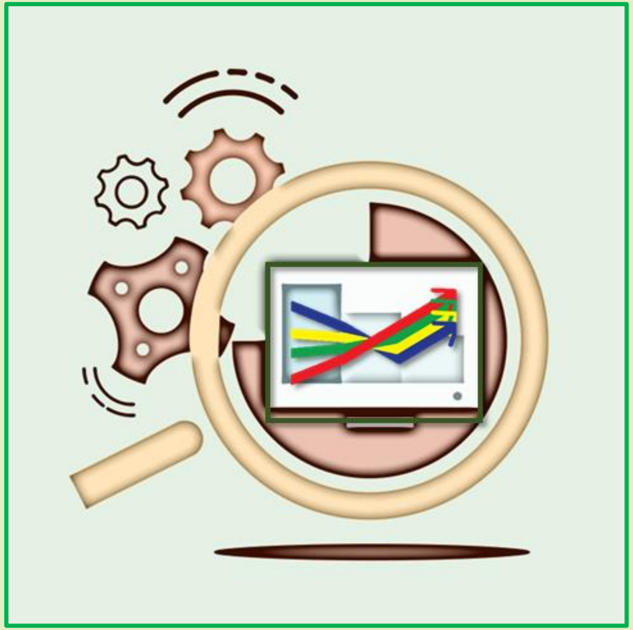
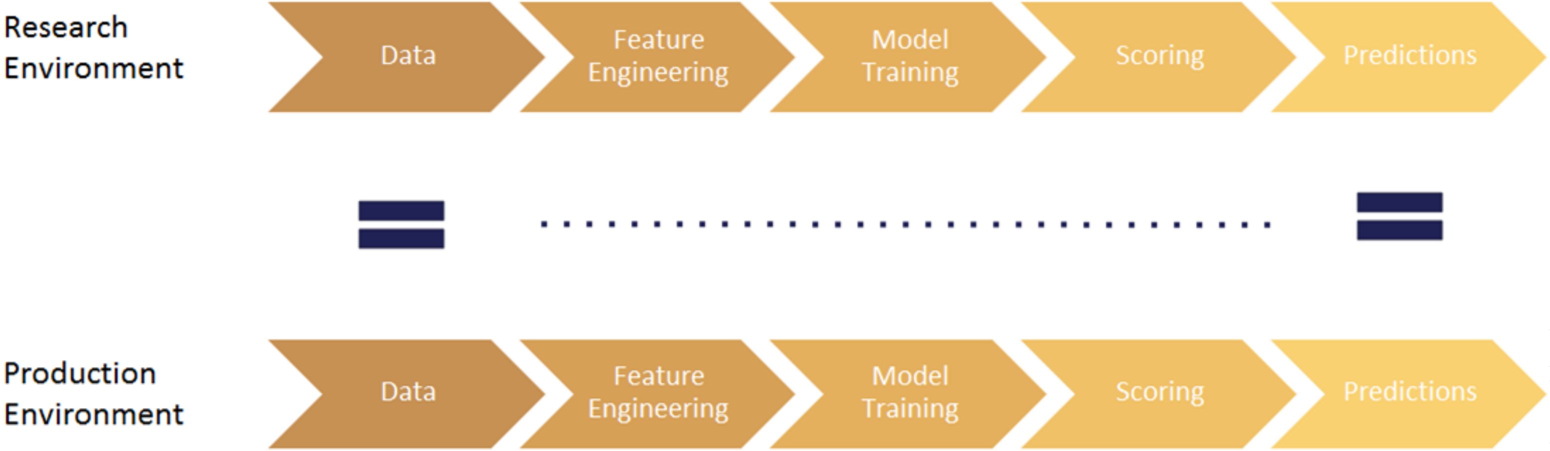
Train In Data

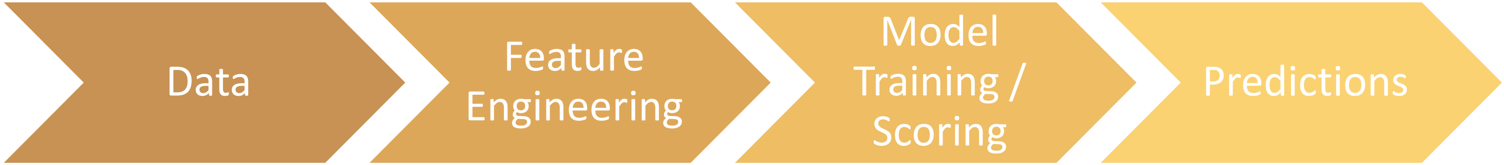


Streamlining Model Deployment with Open-Source

Reproducible ML Pipelines



Machine Learning Pipeline



Data

Feature Engineering

Model Training / Scoring

Predictions

- Missing data imputation
- Categorical encoding
- Discretisation
- Transformation
- Feature extraction
  - Text
  - Datetime
  - Images
  - Time series
- Feature combination

- **Lots to code!**

- Linear Models
- Decision trees
- KNNs
- SVMs
- Neural Networks

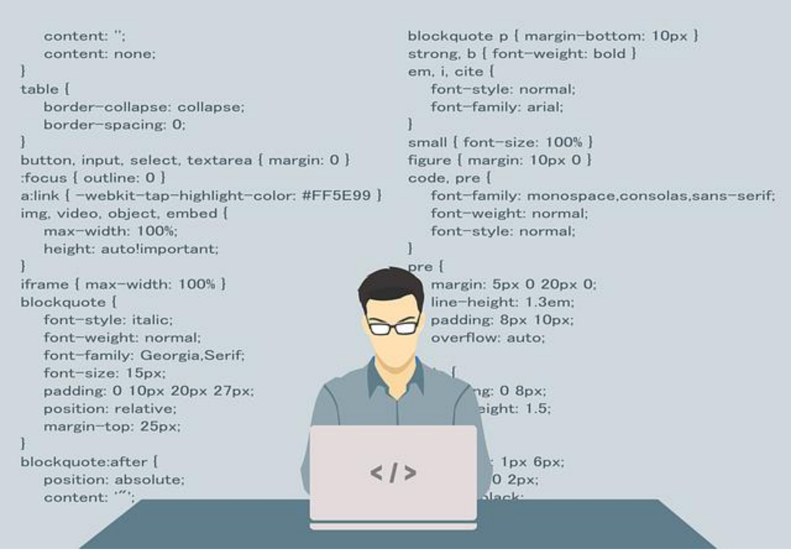
- Stacking

- **Lots of parameters ➔ Reproducibility**

Challenges

- A lot to code
- Repetitive
- Learn and store parameters
- Reproducibility

Lots to code



```
    content: '';
    content: none;
}
table {
    border-collapse: collapse;
    border-spacing: 0;
}
button, input, select, textarea { margin: 0 }
:focus { outline: 0 }
a:link { -webkit-tap-highlight-color: #FF5E99 }
img, video, object, embed {
    max-width: 100%;
    height: auto!important;
}
iframe { max-width: 100% }
blockquote {
    font-style: italic;
    font-weight: normal;
    font-family: Georgia,Serif;
    font-size: 15px;
    padding: 0 10px 20px 27px;
    position: relative;
    margin-top: 25px;
}
blockquote:after {
    position: absolute;
    content: '"';
```

```
blockquote p { margin-bottom: 10px }
strong, b { font-weight: bold }
em, i, cite {
    font-style: normal;
    font-family: arial;
}
small { font-size: 100% }
figure { margin: 10px 0 }
code, pre {
    font-family: monospace,consolas,sans-serif;
    font-weight: normal;
    font-style: normal;
}
pre {
    margin: 5px 0 20px 0;
    line-height: 1.3em;
    padding: 8px 10px;
    overflow: auto;

      {
       ng: 0 8px;
       eight: 1.5;

       1px 6px;
       0 2px;
```

- Time consuming
- Different versions across team

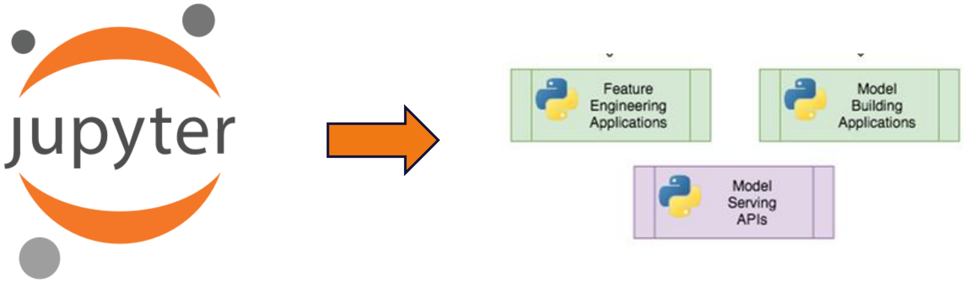
Repetitive

- Multiple copies of same code
- Different versions of same code
- Difficult to keep track

Learn and store parameters

- Multiple intermediate files with parameters
- Config or params file

Reproducibility in Deployment

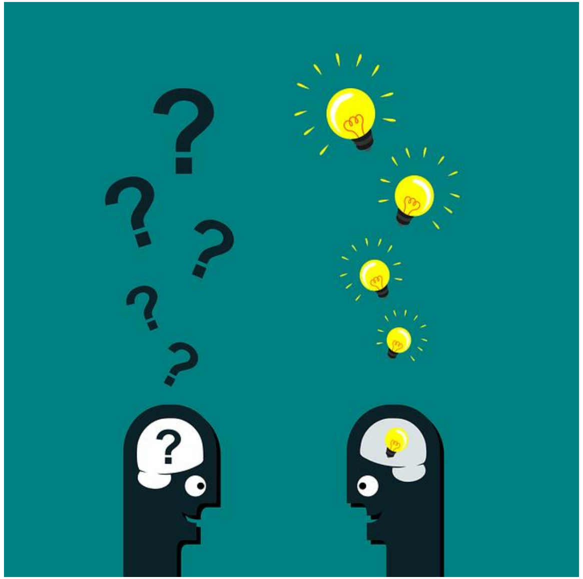


Research Environment

Production Environment

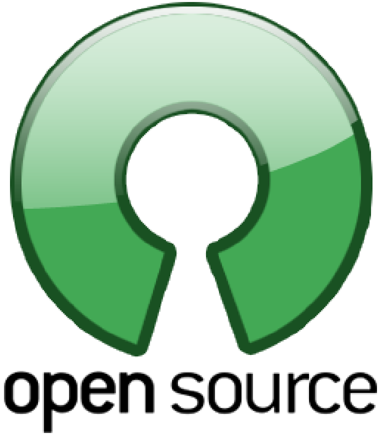
- Re-write code
- Include tests
- Reproducibility

Team Performance



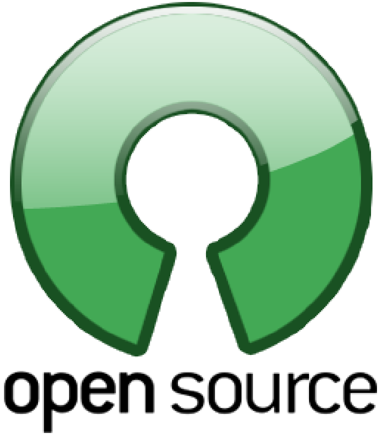
- Decreased Performance
- Frustration
- Lack of reproducibility
- Increased deployment times

Open-source



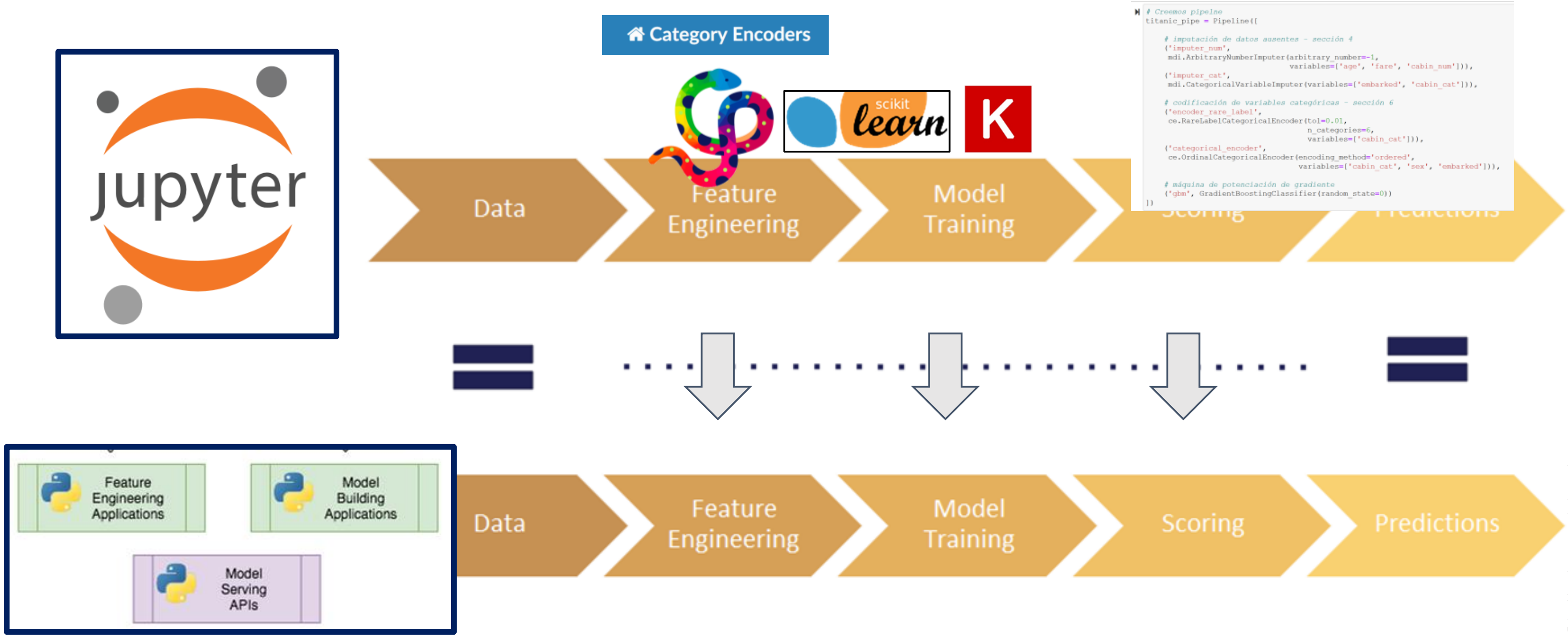
- Increase Performance
- Prevent Frustration
- Maximise reproducibility
- Minimise deployment times

Open-source



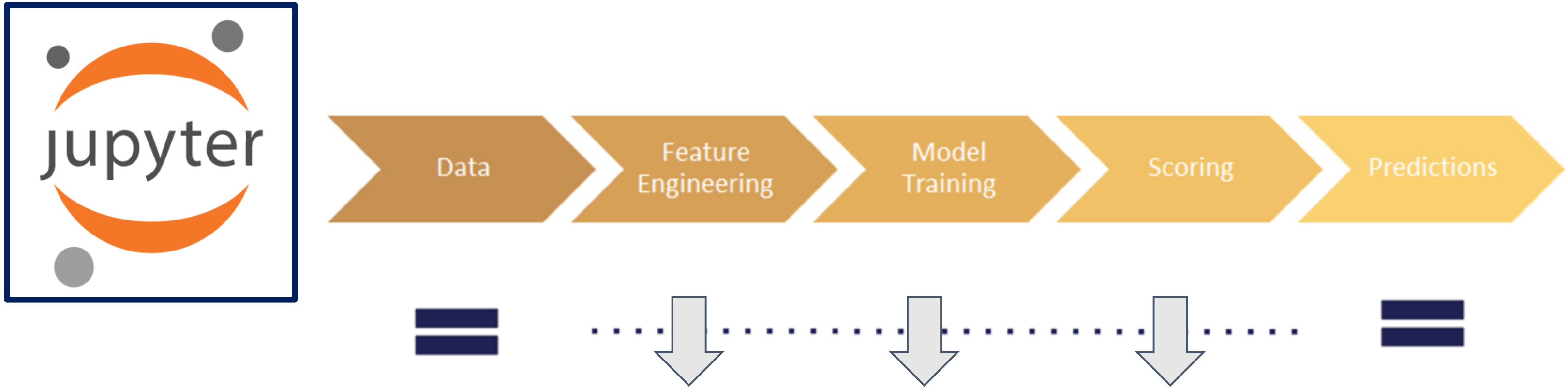
- No more coding
- Version tracking for reproducibility
- Classes and functions include tests – no need to recode for production

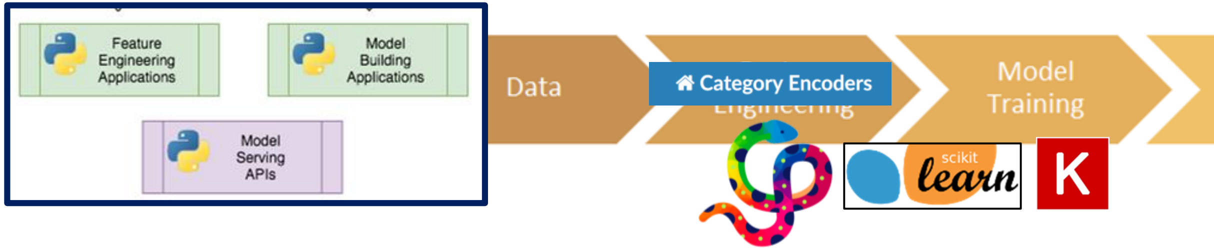
Reproducible ML Pipelines

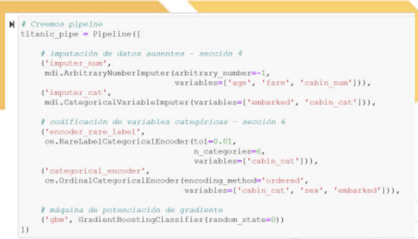


```
# Creemos pipelne
titanic_pipe = Pipeline([

    # imputación de datos ausentes - sección 4
    ('imputer_num',
     mdi.ArbitraryNumberImputer(arbitrary_number=-1,
                                variables=['age', 'fare', 'cabin_num'])),
    ('imputer_cat',
     mdi.CategoricalVariableImputer(variables=['embarked', 'cabin_cat'])),

    # codificación de variables categóricas - sección 6
    ('encoder_rare_label',
     ce.RareLabelCategoricalEncoder(tol=0.01,
                                    n_categories=6,
                                    variables=['cabin_cat'])),
    ('categorical_encoder',
     ce.OrdinalCategoricalEncoder(encoding_method='ordered',
                                  variables=['cabin_cat', 'sex', 'embarked'])),

    # máquina de potenciación de gradiente
    ('gbm', GradientBoostingClassifier(random_state=0))
])
```

Reproducible ML Pipelines







```python
# Creemos pipelne
titanic_pipe = Pipeline([

    # imputación de datos ausentes - sección 4
    ('imputer_num',
     mdi.ArbitraryNumberImputer(arbitrary_number=-1,
                                variables=['age', 'fare', 'cabin_num'])),
    ('imputer_cat',
     mdi.CategoricalVariableImputer(variables=['embarked', 'cabin_cat'])),

    # codificación de variables categóricas - sección 6
    ('encoder_rare_label',
     ce.RareLabelCategoricalEncoder(tol=0.01,
                                    n_categories=6,
                                    variables=['cabin_cat'])),
    ('categorical_encoder',
     ce.OrdinalCategoricalEncoder(encoding_method='ordered',
                                  variables=['cabin_cat', 'sex', 'embarked'])),

    # máquina de potenciación de gradiente
    ('gbm', GradientBoostingClassifier(random_state=0))
])
```