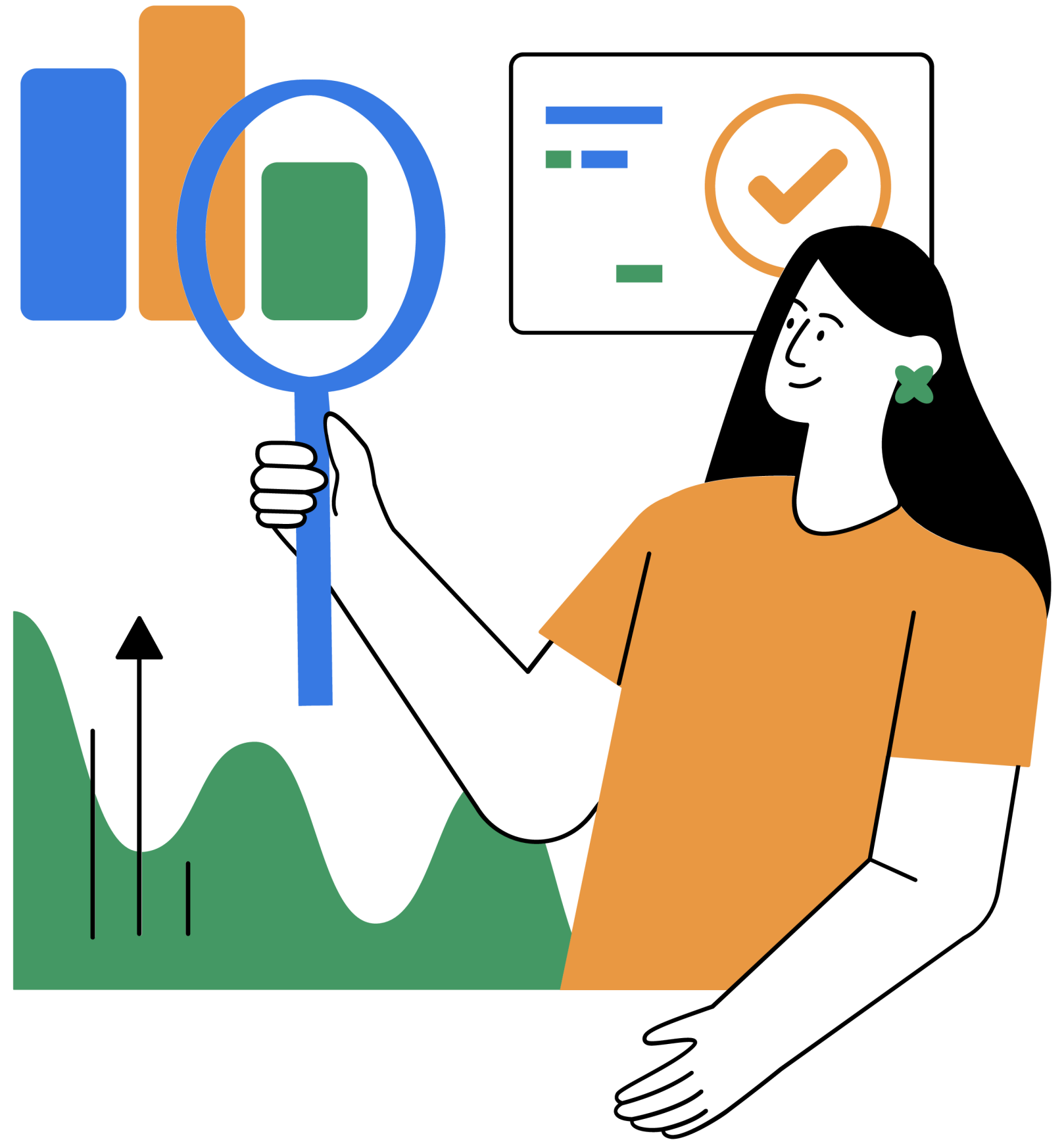




# Credit Risk Prediction & Decision Dashboard



ISM6358 – Decision Making with BI & Analytics  
Seattle Pacific University  
Summer 2025  
Tejashwini Saravanan





Loan Purpose	Checking	Savings	Months Customer	Months Employed	Gender	Marital Status	Age	Housing	Years	Job	Credit Risk
Small Appliance	\$0	\$739	13	12	M	Single	23	Own	3	Unskilled	Low
Furniture	\$0	\$1,230	25	0	M	Divorced	32	Own	1	Skilled	High
New Car	\$0	\$389	19	119	M	Single	38	Own	4	Management	High
Furniture	\$638	\$347	13	14	M	Single	36	Own	2	Unskilled	High
Education	\$963	\$4,754	40	45	M	Single	31	Rent	3	Skilled	Low
Furniture	\$2,827	\$0	11	13	M	Married	25	Own	1	Skilled	Low
New Car	\$0	\$229	13	16	M	Married	26	Own	3	Unskilled	Low
Business	\$0	\$533	14	2	M	Single	27	Own	1	Unskilled	Low
Small Appliance	\$6,509	\$493	37	9	M	Single	25	Own	2	Skilled	High
Small Appliance	\$966	\$0	25	4	F	Divorced	43	Own	1	Skilled	High
Business	\$0	\$989	49	0	M	Single	32	Own	2	Management	High
New Car	\$0	\$3,305	11	15	M	Single	34	Rent	2	Unskilled	Low
Business	\$322	\$578	10	14	M	Married	26	Own	1	Skilled	Low
New Car	\$0	\$821	25	63	M	Single	44	Own	1	Skilled	High
New Car	\$396	\$228	13	26	M	Single	46	Own	3	Unskilled	Low
Used Car	\$0	\$129	31	8	M	Divorced	39	Own	4	Management	Low
Furniture	\$652	\$732	49	4	F	Divorced	25	Own	2	Skilled	High
New Car	\$708	\$683	13	33	M	Single	31	Own	2	Skilled	Low
Repairs	\$207	\$0	28	116	M	Single	47	Own	4	Skilled	Low
Education	\$287	\$12,348	7	2	F	Divorced	23	Rent	2	Skilled	High
Furniture	\$0	\$17,545	34	16	F	Divorced	22	Own	4	Skilled	High
Furniture	\$101	\$3,871	13	5	F	Divorced	26	Rent	4	Skilled	High
Furniture	\$0	\$0	25	23	M	Married	19	Own	4	Skilled	High
Furniture	\$0	\$485	37	23	F	Divorced	27	Own	2	Management	High
New Car	\$0	\$10,723	11	15	M	Single	39	Rent	2	Unskilled	Low
Business	\$141	\$245	22	33	M	Single	26	Own	3	Skilled	Low
Used Car	\$0	\$0	19	58	M	Single	50	Other	4	Skilled	High
Used Car	\$2,484	\$0	49	46	M	Single	34	Other	1	Skilled	Low
Small Appliance	\$237	\$236	37	24	M	Single	23	Rent	4	Skilled	Low

```
import pandas as pd
file_path = r"C:\Users\Tejashwini Saravanan\Credit Risk Data.xlsx"
base_df = pd.read_excel(file_path, sheet_name="Base Data")
base_df.head()
```

E:\New folder\Lib\site-packages\openpyxl\worksheet\\_read\_only.py:81: UserWarning: Unknown extension is not supported and will be removed for idx, row in parser.parse():

	Loan Purpose	Checking	Savings	Months Customer	Months Employed	Gender	Marital Status	Age	Housing	Years	Job	Credit Risk
0	Small Appliance	0	739	13	12	M	Single	23	Own	3	Unskilled	Low
1	Furniture	0	1230	25	0	M	Divorced	32	Own	1	Skilled	High
2	New Car	0	389	19	119	M	Single	38	Own	4	Management	High
3	Furniture	638	347	13	14	M	Single	36	Own	2	Unskilled	High
4	Education	963	4754	40	45	M	Single	31	Rent	3	Skilled	Low



# Project Overview

- Used dataset: Credit Risk Data.xlsx
- Focused on "Base Data" sheet



## Objective -

- Build a machine learning model to predict Credit Risk
- Deploy as a dashboard for decision makers

## Tools used -

- Python (Pandas, Scikit-Learn, Plotly) for data prep & ML
- Dash & ngrok for deployment

# \* Data Cleaning & Preparation

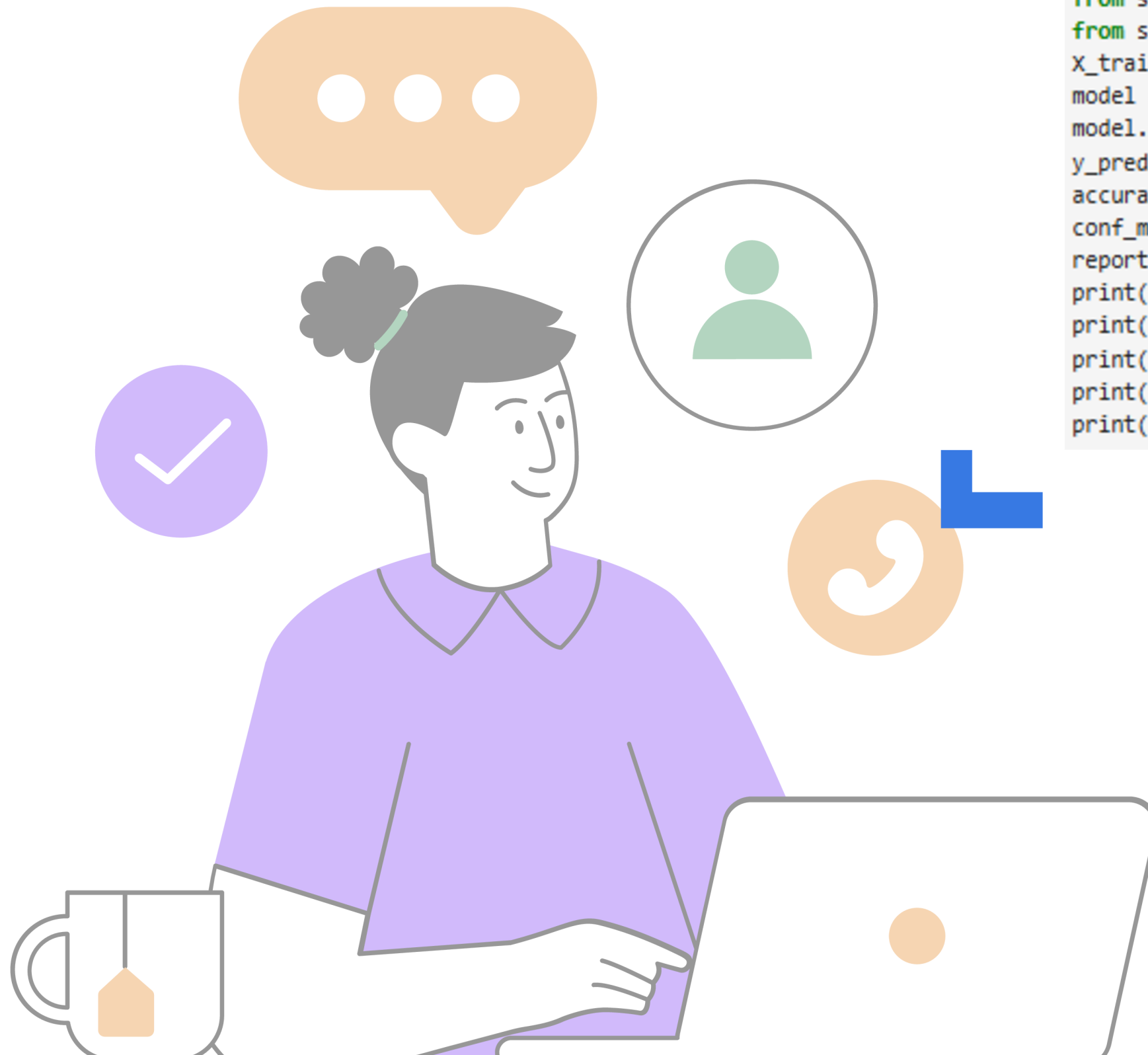
- Loaded Excel data into pandas dataframe
- Dropped rows with missing values using `dropna()` to ensure model reliability
- Applied Label Encoding to convert categorical columns into numeric format
- Used sklearn's `LabelEncoder`
- Stored encoders for each column in a dictionary

```
from sklearn.preprocessing import LabelEncoder
df = base_df.copy()
df.dropna(inplace=True)
label_encoders = {}
for col in df.select_dtypes(include='object').columns:
    le = LabelEncoder()
    df[col] = le.fit_transform(df[col])
    label_encoders[col] = le
X = df.drop(columns=["Credit Risk"])
y = df["Credit Risk"]
df.head()
```

	Loan Purpose	Checking	Savings	Months Customer	Months Employed	Gender	Marital Status	Age	Housing	Years	Job	Credit Risk
0	8	0	739	13	12	1	2	23	1	3	3	1
1	2	0	1230	25	0	1	0	32	1	1	1	0
2	4	0	389	19	119	1	2	38	1	4	0	0
3	2	638	347	13	14	1	2	36	1	2	3	0
4	1	963	4754	40	45	1	2	31	2	3	1	1



# Model Building - Decision Tree



```
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
model = DecisionTreeClassifier(random_state=42)
model.fit(X_train, y_train)
y_pred = model.predict(X_test)
accuracy = accuracy_score(y_test, y_pred)
conf_matrix = confusion_matrix(y_test, y_pred)
report = classification_report(y_test, y_pred)
print(f"Accuracy: {accuracy:.2f}")
print("\nConfusion Matrix:")
print(conf_matrix)
print("\nClassification Report:")
print(report)
```

- Selected Decision Tree Classifier for its interpretability
- Defined target variable: "Credit Risk"
- Remaining columns were treated as features

Split dataset -

- 80% training data, 20% testing data
- Used train\_test\_split with random\_state = 42 for reproducibility
- Trained model on training set



# Model Evaluation

Accuracy: 0.46

Confusion Matrix:

```
[[20 20]
 [26 19]]
```

Classification Report:

	precision	recall	f1-score	support
0	0.43	0.50	0.47	40
1	0.49	0.42	0.45	45
accuracy			0.46	85
macro avg	0.46	0.46	0.46	85
weighted avg	0.46	0.46	0.46	85

- Accuracy on test data: 46%

Confusion Matrix -

- True Negatives (correctly predicted low risk): 20
- False Positives (incorrectly predicted low risk): 20
- False Negatives (incorrectly predicted high risk): 26
- True Positives (correctly predicted high risk): 19

Classification Report metrics -

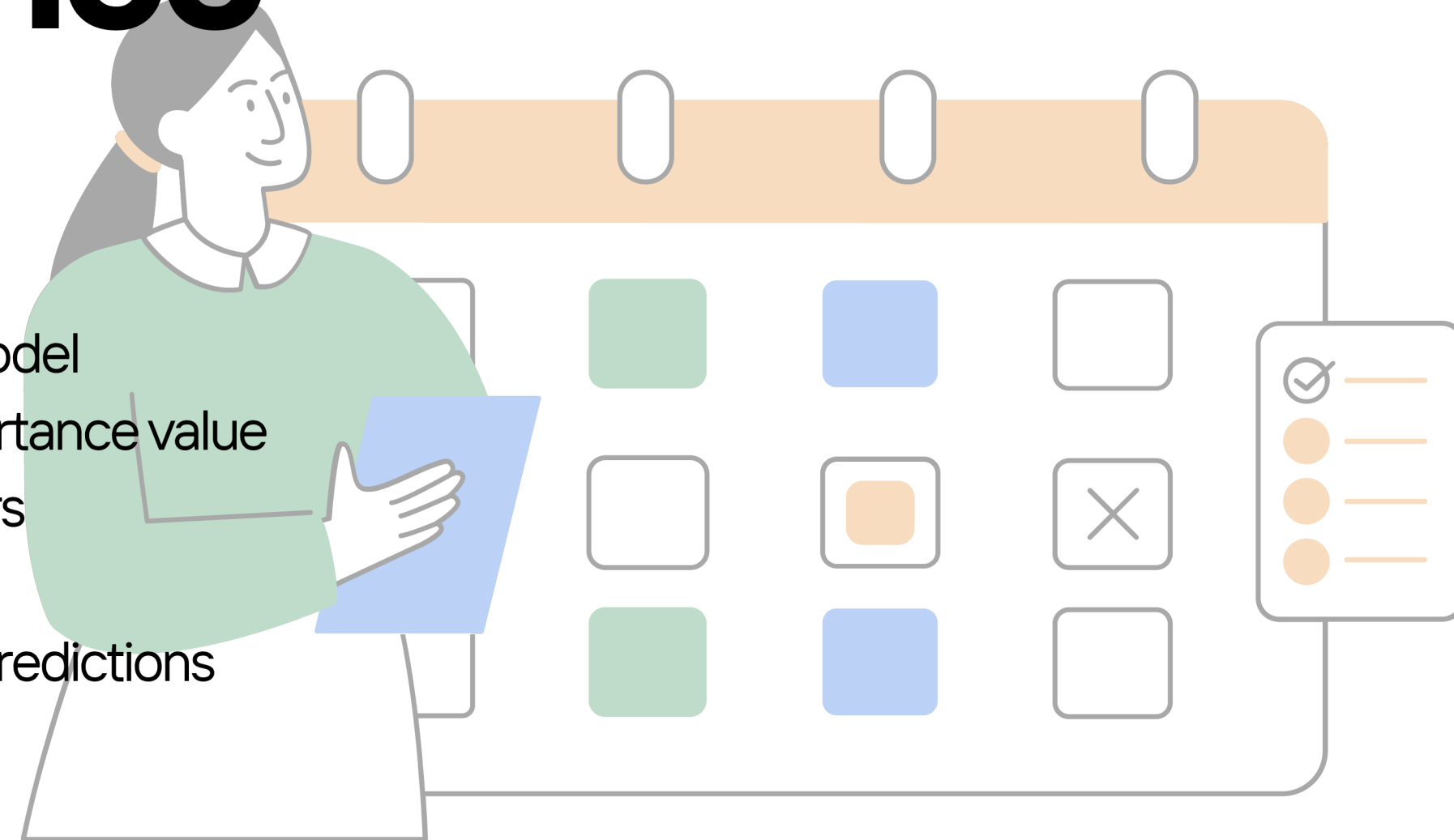
- Class 0 (low risk):
  - Precision: 0.43
  - Recall: 0.50
  - F1-score: 0.47
- Class 1 (high risk):
  - Precision: 0.49
  - Recall: 0.42
  - F1-score: 0.45

Macro & weighted avg F1: 0.46 → shows balanced, but overall low predictive power



# \* Feature Importance Analysis

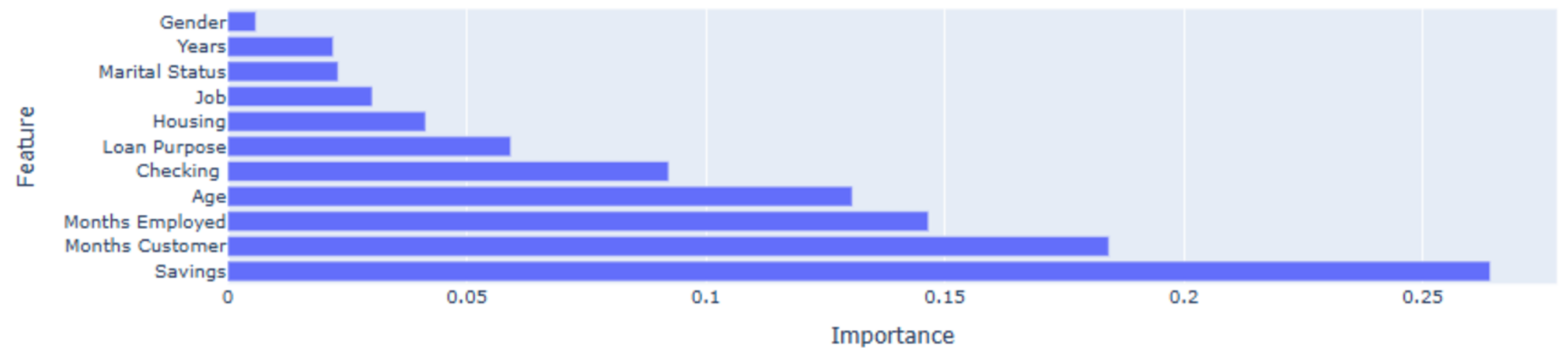
- Extracted feature importances from the trained Decision Tree model
- Created a dataframe to pair each feature with its calculated importance value
- Sorted features by importance (highest first) to identify key drivers
- Built a horizontal bar chart using Plotly Express to visualize this
- This helps understand which factors most influenced credit risk predictions



```
import plotly.express as px
importances = model.feature_importances_
feature_names = X.columns
feat_df = pd.DataFrame({'Feature': feature_names, 'Importance': importances})
feat_df = feat_df.sort_values(by='Importance', ascending=False)
fig = px.bar(feat_df, x='Importance', y='Feature', orientation='h',
            title='Feature Importance - Decision Tree Model')
fig.show()
```



Feature Importance - Decision Tree Model



- Created annotated heatmap of the confusion matrix using plotly.figure\_factory

Labels show-

- True High: correctly predicted high risk
- True Low: correctly predicted low risk
- False High: low risk incorrectly predicted as high
- False Low: high risk incorrectly predicted as low

Used blue color scale for intuitive visual impact

Chart title: "Confusion Matrix - Credit Risk Model"

Why this approach -

- Heatmaps make it easy to spot model strengths & weaknesses at a glance
- Clearly highlights where misclassifications occurred



# Confusion Matrix Visualization

```
import plotly.figure_factory as ff
conf_matrix_labels = [['True High', 'False Low'], ['False High', 'True Low']]
fig = ff.create_annotated_heatmap(
    z=conf_matrix,
    x=['Predicted High', 'Predicted Low'],
    y=['Actual High', 'Actual Low'],
    annotation_text=conf_matrix_labels,
    colorscale='Blues',
    showscale=True
)
fig.update_layout(title_text='Confusion Matrix - Credit Risk Model')
fig.show()
```



Confusion Matrix - Credit Risk Model



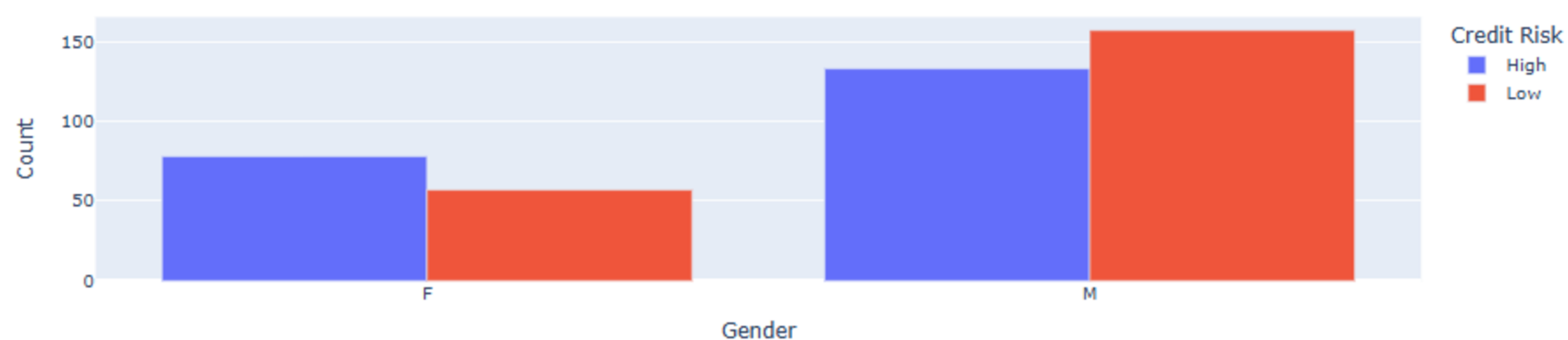
# \* Notebook charts

## Credit Risk by Gender

- Chart type: Grouped bar chart of credit risk split by gender
- Shows: Counts of high vs low credit risk customers across male and female groups
- Why useful: Reveals if there are noticeable differences in credit risk patterns by gender, supporting fair lending analysis.

```
gender_risk = base_df.groupby(["Gender", "Credit Risk"]).size().reset_index(name='Count')
fig = px.bar(
    gender_risk,
    x="Gender",
    y="Count",
    color="Credit Risk",
    barmode="group",
    title="Credit Risk by Gender"
)
fig.show()
```

Credit Risk by Gender



## Savings Distribution by Credit Risk

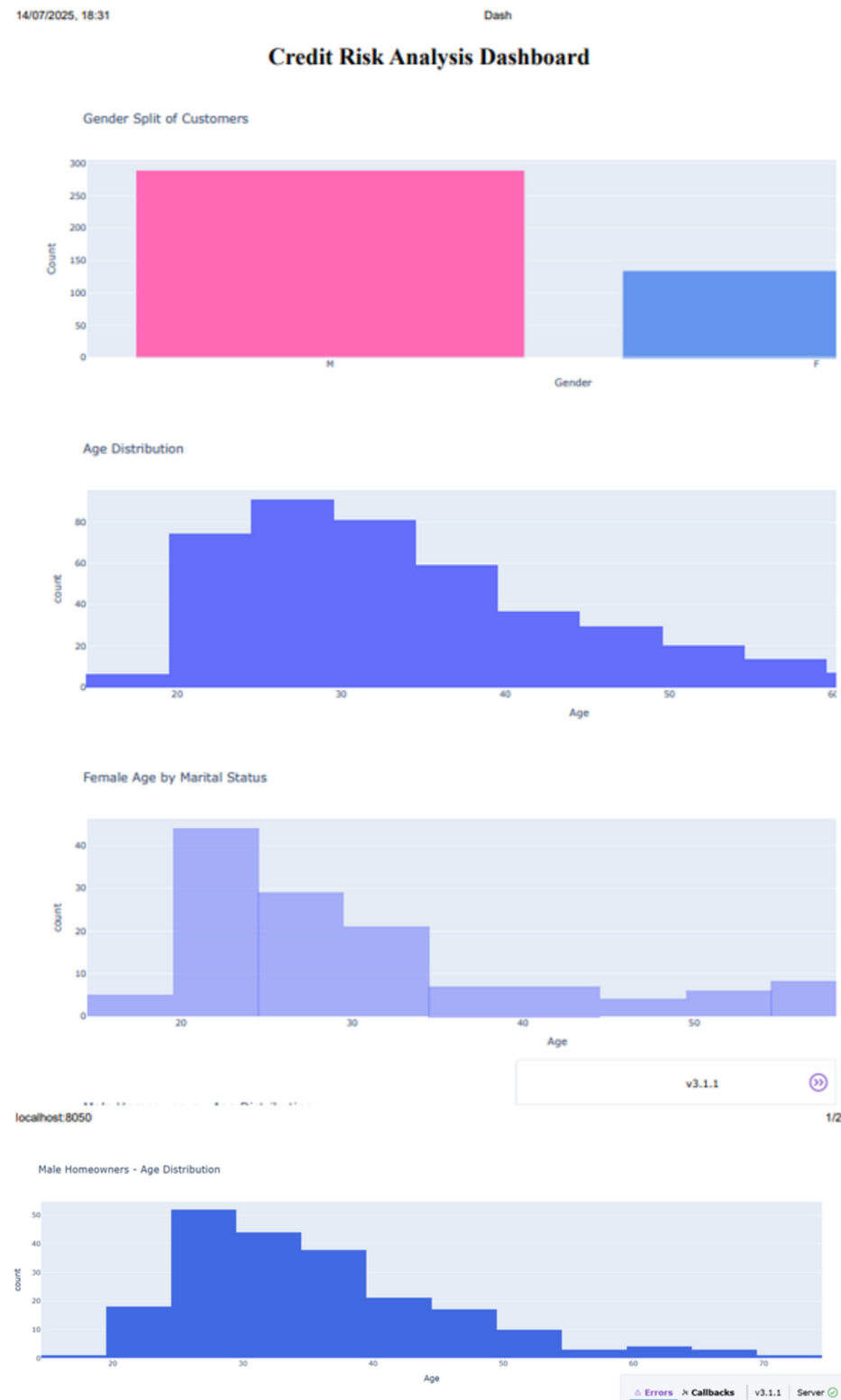
- Chart type: Box plot of savings amounts grouped by credit risk level
- Shows: How savings vary between high and low credit risk customers
- Why useful: Highlights if lower savings is linked to higher credit risk, aiding in designing risk thresholds.

```
fig = px.box(base_df, x="Credit Risk", y="Savings", title="Savings Distribution by Credit Risk")
fig.show()
```

Savings Distribution by Credit Risk



# \* Building & Running the Dashboard



- Used Dash by Plotly to create an interactive Python dashboard
- Dash is a Python framework that integrates pandas data, Plotly graphs, and Flask server into a single web app.
- Loaded cleaned data & embedded feature importance chart
- Ran locally on `http://127.0.0.1:8050/` using Anaconda prompt
- It lets decision makers explore the data visually without needing to run Python code.

```
Anaconda Prompt
(base) C:\Users\Tejashwini Saravanan>cd C:\Users\Tejashwini Saravanan
(base) C:\Users\Tejashwini Saravanan>python credit_dashboard.py
E:\New folder\Lib\site-packages\openpyxl\worksheet\_read_only.py:81: UserWarning: Unknown extension is not supported and will be removed
  for idx, row in parser.parse():
Dash is running on http://127.0.0.1:8050/

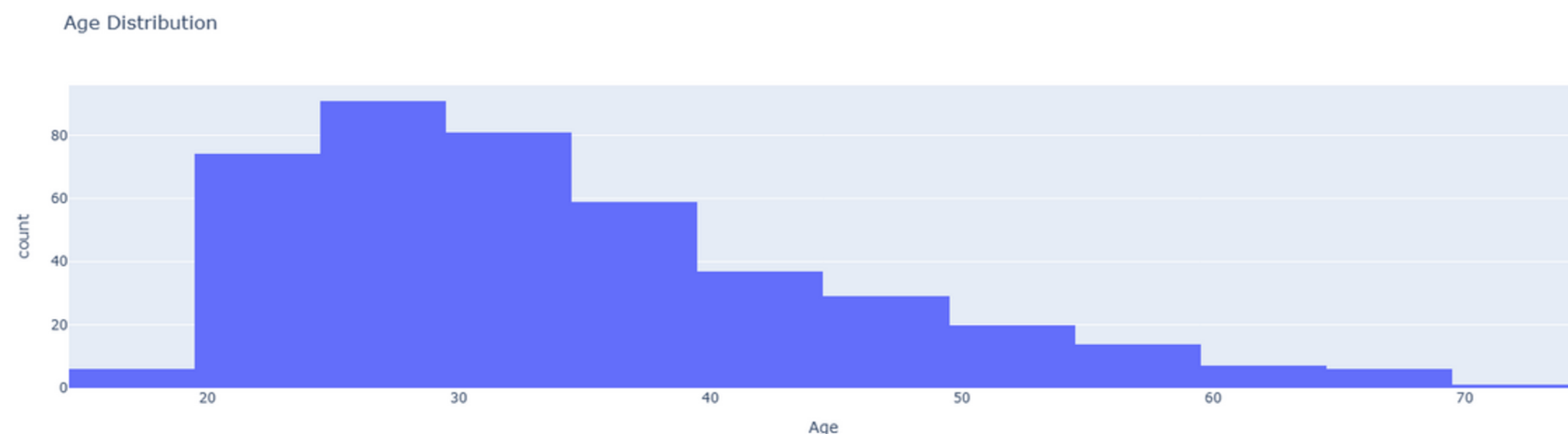
* Serving Flask app 'credit_dashboard'
* Debug mode: on
E:\New folder\Lib\site-packages\openpyxl\worksheet\_read_only.py:81: UserWarning: Unknown extension is not supported and will be removed
  for idx, row in parser.parse():
```

# \* Plotly Dash dashboard charts

## Age Distribution of Customers

- Chart type: Histogram of customer ages
- Shows: Overall spread and common age ranges in the dataset
- Why useful: Helps understand if credit applicants are clustered in certain age groups, guiding marketing & risk profiles.

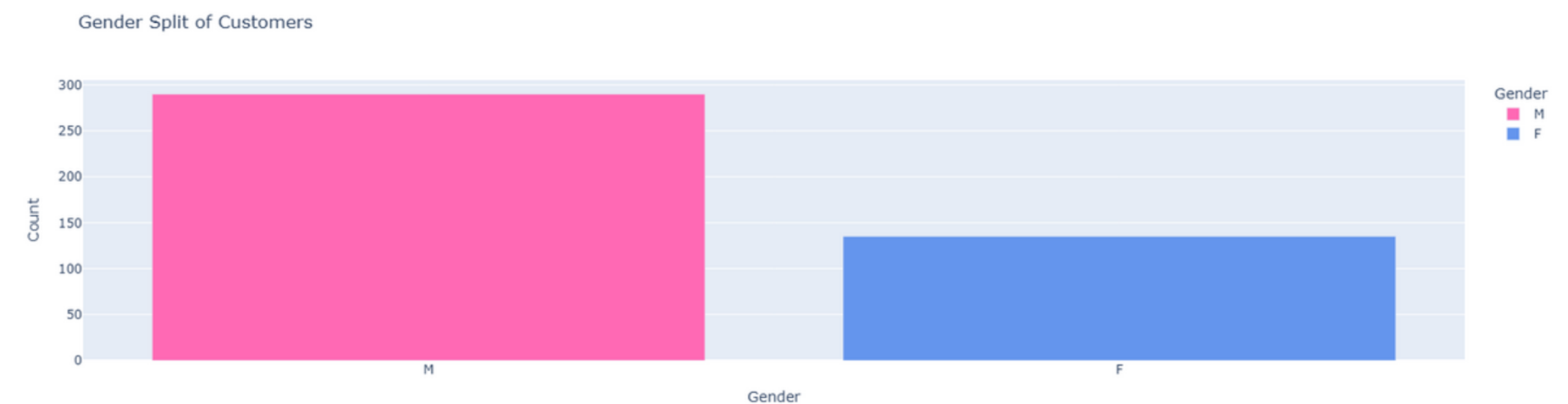
```
fig = px.histogram(base_df, x="Age", nbins=20, title="Age Distribution")  
fig.show()
```



## Gender Split of Customers

- Chart type: Bar chart of male vs female counts
- Shows: Proportion of male and female customers in the data
- Why useful: Identifies demographic makeup, which can influence targeted credit products or outreach.

```
gender_counts = base_df["Gender"].value_counts()  
import plotly.express as px  
fig = px.bar(  
    gender_counts,  
    x=gender_counts.index,  
    y=gender_counts.values,  
    labels={'x': 'Gender', 'y': 'Count'},  
    title="Gender Split of Customers"  
)  
fig.show()
```



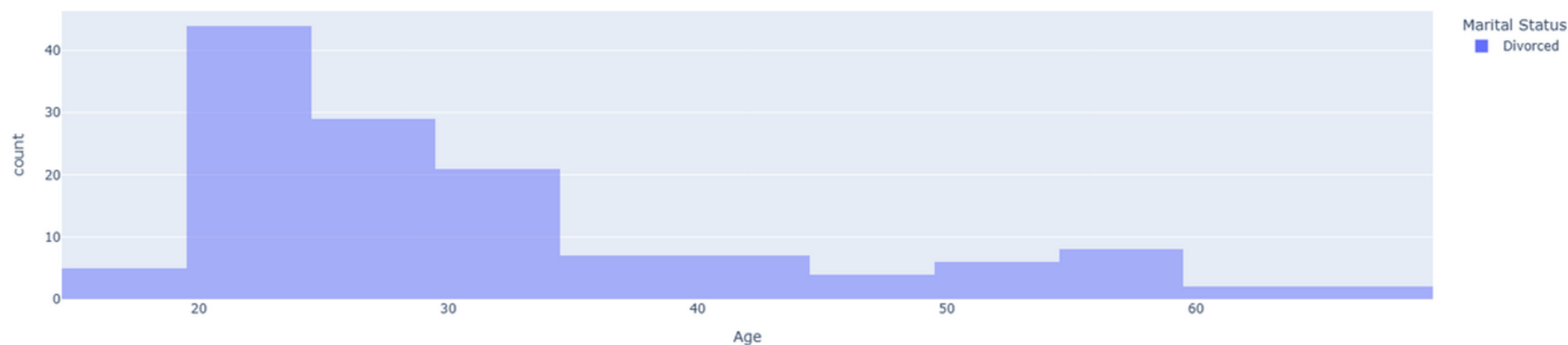
# \* Plotly Dash dashboard charts

## Age & Marital Status of Female Customers

- Chart type: Overlay histogram of ages, colored by marital status
- Shows: Age distribution of female customers, broken down by whether they're single, married, etc.
- Why useful: Lets lenders see if certain age & marital segments are more prevalent, guiding tailored offerings.

```
female_df = base_df[base_df["Gender"] == "F"]
fig = px.histogram(
    female_df,
    x="Age",
    color="Marital Status",
    barmode="overlay",
    nbins=20,
    title="Age Distribution of Female Customers by Marital Status"
)
fig.update_layout(
    xaxis_title="Age",
    yaxis_title="Count",
    bargap=0.2
)
fig.show()
```

Female Age by Marital Status

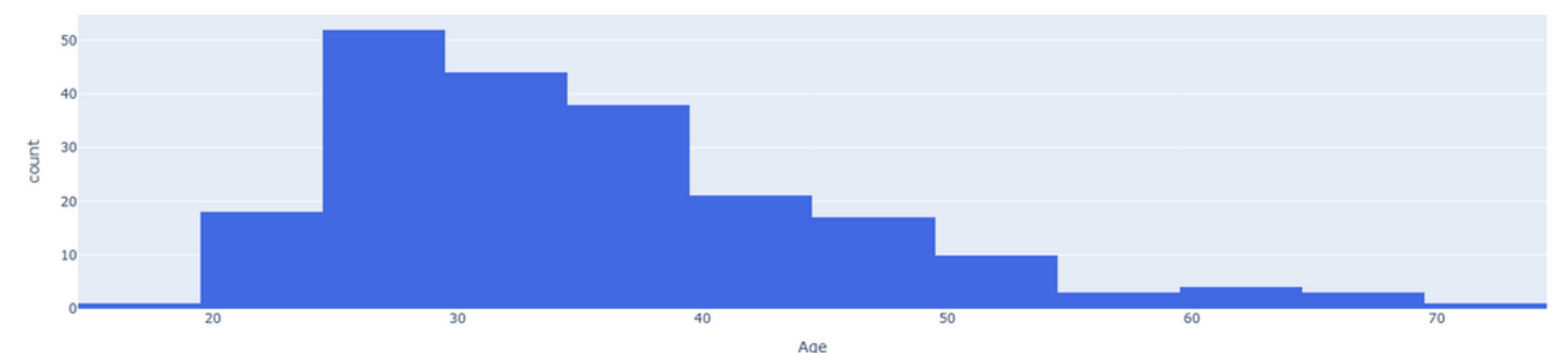


## Age Distribution of Male Home Owners

- Chart type: Histogram of ages for male customers who own their home
- Shows: Specific look at this subgroup's age spread
- Why useful: Home ownership often impacts credit decisions; understanding age profiles helps assess asset-backed lending.

```
male_own_df = base_df[(base_df["Gender"] == "M") & (base_df["Housing"] == "Own")]
fig = px.histogram(
    male_own_df,
    x="Age",
    nbins=20,
    title="Age Distribution of Male Customers Who Own Their Home",
    labels={'Age': 'Customer Age'},
    color_discrete_sequence=['steelblue']
)
fig.update_layout(
    xaxis_title="Age",
    yaxis_title="Count",
    bargap=0.2
)
fig.show()
```

Male Homeowners - Age Distribution





# \* Business Insights & Recommendations

## What I did

- Explored and cleaned the Credit Risk dataset in Jupyter Notebook
- Removed missing values and encoded categorical data
- Built a Decision Tree model to predict customer credit risk
- Evaluated the model with accuracy (46%) & confusion matrix
- Created multiple visualizations to understand data & model results:
- Age, gender, savings distributions and credit risk splits
- Developed an interactive Dash dashboard with 4 key charts
- Used ngrok to make the dashboard publicly accessible for easy review

## Why I did it this way

- Chose Python (Pandas, Plotly, Scikit-Learn) for full control over data, ML, and visuals in one workflow
- Used Dash to build a lightweight, customizable dashboard that directly integrates with Python avoiding extra costs or rigid templates of BI tools
- Leveraged ngrok to share results instantly without deploying to a server
- Focused on visuals & feature importance to help stakeholders quickly identify key drivers of credit risk

## What I learned & what might have gone wrong

### Learned:

- How to move from raw data to a deployed dashboard: data cleaning → modeling → visualization → deployment
- Importance of clearly visualizing model strengths & weaknesses (via confusion matrix & feature importance)

### What might have gone wrong:

- Model performance was low (46% accuracy), likely due to:
- Limited dataset size or features not strongly predictive
- Possible need for more feature engineering (creating debt ratios, income buckets)
- Could improve by testing more advanced models (Random Forest, XGBoost) and using cross-validation



**THANK YOU**