We are using two metrics to determine whether or not a business stands out compared to its local competition. One metric is the businesses that have higher than average total check-ins while also having higher than average ratings. This approach helps us identify which businesses attract the most customers while maintaining their high rating. For example, a McDonald's may have significant foot traffic but will likely not have a high rating, and should not be counted as successful. The other metric tracks the businesses that have a 33% or higher average monthly review count increase. This metric determines if the business is bringing in more customers year-over-year, representing a business that more people are visiting and liking than usual. We set the threshold at 33% more reviews per year on average for a popular business. This growth indicates a thriving business that other Yelp users may like.

To find the average total check-ins for each business category, we joined the business, categories and check-ins tables. Then, we took the average number of check-ins for each business category and the average the star rating for each business category. Since many businesses had multiple categories, we chose to keep only one category per business. This is so that a business would not skew categories by being above average in some categories and below average in others. To filter each business to only one category, we grouped the businesses by their ID, then by their category name and kept the minimum value of the category name. After that we selected the businesses that had higher values for the stars and check-ins than the average for their category. A business is considered successful if it is included in this query. If it is above average in ratings then it is a successful and well-renowned business. If it is above average in check-ins then it is a popular business. This method allows us to gauge both the popularity and quality of businesses. So, if a business is above average in both check-ins and ratings, then it is a popular and successful business.

Our second metric is used to measure how businesses are doing over time and identify those with blossoming new management, menus, or events. We analyzed the positive change in reviews they received each month. First, we counted the reviews for each business for every month and compared this number to the reviews from the month before to see how much it grew. We were especially interested in businesses that had a significant increase in positive reviews, showing they are gaining popularity. We set a target growth rate and looked for businesses that were growing faster than this rate. This measure helps us identify businesses that are gaining momentum compared to those around it, including those that might outperform it in the "successful" metric. We grouped these findings by business name and ID, focusing on those with a higher than 0.329 growth rate in their monthly reviews. We found that this rate generally revealed a select number of businesses which were attracting more customers and clearly capturing the public's interest. Businesses above this rate are considered popular because they're getting more attention and positive feedback from customers. In the future, this rate may need to be adjusted or become variable if none of the businesses in an area meet the threshold.

The following queries do not include WHERE clauses for querying individual states, cities, or zip codes for the sake of brevity, and the fact that we will leave it up to the user how refined their area search is. In the final implementation, we still plan to allow this querying so that users may find the most successful and popular businesses in their desired region.

(Returns the businesses with higher than average checkins and stars for their category)

```sql
WITH CategoryAverages AS (
    SELECT
        cat.cat_name,
        AVG(ch.count) AS avg_checkins,
        AVG(b.stars) AS avg_stars
    FROM
        Checkins ch
        JOIN Categories cat ON ch.business_id = cat.business_id
        JOIN Business b ON ch.business_id = b.business_id
    GROUP BY
        cat.cat_name
),
FirstCategoryPerBusiness AS (
    SELECT
        b.business_id,
        b.name,
        MIN(cat.cat_name) AS first_category
    FROM
        Business b
        JOIN Categories cat ON b.business_id = cat.business_id
    GROUP BY
        b.business_id,
        b.name
),
BusinessCheckins AS (
    SELECT
        ch.business_id,
        SUM(ch.count) AS total_checkins
    FROM
        Checkins ch
    GROUP BY
        ch.business_id
),
BusinessWithFirstCategoryAndCheckins AS (
    SELECT
        fc.business_id,
        fc.name,
        fc.first_category,
        b.stars,
        bc.total_checkins
    FROM
        FirstCategoryPerBusiness fc
        JOIN BusinessCheckins bc ON fc.business_id = bc.business_id
        JOIN Business b ON fc.business_id = b.business_id
)
SELECT
    bfc.business_id,
    bfc.name,
    bfc.first_category,
    bfc.stars,
    bfc.total_checkins
FROM
    BusinessWithFirstCategoryAndCheckins bfc
    JOIN CategoryAverages ca ON bfc.first_category = ca.cat_name
WHERE
    bfc.total_checkins > ca.avg_checkins
    AND bfc.stars > ca.avg_stars
ORDER BY
    bfc.first_category, bfc.name;
```

(Returns the average monthly increase in reviews to show the increase of new customers)

```
WITH MonthlyReviews AS (
    SELECT
        business_id,
        DATE_TRUNC('month', date) AS review_month,
        COUNT(*) AS monthly_reviews
    FROM
        Review
    GROUP BY
        business_id, review_month
),
MonthlyGrowth AS (
    SELECT
        mr.business_id,
        mr.review_month,
        mr.monthly_reviews,
        LAG(mr.monthly_reviews, 1) OVER(PARTITION BY mr.business_id ORDER BY mr.review_month) AS
prev_month_reviews,
        (mr.monthly_reviews - LAG(mr.monthly_reviews, 1) OVER(PARTITION BY mr.business_id ORDER BY
mr.review_month))::FLOAT / NULLIF(LAG(mr.monthly_reviews, 1) OVER(PARTITION BY mr.business_id ORDER BY
mr.review_month), 0) AS growth_rate
    FROM
        MonthlyReviews mr
)
SELECT
    mg.business_id,
    b.name,
    AVG(mg.growth_rate) AS avg_monthly_growth_rate
FROM
    MonthlyGrowth mg
JOIN
    Business b ON mg.business_id = b.business_id
WHERE
    mg.prev_month_reviews IS NOT NULL
GROUP BY
    mg.business_id, b.name
HAVING
    AVG(mg.growth_rate) > 0.329
ORDER BY
    avg_monthly_growth_rate DESC;
```