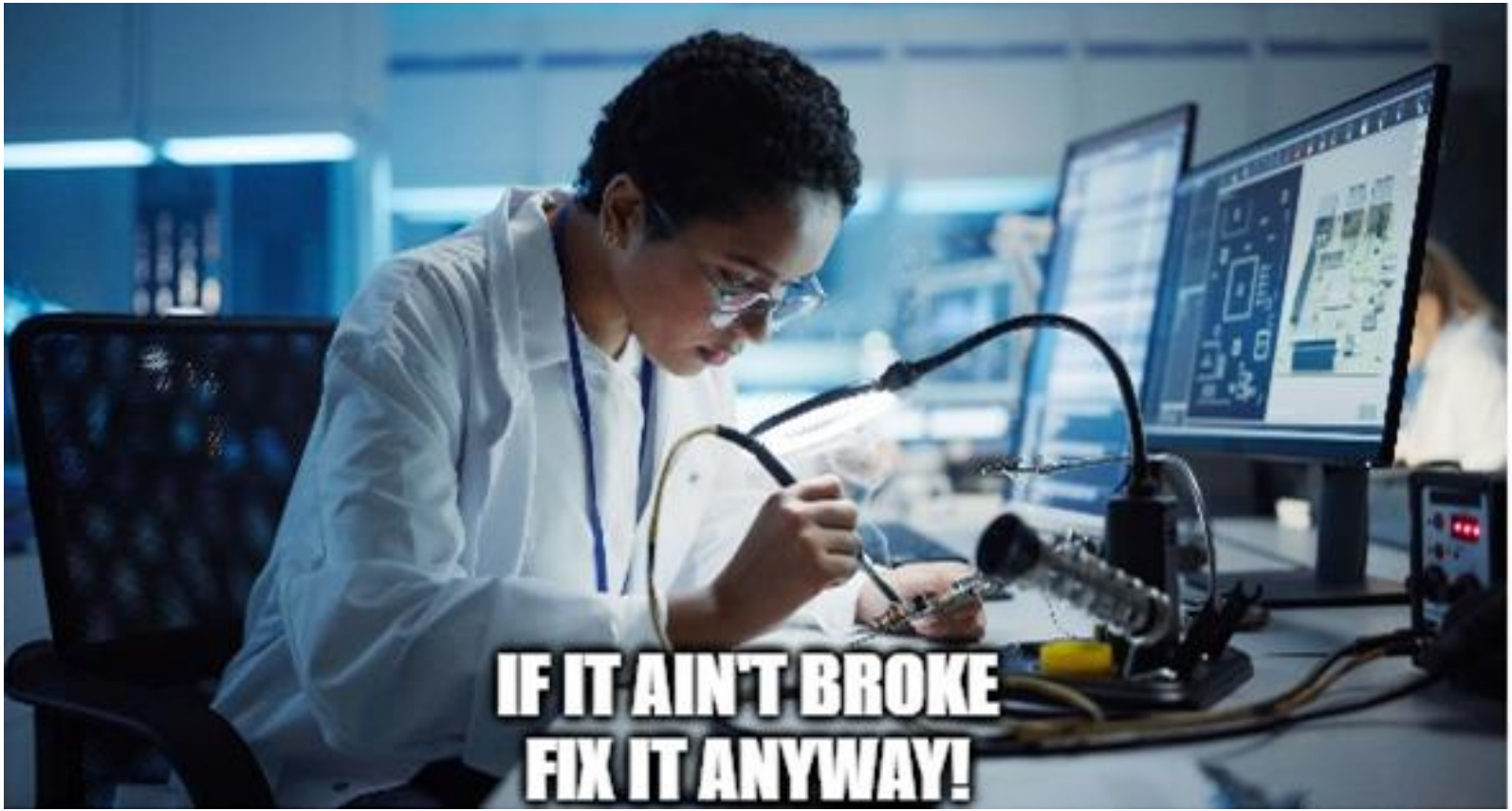



Image Source

# Welcome to Week 10 Lecture 2!

Data Science in Python & Machine Learning

# Review: What's wrong with this code?

There are 7 mistakes.  Use the annotation tool to circle one, type one chat, or say one outloud

```
from sklearn.decomposition import PCA
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
import pandas as pd

df = pd.read_csv('/content/data.csv')
X = df.drop(columns='target')
y = df['target']
X_train, X_test, y_train, y_test = train_test_split(X, y, random_state=42)

pca_pipe = make_pipeline(StandardScaler(), PCA(n_components=.95))
X_train_proc = pca_pipe.fit_transform(X_train)
X_test_proc = pca_pipe.transform(X_test)
```

# Learning Objectives

1. Identify features for engineering
2. Select appropriate engineering strategies
3. Create non-linear feature combinations with PolynomialFeatures
4. Apply feature engineering to a dataset to improve model performance

# Feature Engineering

- Changes features in some way
- Unlock information in your data
- Allow your model to look at data in new ways.


Image thank to slaughterdbc

# Feature Engineering Review

| Engineering Skill | Application |
|---|---|
| Scaling | StandardScaler, MinMaxScaler |
| Encoding | OneHotEncoder, OrdinalEncoder |
| Dimensionality Reduction | PCA, LDA |
| Overloaded Operators | col1 + col2, col2 - col2, col1 * col2, col1**2 |
| String Operators | col1.str.split(), col1.str.strip(), col1 + ‘ ‘ + col2 |
| Datetime | col.dt.hour, col.dt.day_name(), col.dt.month |
| Apply Functions | col.apply(lambda x: 1 if x  > 50 else 0) |

# Overloaded Operator Ideas

**Combine Features**

- df['bed_bath_ratio'] = df['beds'] / df['baths']
- df['total_candy'] = df['chocolate'] \
  + df['bubblegum'] \
  + df['taffy']

**Transform Features**

- df['squared_latitude'] = df['latitude']**2
- df['sqrt_income'] = df['income']**.5

# Datetime Ideas

**Extract More Information**

- df[‘month’] = df[‘datetime’].dt.month_name()
- df[‘hour’] = df[‘datetime’].dt.hour

# New Ideas:

**Binning**

- Group numeric into ranges
- Combine categories
- Change from regression classification
- Change from multiclass to binary

**Polynomial Encoding**

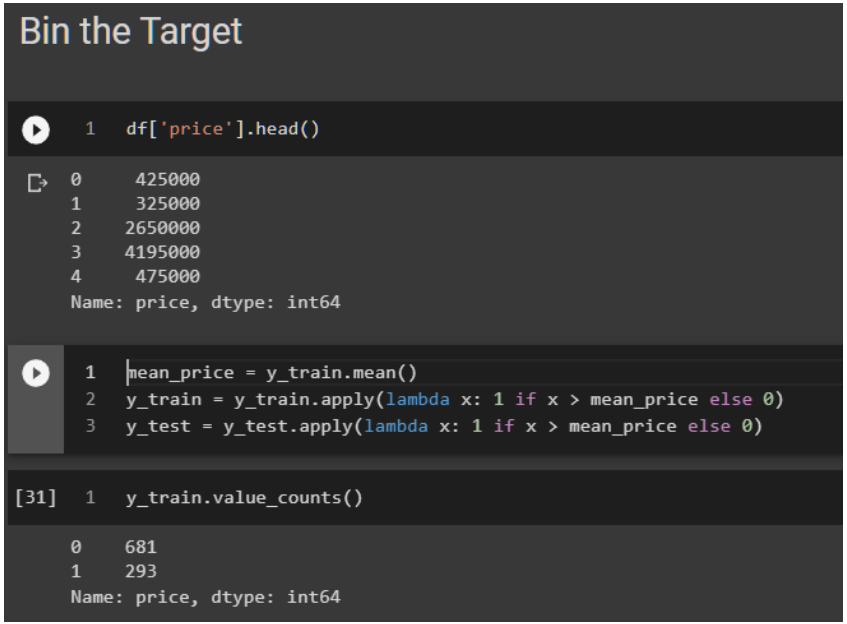
- Numeric features
- Adds products and powers of features as new feature
- Makes numeric columns non-linear

# Binning: What it does

- Combines numeric ranges or groups of categories into or bins or combination categories
  - Think histograms

Uses:

- Change regression to classification, which may be easier to model
- Change multiclass classification to binary classification, which may be easier to model
- Reduce cardinality of features



# Binning: What it does

**Use .apply() to transform features (or targets!)**

```
df['is_mammal'] = df['animal'].apply(lambda x: 1 if x in ['dog', 'cat', 'horse'] else 0)
```

Old Categories →

New Categories →

Mammal (1)

Non-mammal (0)

# Binning: What it does

**Use .apply() to transform features (or targets!)**

```
df['is_tall'] = df['height'].apply(lambda x: 1 if x > 72 else 0)
```

# PolynomialFeatures: What it does

| price | distance_travelled(kms) | price^2 | price distance_travelled(kms) | distance_travelled(kms)^2 | price^3 | price^2 distance_travelled(kms) | price distance_travelled(kms)^2 | distance_travelled(kms)^3 |
|---|---|---|---|---|---|---|---|---|
| -0.613742 | -1.823289 | 0.376679 | 1.119030 | 3.324384 | -0.231184 | -0.686796 | -2.040315 | -6.06131 |
| -0.671885 | 1.275503 | 0.451430 | -0.856992 | 1.626909 | -0.303309 | 0.575800 | -1.093096 | 2.07512 |
| 0.679942 | 0.519931 | 0.462321 | 0.353523 | 0.270328 | 0.314352 | 0.240375 | 0.183807 | 0.14055 |
| 1.578253 | -0.651936 | 2.490883 | -1.028920 | 0.425021 | 3.931244 | -1.623897 | 0.670791 | -0.27708 |
| -0.584671 | -0.712664 | 0.341840 | 0.416674 | 0.507890 | -0.199864 | -0.243617 | -0.296948 | -0.36195 |

- Adds new numeric features
- Products and powers of original features

# PolynomialFeatures: What it does

**Pros:**

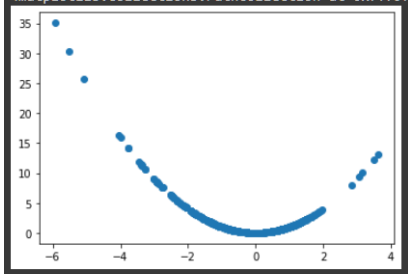
- Makes linear features non-linear
- Improves the power of linear models
- Increases model complexity

**Cons**

- Increased model complexity with more features
- Higher degrees can create overfitting

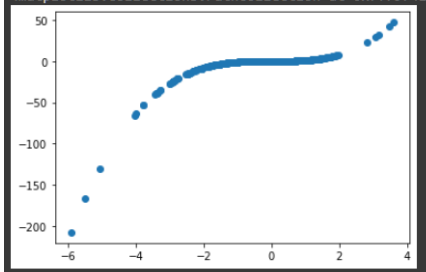
```
1  # Plot distance vs distance^2
2  plt.scatter(data=poly_train, x='distance_travelled(kms)', y='distance_travelled(kms)^2')
```

```
<matplotlib.collections.PathCollection at 0x7f79714f4590>
```




```
1  # Plot distance vs distance^2
2  plt.scatter(data=poly_train, x='distance_travelled(kms)', y='distance_travelled(kms)^3')
```

```
<matplotlib.collections.PathCollection at 0x7f79741b4e50>
```



# Feature Engineering Poll

**Which feature engineering strategies would you use?**

# Feature Engineering Code-along

Code-along Notebook

You can use the link at the top to open the notebook in Colab if desired

# Coding Challenge: Feature Engineering

In your breakout group:

1. Challenge Notebook. You will be predicting house sales in Melbourne, Australia
2. Open the notebook either locally or in Colab
3. Quickly choose a driver to code and share their screen.
4. For each section, choose a feature engineering technique to try
5. If there is time, fit a model on your resulting data and be ready to share your score.

# Assignments Due Friday Morning

- PCA Exercise
- Feature Engineering Exercise
- Project 2 - Part 4

I will be reviewing assignments Friday, Not Monday this week.  Just sayin’

# Announcements:

- Belt Exams are the week after Thanksgiving: December 2nd - 4th
  - Make sure you are caught up on assignments!
  - Content from weeks 9-11 will be on the belt exam
    - Clustering
    - PCA
    - Neural network models
  - Belt Prep and Practice Exam on LP

# Announcements:
# Special Belt Exam Code Reviews

- Next Week's Code Reviews will be devoted to Belt Exam Prep!!

    - Review content
    - Find your weaknesses
    - Create a study plan

Code Review Sign Up

# Week 3 is a Big Week!
# Start Early...like Friday Night!

- Catch up on Weeks 1 and 2
- Week 3 content
- Week 3 assignments
- Finalize Project 2 Presentation
- Prepare for Stack 3 Belt Exam

Daily Schedule

# Next Lecture: Intro to Deep Learning

Please Read:

- Intro to Deep Learning
- Forward Propagation
- Activation Functions
- Backward Propagation
- Neural Networks in Keras