**Data science and ML**

* If data is normally distributed, use mean.
* If data is skewed, use median (less affected by outliers).
* For categorical data, use mode.
* **Predicting Movie Revenue**
  + Based on budget, genre, cast popularity, and release month.
* **Predicting Electricity Usage**
  + Based on time of day, temperature, and appliance usage.
* **Loan Approval Prediction**
  + Based on income, credit score, employment status, and loan amount.

**Movie Genre Classification**

* **Goal**: Predict the genre of a movie (Action, Comedy, Drama, etc.) based on description or poster.
* **Use Case**: Recommendation systems.

**6. Traffic Sign Recognition**

* **Goal**: Classify traffic signs into types like Stop, Yield, Speed Limit, etc.
* **Use Case**: Self-driving cars, driver assistance systems.

Predict component failure

[**Predicting Human Personality**](https://www.kaggle.com/code/rakeshkapilavai/predicting-human-personality)

|  |  |  |
| --- | --- | --- |
| **Predict component failure** | **Based on usage, temperature, vibration** | **Regression or Binary Classification** |
| **🗂️ Classify component type from image** | **Given a cropped image, classify as motor, light, etc.** |  |

**ApplicantIncome,CoapplicantIncome,Gender,Dependents,Education(Graduate/Not Graduate),Self\_Employed(yes,no),Credit\_History(good,bad,medium),Credit\_Score(number),age,Residence\_Type(own/rent),Collateral\_Offered(yes/no),Loan\_Purpose,LoanAmount,Loan\_Amount\_Term(months),loan\_status(approved(0),not approved(1)),**

|  |
| --- |
| Feature approval chance |
| Income more chance more  Co income more more  Gender doesn’t effect  Dependets more less chances  Graduate more  Employment yes more  Credit good more  Credit score good more  Age more depends on loan amount  and loan tenture less  Residences type(rent) less  Collateral offered(yes) more  Loan purpuso doesn’t effect  Loan amount depends on income employment  Loan tenture depends of loan amount and income  Loan status will based on above data |

That’s a great goal! If you want to be able to build a machine learning model **from start to finish** on your own, here’s a **step-by-step roadmap** you can follow. This will help you confidently handle any new dataset your teacher gives you — without needing help from anyone else.

**🧠 Step-by-Step Process to Build a Machine Learning Model**

**🔹 1. Understand the Problem**

* What is the goal? (e.g., Predict loan approval? Predict loan amount?)
* Is it a **classification** or **regression** problem?

**🔹 2. Load and Explore the Data**

* Use pandas to load the dataset.
* Use .head(), .info(), .describe() to understand the structure.
* Check for:
  + Missing values
  + Data types
  + Class imbalance (for classification)

**🔹 3. Clean the Data**

* Handle missing values:
  + Drop or impute (mean, median, mode, etc.)
* Fix data types (e.g., convert strings to categories)
* Remove duplicates if any

**🔹 4. Feature Engineering**

* Encode categorical variables:
  + Label Encoding or One-Hot Encoding
* Create new features if needed (e.g., income-to-loan ratio)
* Normalize or standardize numerical features

**🔹 5. Split the Data**

* Use train\_test\_split from sklearn to divide data into:
  + Training set
  + Testing set

**🔹 6. Choose a Model**

* For classification: Logistic Regression, Decision Tree, Random Forest, XGBoost, etc.
* For regression: Linear Regression, Random Forest Regressor, etc.

**🔹 7. Train the Model**

* Fit the model on the training data using .fit()

**🔹 8. Evaluate the Model**

* Use metrics:
  + Classification: Accuracy, Precision, Recall, F1 Score, Confusion Matrix
  + Regression: R² Score, MAE, RMSE
* Use .predict() on test data

**🔹 9. Tune Hyperparameters**

* Use GridSearchCV or RandomizedSearchCV to find the best parameters

**🔹 10. Save and Deploy (Optional)**

* Save model using joblib or pickle
* Deploy using Flask, Streamlit, or FastAPI (optional)

**🛠️ Tools You Must Learn**

| **Tool** | **Purpose** |
| --- | --- |
| pandas | Data loading and manipulation |
| numpy | Numerical operations |
| matplotlib / seaborn | Data visualization |
| sklearn | ML models, preprocessing, evaluation |
| xgboost / lightgbm | Advanced models |
| joblib / pickle | Save models |

**✅ Practice Plan**

1. **Pick a dataset** (like the one you uploaded)
2. **Follow the steps above**
3. **Repeat** with different datasets until it becomes second nature

Would you like me to generate a **template Python notebook** that includes all these steps using your dataset? You can use it as a reusable guide for any future project.