Linear models are a robust set of algorithms, which can be used for regression and classification problems. One of the major benefits of using linear models is that, in general, they are easier to interpret, train, and deploy. Because of their ease of use, they are used as the baseline model in training workflows. This means they are the first model implemented in a new problem and used to compare all additional work to see what improvements are made. In most cases, however, they are not as powerful as other advanced models.

Code Example of Linear Model

```python
import pandas as pd
from sklearn.linear_model import LinearRegression, LogisticRegression

df = pd.DataFrame(
    [[1, 2, 0], [3, 4, 1], [5, 6, 0], [7, 8, 1]],
    columns=["num", "amount", "target"]
)

# regression model
reg = LinearRegression().fit(df[["num", "amount"]], df["target"])
reg.score(df[["num", "amount"]], df["target"])

#classification model
clf = LogisticRegression().fit(df[["num", "amount"]], df["target"])
clf.score(df[["num", "amount"]], df["target"])
```

Additional Resources

- Scikit-learn **linear models documentation**
- Sagemaker has a **linear learner** algorithm