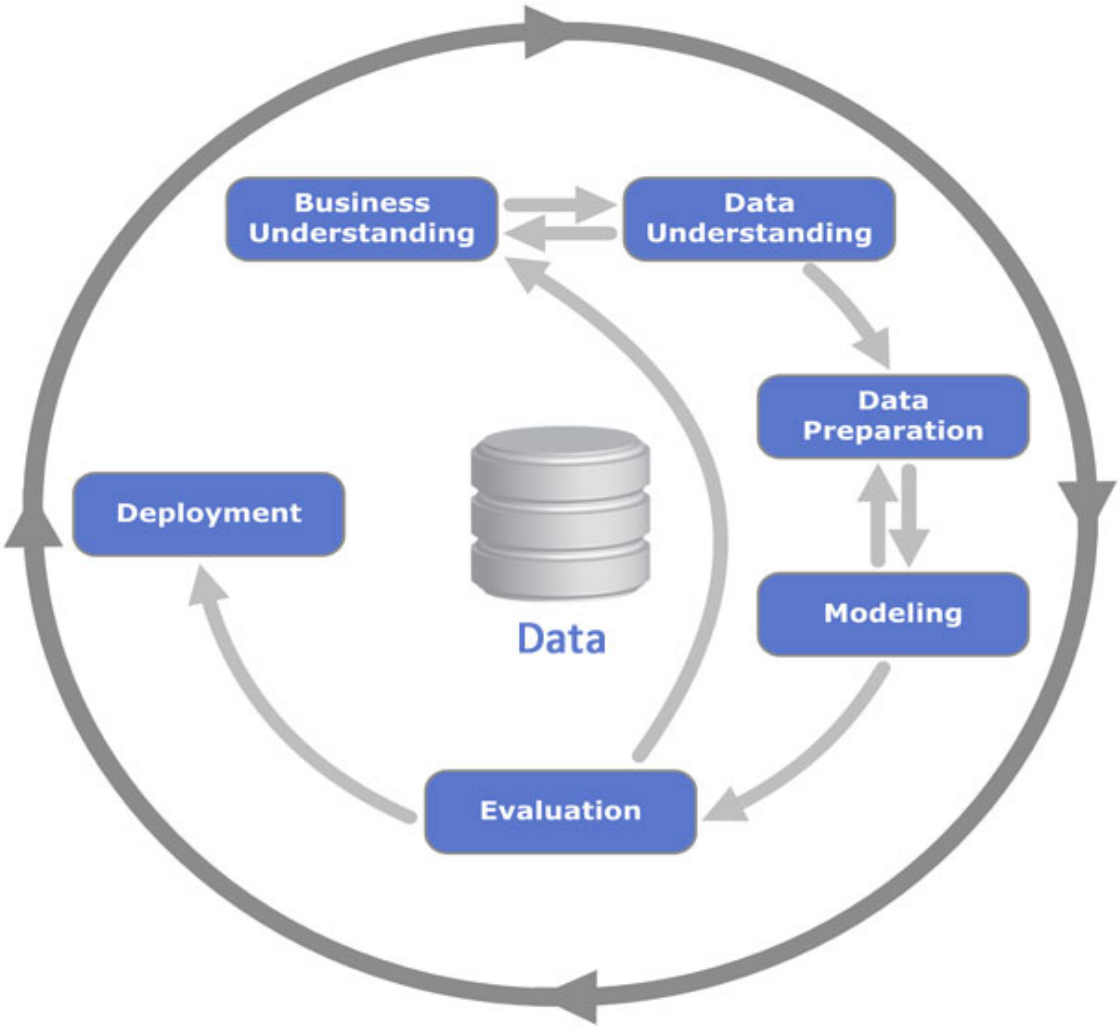
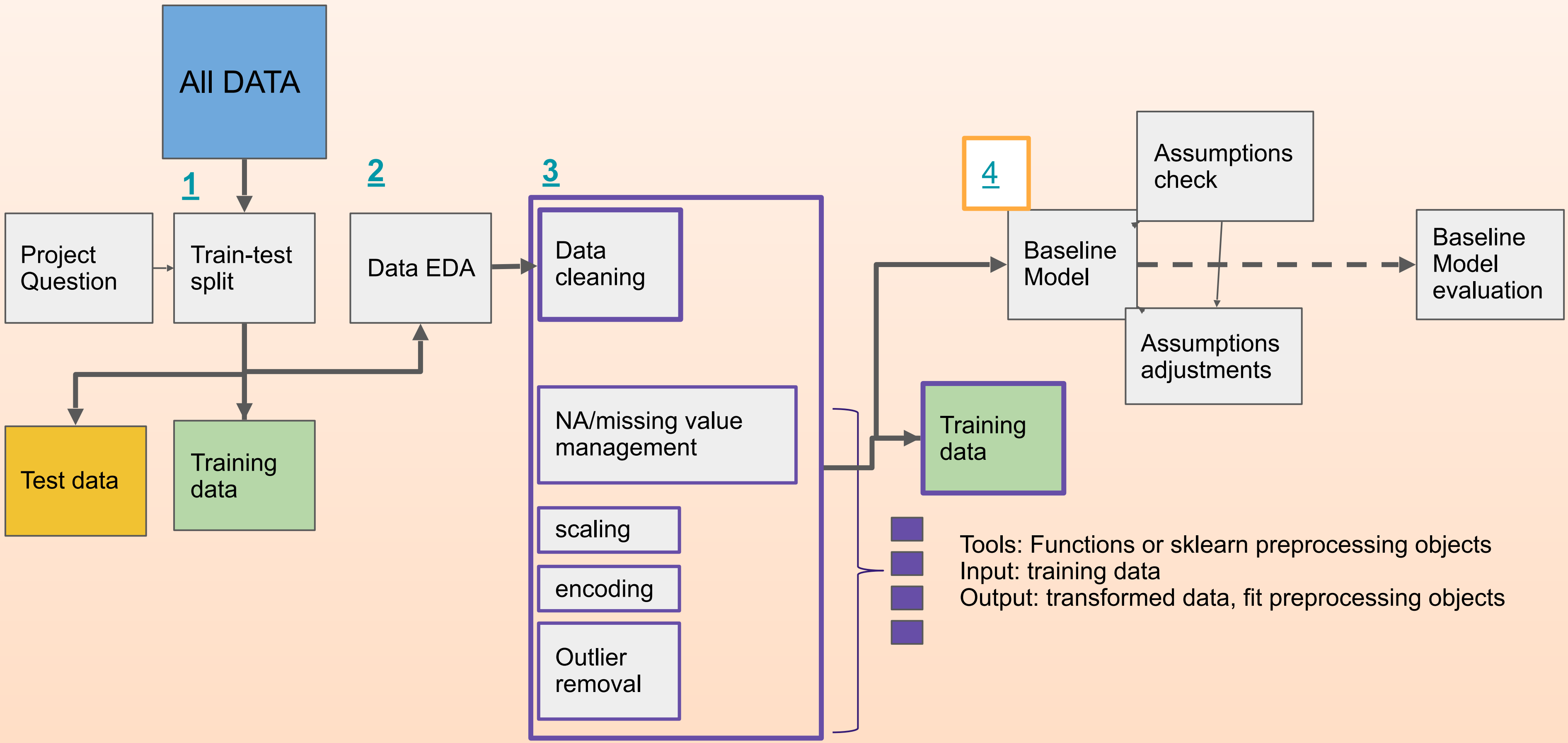
Linear Modeling Workflow

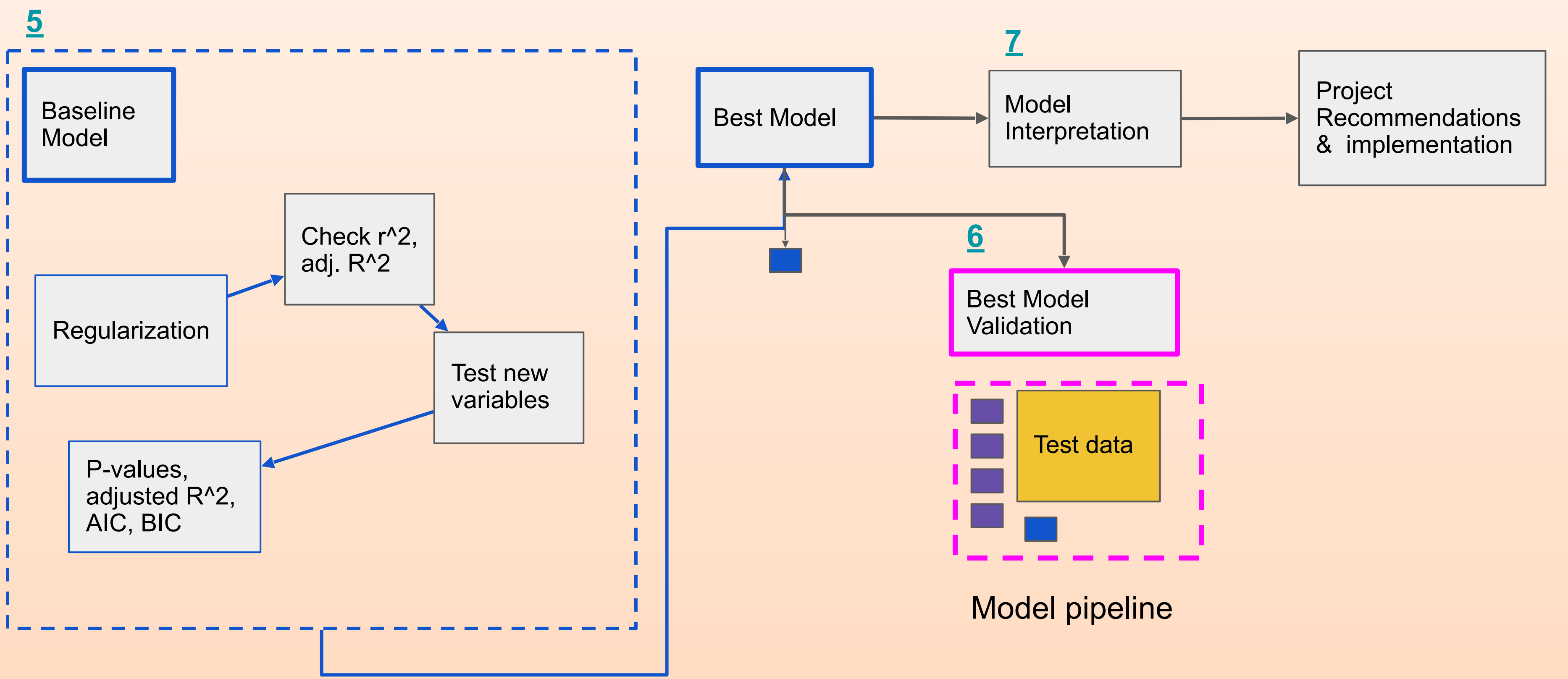
CRISP-DM Process Diagram



Source: Kenneth Jensen

Take 3 minutes with your neighbor and diagram a modeling workflow


All DATA
1
2
3
4
Project Question
Train-test split
Data EDA
Data cleaning
NA/missing value management
scaling
encoding
Outlier removal
Baseline Model
Assumptions check
Assumptions adjustments
Baseline Model evaluation
Test data
Training data
Training data
Tools: Functions or sklearn preprocessing objects
Input: training data
Output: transformed data, fit preprocessing objects


5
Baseline Model
Regularization
Check r^2, adj. R^2
Test new variables
P-values, adjusted R^2, AIC, BIC
Best Model
7
Model Interpretation
Project Recommendations & implementation
6
Best Model Validation
Test data
Model pipeline

1: First Thing's First: Train-Test split

```
from sklearn.model_selection import train_test_split
```

X_train, X_test, y_train, y_test = train_test_split(y, X, test_size = ?, random_state=42)

Split Ratio:
20 - 25%

2: Exploratory Data Analysis (EDA)

Inspect

- Target variable distribution
    - Normality/Skew
    - Make sure target is continuous

- Features
    - Distributions (normality can improve performance but isn't mandatory)
    - Trends
    - Data type
        - Categorical vs. numerical

Watch out for:

- Collinearity
- Outliers
- Null, missing values, n/a values

Useful methods and attributes

- df.shape
- df.describe()
- df.info()
- df.dtypes()
- df.isna().sum()

- sns.pairplot()
- df.hist()/sns.distplot()
- Boxplots/Violinplots (plt/sns)
- sns.regplot()
- plt.scatter()

- df.corr()
- sns.heatmap()

3: Data cleaning/Preprocessing

- Missing Value
  - Dropna: .dropna()
  - Fillna: .fillna()
  - Sklearn: imputation: SimpleImputer(‘mean’, ‘median’, ‘most frequent’)

- Datatypes
  - Check data types, df.info()/df.dtypes
  - pd.to_datetime(), pd.to_numeric() , str.replace

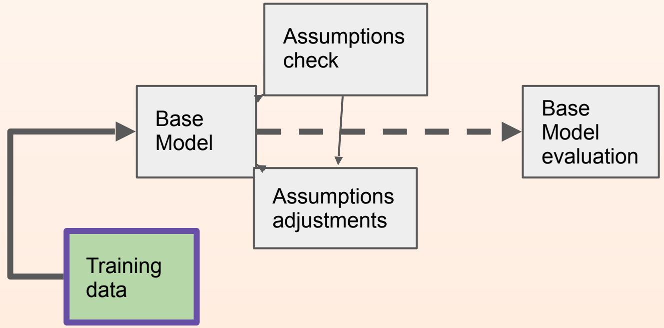
- Categorical/ Encoding
  - sklearn.preprocessing: onehotencoder / get_dummies / Binarizer / LabelEncoder

- Scaling/Normalization
  - Minmax scaling
  - StandardScaling
  - Mean normalization
  - Must scale for regularization to work effectively.

Knowledge Check

Take two minutes and write down the assumptions of linear regression?

4: Base Model Assumption Check

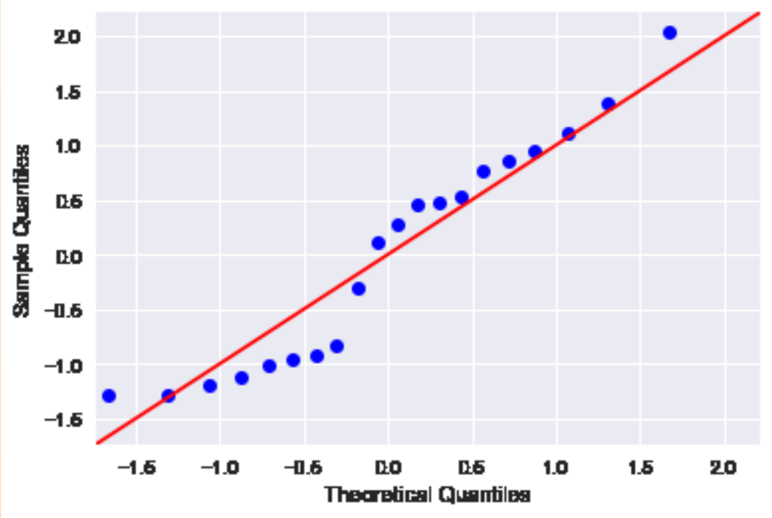


Assumptions:

- **Linearity**: there is a linear relationship between dependent and independent variables
- **Normality**:Residuals are normally distributed
- **Homoscedasticity**: Residuals are evenly distributed around zero and show no visible patterns.
- **Feature independence (no covariance)**
- **No autocorrelation (in linear model)**

Things to use while checking:

- Linearity: sns.pairplot()
- Normality: Histogram, Q-Q plot
- Homoscedasticity: Scatterplots
- covariance: sns.pairplot(), df.corr(), heat maps



5: Modeling - Statsmodels

```
import statsmodels.api as sm
from statsmodels.formula.api import ols
predictors_for_formula  = '+'.join(boston_features.columns)
formula = 'y' + '~' + predictors_for_formula
model = ols(formula, data = boston_data).fit()
model.summary()
```

Outputs: features, p-values,   R^2, adj. R^2, AIC, BIC

5: Modeling: Sklearn

```python
from sklearn.linear_model import LinearRegression, Lasso, Ridge
lr = LinearRegression()
lr.fit(X_train, y_train)
```

lr = LinearRegression()
lr.fit(X_train, y_train)
lr.coef_()
lr.intercept_()
y_hat = lr.predict(X_test)

From sklearn.linear_model import Ridge, Lasso

- lasso = Lasso(alpha)
  - alpha = regularization parameter
  - lasso.fit(), lasso.coef_(), lasso.intercept_(), lasso.predict()
    -same as linear
- ridge = Ridge(alpha)
  - alpha = regularization parameter
  - ridge.fit(), ridge.coef_(), ridge.intercept_(), ridge.predict()
    -same as linear

```python
from sklearn.metrics import mean_squared_error, r2_score
```

- mean_square_error(y_hat, y_test)
- r2_score(y_hat, y_test)

5: Modeling: Transformations

```python
from itertools import combinations
```

Interactions

Polynomials

```python
from sklearn.preprocessing import PolynomialFeatures
```

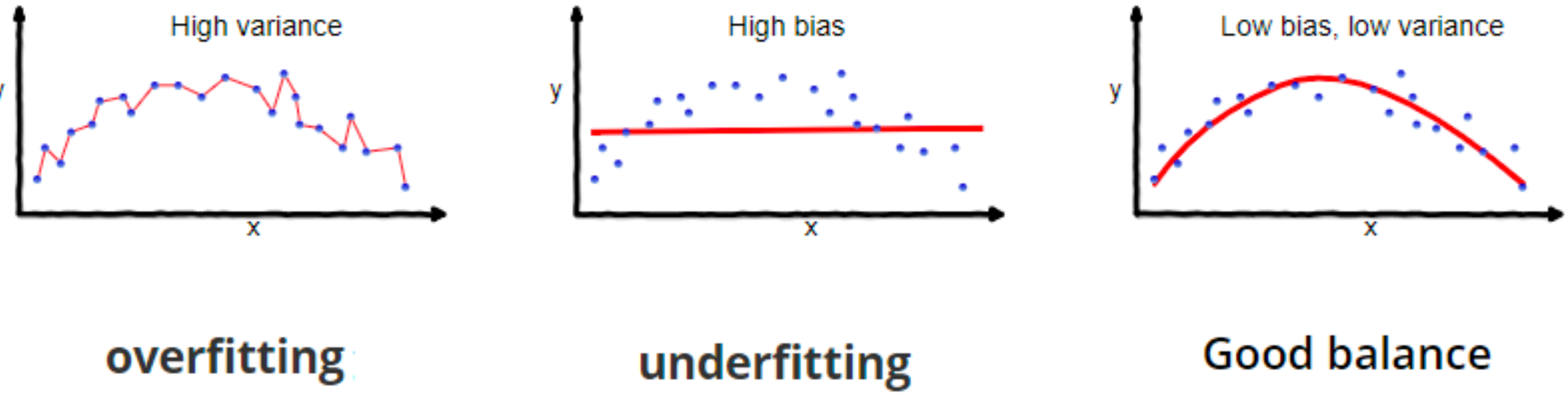
6 - Best Model Validation

To avoid errors while working with the test dataset:

- **Blackbox it!**
  - Make the final parts of steps 3-5 (all data cleaning, regularizing and imputing, transformations) into a single function so you don't skip anything on the testing

- Don't use `.fit()`! (you shouldn't be fitting any of your model to the test data)

Bias-Variance Tradeoff



Compare R^2 for training and test
High train R^2 + low test R^2 = ?
Low train R2 + low test R^2 = ?

https://towardsdatascience.com/understanding-the-bias-variance-tradeoff-165e6942b229