**Libraries and Tools Used in the Analysis**

1. **Pandas**:
   * **Role**: Data manipulation and analysis. Used for reading, writing CSV files, managing, and processing datasets.
2. **OpenPyXL**:
   * **Role**: Reading and writing Excel files. Used to interact with .xlsx files and perform operations such as data extraction and updates.
3. **pybiomart**:
   * **Role**: Accessing data from BioMart databases. Used to retrieve additional information about genes such as descriptions, locations, and associated biological data from external biological databases.
4. **scikit-learn (sklearn)**:
   * **Role**: Machine learning library for model training and evaluation.
   * **Components used**:
     + **DecisionTreeClassifier**: Used for training a decision tree model.
     + **RandomForestClassifier**: Used to train a random forest model to assess feature importance.
     + **train\_test\_split**: For splitting the dataset into training and testing subsets.
     + **StandardScaler**: For normalizing and scaling feature data.
     + **metrics (roc\_curve, auc, confusion\_matrix, precision\_recall\_curve, etc.)**: For evaluating model performance, including ROC-AUC scores and precision-recall metrics.
     + **plot\_confusion\_matrix and ConfusionMatrixDisplay**: Used for visualizing the confusion matrix.
     + **model\_selection**: For cross-validation tasks and optimizing the models.
     + **feature\_importances\_**: Used to extract feature importance from the random forest model.
5. **Joblib**:
   * **Role**: Model serialization. Used for saving and loading trained models efficiently.
6. **Seaborn**:
   * **Role**: Data visualization. Utilized to create visualizations such as bar plots for feature importance and heatmaps for feature correlation.
7. **Matplotlib**:
   * **Role**: Data visualization library used to generate visual plots like ROC curves, precision-recall curves, and confusion matrices.
8. **Graphviz and pydotplus**:
   * **Role**: Visualization of decision tree structures.
   * **pydotplus**: Used for exporting decision tree visualizations to PNG format.
   * **Graphviz**: Generates decision tree diagrams using the export\_graphviz() function.
9. **Plotly**:
   * **Role**: Interactive graph plotting and visualizations.
   * **Component used**: Plotly's graph\_objects (go) library was used for interactive network graph visualizations representing decision trees attractively.
10. **Cytoscape (Exported data for visualization)**:

* **Role**: Network visualization tool for custom network analysis. We exported nodes and edges into a CSV format that can be visualized with Cytoscape to customize the appearance.

1. **NumPy**:

* **Role**: Mathematical operations, although indirectly used by Pandas and Scikit-learn for efficient data handling and computations.

1. **IPython (for visualization purposes)**:

* **Role**: For inline visualization in Jupyter environments. Image and display features were used during the decision tree visualization.

1. **CSV Module (Python standard library)**:

* **Role**: Reading and writing CSV files manually, mainly when Pandas wasn’t the most efficient choice.

**Roles Overview**

* **Data Extraction**:
  + **pybiomart**: To retrieve biological metadata from BioMart.
  + **OpenPyXL**: For reading and interacting with Excel files.
  + **Pandas**: Primary library for extracting data from CSV files and data processing.
* **Data Manipulation and Cleaning**:
  + **Pandas**: Data wrangling, handling missing values, feature engineering, and data formatting.
  + **NumPy**: To assist Pandas with array manipulations.
* **Data Normalization and Scaling**:
  + **StandardScaler (scikit-learn)**: To standardize features by removing the mean and scaling to unit variance.
* **Model Training**:
  + **DecisionTreeClassifier (scikit-learn)**: To build the decision tree for the model.
  + **RandomForestClassifier (scikit-learn)**: To train a more complex model and determine feature importance.
* **Model Evaluation**:
  + **Scikit-learn metrics (roc\_curve, auc, precision\_recall\_curve, confusion\_matrix)**: Used for assessing model accuracy and robustness.
  + **plot\_confusion\_matrix (scikit-learn)**: To generate and plot the confusion matrix.
* **Model Saving and Loading**:
  + **Joblib**: Saving trained models to avoid retraining and for reuse.
* **Feature Importance**:
  + **RandomForestClassifier feature\_importances\_**: To determine which features were most influential in the predictions.
* **Data Visualization**:
  + **Matplotlib**: For creating standard plots such as ROC curves, confusion matrices, and precision-recall plots.
  + **Seaborn**: For creating statistical plots such as feature importance bar graphs and heatmaps for correlations.
  + **Graphviz and pydotplus**: For creating decision tree diagrams.
  + **Plotly**: For an interactive decision tree visualization.
  + **Cytoscape**: Custom network visualization from exported data.
* **Interactive Visualization and Display**:
  + **IPython**: Used to display images within Jupyter-like environments (in some iterations).