



ASCENDVENT | PRACTICAL AI WORKFLOW GUIDE

A PM's Operating Guide to F1 Scores

Learn why F1 matters, how to calculate it and when to use it through one support workflow.

THE PROMISE

Follow a PM investigating a wrong refund answer and an overloaded review queue. Work through the ticket evidence and calculations to decide which model setting deserves a pilot.

For product managers

A worked calculation and a decision about missed referrals and review capacity.

For product leaders

Evidence, operating limits and a controlled pilot recommendation.

A customer gets the wrong answer

You are the PM for a B2B software company's support workflow. A customer asks for a refund under an exception in their contract. The agent checks the standard refund policy and sends an answer. Later, a specialist discovers that the exception required a different response. The customer has to ask again and the team has to undo the first answer.

The support lead shows you another ticket: a customer only wanted a copy of an invoice, but a keyword rule flagged the word "billing" and sent it to a specialist. That customer waited for work a frontline agent could have completed.

The team needs help recognizing which tickets require a specialist before anyone replies. Catching more of those tickets could prevent rework. Sending every ticket to a specialist would overload the queue and delay routine requests.

Why you would care about F1

Engineering proposes a model that suggests whether to escalate each ticket. Before you put that suggestion into the agent's workflow, you need to understand two kinds of error: required referrals it misses and unnecessary referrals it adds. Precision and recall measure those two sides. F1 combines them into a score you can use when comparing versions.

Use these measures before a pilot to compare the model with the existing rule, then keep checking reviewed tickets after launch. Include the error counts with F1 so the support lead can connect changes in the score to missed referrals and review work.

YOUR JOB IN THIS TUTORIAL

Work out whether this escalation suggestion deserves a controlled pilot. You will build the comparison, calculate F1 and use the results to explain a product decision.

The company, tickets and numerical results are invented for teaching. The example evaluates a support copilot's escalation suggestions.

1. Decide what the model should recognize

Begin by agreeing with the support lead on when an agent needs help: the refund exception requires a specialist's authority, while the invoice request is work the agent can complete. That distinction gives you a rule for judging each suggestion.

Together you write: "Escalation is required when the agent cannot resolve the ticket using the approved policy and their own authority before replying." Now the proposed feature has a specific job: suggest specialist review when that rule applies. The agent will still inspect every ticket and own the reply.

Turn the policy into examples people can judge

TICKET	EXPECTED DECISION	REASON
"Please resend my invoice."	No escalation	The agent can complete the request.
"My contract has a refund exception missing from the help center."	Escalation	A specialist must interpret the exception.
"Two approved policies give different answers to my refund request."	Escalation	The agent cannot settle the policy conflict.

For the calculation, encode "escalation required" as 1, the positive class, and "no escalation required" as 0, the negative class. Positive is the technical term for the condition you are trying to detect.

Agree on acceptable errors before seeing results

For this assisted workflow, the support lead agrees that the model must catch at least 95 of every 100 required referrals. The team can inspect up to 150 suggestions per 1,000 incoming tickets, and at least 65% of the suggestions should be necessary.

Existing privacy, security and high-consequence routes stay mandatory and are checked separately. This tutorial evaluates the in-scope billing and refund queue.

To check whether the model meets these limits, compare its predictions with independently labeled examples.

2. Build a comparison you can trust

You ask for historical tickets from the queue the feature will serve. Give the model the ticket and policy information available when the agent first saw it. Later resolution notes would reveal answers it will not have when the feature is running.

The support lead and policy owner apply the rule without seeing the model's predictions. Have both label a subset independently so you can identify disagreements, and record unresolved contract exceptions for policy review before assigning them a reference label.

Reserve different examples for different questions

SET	WHAT YOU DO WITH IT	WHY YOU KEEP IT SEPARATE
Development	Improve the rubric, model or prompt.	The team learns from these examples.
Validation	Compare thresholds on a fixed model.	Choose how readily it should flag a ticket.
Unseen test	Check the chosen setup once.	Find out whether the choice works on examples it was not tuned against.

The validation set contains 1,000 tickets: 100 require escalation and 900 do not. Reserve 1,000 different tickets for the unseen test, with the same 10% escalation rate. When building your own set, use a sample size and class mix that reflect the queue you plan to serve.

Keep duplicates and tickets from the same incident or conversation in one set. Record the source, period, languages, policy version and unresolved count. Include difficult cases instead of selecting only clear ones.

Engineering runs the first model on the validation tickets. It gives every ticket an escalation score and flags scores at or above 0.50. This cutoff turns a score into a yes-or-no suggestion. You will revisit it after you understand the errors.

TWO SCORES WITH DIFFERENT JOBS

The model's score determines whether one ticket is flagged. F1 measures how well the resulting flags match the agreed answers across the evaluation set.

3. Look inside the "96% accurate" result

The engineer returns with a promising headline: "The model got 96% of tickets right." You check it against the question support asked. How many of the tickets that needed a specialist did it catch?

Compare each prediction with its reference label. The model flagged 100 tickets, but only 80 of those actually required escalation. It also left 20 required referrals unflagged. The matching totals hid different mistakes.

MODEL SUGGESTION	ESCALATION REQUIRED	ESCALATION NOT REQUIRED	TOTAL
Flag for review	80 true positives (TP)	20 false positives (FP)	100
No flag	20 false negatives (FN)	880 true negatives (TN)	900
Total	100	900	1,000

Connect each count to the work it creates

A true positive catches a ticket like the contractual refund exception. A false positive sends a routine invoice request into review. A false negative misses a case like conflicting refund policies. A true negative correctly leaves a routine request in normal triage. These four counts form the confusion matrix.

Accuracy = (TP + TN) / all tickets

Accuracy = (80 + 880) / 1,000 = 96%

Because 900 of these tickets do not need escalation, correctly leaving routine tickets alone makes overall accuracy high. Even a rule that never flags anything would get 90% right while missing all 100 required referrals.

WHAT YOU HAVE LEARNED

This model misses 20 of the 100 required referrals. It fails the support lead's limit of five misses even though overall accuracy is 96%.

To explain both the missed referrals and the review burden consistently, you need two more measures. They use different denominators because they answer different questions.

4. Ask two questions about the same tickets

Use the matrix to answer the specialist lead's question about useful flags and the frontline lead's concern about missed referrals.

Of the tickets we flagged, how many needed review?

$$\text{Precision} = \text{TP} / (\text{TP} + \text{FP})$$

$$\text{Precision} = 80 / (80 + 20) = 0.80 = 80\%$$

Start with the 100 flagged tickets because you are evaluating the specialist's incoming work. Eighty flags were necessary and 20 were avoidable. Higher precision means a greater share of the flags are useful.

Of the tickets needing review, how many did we catch?

$$\text{Recall} = \text{TP} / (\text{TP} + \text{FN})$$

$$\text{Recall} = 80 / (80 + 20) = 0.80 = 80\%$$

Start with the 100 tickets that required escalation, including the ones the model missed. It caught 80 and missed 20. Higher recall means fewer required referrals escape the suggestion.

The two denominators happen to be 100 here. Precision always starts with what the model flagged; recall starts with what actually needed a flag. They will differ when you change the cutoff.

TRANSLATE THE TEAM'S LIMITS

"Catch 95 out of 100 required referrals" means recall of at least 95%. "At least 65% of flags should be useful" means precision of at least 65%. The 150-flag capacity limit is a separate workload check.

PRACTICE: PRECISION AND RECALL

On a separate 20-ticket practice set, a model catches six required referrals, adds two unnecessary flags, misses four required referrals and correctly leaves eight routine tickets unflagged. Calculate precision and recall. Use the flagged total for precision and the required-referral total for recall.

Check: precision = $6 / 8 = 75\%$. Recall = $6 / 10 = 60\%$. You will use these values for an F1 calculation on the next page.

5. Calculate F1 and understand its job

The first model has 80% precision and 80% recall, but other settings may improve one while reducing the other. Calculate F1 to compare how each setting combines useful flags with required referrals caught.

Start with the precision and recall you calculated

F1 uses the harmonic mean to combine precision and recall with equal importance. Multiply the two values, double the result and divide by their sum. Use 0.80 for each of the 80% scores.

$$F1 = 2 \times \text{precision} \times \text{recall} / (\text{precision} + \text{recall})$$

$$F1 = 2 \times 0.80 \times 0.80 / (0.80 + 0.80) = 0.80$$

That gives this setting an F1 of 0.80. The score ranges from 0 to 1 when defined and falls when either precision or recall is low.

Calculate it directly from the counts

The count-based formula gives the same result: double the 80 true positives, then divide by that total plus the 20 false positives and 20 false negatives. The 880 true negatives do not enter F1.

$$F1 = 2 \times TP / (2 \times TP + FP + FN)$$

$$F1 = 160 / (160 + 20 + 20) = 160 / 200 = 0.80$$

Both formulas give 0.80 because they summarize the same predictions. Accuracy measures the share of tickets classified correctly; F1 combines precision and recall.

Finish the practice example

For the 20-ticket exercise at the end of section 4 on the previous page, use six true positives, two false positives and four false negatives. Substitute those counts into the same formula.

$$F1 = 2 \times 6 / (2 \times 6 + 2 + 4) = 12 / 18 = 0.667$$

For that set, 75% precision and 60% recall give an F1 of about 0.667. An ordinary average would give 67.5%; F1's harmonic mean gives more pull to the lower value.

HOW YOU USE THE SCORE

Compare F1 alongside precision, recall and the four counts, using the team's miss limit and queue capacity to judge each setting. This setting misses 20 required referrals against a limit of five.

The first model still misses 20 required referrals, so engineering proposes lowering the cutoff to catch more of them.

6. Change the cutoff and examine the tradeoff

To catch more of the missed referrals, lower the cutoff that triggers a suggestion. The model's scores stay fixed: at 0.50 it flagged scores of 0.50 and above, while at 0.30 it also flags tickets scoring at least 0.30 and below 0.50.

Start with the required referrals. At 0.50, the model caught 80 and missed 20. At 0.30, it catches 95 and misses five, meeting the support lead's limit. To decide whether that gain is usable, look next at the unnecessary flags and total review load.

THRESHOLD	TP / FP / FN / TN	PRECISION	RECALL	F1	FLAGS
0.70	54 / 6 / 46 / 894	90.0%	54.0%	0.675	60
0.50	80 / 20 / 20 / 880	80.0%	80.0%	0.800	100
0.30	95 / 50 / 5 / 850	65.5%	95.0%	0.776	145

At 0.30, false positives rise from 20 to 50, bringing the total to 145 flags. Precision falls to 65.5%, which still meets the 65% floor, and the flags fit within the 150-ticket limit. The more selective 0.70 setting misses 46 required referrals.

Calculate F1 for the changed predictions

The earlier F1 of 0.800 described the predictions at 0.50. Lowering the cutoff changes those predictions, so calculate F1 again using the new counts: 95 true positives, 50 false positives and five false negatives.

$$F1 = 2 \times 95 / (2 \times 95 + 50 + 5) = 190 / 245 = 0.776$$

The lower cutoff catches more required referrals while adding enough unnecessary flags to bring F1 down from 0.800 to 0.776. The team can accept that tradeoff because the setting meets its recall requirement and keeps review demand within capacity.

Math check: the ordinary average of 95% recall and 65.5% precision is about 80.3%. F1 uses the harmonic mean, so the lower precision pulls its result down to 0.776.

YOUR THRESHOLD CHOICE

Choose 0.30 for the next check. It meets the previously agreed 95% recall floor, 65% precision floor and 150-flag limit. Choosing the highest F1 would have violated the miss limit.

Next, convert the 145 flags into review time to check that the team can handle them.

7. Turn the extra flags into review work

The recall gain of 15 percentage points also adds work for specialists. To check whether they can handle it, count the extra tickets and estimate the review time.

Moving from 0.50 to 0.30 adds 45 flags, bringing 15 more required referrals to a specialist along with 30 unnecessary reviews. Include both in the staffing estimate.

FIRST-PASS REVIEW	AT 0.50	AT 0.30
Flagged tickets	100	145
Assumed minutes per flag	3	3
Total review minutes	300	435
Required referrals missed	20	5

The lower threshold adds 135 minutes of review per 1,000 incoming tickets. The team's assumed capacity is 450 minutes, so the estimate fits with only 15 minutes of headroom. A busier queue or longer review time could remove that margin.

Check review time during the pilot

Use three minutes per flag to estimate the initial review, then track review time and queue age during the pilot. Record the specialist's resolution work separately so both kinds of effort appear in the staffing plan.

READY FOR THE UNSEEN TEST

The 0.30 setting meets the quality and review limits on the validation set. Test the same setting on the reserved tickets before changing the workflow.

8. Check the choice on unseen tickets

You chose 0.30 because it worked well on the validation tickets. That result may depend on the particular cases in that set, so use the reserved tickets to check whether it still catches enough required referrals and keeps review demand within capacity.

Keep the chosen model, prompt, policy context, label rule and threshold fixed during this check. Run it and the existing keyword rule on the same 1,000 unseen tickets so their results can be compared fairly. The agreed labels identify 100 required referrals.

UNSEEN TEST	KEYWORD RULE	MODEL AT 0.30
TP / FP / FN / TN	62 / 88 / 38 / 812	96 / 48 / 4 / 852
Precision	41.3%	66.7%
Recall	62.0%	96.0%
F1	0.496	0.787
Review suggestions	150	144

Start with missed referrals: the model misses four compared with 38 for the keyword rule. Unnecessary flags also fall, from 88 to 48, so F1 rises from 0.496 to 0.787 as both measures improve.

The model meets the three screening limits with 96% recall, 66.7% precision and 144 flags. At three minutes per flag, review demand is 432 minutes compared with 450 for the current rule. Measure customer response time during the pilot to find out whether the better referral suggestions help customers.

Check how much evidence supports the result

Recall is based on 100 required referrals, so each additional miss changes it by one percentage point. Report a statistical uncertainty estimate with the 96% result to show how much it might vary on new tickets.

If the test fails, investigate and revise using development data. Reserve new unseen examples before confirming the revision; repeated tuning against this test would use up its independence.

The test supports a pilot in the real queue, where you can observe whether agents act on the suggestions and complete the right handoffs.

9. Find out whether the suggestion helps people

Now check what happens after the model flags the refund-exception ticket. It counts as a true positive in the evaluation, but the customer can still get the wrong answer if the agent overlooks the flag or sends it to the wrong specialist. The pilot lets you observe whether correct suggestions lead to completed handoffs.

Begin in shadow mode: the model records suggestions while the existing process handles tickets. The support lead compares them with completed reviews, checks missed cases and measures the review demand. After that evidence is reviewed, a limited assisted pilot can make the suggestion visible while the agent keeps control.

QUESTION IN THE PILOT	EVIDENCE YOU COLLECT
Are required referrals still being caught?	Labeled flagged and unflagged tickets, the four counts and precision, recall and F1
Can the team handle the suggestions?	Actual review minutes, flag volume and specialist queue age
Does the required work reach the right person?	Correctly completed handoffs before a reply
Does the customer get a correct response sooner?	Time to a correct response, rework and reopened cases compared with the prior workflow

Look for misses where they can occur

Audit a random sample from the entire incoming queue so you can find required referrals among unflagged tickets as well as assess the flags. Calculate recall using all reference-positive cases in that sample, and track unresolved or not-yet-reviewed tickets until they have labels.

Review results by ticket type and language as well as overall. If policy conflicts perform poorly while routine questions dominate the sample, an overall score can hide the failure. Rerun checks after model, prompt or policy changes and watch for changes in the queue mix.

If you oversample flags or rare cases, use appropriate sampling weights for population estimates. A balanced diagnostic set is useful for finding errors, but its precision may not represent the live queue.

Use the test results to recommend a pilot, with named owners for the measurements and stop conditions.

10. Recommend a pilot to support leadership

At the review, explain how the proposed pilot addresses missed specialist referrals while staying within the team's review capacity. Use the test results to support the recommendation and name what the pilot will measure.

THE RECOMMENDATION

Approve a two-week shadow pilot of escalation suggestions at threshold 0.30. Keep the current workflow active. Use the pilot to judge whether limited assisted use is justified.

WHAT LEADERSHIP NEEDS	WHAT YOU CAN SAY
Evidence against the current rule	On the same unseen 1,000 tickets, misses fall from 38 to 4 and false alarms from 88 to 48. Flags fall from 150 to 144.
How F1 helps explain the result	F1 rises from 0.496 to 0.787. Precision is 66.7% and recall is 96.0%; report the 100 required-referral cases and uncertainty with them.
Why you chose this threshold	The highest validation F1 failed the required recall floor. The 0.30 setting met the agreed recall, precision and review-volume screens.
What remains unproven	Actual review time, completed handoffs and time to a correct customer response

Give the pilot owners concrete responsibilities

The support lead checks queue capacity and missed referrals. The policy owner resolves disputed labels. Engineering keeps the previous rule available for restoration. You own the readout and the exposure decision. Agree on the monitoring sample and review periods before the pilot starts.

Stop expansion on a critical policy failure or a breach of a mandatory route. If the reviewed evidence fails the agreed quality or capacity limits, return affected work to the prior route and keep those limits fixed while the team investigates.

At the two-week readout, bring review effort, completed handoffs, customer response outcomes and classification results by segment. If too few required referrals have been labeled to judge recall, continue collecting labels in shadow mode until the support lead can assess it.

Attach the source, period, sample definition, versions and uncertainty to the results in the brief.

Try this with a decision in your own product

Choose a yes-or-no decision your feature makes, such as whether a ticket needs review or an alert needs attention. Write what a positive result causes someone to do and the consequences of missing one or flagging one unnecessarily.

When both error types matter, use precision and recall to describe them and F1 to summarize the comparison. Set acceptance limits from the consequences of each error and the work reviewers can handle.

Build a small calculation you can inspect

Put one evaluated item on each row. In column B, enter the reference label; in column C, the prediction. Use numeric 1 for the positive class and 0 for the negative class. For 1,000 labeled items in rows 2 through 1001, use these formulas.

CELL	VALUE	FORMULA
F2	TP	=COUNTIFS(B2:B1001,1,C2:C1001,1)
F3	FP	=COUNTIFS(B2:B1001,0,C2:C1001,1)
F4	FN	=COUNTIFS(B2:B1001,1,C2:C1001,0)
F5	TN	=COUNTIFS(B2:B1001,0,C2:C1001,0)
F7	Precision	=IF(F2+F3=0,NA(),F2/(F2+F3))
F8	Recall	=IF(F2+F4=0,NA(),F2/(F2+F4))
F9	F1	=IF(2*F2+F3+F4=0,NA(),2*F2/(2*F2+F3+F4))

Use the full evaluated range for your data. Keep missing and unresolved labels outside that range and report their count. Check that TP + FP + FN + TN equals the number of included items.

Use the errors to choose your next task

Read false positives and false negatives with the person who owns the work. Agree whether the next change should repair the label rule, adjust a threshold, improve the model or fix the handoff. Record that choice with the configuration, sample, four counts and scores so the next comparison has a clear baseline.

YOUR FIRST PRACTICE RUN

Calculate the four counts on a small labeled set, explain two actual errors and name the product decision they affect. Before launch, repeat the comparison on reserved, unseen examples.

Keep these details beside your calculations

Handle the cases the main example did not need

CASE	WHAT TO DO
No predicted positives	If $TP + FP = 0$, precision is undefined. Show the counts and reporting convention.
No actual positives	If $TP + FN = 0$, recall is undefined. This sample cannot show how well required cases are detected.
No positive labels or predictions	If $2TP + FP + FN = 0$, F1 is undefined. Tools may substitute a value; document that choice.
Several exclusive classes	Report each class. Macro F1 averages class F1 values equally; weighted F1 weights by actual class size. Micro F1 pools counts and equals accuracy for single-label multiclass tasks over all classes.
Unequal priorities	F-beta can emphasize recall or precision, but operating limits and error costs still need separate treatment.

For a support copilot, use F1 to evaluate escalation suggestions alongside separate checks for reply accuracy, policy support and usefulness. Keep the label rule, threshold and dataset mix with the results.

Use a threshold with an evaluated model score; for yes-or-no outputs, evaluate the labels directly. Validate an LLM's stated confidence before interpreting it as a probability.

Sources for the methods

[Google: Classification metrics](#)

Precision, recall, accuracy and F1 definitions used in the calculations.

[Google: Thresholds and the confusion matrix](#)

Prediction outcomes and the effect of changing a threshold.

[scikit-learn: f1_score](#)

F1 calculation and undefined-score conventions.

[scikit-learn: Tuning the decision threshold](#)

Threshold selection and separation of tuning from testing.

[scikit-learn: Metrics and scoring](#)

Multiclass averaging and F-beta.

[Ascendvent: A PM's Operating Guide to Comparing Models](#)

Companion tutorial for the broader support copilot model decision.

[Ascendvent: Practical AI Workflow Guide](#)

Companion guide to workflow boundaries and evaluation.

Ryan K. McDonald | Ascendvent | ascendvent.life

Working draft | 12 October 2026.