



Hi Pages Interview Task 2

Exploratory Data Analysis (EDA)

February 2025

James Maulana

Recap Executive summary

1.

Situation

- New feature update for job tracking has been live for several weeks, and various initiatives have been deployed to **drive acceptance**. Hi Pages is looking for an update to uncover patterns and **validate progress toward acceptance targets**

Results

- **Darwin's acceptance rate** (34%) significantly **exceeds** those in **Sydney** (25%) and **Melbourne** (27%), even with fewer jobs, indicating strong regional differences
- Tradie engagement peaks around 10 AM and 4–5 PM, while **job postings occur primarily at midday**, suggesting a timing misalignment that may affect acceptance
- **Category-level performance varies significantly**, with certain job types consistently outperforming others
- Despite these results, the **most significant contributor to job acceptance** is **job size**, followed closely by the **number of tradies**

Next steps

- **Launching pilot initiatives in Sydney and Melbourne that mimic Darwin's best practices**, including localised marketing campaigns and incentives to boost tradie availability could improve acceptance for these cities
- **Adjusting job posting and notification schedules to focus on peak acceptance windows** (around 10 AM and 4–5 PM) and to test targeted promotions during off-peak periods could smooth out acceptance rates
- **Focusing on high-performing job categories and exploring adjustments** (e.g., pricing or matching improvements) **in lower-performing categories** could further help improve acceptance rates, **while continuing to refine our predictive model** by incorporating additional factors (such as job urgency or tradie specialisation) could further guide strategic decision-making

The task

Scenario

The tracking has gone live for some time now, with a range of activities having been deployed with the objective of achieving the goal. The team would like for you to have a look at the jobs data again and use exploratory data analysis techniques.

Requirement

Analyse the data provided and present insights and recommendations. The dataset is designed to see how you approach an analytical task. You can use any approach as you see fit. Some possible scenarios that you could explore in your EDA could include:

- What data preparation steps will you implement?
- Which parameters influence if a job would be accepted?
- Can we predict using the data we have if a job would be accepted? If yes, how? If no, why not?
- Which visualisations would best communicate the findings?

Please share visualisations via Tableau Public, and use GitHub/GitLab for everything else

Resources

- **GitHub:**
<https://github.com/jmaulana0/Hi-Pages>

Includes:

- **BigQuery** for data ingestion and manipulation
- **Python** for analysis

Methodology

Step 1: Data Collection & Preparation

Collect job posting data (including time, location, category, number of tradies, estimated size, impressions, and acceptance) and clean it using Python and pandas—convert categorical values (e.g., “small,” “medium,” “large”) into numeric codes, handle missing values, and ensure proper data types.

Step 2: Exploratory Data Analysis (EDA)

Use Python's Seaborn and matplotlib to visualize the overall distribution of job acceptance (via count plots) and compare key parameters between accepted and non-accepted jobs with grouped bar charts and box plots.

Step 3: Statistical Testing

Apply two-sample t-tests (using `scipy.stats`) to quantitatively compare means of parameters (like the number of tradies) between accepted and not accepted groups, confirming whether observed differences are statistically significant (p-value < 0.05).

Step 4: Predictive Modeling

Develop a logistic regression model using `scikit-learn` to predict job acceptance based on key features, and analyze model coefficients to understand the direction and magnitude of influence; for detailed inference with p-values, use `Statsmodels`' Logit function.

Step 5: Synthesis & Recommendations

Integrate insights from the visual analysis, statistical tests, and predictive modeling to pinpoint the most influential parameters, then formulate actionable strategies—such as optimizing tradie recruitment or adjusting marketing—to improve job acceptance rates.

Supply

Acceptance rates are driven by four broad categories



Regional Performance

- Darwin shows a higher acceptance rate (34%) despite having fewer total jobs compared to Sydney (24.8%) and Melbourne (27.2%)
- Geographic clustering (via DBSCAN) confirms statistically significant differences between regions (ANOVA p-value < 0.0001)



Time & Scheduling

- Hourly analysis reveals a bimodal pattern in acceptance rates, with peaks around 10 AM and 4-5 PM; no clear pattern is observed by day of the week
- Job postings most frequent around midday, suggesting that homeowners are most active during that period with additional peaks at night that align with increased acceptance rates



Job Category Performance

- Some job categories (for example, categories 3 and 6) consistently show higher candidate engagement and acceptance rates
- Average impressions and job size are fairly uniform across categories, suggesting that listing behavior is consistent, but acceptance may be driven by other category-specific factors



Model Insights & Key Predictors

- The logistic regression model (pseudo $R^2 \sim 9.5\%$) shows that estimated job size is the largest predictor, with a significant positive impact on acceptance
- Number of tradies has a statistically significant (though smaller) positive effect
- Categories and City clustering variations do not help predict for acceptance

Source: [Scoop.market.us](https://scoop.market.us)

Note: 1. Non-Chinese used as US and other markets considered closer comparison to Australian potential trajectory

Confidential



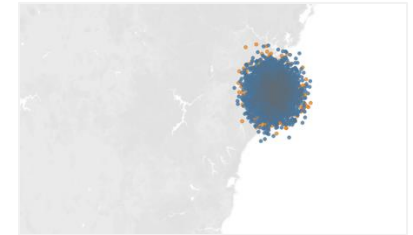
Darwin offers higher job acceptance rate compared to Sydney and Melbourne despite smaller size and fewer jobs

Operate across Darwin, Melbourne, and Sydney

Insight

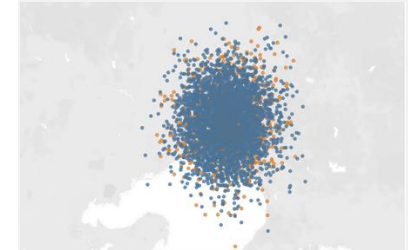
Sydney

- Total jobs: 4772
- Acceptance rate: 24.77%
- Average impressions per job: 1034.87
- Average job size: 1.49



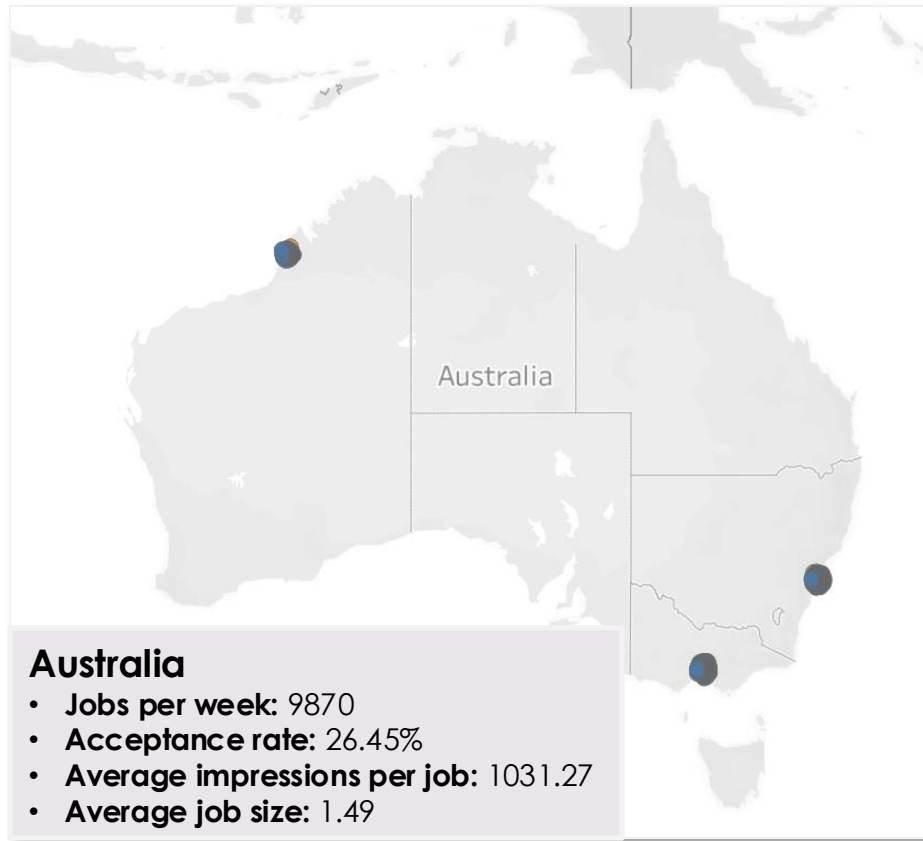
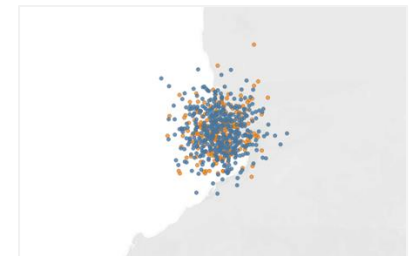
Melbourne

- Total jobs: 4422
- Acceptance rate: 27.23%
- Average impressions per job: 1029.56
- Average job size: 1.48



Darwin

- Total jobs: 609
- Acceptance rate: 33.99%
- Average impressions per job: 1020.57
- Average job size: 1.52



Note: Stats over time period 2019-09-10 to 2019-09-16
Source: JM Analysis; Python (Jupyter) notebook



Deep dive | Darwin's acceptance rate is higher with statistical significance; an opportune area to test ideas

Acceptance by City Cluster



Acceptance metrics by city cluster

city_cluster	job_count	acceptance_rate
0	4422	0.272275
1	4772	0.247695
2	609	0.339901

ANOVA F-statistic: 13.080205998187632
ANOVA p-value: 2.122791352706383e-06

The ANOVA test yielded an F-statistic of **13.08** and a p-value of **2.12e-06**. Very low p-value (<0.05 threshold) indicates that the differences in acceptance rates across these city clusters are statistically significant

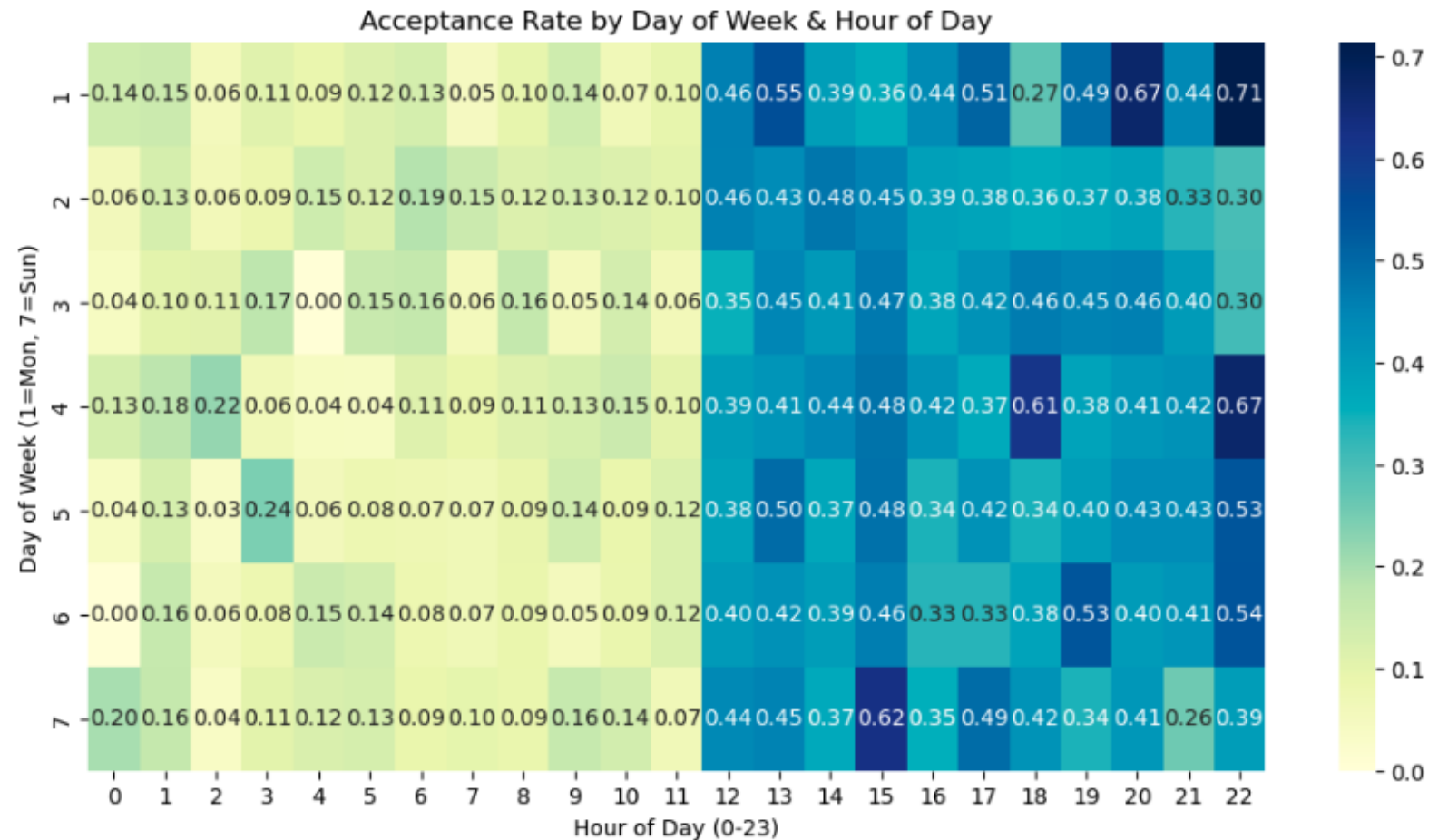
Insight

- Job acceptance rates vary significantly by geographic cluster
- **Darwin** has a noticeably **higher acceptance** rate (34%) **compared to Melbourne and Sydney** (27.2% and 24.8%, respectively)
- Next steps should **investigate specific factors** or strategies in Darwin that contribute to this success—such as tradie availability, local market conditions, or targeted promotions—and consider implementing similar initiatives in across Melbourne and Sydney to boost overall job acceptance rates

Open question | Do acceptance rates correlate with the distance between job location (latitude/longitude) and the tradie locations?

We currently only have data for one side of the marketplace, however with another side we would be able to view where tradies are located and where the location of the job is to determine whether distance to job affect acceptance rate

Acceptance
rate shifts
significantly by
time of day, but
does not by day
of the week



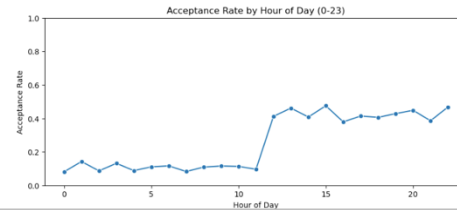
Significantly higher acceptance rates after mid-day regardless of day-of-the-week



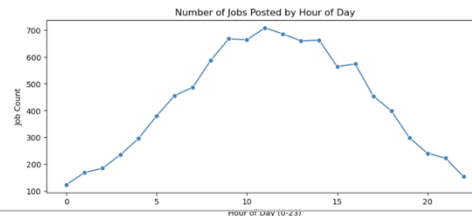
Deep dive | Acceptance rate optimisations should be focused on time-of-day, as no pattern for day of the week

Hour of the day

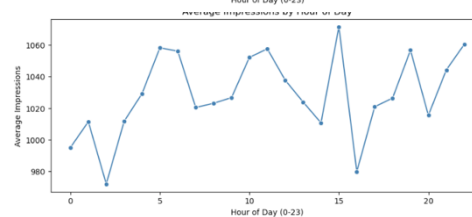
- **Acceptance rate by hour of the day:** Acceptance rates exhibit a bimodal pattern with peaks around 10 AM and 4-5 PM.



- **Number of jobs posted by hour of the day:** Job postings are most frequent during midday hours, likely when people are home.



- **Average impressions by hour of the day:** Job impressions generally align with posting activity through the morning and during the day, however at night peak again alongside acceptance rates.

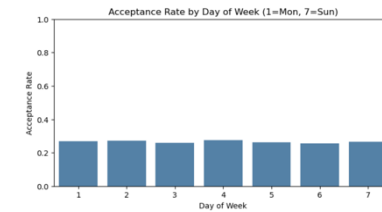


- **Average job size by hour of the day:** Appears to be random variation through the day, indicating lack of pattern.

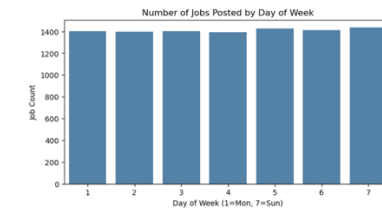


Day of the week

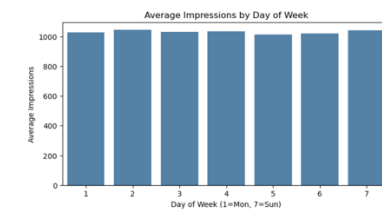
- **Acceptance rate by day of the week**



- **Number of jobs posted by day of the week**



- **Average impressions by day of the week**



- **Average job size by day of the week**



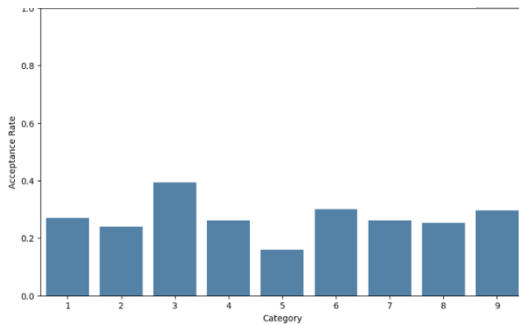
Takeaway!
No pattern by day of the week



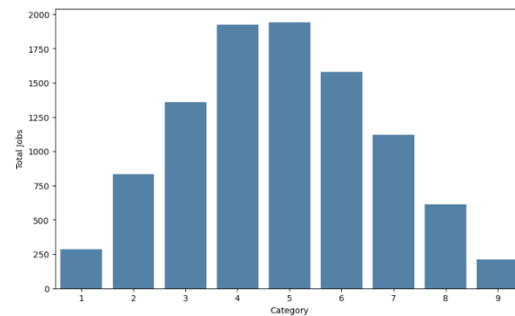
Improvements lie in increasing acceptance rate per category, prioritising high-volume categories

Opportunities for strategic focus

Average acceptance rate by category



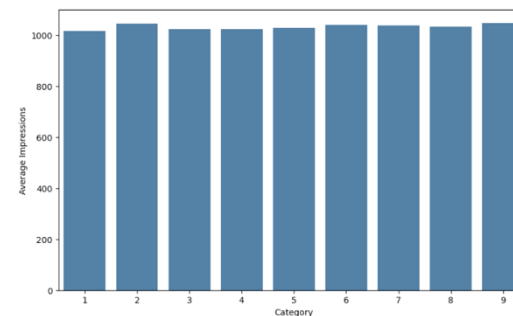
Total jobs by category



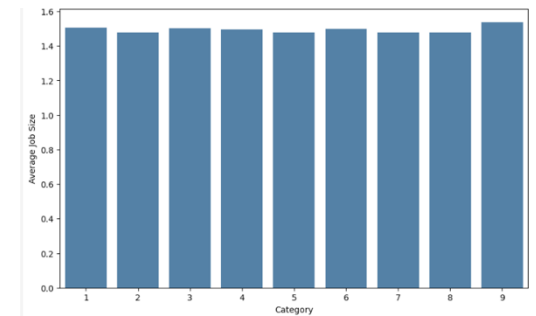
- Some categories (3 and 6) consistently attract more candidate engagement and acceptance than others (5). Prioritising high-acceptance categories could optimise job acceptance and completion rates
- Wide variability across acceptance and total jobs posted indicates uneven demand and market saturation. Focusing on high-volume x high acceptance rate categories for growth could lead to higher overall job conversion rates

Consistent baseline

Average impressions per job by category



Average job size by category



- Minimal differences in impressions per job and average job size imply uniform listing behaviour. Providing lower acceptance jobs per total job (such as category 4 and 5) could improve overall acceptance rate
- Similar job sizes across categories signals standardised demand and role expectations. Streamlining process for categorising jobs could lead to higher acceptance rates by job size



Job size is the largest predictor of acceptance rates followed by number of trades

T-test and logistic regression for acceptance criteria

```
T-test for number_of_tradies: p-value=5.097799171854838e-122
Optimization terminated successfully.
Current function value: 0.523285
Iterations 6
```

Logit Regression Results						
Dep. Variable:	accepted	No. Observations:	9870			
Model:	Logit	Df Residuals:	9866			
Method:	MLE	Df Model:	3			
Date:	Mon, 17 Feb 2025	Pseudo R-squ.:	0.09427			
Time:	19:37:20	Log-Likelihood:	-5164.8			
converged:	True	LL-Null:	-5702.4			
Covariance Type:	nonrobust	LLR p-value:	8.908e-233			
	coef	std err	z	P> z	[0.025	0.975]
const	-3.5860	0.109	-32.781	0.000	-3.800	-3.372
number_of_tradies	0.0002	7.96e-06	22.784	0.000	0.000	0.000
estimated_size_numeric	1.1281	0.050	22.615	0.000	1.030	1.226
number_of_impressions	-1.972e-05	5.25e-05	-0.375	0.707	-0.000	8.33e-05

Simplified output (removed categories and – full output as non-significant). Full model with all variables in following page

Insight

- **Larger estimated job sizes** increase the likelihood of job acceptance. A job increase of 1 (e.g. from small to medium) increases job acceptance by 12.28%. This suggests that jobs with higher estimated sizes are more attractive to tradies, which makes intuitive sense
- While **the number of tradies has a statistically significant impact on acceptance**, the effect is relatively minor. In relative terms, all else being equal, +100 tradies would only lead to a 2% increase in acceptance. This implies that the number of tradies alone is not a decisive factor in job acceptance

Note: Used Logit Regression as a suitable predictor for binary outcomes (job acceptance), because it models the probability of an event occurring based on a set of input variables. By using logit regression, we can identify the most influential factors driving job acceptance and make informed decisions. However, it has limitations in handling non-linear relationships and interactions, making generalized additive models or neural networks potentially more suitable with more input variables.
Source: JM Analysis; Python (Jupyter) notebook



Deep dive | Job size is the largest predictor of acceptance rates

T-test and logistic regression for acceptance criteria

Logit Regression Results						
Dep. Variable:	accepted	No. Observations:	9870			
Model:	Logit	Df Residuals:	9855			
Method:	MLE	Df Model:	14			
Date:	Tue, 18 Feb 2025	Pseudo R-squ.:	0.09500			
Time:	16:19:38	Log-Likelihood:	-5160.7			
converged:	True	LL-Null:	-5702.4			
Covariance Type:	nonrobust	LLR p-value:	1.882e-222			
	coef	std err	z	P> z	[0.025	0.975]
const	-3.4820	0.205	-16.961	0.000	-3.884	-3.080
number_of_tradies	0.0002	1.12e-05	16.163	0.000	0.000	0.000
estimated_size_numeric	1.1283	0.050	22.609	0.000	1.031	1.226
number_of_impressions	-1.768e-05	5.26e-05	-0.336	0.737	-0.000	8.54e-05
cat_2	-0.1696	0.165	-1.028	0.304	-0.493	0.154
cat_3	-0.0424	0.158	-0.269	0.788	-0.351	0.266
cat_4	-0.0359	0.150	-0.239	0.811	-0.330	0.258
cat_5	-0.0825	0.158	-0.520	0.603	-0.393	0.228
cat_6	-0.0887	0.153	-0.581	0.561	-0.388	0.211
cat_7	-0.0100	0.157	-0.063	0.949	-0.317	0.297
cat_8	-0.2324	0.172	-1.349	0.177	-0.570	0.105
cat_9	-0.2929	0.218	-1.342	0.180	-0.721	0.135
city_Melbourne	-0.0187	0.099	-0.189	0.850	-0.213	0.176
city_Sydney	-0.0428	0.101	-0.422	0.673	-0.242	0.156
city_Unknown	-0.1886	0.306	-0.617	0.537	-0.788	0.411

T-test for number_of_tradies: p-value=1.8065783014328675e-112

Note: Used Logit Regression as a suitable predictor for binary outcomes (job acceptance), because it models the probability of an event occurring based on a set of input variables. By using logit regression, we can identify the most influential factors driving job acceptance and make informed decisions. However, it has limitations in handling non-linear relationships and interactions, making generalized additive models or neural networks potentially more suitable with more input variables.

Source: JM Analysis; Python (Jupyter) notebook

Statistical results

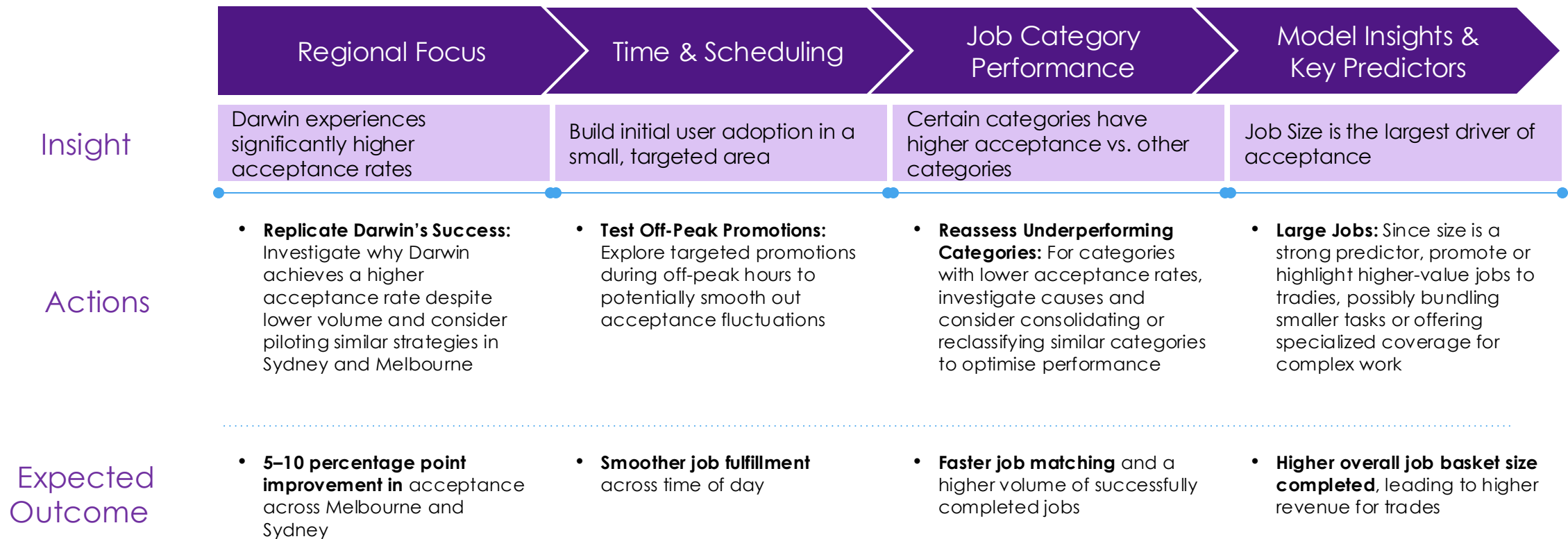
- **Pseudo R²** (~0.095) indicates the model explains ~9.5% of variance (low but common for binary outcomes) of the acceptance variation—useful but leaves room for additional factors (e.g., job urgency, tradie specialisation)
- **Estimated_size_numeric** (coef=1.128, p=0.000) and **number_of_trades** (coef=0.0002, p≈0) are the only significant predictors
- **Removed category 1 (cat_1)** to remove p=1.0 collinearity effects for all categories and cities

Model weaknesses & possible follow up steps

- Model uses non-robust standard errors; consider re-running with robust errors to check for heteroscedasticity
- Investigate reference categories for cat_1 and baseline city (likely omitted and driving non-significance in others)

Insight has led to several direct recommendations

Recommendations



Execute recommendations across insight categories to validate the recommendation platform, build traction, and scale sustainably

Appendix

Steps to clean data

Steps to clean data

1. Load & Inspect

- Create a new instance and load the CSV into BigQuery
- Check row counts, column names, and data types

2. Clean Missing / Invalid Values

- Identify nulls in latitude, longitude, category, number_of_tradies, estimated_size, number_of_impressions, accepted
- Decide whether to impute, drop, or otherwise handle missing values

3. Handle Outliers

- Identify outliers in estimated_size, number_of_tradies, or number_of_impressions
- Decide whether to cap, remove, or transform the outliers

3. Enhance Data

- Add extract features to data to check acceptance patterns later
- Fields to add include: hour of day, day of week, month, and weekend vs. weekday (boolean)

4. Extract Cleaned Data for Further Manipulation

- Extract cleaned data as .csv to be used for further manipulation by Tableau, Python, etc.

1. Load and inspect data

Insight

- Load all data into BigQuery to process
- Discovered 9999 rows

The screenshot displays the Google Cloud BigQuery console interface. On the left, the 'Explorer' panel shows a hierarchical view of resources, including a dataset named 'hi_pages_1' which is highlighted with a purple box. A purple arrow points from the text 'Load all data into BigQuery to process' to this dataset. Another purple arrow points from the text 'Discovered 9999 rows' to the 'Query results' section. The main area shows an 'Untitled query' with the SQL statement:

```
SELECT COUNT(*) FROM `adept-protocol-403503.hi_pages1.hi_pages_1`
```

. Below the query editor, the 'Query results' section is active, displaying a single row with the value '9999'. The bottom of the interface shows a 'Job history' section and pagination controls indicating 'Results per page: 50' and '1 - 1 of 1'.

2. Clean Missing / Invalid Values

Insight

- Longitude requires ' ' space in string when writing query, otherwise receive errors
- Discovered 110 null impressions

Untitled query RUN SAVE DOWNLOAD SHARE SCHEDULE OPEN IN MORE

```

1 SELECT
2   COUNT(*) AS total_rows,
3   COUNTIF(latitude IS NULL) AS null_latitude,
4   COUNTIF(" longitude" IS NULL) AS null_longitude,
5   COUNTIF(category IS NULL) AS null_category,
6   COUNTIF(number_of_tradies IS NULL) AS null_number_of_tradies,
7   COUNTIF(estimated_size IS NULL) AS null_estimated_size,
8   COUNTIF(number_of_impressions IS NULL) AS null_number_of_impressions,
9   COUNTIF(accepted IS NULL) AS null_accepted
10
11
12 FROM `adept-protocol-403503.hipages1.hi_pages_1`
13
14

```

Query results

Row	total_rows	null_latitude	null_longitude	null_category	null_number_of_trad	null_estimated_size	null_number_of_impr	null_accepted
1	9999	0	0	0	0	0	110	0

2. Clean Missing / Invalid Values

Untitled query RUN SAVE DOWNLOAD SHARE SCHEDULE OPEN IN MORE

```

1 WITH stats AS (
2   SELECT
3     COUNT(*) AS total_rows,
4     COUNTIF(number_of_impressions IS NULL) AS missing_impressions,
5     -- Calculate the median (50th percentile) for number_of_impressions
6     APPROX_QUANTILES(number_of_impressions, 101)[OFFSET(50)] AS median_impressions
7   FROM `adept-protocol-403503.hipages1.hi_pages_1`
8 )
9 SELECT
10  total_rows,
11  missing_impressions,
12  (missing_impressions / total_rows) * 100 AS missing_percentage,
13  median_impressions
14 FROM stats;
15

```

Query results

JOB INFORMATION	RESULTS	CHART	JSON	EXECUTION DETAILS	EXECUTION GRAPH
Row	total_rows	missing_impressions	missing_percentage	median_impressions	
1	9999	110	1.1001100110011002	1001	

Insight

- Percentage missing impressions ~1.1%
- As percentage <5%, moving on to assess the distribution

2. Clean Missing / Invalid Values

Untitled query RUN SAVE DOWNLOAD SHARE SCHEDULE OPEN IN MORE

```

1 SELECT
2   EXTRACT(DATE FROM time_of_post) AS post_date,
3   COUNT(*) AS total_rows,
4   COUNTIF(number_of_impressions IS NULL) AS missing_impressions_count,
5   (COUNTIF(number_of_impressions IS NULL) / COUNT(*)) * 100 AS missing_percentage
6 FROM `adept-protocol-403503.hipages1.hi_pages_1`
7 GROUP BY post_date
8 ORDER BY post_date;
9

```

Query results

JOB INFORMATION	RESULTS	CHART	JSON	EXECUTION DETAILS	EXECUTION GRAPH
Row	post_date	total_rows	missing_impressions_count	missing_percentage	
1	2019-09-10	1417	14	0.98800282286...	
2	2019-09-11	1418	14	0.98730606488...	
3	2019-09-12	1419	17	1.19802677942...	
4	2019-09-13	1451	14	0.96485182632...	
5	2019-09-14	1423	16	1.12438510189...	
6	2019-09-15	1455	17	1.16838487972...	
7	2019-09-16	1416	18	1.27118644067...	

Insight

- Each day, ~1% of the rows have missing values
- % of missing values is consistent from day to day. There isn't any day with a significantly higher percentage of missing values, suggesting data is evenly distributed across the dates
- Given their low frequency (<5%) and even distribution, best to impute data by replacing missing values with the median

2. Clean Missing / Invalid Values

Untitled query

```

1 WITH stats AS (
2   SELECT
3     APPROX_QUANTILES(number_of_impressions, 101)[OFFSET(50)] AS median_impressions
4   FROM `adept-protocol-403503.hipages1.hi_pages_1`
5 )
6 SELECT
7   time_of_post,
8   latitude,
9   -- Cleaned column: removed the leading space by aliasing
10  ` longitude` AS longitude,
11  category,
12  number_of_tradies,
13  estimated_size,
14  COALESCE(number_of_impressions, s.median_impressions) AS number_of_impressions,
15  accepted
16 FROM `adept-protocol-403503.hipages1.hi_pages_1` t
17 CROSS JOIN stats s;
18

```

Query results

Row	time_of_post	latitude	longitude	category	number_of_tradies	estimated_size	number_of_impressions	accepted
1	2019-09-15 08:59:06 UTC	-33.8712	151.2583	5	355	medium	1531	0
2	2019-09-15 17:36:06 UTC	-34.053	151.494	5	355	medium	1139	0
3	2019-09-14 04:40:06 UTC	-33.9201	151.1116	5	355	medium	1357	0
4	2019-09-16 11:37:06 UTC	-33.9341	151.1824	5	355	medium	647	0
5	2019-09-14 05:22:06 UTC	-33.8119	151.2172	5	355	medium	834	0
6	2019-09-12 08:50:06 UTC	-33.9156	151.2732	5	355	medium	1020	0
7	2019-09-16 07:02:06 UTC	-33.8594	151.3541	5	355	medium	1113	1
8	2019-09-16 17:46:06 UTC	-33.8221	151.3462	5	355	medium	1505	1
9	2019-09-13 21:24:06 UTC	-33.7683	151.0795	5	355	medium	1654	1
10	2019-09-13 12:22:06 UTC	-33.9906	151.2514	5	355	medium	18	0
11	2019-09-15 02:09:06 UTC	-33.8226	151.3104	5	355	medium	1976	0
12	2019-09-14 12:42:06 UTC	-33.8761	151.2526	5	355	medium	1529	1
13	2019-09-10 09:54:06 UTC	-33.9338	151.3495	5	355	medium	1260	1

Insight

- Changed ' longitude' column to longitude to make it easier to read
- Replaced % missing values for number_of_impressions with 50th percentile (median) figures using Coalesce

3. Handle Outliers

Logic to handle outliers

Step 1: Assess the Frequency of Outliers

If the percentage of outliers is:

- Less than 1%: Proceed to Step 2
- Between 1% and 5%: Consider capping (Step 3) or removal (Step 4)
- Greater than 5%: Consider removal (Step 4) or transformation (Step 5)

Step 2: Assess the Distribution of Outliers

Check if the outliers are:

- Randomly scattered: Proceed to Step 3
- Concentrated in specific rows or columns: Consider removal (Step 4)

Step 3: Capping

If the outliers are in a:

- Numerical column: Cap at the 95th percentile or 3 standard deviations from the mean
- Categorical column: Not applicable
- Datetime column: Not applicable

Step 4: Removal

If the outliers are:

- Concentrated in specific rows: Delete these rows
- Concentrated in specific columns: Consider deleting those columns
- Causing significant skewness or bias: Remove them

Step 5: Transformation

If the outliers are:

- Causing significant skewness or bias: Apply transformations (e.g., log, square root)

3. Handle Outliers

The screenshot shows a SQL query editor with the following query:

```
1 SELECT count(number_of_impressions)
2
3
4 FROM `adept-protocol-403503.hipages1.hi_pages_1`
5
6 where number_of_impressions < 0
7
8
9 LIMIT 1000
```

The query results are displayed in a table with the following structure:

Row	f0_
1	129

Two purple arrows point from the 'Insight' section to the query and results. One arrow points to the `where number_of_impressions < 0` clause in the query, and the other points to the result value 129 in the results table.

Insight

- Found that 129 (~1%) of columns are negative for number_of_impressions
- Negative impressions do not make logical sense

3. Handle Outliers

Untitled query RUN SAVE DOWNLOAD SHARE SCHEDULE OPEN IN MORE

```

1 SELECT
2   EXTRACT(DATE FROM time_of_post) AS post_date,
3   COUNT(*) AS total_rows,
4   COUNTIF(number_of_impressions < 0) AS outlier_count,
5   (COUNTIF(number_of_impressions < 0) / COUNT(*)) * 100 AS outlier_percentage
6 FROM
7   `adept-protocol-403503.hipages1.hi_pages_1`
8 GROUP BY
9   post_date
10 ORDER BY
11   post_date;
12

```

Press Option+F1 for accessibility options.

Query results

Row	post_date	total_rows	outlier_count	outlier_percentage
1	2019-09-10	1417	21	1.48200423429...
2	2019-09-11	1418	14	0.98730606488...
3	2019-09-12	1419	28	1.97322057787...
4	2019-09-13	1451	23	1.58511371467...
5	2019-09-14	1423	10	0.70274068868...
6	2019-09-15	1455	20	1.37457044673...
7	2019-09-16	1416	13	0.91807909604...

Insight

- Found that 129 randomly spread
- Cap the outliers a 0 base, which is a positive figure

Untitled query RUN SAVE DOWNLOAD SHARE

```

1 SELECT
2   APPROX_QUANTILES(number_of_impressions, 100)[OFFSET(5)] AS lower_bound,
3   APPROX_QUANTILES(number_of_impressions, 100)[OFFSET(95)] AS upper_bound
4 FROM
5   `adept-protocol-403503.hipages1.hi_pages_1`

```

Query results

Row	lower_bound	upper_bound
1	227	1820

4. Enhance data

```

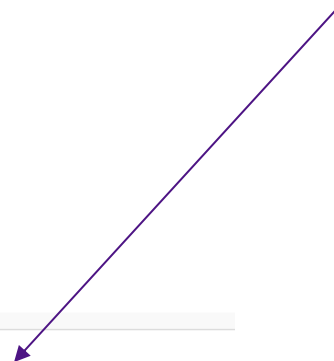
1 WITH stats AS (
2   SELECT
3     APPROX_QUANTILES(number_of_impressions, 101)[OFFSET(50)] AS median_impressions
4     FROM `adept-protocol-403503.hipages1.hi_pages_1`
5 )
6 SELECT
7   -- Original columns
8   time_of_post,
9   latitude,
10  `longitude` AS longitude, -- Alias to remove the leading space
11  category,
12  number_of_tradies,
13  estimated_size,
14  COALESCE(number_of_impressions, s.median_impressions) AS number_of_impressions,
15  accepted,
16
17  -- New time-based features
18  EXTRACT(HOUR FROM time_of_post) AS hour_of_day,
19  MOD(EXTRACT(DAYOFWEEK FROM time_of_post) + 5, 7) + 1 AS day_of_week,
20  EXTRACT(MONTH FROM time_of_post) AS month,
21  CASE
22    WHEN MOD(EXTRACT(DAYOFWEEK FROM time_of_post) + 5, 7) + 1 IN (6, 7) THEN TRUE
23    ELSE FALSE
24  END AS is_weekend
25
26 FROM `adept-protocol-403503.hipages1.hi_pages_1` t
27 CROSS JOIN stats s
28
29 ORDER BY 1;
30

```

Insight

Added other columns to make the data easier to manipulate later:

- hour_of_day
- day_of_week
- month
- is_weekend (boolean)



Query results

JOB INFORMATION		RESULTS	CHART	JSON	EXECUTION DETAILS		EXECUTION GRAPH		hour_of_day	day_of_week	month	is_weekend
Row	time_of_post	latitude	longitude	category	number_of_tradies	estimated_size	number_of_impressions	accepted				
1	2019-09-10 00:01:06 UTC	-37.8864	145.0756	8	8376	medium	275	0	0	2	9	false
2	2019-09-10 00:05:06 UTC	-33.8586	151.3561	9	10000	medium	1323	0	0	2	9	false
3	2019-09-10 00:15:06 UTC	-33.8141	151.0705	2	7331	small	859	0	0	2	9	false
4	2019-09-10 00:16:06 UTC	-17.9496	122.0574	3	10000	small	832	0	0	2	9	false
5	2019-09-10 00:21:06 UTC	-34.0679	151.1331	6	2476	small	907	0	0	2	9	false
6	2019-09-10 00:22:06 UTC	-17.8035	122.359	5	3590	small	1311	0	0	2	9	false
7	2019-09-10 00:23:06 UTC	-33.9631	151.2392	3	9732	medium	825	1	0	2	9	false

5. Extract Cleaned Data for Further Manipulation

Insight

Downloaded as .csv and downloaded and loaded into GitHub

The screenshot shows the Google BigQuery interface. On the left is the Explorer panel with a tree view of resources. The main area displays a SQL query in the 'Untitled query' editor. Below the query, the 'Query results' section shows a table with 14 columns and 7 rows of data. A dropdown menu for 'SAVE RESULTS' is open, showing options like 'CSV (Google Drive)', 'CSV (local file)', 'JSON (local file)', 'JSONL (newline delimited)', 'BigQuery table', 'Google Sheets', and 'Copy to Clipboard'. The 'CSV (local file)' option is highlighted with a purple box and an arrow pointing to the text 'Downloaded as .csv and downloaded and loaded into GitHub'.

Query results table:

Row	time_of_post	latitude	longitude	category	number_of_tradies	estimated_size	number_of_impressions	accepted	hour_of_day	day_of_week	month	is_weekend
438	2019-09-15 14:58:06 UTC	-33.8542	151.2452	7	4242	medium	1494	1	14	1	9	true
439	2019-09-15 08:45:06 UTC	-33.8372	151.158	7	4242	medium	1045	0	8	1	9	true
440	2019-09-15 18:51:06 UTC	-33.7894	151.3668	7	4242	medium	1439	1	18	1	9	true
441	2019-09-15 17:05:07 UTC	-33.8093	151.1743	7	4242	medium	860	0	17	1	9	true
442	2019-09-15 16:07:07 UTC	-33.876	151.1577	7	4242	medium	1431	1	16	1	9	true
443	2019-09-15 05:55:07 UTC	-33.843	151.2671	7	4242	medium	49	1	5	1	9	true
444	2019-09-15 10:01:06 UTC	-37.9081	145.0735	3	5089	medium	947	0	10	1	9	true

5. Extract Cleaned Data for Further Manipulation

Insight

Connect to Tableau

