Random Forest Regression

Frank Ganza

CPSC 4383

Artificial Intelligence

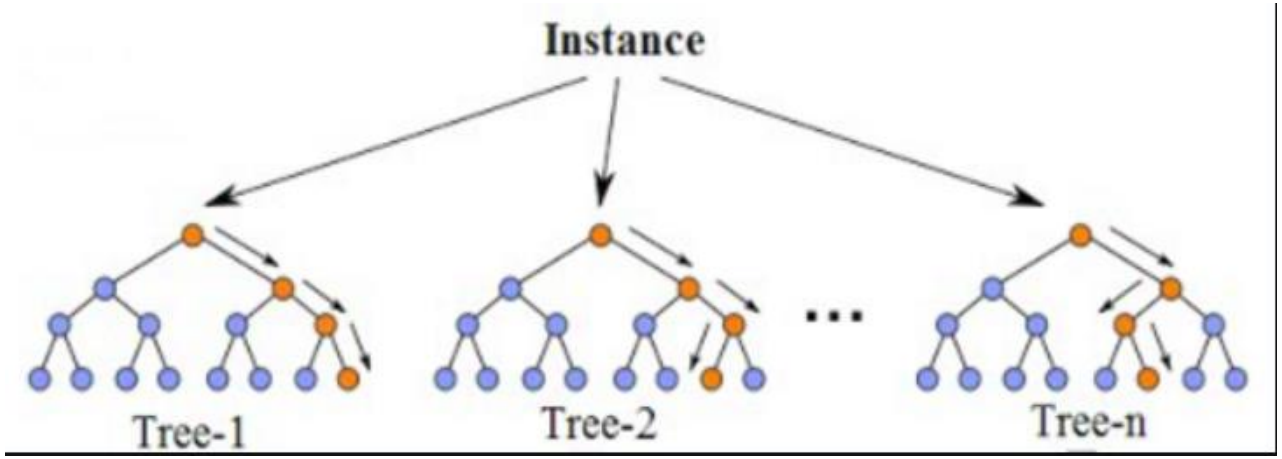
Outline

- Random Forest Regression model
- Model training
- Evaluations of Metrics for Regression
- Implementation of the RF Regression
- Visualization

What is Random Forest Regression?

Random forest regression is a machine learning algorithm that combines the results from multiple decision trees to make predictions based on various features.

- RFG handles non-linear relationships in the data
- Reduces the risk of overfitting
- Feature of importance
- No slope/intercept



https://levelup.gitconnected.com

Random Forest Regression in A.I/ML

To effectively analyze complex and large datasets to predict unknown values in ML, it is essential to meet the underlying assumptions:

1. **Data Quality**– ensure that the forest has sufficient/complete data for accurate predictions.

2. **Model Structure** – assume that predictions from each tree have a very low correlation with each other.

3. **Noisy data handling** – assume that any unwanted data is not systematic but is randomly distributed instead.

Key Differences

Random Forest

- Random Forest can handle overfitting issues regardless of the subset of data.
- It is relatively slower.
- It selects data to build a decision tree, using the average of accumulated results.

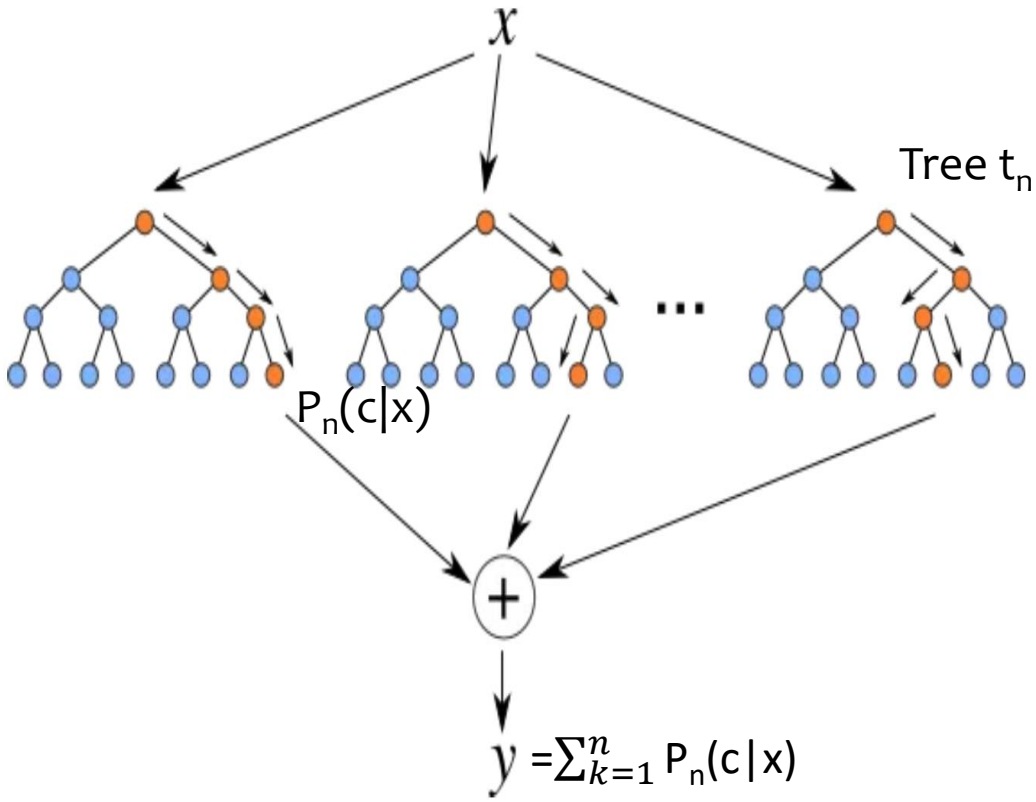
Decision Tree

- Decision trees tend to incur overfitting problems if they grow without control.
- Single decision trees have faster computations.
- Decision trees take features of a dataset as input, to generate predictions.

Model Training and Evaluation

Ensemble Learning – a technique that combines multiple decision trees to enhance accuracy.

- X – instance/dataset
- Each tree makes its own prediction.
- The final prediction is the average of all individual trees.
- Y - combination of decision tree (Predictor)

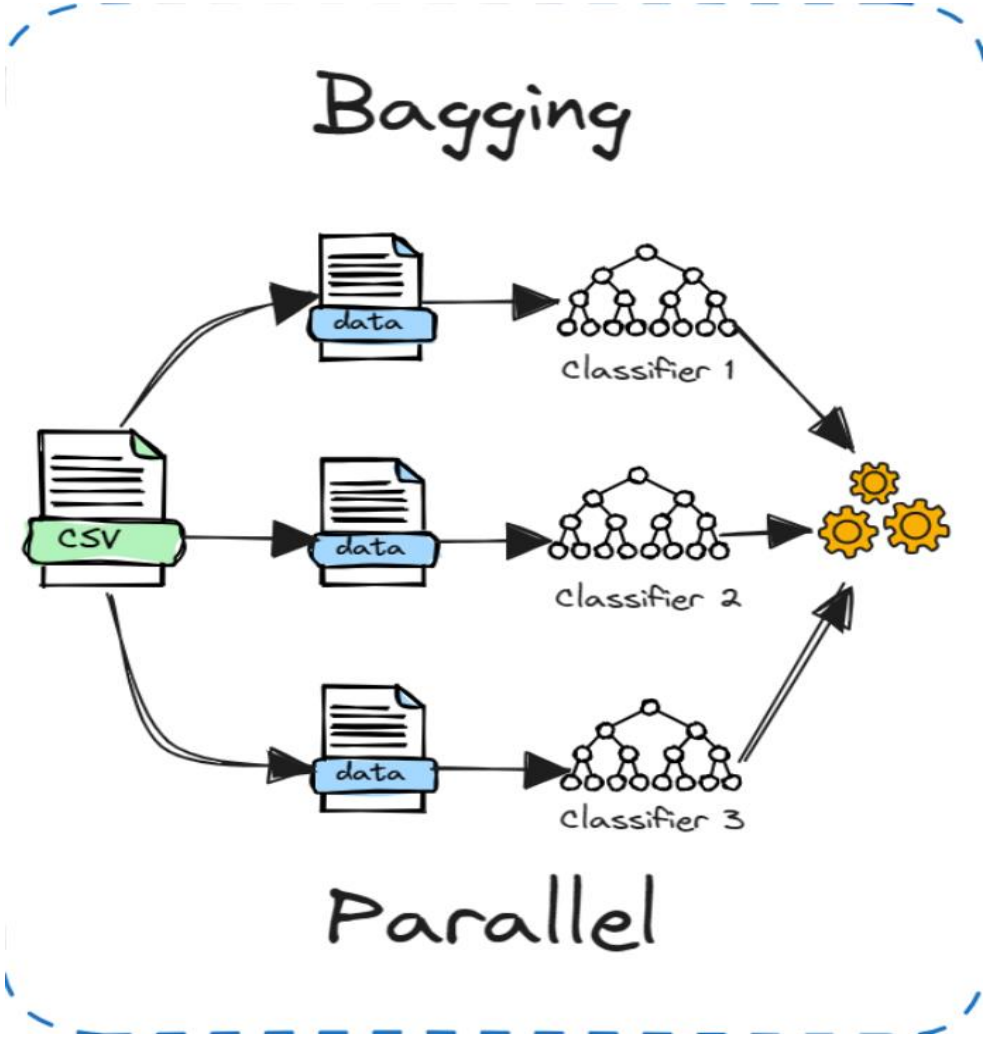


Wisdom of The Crowds

Types of Ensembles

Bagging – a primary ensemble method for Random Forest algorithm.

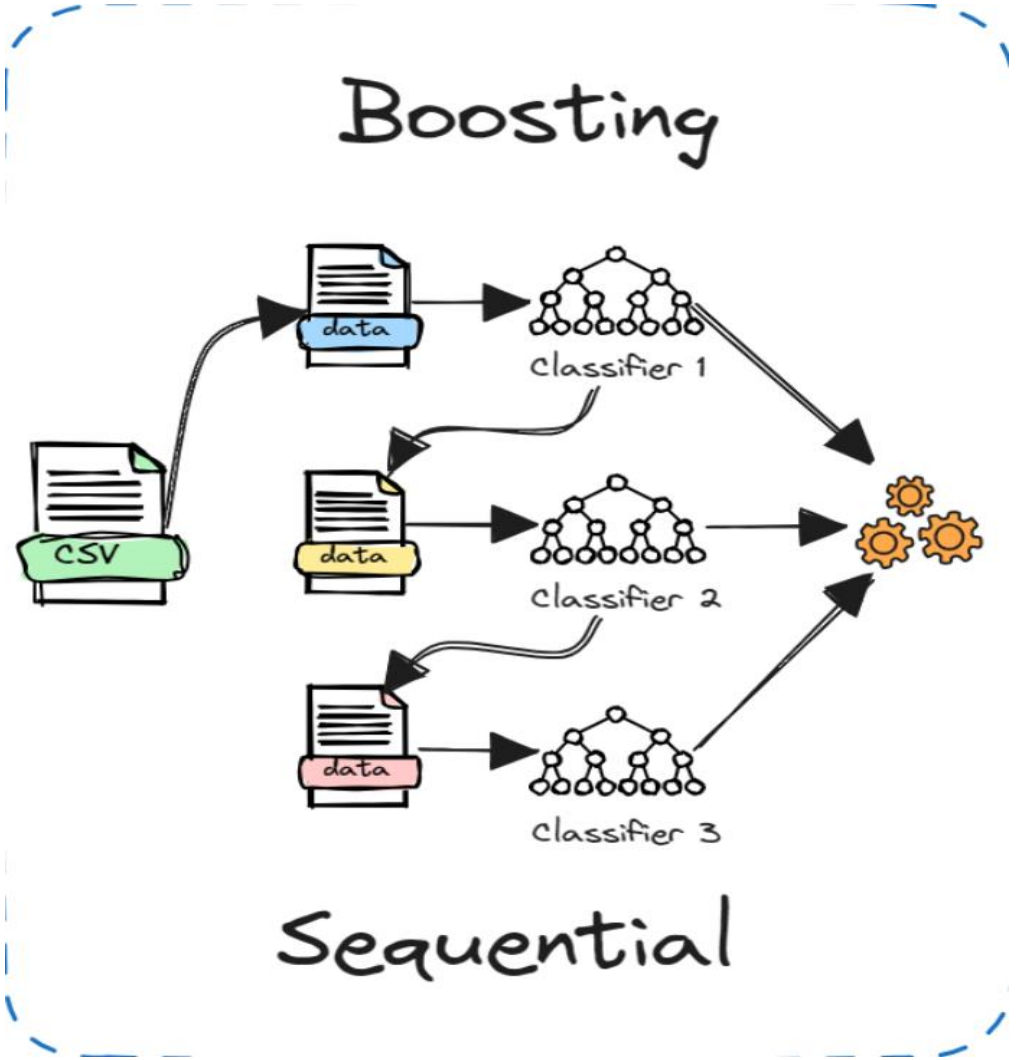
- Uses bootstrap aggregating to create training data sets
- Create multiple subsets of the training data through random sampling with replacement.
- Train individual models independently in parallel
- Aggregate all predictions to make a single prediction.



DataCamp – https://www.datacamp.com/tutorial/what-bagging-in-machine-learning-a-guide-with-examples

Boosting – involves multiple trained models or base estimators, with each new model learning from the previous one.

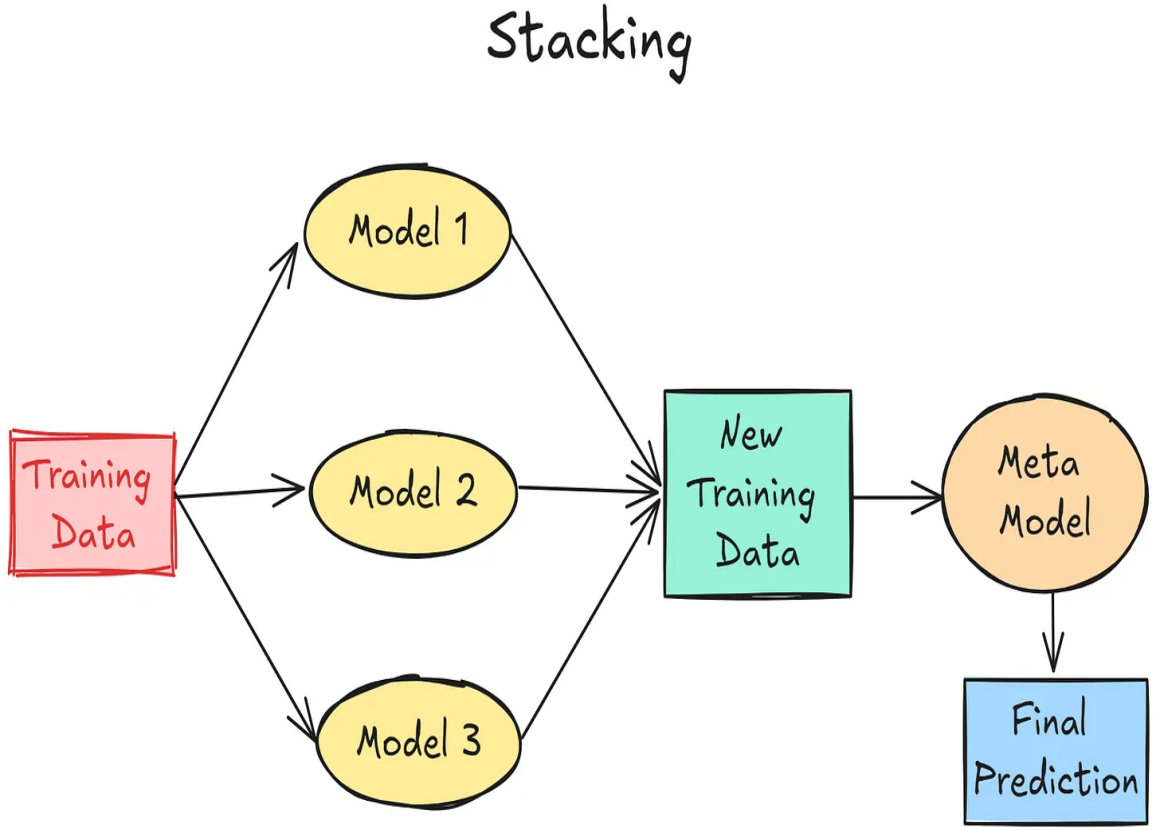
- Use weak model series
- Build models/decision trees sequentially
- Each model determines which feature the next model will focus on.
- Merge the selected features using a weighted-averaging.



DataCamp – https://www.datacamp.com/tutorial/what-bagging-in-machine-learning-a-guide-with-examples

Stacking – an ensemble method that combines multiple base models to create a more powerful predictive model (Meta).

- Prepare the data
- Select the model for stacking
- Train the base models.
- Base models make predictions with cross-validation.
- Collect the predictions that become features/inputs.
- The predictions are fed as input to the meta-learner.
- The meta-learner produces the final prediction.

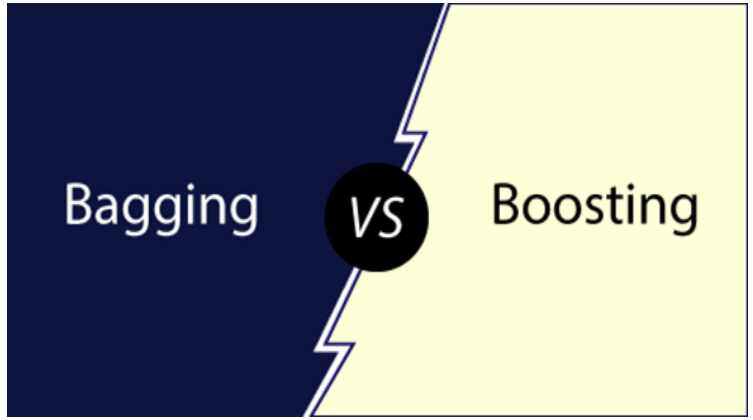


NB Data- https://www.nb-data.com/p/comparing-model-ensembling-bagging

Which Learning Method do we use?

The choice of the ensemble depends on the requirements for the RFR task:

- Memory optimization
- Efficient with noisy data
- Parallel processing
- The size of the dataset
- Cluster networking support



JT- https://www.javatpoint.com/bagging-vs-boosting

Metrics for Random Forest Regression

- Mean Absolute Error – evaluates the average absolute difference between the predicted and actual values.

  $MAR = (1/n)\sum_{i=1}^{n} |yi - y'_i|$

- Mean Square Error – evaluates the average of the squared differences between the actual and predicted values for all data sets.

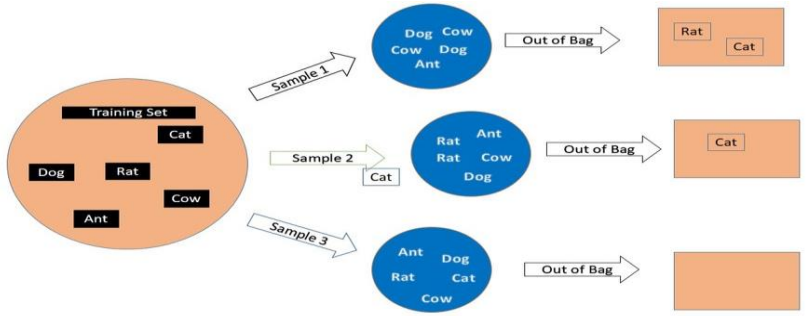
  $MSE = (1/n)\sum_{i=1}^{n} (yi - y'_i)^2$

- Coefficient of Determination (R^2)

  $R^2 = 1 - [(\sum (yi - y'_i)^2) / \sum ((yi - y')^2)]$

Where,
N – is the number of elements
Y_i- is the actual observed value
Y'_i – is the predicted value for the ith term

- Out-of-Bag Score(OOB) – evaluates the average prediction error of each test sample in its respective decision tree.



https://www.analyticsvidhya.com/blog/2020/12/out-of-bag-oob-score-in-the-random-forest-algorithm/

Python Implementation of RFR

- Import the necessary libraries

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score
from sklearn.tree import plot_tree
from sklearn.tree import DecisionTreeRegressor
```

- pandas for data manipulation
- numpy for numerical operations
- seaborn for statistical data visualization
- matplotlib for visualization (bar charts, plots)
- Various modules from scikit-learn for machine learning tasks.

- Load the weather dataset

```
print(df.head())
print(df.info())
print(df.describe())
```

- Handle missing data (f.fill)

```
df['date'] = pd.to_datetime(df['date'])
df = df.set_index('date').ffill().reset_index()
print("\n missing values after fill:")
print(df.isnull().sum())
```

'**ffill()**' is a method that handles incomplete data by propagating forward the last valid observation to the empty cell. "last observation carried forward" (**LOCF**)

- Create a feature matrix and response vector

```
X = df[['wind_speed', 'humidity', 'meanpressure']] #features
y = df['meantemp'] #Response vector
```

- Split the dataset

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.30, random_state=42)
```

- Make and train the Random Forest model

```
rf_model = RandomForestRegressor(n_estimators=100, random_state=42, oob_score=True)
rf_model.fit(X_train, y_train)
```

- Generate predictions

```
y_pred = rf_model.predict(X_test)
print('Predictions:\n', y_pred)
```

- Compare the actual and predicted data

```
comparison_df = pd.DataFrame({"Actual": y_test, "Predicted": y_pred})
print('Actual test data vs predicted: \n', comparison_df.head())
```

- Evaluate the RF metrics

```
oob_score = rf_model.oob_score_
r2 = r2_score(y_test, y_pred)
print(f'Out-of-Bag Score: {oob_score}')
print("MSE:", mean_squared_error(y_test, y_pred))
print('MAE:', mean_absolute_error(y_test, y_pred))
print(f'R-squared Score: {r2}')
```

- Print the Random Forest model over Time

```
plt.figure(figsize=(10, 5))
plt.scatter(df['date'].loc[y_test.index], y_test, color='green', label='Actual', alpha=0.5)
plt.scatter(df['date'].loc[y_test.index], y_pred, color='red', label='Predicted', alpha=0.5)
plt.xlabel('Time')
plt.ylabel('Mean_Temperature')
plt.title('actual vs predicted temperatures over time')
plt.legend()
```

- Visualize feature importance

```
feature_importance = pd.DataFrame({
    'feature': X.columns,
    'importance': rf_model.feature_importances_
}).sort_values('importance', ascending=False)

plt.figure(figsize=(10, 5))
sns.barplot(x='importance', y='feature', data=feature_importance)
plt.title('feature importance for predicted temperature')
plt.xlabel('relative importance')
plt.ylabel('features')
plt.tight_layout()
plt.show()
```

- Visualize a decision tree

```
plt.figure(figsize=(80, 40))
tree_to_plot = rf_model.estimators_[7]
plot_tree(tree_to_plot,
          feature_names=X.columns,
          filled=True,
          rounded=True,
          fontsize=10)
plt.title("Decision tree from Random Forest")
plt.tight_layout()
plt.show()
```

References

- Sumbatilinda, M. “Random Forests Regression by Example.” *Medium*, Medium, 26 Mar. 2024, medium.com/@sumbatilinda/random-forests-regression-by-example-1baa062506f5.
- Mwiti, Derrick. “Random Forest Regression: When Does It Fail and Why?” *Neptune.Ai*, 1 Sept. 2023, neptune.ai/blog/random-forest-regression-when-does-it-fail-and-why.
- AnalytixLabs. “Random Forest Regression - How It Helps in Predictive Analytics?” *Medium*, Medium, 26 Dec. 2023, medium.com/@byanalytixlabs/random-forest-regression-how-it-helps-in-predictive-analytics-01c31897c1d4.
- B, Nima. “Random Forest Regression.” *Medium*, Towards Data Science, 2 Mar. 2022, towardsdatascience.com/random-forest-regression-5f605132d19d.
- Ibm. “What Is Random Forest?” *IBM*, 2 Oct. 2024, www.ibm.com/topics/random-forest.
- FLP. “Bagging vs Boosting - Javatpoint.” *Www.Javatpoint.Com*, www.javatpoint.com/bagging-vs-boosting. Accessed 19 Oct. 2024.
- https://aiml.com/what-are-the-advantages-and-disadvantages-of-random-forest/

END