**Briefing Document: Machine Learning and Hellbender Salamanders**

**I. Introduction to Machine Learning**

**A. Core Concepts**

* **Definition:** Machine learning (ML) is a field that focuses on building systems that can learn from data, rather than relying on explicitly programmed rules. The goal is to create systems that can make predictions or decisions based on patterns found in data.
* **Applications:** ML is widely used in commercial applications and research, including medical diagnosis, social networking, and climate change predictions. The possibilities are "mostly limited by your imagination." (Preface)
* **Accessibility**: This book aims to demonstrate that machine learning is not exclusive to large companies, and can be learned and applied by individuals.

**B. Types of Machine Learning**

* **Supervised Learning:** This type of ML involves learning from labeled data, where the algorithm is given both the input features and the desired output. It can be further divided into:
* **Classification:** Predicting a category or class label (e.g., whether a customer will buy a boat).
* **Regression:** Predicting a continuous numerical value (e.g., predicting house prices or temperature).
* "For regression, the general prediction formula for a linear model looks as follows: ŷ = w[0] \* x[0] + w[1] \* x[1] + ... + w[p] \* x[p] + b" where x are the features, w are the weights and b is the bias.
* **Unsupervised Learning:** This type of ML involves learning from unlabeled data, where the algorithm tries to find structure or patterns in the data itself.
* **Clustering:** Grouping similar data points together.

**C. Key ML Challenges** \* **Generalization:** The goal of ML is not to memorize the training data, but to create models that perform well on new, unseen data. \* **Overfitting:** When a model is too complex and learns the noise in the training data, it performs poorly on new data. \* **Underfitting:** When a model is too simple and does not capture the underlying patterns in the data, it performs poorly on both training and new data. \* A naive rule that performs well on training data does not guarantee good generalization; "we are not interested in making predictions for this dataset; we already know the answers for these customers."

**D. Essential Libraries and Tools**

* **Python:** The primary programming language used for machine learning.
* **NumPy:** A library for numerical computation in Python.
* **SciPy:** A library for scientific computing that builds on NumPy.
* **pandas:** A library for data manipulation and analysis.
* It is "built around a data structure called the DataFrame that is modeled after the R DataFrame," which is similar to a table.
* **matplotlib:** A library for creating visualizations.
* **scikit-learn:** A widely used machine learning library in Python.

**II. Supervised Learning Algorithms**

**A. K-Nearest Neighbors (KNN)** \* **Classification:** Classifies a data point based on the majority class of its nearest neighbors. \* **Regression:** Predicts a continuous value based on the average of its nearest neighbors. \* The number of neighbors, "n\_neighbors", is a hyperparameter that controls model complexity; "Figure 2-7. Comparison of training and test accuracy as a function of n\_neighbors". \* "The k-nearest neighbors algorithm for regression is implemented in the KNeighbors Regressor class in scikit-learn."

**B. Linear Models**

* **Linear Regression:** Predicts a continuous value based on a linear combination of features.
* Equation for one feature: "ŷ = w[0] \* x[0] + b" where w[0] is the slope, and b is the y intercept.
* **Logistic Regression:** Predicts a class label by modeling the probability of belonging to a class based on a linear combination of features.
* **Support Vector Machines (SVMs):Linear SVM:** Find the linear decision boundary that maximizes the margin between classes.
* **Kernelized SVM:** Use kernel functions to perform classification in a higher-dimensional feature space, allowing for nonlinear decision boundaries.
* The Gaussian kernel measures distance between points as "krbf(x1, x2) = exp (ɣǁx1 - x2ǁ 2)" and gamma controls the radius of the gaussian kernel.
* "To make a prediction for a new point, the distance to each of the support vectors is measured."

**C. Decision Trees**

* Built by recursively splitting the data into subsets, based on feature values that best separate target classes.
* "To build a tree, the algorithm searches over all possible tests and finds the one that is most informative about the target variable".
* Feature importance is a score between 0 and 1, where 1 means that the feature "perfectly predicts the target."
* Visualizations include decision trees and corresponding decision boundaries ("Figure 2-24. Decision boundary of tree with depth 1 (left) and corresponding tree (right)")

**D. Random Forests**

* An ensemble of decision trees, each trained on a random subset of the training data and a random subset of features.
* Trees are built from bootstrap samples, e.g. "['b', 'd', 'd', 'c']" from "['a', 'b', 'c', 'd']".
* Feature selection is repeated separately in each node of each tree
* Reduces the risk of overfitting and provides more robust predictions.
* "The trees that are built as part of the random forest are stored in the estimator\_ attribute."

**E. Neural Networks** \* Uses interconnected layers of nodes (neurons) to learn complex patterns. \* Uses activation functions such as "tanh" and "relu". \* Example of how neural networks compute regression ŷ with "tanh": \* "h[0] = tanh(w[0, 0] \* x[0] + w[1, 0] \* x[1] + w[2, 0] \* x[2] + w[3, 0] \* x[3])" \* "h[1] = tanh(w[0, 0] \* x[0] + w[1, 0] \* x[1] + w[2, 0] \* x[2] + w[3, 0] \* x[3])" \* "h[2] = tanh(w[0, 0] \* x[0] + w[1, 0] \* x[1] + w[2, 0] \* x[2] + w[3, 0] \* x[3])" \* "ŷ = v[0] \* h[0] + v[1] \* h[1] + v[2] \* h[2]"

**F. Gradient Boosting** \* Combines multiple weak learners (usually decision trees) to create a strong model.

**III. Model Evaluation and Improvement**

**A. Cross-Validation** \* Technique used to estimate how well a model will generalize to unseen data by partitioning a dataset into multiple training and testing sets. \* **k-fold cross-validation:** The dataset is split into k folds, each used as a test set once, and the remaining folds as training sets. \* "Splitting the dataset into k folds by starting with the first one-k-th part of the data...might not always be a good idea." \* **Stratified k-fold cross-validation:** Ensures that the class proportions are preserved in each fold (useful for imbalanced datasets).

**B. Grid Search** \* A technique to tune model parameters by systematically searching through a grid of possible values and evaluating model performance using cross-validation.

**C. Evaluation Metrics** \* **Accuracy:** The proportion of correctly classified instances. \* **Precision:** The proportion of true positives out of all predicted positives. \* **Recall:** The proportion of true positives out of all actual positives. \* **F1-score:** The harmonic mean of precision and recall. \* **Area Under the ROC Curve (AUC):** Measure the performance of a classifier across all possible thresholds. \* **Confusion Matrix**: A table showing the true and predicted values of a model

**IV. Data Representation and Feature Engineering**

**A. Categorical Variables** \* **One-Hot Encoding:** Converting categorical variables into binary features, one for each possible value. "One-Hot-Encoding (Dummy Variables)". \* e.g. "get\_dummies" creates a column for each class of "workclass" that could be 'workclass\_ ?', 'workclass\_ Federal-gov', 'workclass\_ Local-gov' etc... \* **Integer Encoding:** Representing categorical values using integers.

**B. Binning and Discretization**

* Converting continuous numerical features into discrete bins to improve performance for specific models.
* "We imagine a partition of the input range for the feature (in this case, the numbers from –3 to 3) into a fixed number of bins—say, 10."
* Different kinds of models may work best with different encodings, and "The best way to represent data depends not only on the semantics of the data, but also on the kind of model you are using."
* Linear models can benefit from interactions with binning. For example, creating product features that indicate both the bin and the original feature value.

**C. Interactions and Polynomials** \* Adding interactions of features to the dataset allows for modelling complex relationships. \* Polynomial features allow for the modeling of nonlinear relationships.

**D. Univariate Nonlinear Transformations** \* Applying nonlinear functions (e.g., log, exponential, sine, cosine) to features to alter their distributions, and the subsequent behavior of the model.

**E. Automatic Feature Selection** \* Methods for selecting the most informative features, which can include: \* Univariate statistics \* Model based feature selection (using feature importance) \* Iterative feature selection

**F. Scaling and Normalization** \* Scaling and Normalization are used to put the features on similar scales \* **MinMaxScaler:** Scales features to be within a specified range (usually 0 to 1). \* **StandardScaler:** Scales features to have zero mean and unit variance. \* **Normalizer** : Scales each data point such that the feature vector has a Euclidean length of 1.

**V. Unsupervised Learning**

**A. Clustering**

* **k-means clustering:** Partitions data into k clusters, where each data point belongs to the cluster with the closest mean.
* "Each new point is assigned to the closest cluster center when predicting, but the existing model is not changed."
* **DBSCAN:** A density-based clustering algorithm, identifying clusters by the density of points in space.
* It identifies core points that have a minimum number of points near them, then it finds other points to include in a cluster with those core points, labeling remaining points as noise.

**B. Dimensionality Reduction** \* Techniques used to reduce the number of features in a dataset while preserving important information. \* **Manifold learning**: Used for visualization or exploratory data analysis, but rarely for training purposes as it does not allow transformations of new data. \* **t-SNE**: Creates a two dimensional representation of data in a way that attempts to preserve distances between points, "trying to preserve the information indicating which points are neighbors to each other."

**VI. Text Data** **A. Bag-of-Words**

* A way of representing text data by counting the occurrences of each word.
* Tokenization is the process of separating words. e.g. "The fool doth think he is wise" is tokenized into 13 words, from "be" to "wise".
* "The vocabulary consists of 13 words, from 'be' to 'wise'."
* Word order is ignored in bag-of-words representations
* "The CountVectorizer extracts tokens using a regular expression." By default it is "\b\w\w+\b". **B. N-grams**
* Captures word context in bag-of-words by counting sequences of words e.g. "ngram\_range=(2,2)" to represent bigrams such as "but the". **C. Stop Words**
* Common words (e.g., "the," "a," "is") that often do not carry much meaning.
* They can be removed by using stop\_words="english" **D. Term Frequency–Inverse Document Frequency (tf–idf)**
* Assigns a weight to each word based on its frequency within a document and its overall importance in the entire collection of documents.
* Words that appear frequently are given less weight.
* "The inverse document frequency values found on the training set are stored in the idf\_ attribute" **E. Lemmatization and Stemming**
* Techniques used to normalize words, by reducing words to a common base form.
* **Stemming:** Rules based approaches for removing suffixes (e.g. "worse" becomes "wors").
* **Lemmatization:** Dictionary based approaches using sentence context to identify root words (e.g. "worse" becomes "bad")

**VII. The Hellbender Salamander**

**A. Physical Characteristics** \* Third-largest aquatic salamander in the world, growing up to 29 inches long. \* Flat bodies with thick skin folds along the sides. \* Breathes primarily through skin folds, rather than gills. \* Has a single gill on each side of the neck.

**B. Habitat and Diet** \* Prefers clear, fast-moving, shallow streams with high oxygen content. \* Shelters under rocks. \* Hunts by sense of smell and detects vibrations. \* Diet includes crayfish, small fish, and sometimes its own eggs.

**C. Ecological Role** \* Acts as both predator and prey within its ecosystem. \* Preyed upon by various fish, snakes, and turtles.

**VIII. Conclusion**

This briefing document provided a summary of key machine learning concepts and how various models and preprocessing techniques work. It also provides an overview of the Hellbender salamander. This understanding of both machine learning and real world ecological examples will be valuable in further exploration and application.