המכללה האקדמית כנרת
הפקולטה להנדסה

קורס: מעבדה בלמידת מכונה
סמסטר ב' תשפ"ד – עבודה סופית
הגשה עד 08/08/2024

תרגיל 1 – עצי החלטה

בתור מנהל שיווק ,אתה רוצה קבוצה של לקוחות בעלי הסבירות הגבוהה ביותר לרכוש את המוצר שלך .כך תוכל לחסוך בתקציב השיווק שלך על ידי מציאת קהל היעד שלך .כמנהל הלוואות ,עליך לזהות בקשות הלוואה מסוכנות כדי להשיג שיעור חדלות פירעון נמוך יותר .תהליך זה של סיווג לקוחות לקבוצה של לקוחות פוטנציאליים ולא פוטנציאליים או בקשות הלוואה בטוחות או מסוכנות ידוע כבעיית סיווג.
סיווג הוא תהליך דו-שלבי; שלב למידה ושלב חיזוי .בשלב הלמידה ,המודל מפותח על סמך נתוני אימון נתונים .בשלב החיזוי ,המודל משמש לחיזוי התגובה לנתונים נתונים .עץ החלטה הוא אחד מאלגוריתמי הסיווג הקלים והפופולריים ביותר המשמשים להבנה ופירוש נתונים .ניתן להשתמש בו הן לבעיות סיווג והן לבעיות רגרסיה.

אלגוריתם עץ ההחלטה

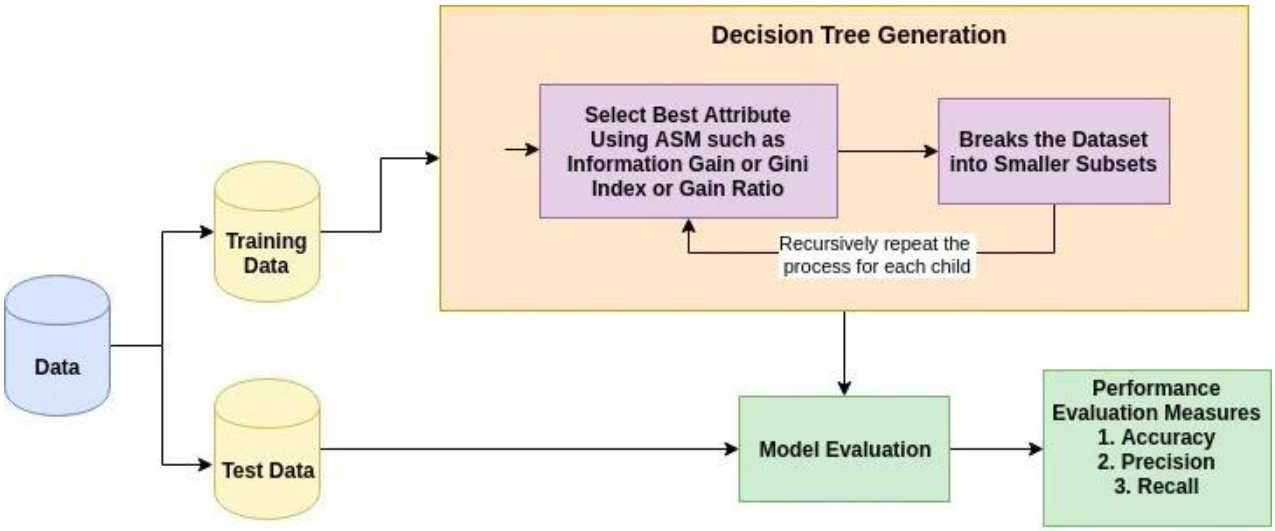
עץ החלטה הוא מבנה דמוי תרשים זרימה ,שבו כל צומת פנימי מייצג תכונה (או מאפיין) ,הענף מייצג כלל החלטה ,וכל צומת עלה מייצג את התוצאה.
הצומת העליון ביותר בעץ החלטה ידוע כצומת השורש .הוא לומד לחלק את הנתונים על בסיס ערך התכונה .הוא מחלק את העץ באופן רקורסיבי ,הנקרא "חלוקה רקורסיבית ."מבנה זה ,דמוי תרשים זרימה ,עוזר לך בקבלת החלטות .הוא מוצג בצורה ויזואלית ,כמו תרשים זרימה ,אשר מחקה בקלות את החשיבה האנושית .זו הסיבה לכך שעצי החלטה קלים להבנה ולפרשנות.

עץ החלטה הוא אלגוריתם מסוג "קופסה לבנה" (white box) בלמידת מכונה .הוא חולק את הלוגיקה הפנימית שלו לקבלת החלטות ,שאינה זמינה באלגוריתמים מסוג "קופסה שחורה"(black box) , כמו רשתות נוירונים .זמן האימון שלו מהיר יותר בהשוואה לאלגוריתם רשת הנוירונים.
סיבוכיות הזמן של עצי החלטה היא פונקציה של מספר הרשומות והתכונות בנתונים הנתונים .עץ ההחלטה הוא שיטה ללא התפלגות או לא פרמטרית ,שאינה תלויה בהנחות התפלגות הסתברות .עצי החלטה יכולים להתמודד עם נתונים בעלי ממדים גבוהים עם דיוק טוב.

כיצד אלגוריתם עץ ההחלטה פועל?

הרעיון הבסיסי מאחורי כל אלגוריתם של עץ החלטה הוא כדלקמן:

1. <u>בחירת התכונה הטובה ביותר</u>: בחר את התכונה הטובה ביותר לביצוע פיצול הרשומות באמצעות מדדי בחירת תכונות.(ASM - Attribute Selection Measures)
2. <u>יצירת צומת החלטה</u>: הפוך את התכונה שנבחרה לצומת החלטה, וחלק את מערך הנתונים לתת-קבוצות קטנות יותר.
3. <u>בניית העץ</u>: התחל בבניית העץ על ידי חזרה על התהליך הזה באופן רקורסיבי עבור כל צומת בן, עד שיתקיים אחד מהתנאים הבאים:
   - כל הרשומות שייכות לאותו ערך תכונה.
   - לא נותרו עוד תכונות.
   - לא נותרו עוד מופעים של נתונים.



בחירת תכונות Attribute Selection Measures

אמצע בחירת תכונה הוא פונקציה היוריסטית המשמשת לבחירת הקריטריון לפיצול הנתונים בצורה הטובה ביותר. הוא מוכר גם בשם "כללי פיצול" מכיוון שהוא עוזר לנו לקבוע את נקודות הפיצול עבור דוגמאות (tuples) בצומת מסוים. אמצעי בחירת התכונה מעניק ציון לכל תכונה (או מאפיין) על ידי כך שהוא מסביר את סט הנתונים הנתון. התכונה בעלת הציון הגבוה ביותר תיבחר כתכונה לפיצול (מקור). במקרה של תכונה בעלת ערכים רציפים, יש צורך להגדיר גם את נקודות הפיצול עבור הענפים. אמצעי הבחירה הפופולריים ביותר הם רווח האינפורמציה Information Gain.

רווח האינפורמציה Information Gain

אנטרופיה entropy, מודד את חוסר הטוהר של קבוצת הקלט. בפיזיקה ובמתמטיקה, אנטרופיה מתייחסת לראנדומליות או לחוסר הטוהר במערכת. בתורת האינפורמציה, היא מתייחסת לחוסר הטוהר בקבוצת דוגמאות. רווח האינפורמציה הוא הירידה באנטרופיה. רווח האינפורמציה מחשב את ההפרש בין האנטרופיה לפני הפיצול לבין האנטרופיה הממוצעת לאחר פיצול קבוצת הנתונים על בסיס ערכי תכונה נתונים. אלגוריתם עץ ההחלטות (Iterative Dichotomiser) ID3 משתמש ברווח האינפורמציה.

$$\text{Info}(D) = -\sum_{i=1}^{m} pi \log_2 pi$$

כאשרק Pi מייצג את ההסתברות שדוגמה (tuple) שרירותית ב-D שייכת למחלקה Ci.

$$\text{Info}_A(D) = \sum_{j=1}^{V} \frac{|Dj|}{|D|} \times \text{Info}(D_j)$$

$$\text{Gain}(A) = \text{Info}(D) - \text{Info}_A(D)$$

כאשר:

Info(D): כמות המידע הממוצעת הדרושה כדי לזהות את תווית המחלקה של דוגמה ב-D.

|Dj|/|D|: פועל כמשקל של החלק ה j- בחלוקה.

InfoA(D): כמות המידע הצפויה הנדרשת כדי לסווג דוגמה מ-D על סמך החלוקה לפי A.

דרישות:

Decision Tree Classifier Building in Scikit-learn

1. **Importing Required Libraries:** Let's first load the required libraries.

2. **Loading Data:** Let's first load the required Pima Indian Diabetes dataset using pandas' read CSV function from attached file "diabetes.csv"

3. **Feature Selection:** Here, you need to divide given columns into two types of variables dependent(or target variable) and independent variable(or feature variables).

4. **Splitting Data:** To understand model performance, dividing the dataset into a training set and a test set is a good strategy.
   Let's split the dataset by using the function `train_test_split()`. You need to pass three parameters features; target, and test_set size

5. **Building Decision Tree Model:** Let's create a decision tree model using Scikit-learn

6. **Evaluating the Model:** Let's estimate how accurately the classifier or model can predict the type of cultivars. Accuracy can be computed by comparing actual test set values and predicted values.

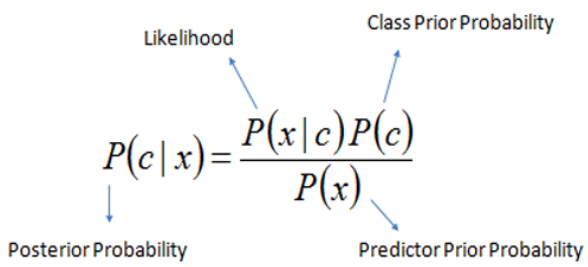
7. **Visualizing Decision Trees:** You can use Scikit-learn's *export_graphviz* function for display the tree within a Jupyter notebook. For plotting the tree, you also need to install graphviz and pydotplus.
   ```
   pip install graphviz
   pip install pydotplus
   ```

תרגיל 2 – Naïve Bayes Classification

אלגוריתם סיווג Naïve Bayes מבוסס על תיאורמת בייס

תיאוריית בייס קובעת כי ההסתברות של מאורע שווה להסתברות המקדמית של המאורע כפול ההסתברות שהמאורע יתקיים בהינתן ראיה מסוימת. בהקשר של סיווג, משמעות הדבר היא שאנו מנסים למצוא את המחלקה בעלת ההסתברות הגבוהה ביותר בהינתן סט של תכונות או מאפיינים.

Naive Bayes מניח שהתכונות בלתי תלויות זו בזו, כלומר נוכחות או היעדרות של תכונה אחת לא משפיעות על הנוכחות או ההיעדרות של תכונה אחרת. הנחה זו מפשטת את חישוב ההסתברות של התכונות, מכיוון שאנחנו יכולים לחשב את ההסתברות של כל תכונה בנפרד ולאחר מכן להכפיל אותן זו בזו.



$$P(c \mid X) = P(x_1 \mid c) \times P(x_2 \mid c) \times \cdots \times P(x_n \mid c) \times P(c)$$

<u>דרישות:</u>

Step 1: Import libraries
We need *Pandas* for data manipulation, *NumPy* for mathematical calculations, *MatplotLib, and Seaborn* for visualizations. *Sklearn* libraries are used for machine learning operations

Step 2: Import data
Download the dataset "Social_Network_Ads.csv" and upload it to your notebook and read it into the pandas dataframe.

Step 3: Data Analysis / Preprocessing
Exploratory Data Analysis (EDA) is a process of analyzing and summarizing the main characteristics of a dataset, with the goal of gaining insight into the underlying structure, relationships, and patterns within the data. EDA helps to identify important features, anomalies, and trends in the data that can inform further analysis and modeling

Step 4: Split data
Splitting data into independent and dependent variables involves separating the *input features* ***(independent variables)*** from the *target variable* ***(dependent variable)***. The independent variables are used to predict the value of the dependent variable.

The data is then split into a training set and a test set, with the training set used to fit the model and the test set used to evaluate its performance.

Step 5: Feature scaling

Feature scaling is a method of transforming the values of numeric variables so that they have a common scale as machine learning algorithms are sensitive to the scale of the input features. There are two common methods of feature scaling: *normalization* and *standardization*.

- **Normalization** scales the values of the variables so that they fall between 0 and 1. This is done by subtracting the minimum value of the feature and dividing it by the range (max-min).
- **Standardization** transforms the values of the variables so that they have a mean of 0 and a standard deviation of 1. This is done by subtracting the mean and dividing it by the standard deviation.

Feature scaling is usually performed before training a model, as it can improve the performance of the model and reduce the time required to train it, and helps to ensure that the algorithm is not biased towards variables with larger values.

Step 6: Train model

Training a machine learning model involves using a training dataset to estimate the parameters of the model. The training process uses a learning algorithm that iteratively updates the model parameters, minimizes a loss function, which measures the difference between the predicted values and the actual values in the training data, and updates the model parameters to improve the accuracy of the model.

Step 7: Predict result / Score model

Once the likelihood of the features for each class is calculated, the algorithm multiplies the likelihood by the prior probability of each class, which is estimated from the training data. The class with the highest probability is then selected as the predicted class.

The accuracy of the model can be evaluated on a test set, which was previously held out from the training process.

Step 8: Evaluate model

Accuracy is a useful metric for assessing the performance of a model, but it can be misleading in some cases. For example, in a highly imbalanced dataset, a model that always predicts the majority class will have high accuracy, even though it may not be performing well. Therefore, it is important to consider other metrics, such as confusion matrix, precision, recall, F1-score, and ROC-AUC, along with accuracy, to get a more complete picture of the performance of a model

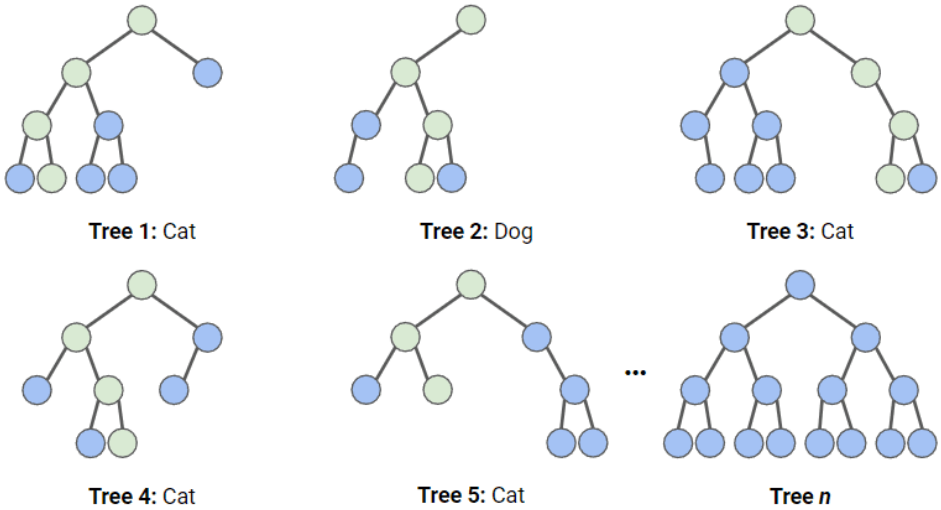
תרגיל 3 – Random Forest Classification

יערות אקראיים: אלגוריתם למידה חישובית "
יערות אקראיים משמשים ב"למידה מלומדת supervised learning ", שבה יש משתנה יעד מסומן (תווית).
יערות אקראיים יכולים לשמש לפתרון בעיות רגרסיה (משתנה יעד מספרי) וסיווג (משתנה יעד קטגורי).
יערות אקראיים הם שיטת אנסמבל, כלומר הם משלבים תחזיות מדגמים אחרים.
כל אחד מהדגמים הקטנים יותר באנסמבל של היער האקראי הוא עץ החלטות.

איך עובדת סיווג באמצעות יער אקראי
דמיינו שיש לכם בעיה מורכבת לפתור, ואתם מגבשים קבוצת מומחים מתחומים שונים כדי לספק את הקלט שלהם. כל מומחה נותן את חוות דעתו בהתבסס על המומחיות והניסיון שלו. לאחר מכן, המומחים יצביעו כדי להגיע להחלטה סופית.

בסיווג באמצעות יער אקראי, נוצרים מספר עצי החלטות באמצעות תת-קבוצות אקראיות שונות של הנתונים והתכונות. כל עץ החלטות דומה למומחה, ומספק את חוות דעתו על אופן סיווג הנתונים. תחזיות מתבצעות על ידי חישוב התחזית עבור כל עץ החלטות, ולאחר מכן לקיחת התוצאה הנפוצה ביותר. (לצורך רגרסיה, תחזיות משתמשות בטכניקת ממוצע במקום זאת).

בתרשים למטה, יש לנו יער אקראי עם n עצי החלטות, והצגנו את ה-5 הראשונים, יחד עם התחזיות שלהם (כלב או חתול). כל עץ נחשף למספר שונה של תכונות ולדוגמה שונה מהקבוצה המקורית, ולכן כל עץ יכול להיות שונה. כל עץ מבצע תחזית. בהסתכלות על 5 העצים הראשונים, אפשר לראות ש-4 מתוך 5 חזו שהדוגמה היא חתול. העיגולים הירוקים מסמנים נתיב היפותטי שהעץ עבר כדי להגיע להחלטה שלו. יער אקראי היה סופר את מספר התחזיות מעצי ההחלטות עבור חתול ועבור כלב, ובוחר את התחזית הנפוצה ביותר.



דרישות:

The Dataset
This dataset consists of direct marketing campaigns by a Portuguese banking institution using phone calls. The campaigns aimed to sell subscriptions to a bank term deposit. We are going to store this dataset in a variable called `bank.csv`.

The columns we will use are:

- `age`: The age of the person who received the phone call
- `default`: Whether the person has credit in default
- `cons.price.idx`: Consumer price index score at the time of the call
- `cons.conf.idx`:Consumer confidence index score at the time of the call
- `y`: Whether the person subscribed (this is what we're trying to predict)

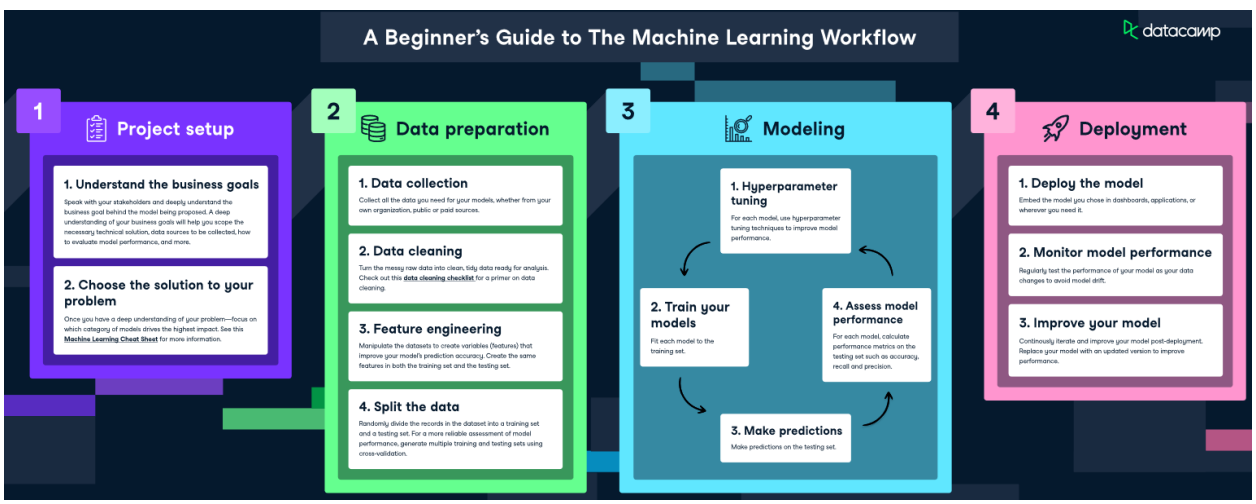
Importing Packages

- The following packages and functions are used in this tutorial:

Random Forests Workflow

To fit and train this model, we'll be following [The Machine Learning Workflow]() infographic; however, as our data is pretty clean, we won't be carrying out every step. We will do the following:

- Feature engineering
- Split the data
- Train the model
- Hyperparameter tuning
- Assess model performance



Preprocessing Data for Random Forests

Tree-based models are much more robust to outliers than linear models, and they do not need variables to be normalized to work. As such, we need to do very little preprocessing on our data.

- We will map our 'default' column, which contains `no` and `yes`, to `0`s and `1`s, respectively. We will treat `unknown` values as `no` for this example.

Splitting the Data

- When training any supervised learning model, it is important to split the data into training and test data. The training data is used to fit the model. The algorithm uses the training data to learn the relationship between the features and the target. The test data is used to evaluate the performance of the model.

Fitting and Evaluating the Model

- We first create an instance of the Random Forest model, with the default parameters. We then fit this to our training data. We pass both the features and the target variable, so the model can learn.

Visualizing the Results

- We can use the following code to visualize our first 3 trees.