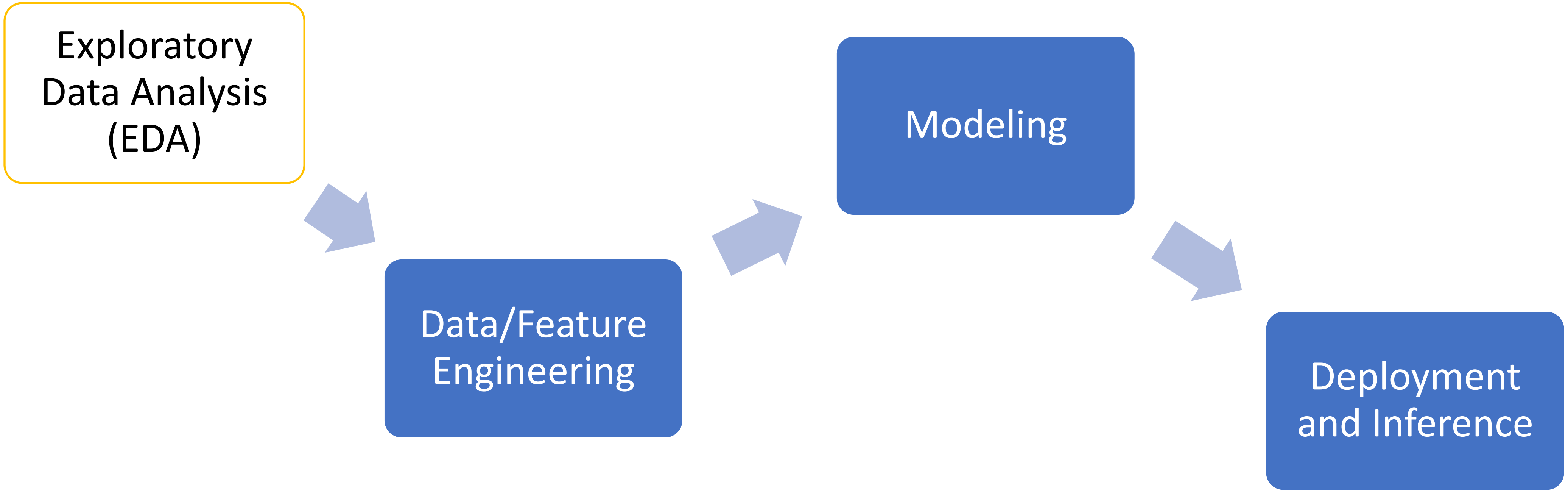
Additional Resources, Page #61:

http://www-bcf.usc.edu/~gareth/ISL/ISLR%20Seventh%20Printing.pdf



Dr. Ryan Ahmed

EDA


Exploratory Data Analysis (EDA)

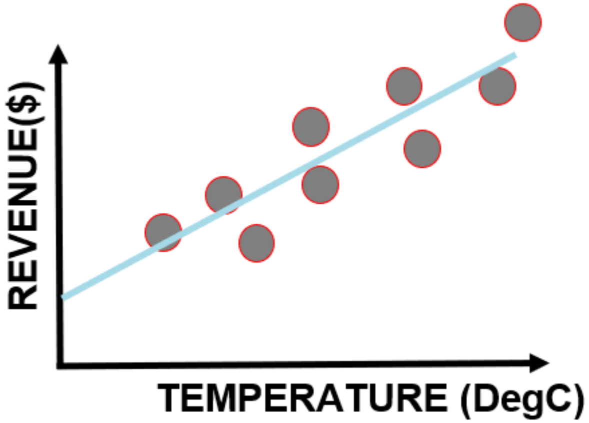
Data/Feature Engineering

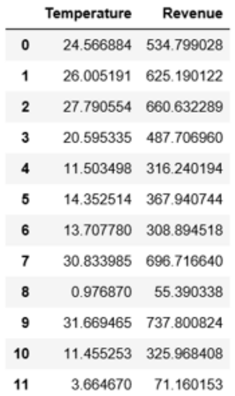
Modeling

Deployment and Inference

Linear Regression

- **Introduction to Variables**:
  - In simple linear regression, the value of a dependent variable, denoted as Y, is predicted based on the value of an independent variable, denoted as X. The independent variable, X, serves as the predictor, while the dependent variable, Y, is the outcome of interest.
- **Understanding 'Simple' in Context**:
  - The term "simple" is used because the analysis involves only two variables. This simplicity allows for a focused examination of the relationship between the independent variable (X) and the dependent variable (Y).
- **Explanation of 'Linear' Relationship**:
  - The adjective "linear" describes the nature of the relationship between the variables. As the independent variable, X, experiences changes (either increases or decreases), the dependent variable, Y, responds in a linear manner. This linear response indicates a consistent, proportional change in Y relative to changes in X.





	Temperature	Revenue
0	24.566884	534.799028
1	26.005191	625.190122
2	27.790554	660.632289
3	20.595335	487.706960
4	11.503498	316.240194
5	14.352514	367.940744
6	13.707780	308.894518
7	30.833985	696.716640
8	0.976870	55.390338
9	31.669465	737.800824
10	11.455253	325.968408
11	3.664670	71.160153

Available Algorithms within Amazon SageMaker

Classification: This category includes algorithms designed for sorting data into predefined categories. Algorithms listed are:

- Linear Learner: A type of algorithm for binary and multiclass classification problems.
- XG-Boost: An optimized gradient boosting library that is effective for various classification challenges.
- K-Nearest Neighbors (KNN): A non-parametric method used for classification (and regression) tasks.
- Factorization Machines: A general-purpose supervised learning algorithm that can be used for both classification and regression tasks.

Regression: This section contains algorithms used for predicting continuous outcomes. Included are:

- Linear Learner: It can also be applied to regression tasks.
- XG-Boost: Capable of solving regression problems.
- K-Nearest Neighbors (KNN): Also useful for regression.

Computer Vision: Algorithms here are focused on image processing tasks such as:

- Image Classification: Assigning a label to an image from a predefined set of categories.
- Object Detection: Identifying objects within an image and determining their boundaries.
- Semantic Segmentation: Assigning a label to every pixel in an image, thus differentiating between objects.

Available Algorithms within Amazon SageMaker

Time Series Forecasting:

- DeepAR: An algorithm for forecasting time series using autoregressive recurrent networks.

Recommendation:

- Factorization Machines: Useful in recommendation systems for capturing interactions between variables within high dimensional sparse datasets.

Text and Topic Modeling: Algorithms for processing and understanding text data. These include:

- Blazing Text: For text classification and word embeddings.
- Neural Topic Modelling (NTM): A neural network-based approach for discovering topics within documents.
- Latent Dirichlet Allocation (LDA): A classic topic modeling technique.
- Sequence2Sequence: Used in tasks like machine translation, where an input sequence (like a sentence) is transformed into a new output sequence.

Dimensionality Reduction: Techniques to reduce the number of random variables to consider.

- Principal Component Analysis (PCA): A statistical procedure that uses orthogonal transformation to convert a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables.
- Object2Vec: A neural embedding algorithm trained to understand the similarity between objects.

Anomaly Detection: Identifying unusual patterns that do not conform to expected behavior.

- Random Cut Forest: An unsupervised algorithm for detecting anomalies.
- IP Insights: An algorithm designed to learn the usage patterns of IP addresses, which can be useful for detecting anomalous behavior.

Clustering:

- K-Means: A popular algorithm that organizes data into k number of clusters.
- KNN: Mentioned here again, although it is more commonly associated with classification, it can be adapted for clustering.

SageMaker Linear Learner

- **Preprocessing**:
- The Linear Learner algorithm includes a feature scaling or normalization option, which is beneficial for ensuring that no single feature disproportionately influences the model due to scale variance.
- **Training**:
- The algorithm employs stochastic gradient descent for optimizing the learning process. Users have the flexibility to choose from various optimization algorithms like Adam, AdaGrad, or SGD (Stochastic Gradient Descent).
- Hyperparameters, which determine the learning structure and behavior of the model, can be adjusted. An example is the learning rate.
- Regularization techniques such as L1 and L2 regularization are used to prevent overfitting, ensuring the model doesn't overly adapt to the training data at the cost of generalization to new data.
- **Validation**:
- Models are tested against a validation dataset to assess their performance, using different metrics depending on the type of task:
  - For regression tasks, metrics such as mean squared error (MSE), root mean squared error (RMSE), and mean absolute error (MAE) are used.
  - For classification tasks, metrics such as the F1 score, precision, recall, and accuracy are considered to evaluate the model's performance.

SageMaker Linear Learner Hyperparameters

- **Predictor type**: This specifies the type of task the Linear Learner is performing. 'Regressor' indicates it is used for regression tasks, predicting continuous outcomes.
- **Learning Rate**: This is a hyperparameter that determines the step size during gradient descent optimization. It controls how much the model weights should be updated during training.
- **L1**: The L1 regularization parameter introduces a penalty equal to the absolute value of the magnitude of coefficients. Regularization techniques like L1 help prevent the model from overfitting by reducing the complexity of the model.
- **Mini batch size**: This defines the number of observations used in each mini-batch of optimization. Mini-batch gradient descent is a variation of the gradient descent algorithm that splits the training dataset into small batches that are used to calculate model error and update model coefficients.
- **Wd**: This stands for weight decay, which is another term for the L2 regularization parameter. L2 regularization adds a penalty term to the loss function equal to the squared value of the magnitude of coefficients, which helps in preventing overfitting by keeping the weights small.

SageMaker Linear Learner Input/output

- **Supported Input Data Types**:
  - **RecordIO-wrapped protobuf**: This format refers to a binary protocol designed by Google that serializes structured data. RecordIO is a data storage format that wraps protobuf messages for efficient storage and I/O operations. The note specifies that only 32-bit floating-point tensors are supported.
  - **Text/CSV**: The algorithm can accept text files or CSV (Comma Separated Values) files. It is noted that the first column in the provided data is assumed to be the target variable, which is the dependent variable in the context of supervised learning.
  - **File or Pipe mode**: These modes refer to how data is provided to the training job. File mode means that the complete dataset is available before the training job starts, while Pipe mode streams data directly into the algorithm while it is running.
- **Supported Output Formats for Inference**:
  - The Linear Learner algorithm can produce predictions (inferences) in various data formats such as **application/json**, **application/x-recordio-protobuf**, and **text/csv**. This allows the model to be integrated with different types of applications and services that might require different data formats.
- **Prediction Scoring for Regression**:
  - When the Linear Learner is set to **predictor_type='regressor'**, indicating a regression task, the output score from the model represents the prediction made by the model. In regression tasks, this score typically corresponds to a continuous value that the model predicts based on the input features.

SageMaker Linear Learner Input/output

- **Supported Input Data Types**:
  - **RecordIO-wrapped protobuf**: This format refers to a binary protocol designed by Google that serializes structured data. RecordIO is a data storage format that wraps protobuf messages for efficient storage and I/O operations. The note specifies that only 32-bit floating-point tensors are supported.
  - **Text/CSV**: The algorithm can accept text files or CSV (Comma Separated Values) files. It is noted that the first column in the provided data is assumed to be the target variable, which is the dependent variable in the context of supervised learning.
  - **File or Pipe mode**: These modes refer to how data is provided to the training job. File mode means that the complete dataset is available before the training job starts, while Pipe mode streams data directly into the algorithm while it is running.
- **Supported Output Formats for Inference**:
  - The Linear Learner algorithm can produce predictions (inferences) in various data formats such as **application/json**, **application/x-recordio-protobuf**, and **text/csv**. This allows the model to be integrated with different types of applications and services that might require different data formats.
- **Prediction Scoring for Regression**:
  - When the Linear Learner is set to **predictor_type='regressor'**, indicating a regression task, the output score from the model represents the prediction made by the model. In regression tasks, this score typically corresponds to a continuous value that the model predicts based on the input features.

SageMaker Linear Learner EC2 instance

- **EC2 Instance Types for Training**:
  - The Linear Learner algorithm can be trained on various types of EC2 instances. These include instances that are equipped with a single Central Processing Unit (CPU) or Graphics Processing Unit (GPU), which are suitable for different scales and complexities of tasks.
  - The algorithm can also be trained on EC2 instances that span multiple machines, implying the use of a cluster of CPUs and/or GPUs. This setup is beneficial for distributed training on larger datasets.
- **Considerations During Testing**:
  - The slide notes that during the testing phase, utilizing multi-GPU computers may not be necessary. This is likely due to the fact that the benefits of parallel processing power that GPUs provide may not translate into significant improvements during the testing phase, which is generally less computationally intensive than the training phase.
  - The use of multi-GPU instances without experiencing corresponding performance improvements would add to costs without providing value, which is an important consideration for optimizing cloud computing expenses.