Deliverables Implementation

1. Interactive Analysis Notebooks

The Jupyter notebooks will include:

Executive Summary Sections: Markdown cells with business-friendly interpretations of key findings

Technical Analysis: Detailed code and statistical validation as shown above

Interactive Visualizations: Using Plotly for zooming, filtering capabilities

Methodology Documentation: Explanation of statistical tests and assumptions

2. Statistical Analysis Report

Key components:

Hypothesis Testing Summary Table:

Null and alternative hypotheses

Test statistics

p-values

Effect sizes

Confidence intervals

Effect Size Calculations:

Cohen’s d for continuous variables

Odds ratios for categorical variables

95% confidence intervals for all estimates

Power Analysis:

python

from statsmodels.stats.power import TTestIndPower

effect\_size = (won\_deals[’deal\_value’].mean() - lost\_deals[’deal\_value’].mean()) / np.sqrt(

(won\_deals[’deal\_value’].std()\*\*2 + lost\_deals[’deal\_value’].std()\*\*2) / 2

)

power\_analysis = TTestIndPower()

power = power\_analysis.solve\_power(

effect\_size=effect\_size,

nobs1=len(won\_deals),

alpha=0.05,

ratio=len(lost\_deals)/len(won\_deals)

)

print(f”Power of the test: {power:.2f}”)

3. Visualization Portfolio

Professional-quality visualizations will include:

Small Multiples for comparing distributions across segments

Interactive Time Series with range selectors

Annotated Heatmaps for correlation matrices

Faceted Plots for multidimensional analysis

Example of publication-ready visualization:

python

# Example of a polished visualization

fig = px.box(

acts\_per\_deal,

x=’outcome’,

y=’activity\_count’,

color=’outcome’,

points=’all’,

title=’Total Activities per Deal by Outcome<br><sup>Won deals typically have more activities (p < 0.001)</sup>’,

labels={’activity\_count’: ‘Number of Activities’, ‘outcome’: ‘Deal Outcome’}

)

fig.update\_layout(

font\_family=”Arial”,

font\_size=12,

title\_font\_size=16,

plot\_bgcolor=’white’,

paper\_bgcolor=’white’,

boxgap=0.3

)

fig.update\_traces(

marker=dict(size=4, opacity=0.5),

line=dict(width=1.5)

)

fig.show()

Data Quality and Edge Case Handling

python

# Missing data analysis

msno.matrix(activities)

plt.title(’Missing Data Pattern in Activities’)

plt.show()

# Outlier detection

def detect\_outliers(df, column):

q1 = df[column].quantile(0.25)

q3 = df[column].quantile(0.75)

iqr = q3 - q1

lower\_bound = q1 - 1.5 \* iqr

upper\_bound = q3 + 1.5 \* iqr

return df[(df[column] < lower\_bound) | (df[column] > upper\_bound)]

deal\_value\_outliers = detect\_outliers(all\_deals, ‘deal\_value’)

print(f”Found {len(deal\_value\_outliers)} outliers in deal values”)

# Temporal consistency check

date\_consistency = all\_deals[all\_deals[’closed\_at’] < all\_deals[’created\_at’]]

print(f”Found {len(date\_consistency)} deals with closed date before creation date”)

Feature Engineering Recommendations

Based on EDA findings, recommend features for predictive modeling:

Activity Composition Features:

Ratio of meetings to emails

Percentage of activities that are calls

Note sentiment variance

Temporal Features:

Activity velocity (activities per week)

Days since last activity

Time between first and second meeting

Sequence Features:

Last activity type before closure

Presence of specific activity patterns (e.g., meeting → note → call)

Text Features:

Presence of key phrases in notes

Topic distribution from LDA

Sentiment trend over time

This comprehensive EDA plan ensures all technical requirements and acceptance criteria are met while providing actionable insights for improving sales performance.