Objective & Data

The competition goal is to predict sale prices for homes. You're given a training and testing data set in csv format as well as a data dictionary.

Training: Our training data consists of 1,460 examples of houses with 81 features describing every aspect of the house. We are given sale prices (labels) for each house. The training data is what we will use to "teach" our models.

Testing: The test data set consists of 292 examples with the same number of features as the training data. Our test data set excludes the sale price because this is what we are trying to predict..

Task: Machine learning tasks are usually split into three categories; supervised, unsupervised and reinforcement. For this competition, our task is supervised learning.

Supervised learning uses examples and labels to find patterns in data

It's easy to recognise the type of machine learning task in front of you from the data you have and your objective. We've been given housing data consisting of features and labels, and we're tasked with predicting the labels for houses outside of our training data.

Tools

I used Python and Jupyter notebooks for the competition. Jupyter notebooks are popular among data scientist because they are easy to follow and show your working steps.

Please be aware this code is not for production purposes, it doesn't follow software engineering best practices. I've sacrificed that somewhat for explainability.

Libraries: These are frameworks in python to handle commonly required tasks. I Implore any budding data scientists to familiarise themselves with these libraries:

Pandas — For handling structured data

Scikit Learn — For machine learning

NumPy — For linear algebra and mathematics

Seaborn — For data visualization

Project Pipeline

Generally speaking, machine learning projects follow the same process. Data ingestion, data cleaning, exploratory data analysis, feature engineering and finally machine learning.

Data Cleaning.

Duplicates & NaNs: I started by removing duplicates from the data, checked for missing or NaN (not a number) values. It's important to check for NaNs (and not just because it's socially moral) because these cause errors in the machine learning models.

Categorical Features: There are a lot of categorical variables that are marked as N/A when a feature of the house is nonexistent. For example, when no alley is present. I identified all the cases where this was happening across the training and test data and replaced the N/As with something more descriptive. N/As can cause errors with machine learning later down the line so get rid of them.

Date Features: For this exercise dates would be better used as categories and not integers. After all, it's not so much the magnitude that we care about but rather that the dates represent different years. Solving this problem is simple, just convert the numeric dates to strings.

Decoded Variables: Some categorical variables had been number encoded. See the example below.

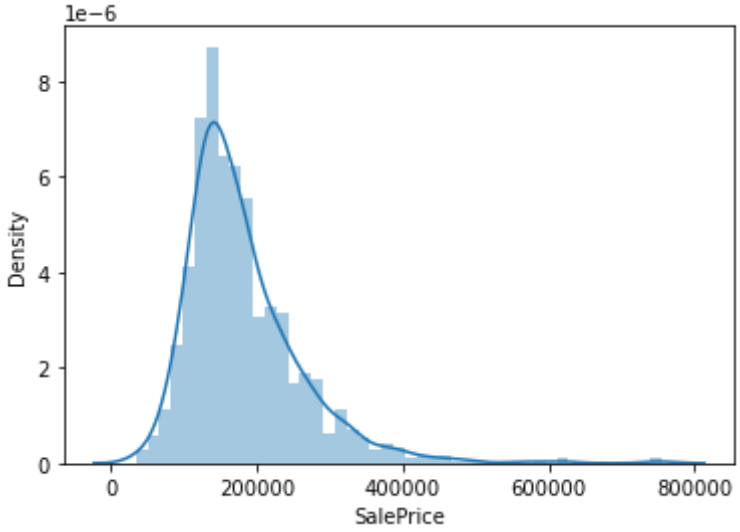
```
from sklearn.preprocessing import LabelEncoder
```

```
lb=LabelEncoder()

for i in df.columns:

  if df[i].dtypes=="object":

    df[i]=lb.fit_transform(df[i].values.reshape(-1,1))
```

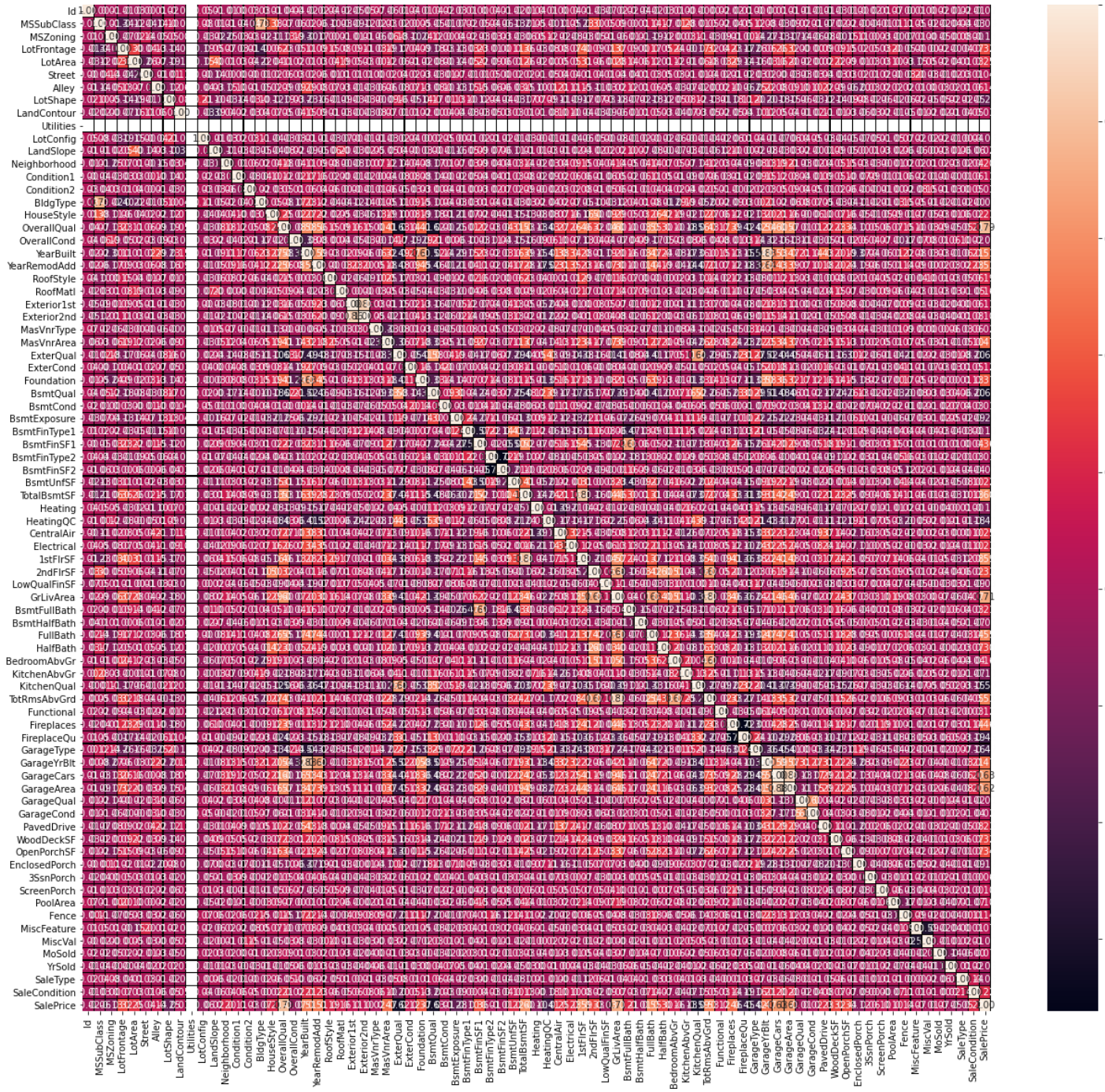
Exploratory Data Analysis (EDA)

This is where our data visualisation journey often begins. The purpose of EDA in machine learning is to explore the quality of our data.

Labels: I plotted sales price on a histogram. The distribution of sale prices is right skewed, something that is expected. In your neighborhood it might not be unusual to see a few houses that are relatively expensive.



Correlations: It's often good to plot a correlation matrix to give you an idea of relationships that exist in your data. It can also guide your model building. For example, if you see a lot of your features are correlated with each other you might want to avoid linear regression.
The correlation measure used here is Pearson's correlation. In our case the lighter the square the stronger the correlation between two variables.
Features related to space such as lot frontage, garage area, ground living area were all positively correlated with sale price as one might expect. The logic being that larger properties should be more expensive. No correlations look suspicious here.



Model Selection

As mentioned at the start of the article the task is supervised machine learning. We know it's a regression task because we are being asked to predict a numerical outcome (sale price).

Therefore, I approached this problem with three machine learning models. Decision tree, Random forest. I used the decision tree as my baseline model then built on this experience to tune my candidate models. This approach saves a lot of time as decision trees are quick to train and can give you an idea of how to tune the hyperparameters for my candidate models.

Model mechanics: I will not go into too much detail about how each model works here. Instead I'll drop a one-liner and link you to articles that describe what they do "under the hood".

Decision Tree — A tree algorithm used in machine learning to find patterns in data by learning decision rules.

Random Forest — A type of bagging method that plays on 'the wisdom of crowds' effect. It uses multiple independent decision trees in parallel to learn from data and aggregates their predictions for an outcome.