Loan Approval Predictor

Automating decision-making with Machine Learning and Streamlit

Python Streamlit Scikit-Learn Pandas

The Challenge: Efficiency

Manual Processing

Traditional loan approval involves sifting through piles of paperwork, manual data entry, and subjective decision-making.

The Solution

By training a **Decision Tree** model on historical applicant data, we can provide instant, data-driven predictions on loan eligibility.

Technology Stack

- ✓ **Pandas:** For data manipulation and creating the structured dataset (DataFrames).
- ✓ **Scikit-Learn:** To build and train the `DecisionTreeClassifier` model.
- ✓ **Streamlit:** To create the interactive web interface (Sidebar, Buttons, Display).
- ✓ **NumPy:** For efficient numerical array handling during prediction.

Training Data

The model is trained on a sample dataset representing historical loan applicants.

Gender	Married	Credit History	Income	Status (Target)
Male	Yes	1.0 (Good)	$5000	Y
Female	No	1.0 (Good)	$4500	N
Male	Yes	0.0 (Poor)	$3500	N

∗Note: Missing credit history is filled with default values during preprocessing.

Preprocessing Logic

Encoding Categoricals

Machine learning models require numerical input. We map text data to binary integers before training.

$$\text{Male} \rightarrow 1 , \quad \text{Female} \rightarrow 0$$

$$\text{Married} \rightarrow 1 , \quad \text{Single} \rightarrow 0$$

Code Snippet

```
df['Gender'] = df['Gender'].map(
    {'Male': 1, 'Female': 0}
)

df['Married'] = df['Married'].map(
    {'Yes': 1, 'No': 0}
)
```

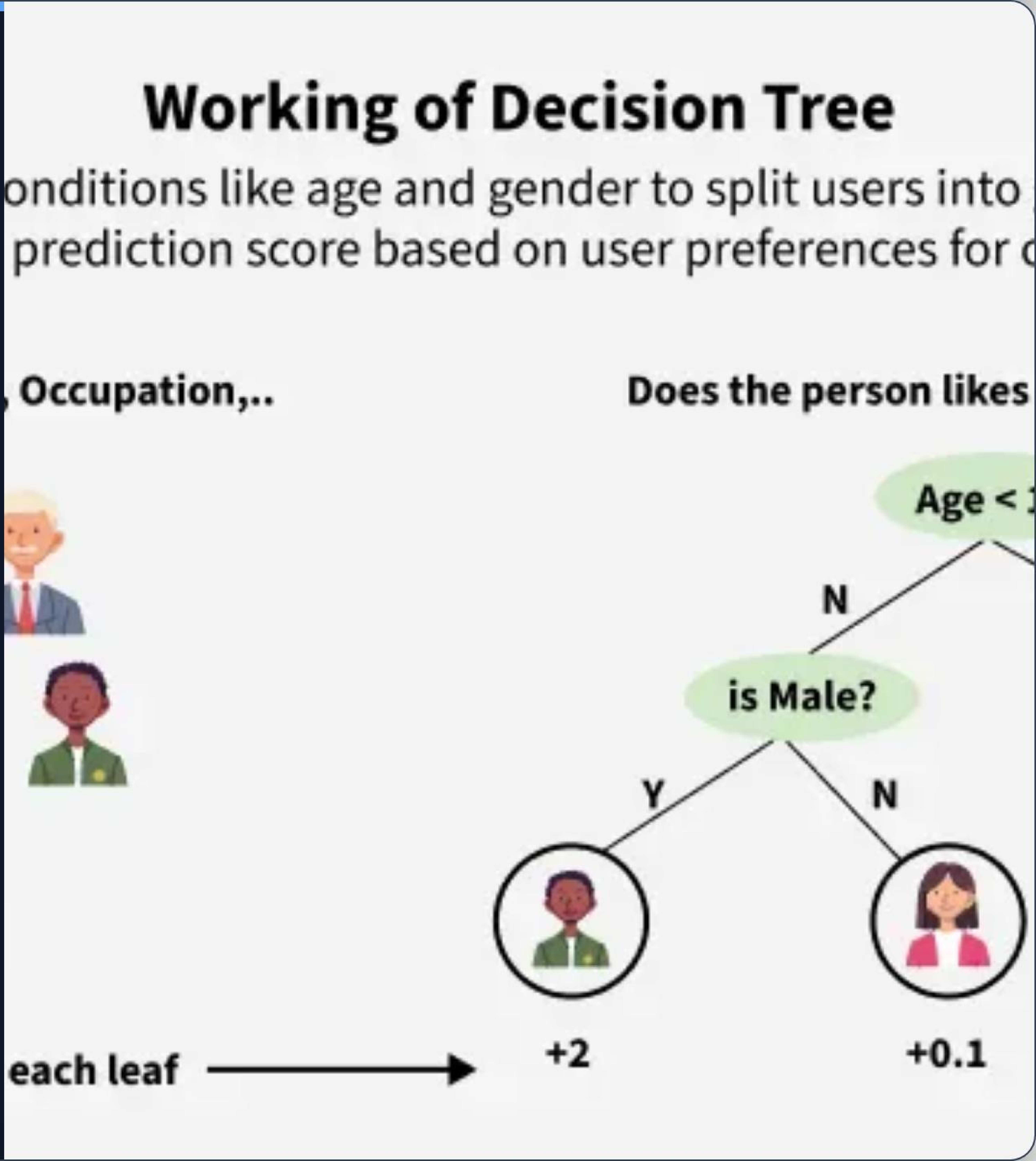
Decision Tree Classifier

How It Works

The model uses a **Decision Tree**. Think of it as a flowchart where the model asks a series of questions to reach a conclusion.

For example: *"Is Credit History == 1.0?"*. If Yes, check Income. If No, Reject Loan.

We limit the tree depth (`max_depth=3`) to prevent overfitting.


Working of Decision Tree
onditions like age and gender to split users into
prediction score based on user preferences for
, Occupation,..
Does the person likes
Age <
N
is Male?
Y
N
+2
+0.1
each leaf

User Interface

Sidebar

Inputs are grouped in `st.sidebar` for a clean layout, separating data entry from the main results.

Inputs

We use `st.selectbox` for categorical options (Gender, Marital Status) and `st.number_input` for Income.

Interactivity

The `st.button('Predict')` triggers the entire prediction workflow only when the user is ready.

Prediction Logic

1. **Gather Inputs:** Collect values from the Streamlit widgets.
2. **Encode:** Convert user inputs into the `[1, 0, 1.0, 5000]` format the model expects.
3. **Predict:** Pass the array to `model.predict()`.
4. **Display:** Show Success (Green) or Error (Red) message based on the 'Y' or 'N' result.

Input Array

```
[[ gender, married, history,
income ]]
```

Output

```
['Y'] (Loan Approved)
```

The Result

Instant Feedback

Upon clicking predict, the user sees an immediate visual confirmation.

- ✓ `st.success()` displays a green notification box.
- ✓ `st.balloons()` adds a celebratory animation for approvals.
- ✓ Summary dataframe confirms exactly what data was sent to the model.

Questions?

Thank you for exploring the Loan Predictor.

Powered by Python

Machine Learning

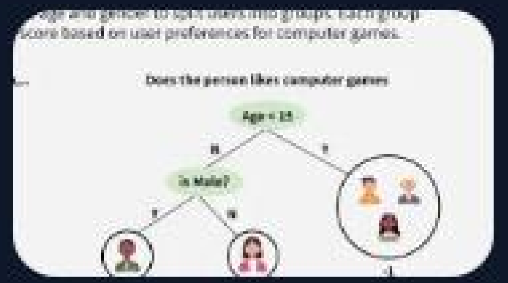
Image Sources

https://static.vecteezy.com/system/resources/previews/008/066/906/large_2x/loan-approved-on-loan-application-form-paper-with-rubber-stamp-on-table-loan-approval-business-finance-economy-concept-free-photo.JPG

Source: www.vecteezy.com

https://static.vecteezy.com/system/resources/previews/054/088/420/non_2x/computer-monitor-displaying-python-programming-language-code-on-blue-background-png.png

Source: www.vecteezy.com



https://media.geeksforgeeks.org/wp-content/uploads/20250514105137227681/Working-of-Decision-Tree.webp

Source: www.geeksforgeeks.org

https://en.pimg.jp/039/795/943/1/39795943.jpg

Source: www.pixtastock.com