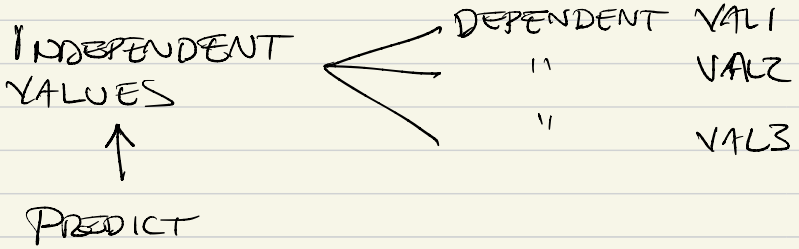
MODEL DEVELOPMENT

- SIMPLE AND MULTIPLE LINEAR REGRESION
- MODEL EVALUATION USING VISUALIZATION
  - POLINOMIAL REGRESION AND PIPELINES
  - R-SQUARED AND MSE IN-SAMPLE EVALUATION
  - PREDICTION AND DECISION MAKING

MODEL ESTIMATOR



LINEAR

1 INDEPENDEN VARIABLE

MULTIPLE LINEAR

2 OR MORE INDEPENDENT VARIABLES

SINGLE LINEAR REGRESION

$$y = b_0 + b_1 X$$

INTERCEPT SLOPE

A LOT OF MATHEMATHICS TO DETERMINE THESE VALUES

DATA IN NUMPY ARRAYS
X AND Y

PROCESS

DATAPOINTS → FIT → (b_0, b_1)

PREDICT

$\hat{y} = b_0 + b_1 X$

```
from sklearn.linear_model
    import LinearRegression
lm = LinearRegression()
X = df["indVar"]
y = df["depVar"]
lm.fit(X,y) #Find b0 and b1
yhat = lm.predict(X)
```

lm.intercept
lm.coef

MULTIPLE LINEAR REGRESION

RELATIONSHIP BETWEEN
ONE CONTINUOUS TARGET AND
TWO OR MORE PREDICTOR VARIABLES

$$\hat{y} = \underset{b_0}{1} + \underset{b_1}{2}X_1 + \underset{b_2}{3}X_2$$

```
lm.intercept
lm.coef #array([2,3])
```

MODEL EVALUATION USING VISUALIZATION

1. The relationship between 2 variables
2. The strength of the correlation
3. The direction of the correlation (+ or -)

REGRESSION PLOT

- Scatter plot
- Fitted linear regression line ($\hat{y}$)

```
import seaborn as sns
sns.regplot(x="indVar", y="depVar", data=df)
plt.ylim(0,)
```

- DISTRIBUTED EVENLY AROUND THE X AXIS WITH SIMILAR VARIANCE SUGGESTS A LINEAR PLOT IS APPROPRIATE

- IF THE RESIDUALS ARE NOT RANDOMLY SEPARATED, THIS SUGGESTS THE LINEAR ASSUMPTION IS INCORRECT.

- IF THE VARIANCE INCREASES WITH X, THE MODEL IS INCORRECT.

RESIDUAL PLOT

```
import seaborn as sns

sns.resiplot(df["indVar"], df["depVar"])
                    X            Y
```

DISTRIBUTION PLOT

VISUALIZING MODELS WITH MORE THAN 1 INDEPENDENT VARIABLE

```
import seaborn as sns

ax1 = sns.displot(df["depVar"], hist=False,
                  color="r", label="Actual Value")

sns.displot(Yhat, hist=False, color="b",
            label="Fitted Values", ax=ax1)
```

POLYNOMIAL REGRESION
MORE THAN 1 DIMENSION

- TRANSFORM DATA INTO POLYNOMIAL
- USE LINEAR REGRESSION TO FIT THE PARAMETER

PR IS A SPECIAL CASE OF LR

USEFUL FOR DESCRIBING CURVILINEAR RELATIONSHIPS

↑ SQUARING OR SETTING A HIGHER ORDER

2ND ORDER

$$\hat{y} = b_0 + b_1 X_1 + b_2 X_1^2$$

3RD ORDER

$$\hat{y} = b_0 + b_1 X_1 + b_2 X_1^2 + b_3 X_1^3$$

```
f = np.polyfit(x, y, 3)
p = np.poly1d(f)
print(p)
```

WE CAN ALSO HAVE MULTIDIMENSIONAL POLYNOMIAL LINEAR REGRESSION

NUMPY'S POLYFIT FUNCTION CANNOT PERFORM THIS TYPE OF REGRESSION

HAVE TO USE PREPROCESSING LIB IN SCI-KIT

```
from sklearn.preprocessing import PolynomialFeatures
pr = PolynomialFeatures(degree=2, include_bias=False)
x_poly = pr.fit_transform(x[["hp", "cw"]])
```

BECAUSE OF MULTIPLE DIMENSIONS IT IS CONVENIENT TO SCALE THE FEATURES:

```
from sklearn.preprocessing import StandardScaler
SCALE = StandardScaler()
SCALE.fit(x_data[["hp", "hm"]])
x_scale = SCALE.transform(x_data[["hp", "hm"]])
```

THERE ARE MANY STEPS TO GETTING A PREDICTION:

NORMALIZATION → POLYNOMIAL TRANSFORM → LINEAR REGRESSION

USE A PIPELINE

PIPELINE

1. Import library Pipeline
2. Create a list of tuples
3. Input the list of steps in the pipeline constructor to create the pipeline object.
4. Apply the train method to the pipeline object.
5. Produce a prediction in the pipeline object.

```
Input = [("scale", StandardScaler()), ("polynomial",
          PolynomialFeatures(degree = 2)),
          "model", LinearRegression())]

pipe = Pipeline(Input)
```

MEASURES FOR IN-SAMPLE EVALUATION

A WAY TO NUMERICALLY DETERMINE HOW GOOD THE MODEL FITS ON DATASET

- COEFFICIENT OF DETERMINATION R^2
- MEAN SQUARE ERROR MSE

MSE

```
from sklearn.metrics import mean_squared_error
mean_squared_error(df['price'], Y_predict_simple_fit)
```

↑ ACTUAL VALUE OF TARGET VARIABLE

↑ PREDICTED VALUE OF TARGET VAR

R^2

THE % OF VARIATION OF THE TARGET VAR(Y) THAT IS EXPLAINED BY THE LINEAR MODEL.

$R^2 \approx 1$ GOOD FIT

$R^2 \approx 0$ BAD FIT

} CLOSER TO 1 INDICATES A BETTER FIT

```
X = df[["highway-mpg"]]
y = df['price']
lm.fit(X, y)
lm.score(X, y)
```

$R^2 < 0$ OVERFITTING

EXAMPLE

$R^2 = 0.49$

MEANS APROX 49% OF THE VARIATION OF PRICE IS EXPLAINED BY THIS SIMPLE LINEAR MODEL.

PREDICTION AND DECISION MAKING

HOW CAN WE DETERMINE IF OUR MODEL IS CORRECT?

1. DO THE PREDICTED VALUES MAKES SENSE?
2. VISUALIZATION
3. NUMERICAL MEASURES FOR EVALUATION
4. COMPARING MODELS

Ex.

```
lm.fit(x,y)
lm.predict(value)   # HOW THE PREDICTED VALUE LOOKS LIKE?. IN BETWEEN A REASONABLE RANGE?
```

TO GENERATE A SEQUENCE OF VALUES FOR TESTING THE MODEL:

```
import numpy as np
new_input=np.arange(1,101,1).reshape(-1,1)
```

STARTING — ENDING +1 — STEP SIZE

```
yhat=lm.predict(new_input)
```

USING A REGRESSION PLOT SHOULD BE TRIED
RESIDUAL PLOT
DISTRIBUTION PLOT

SHOULD BE

- SOME AUTHORS SUGGEST $R^2 \geq 0.1$
- IS A LOWER MSE ALWAYS IMPLYING A BETTER FIT? NOT NECESSARILY

$MSE_{MLR} < MSE_{SLR}$

$MSE_{PR} < MSE_{LR}$

} A SIMILAR INVERSE RELATIONSHIP HOLDS FOR R^2