package com.twitter.visibility.rules

import com.twitter.visibility.common.ModelScoreThresholds

import com.twitter.visibility.common.actions.AvoidReason

import com.twitter.visibility.common.actions.AvoidReason.MightNotBeSuitableForAds

import com.twitter.visibility.common.actions.LimitedEngagementReason

import com.twitter.visibility.common.actions.TweetVisibilityNudgeReason

import com.twitter.visibility.configapi.configs.DeciderKey

import com.twitter.visibility.configapi.params.FSRuleParams.HighSpammyTweetContentScoreSearchLatestProdTweetLabelDropRuleThresholdParam

import com.twitter.visibility.configapi.params.FSRuleParams.HighSpammyTweetContentScoreSearchTopProdTweetLabelDropRuleThresholdParam

import com.twitter.visibility.configapi.params.FSRuleParams.HighSpammyTweetContentScoreTrendLatestTweetLabelDropRuleThresholdParam

import com.twitter.visibility.configapi.params.FSRuleParams.HighSpammyTweetContentScoreTrendTopTweetLabelDropRuleThresholdParam

import com.twitter.visibility.configapi.params.FSRuleParams.SkipTweetDetailLimitedEngagementRuleEnabledParam

import com.twitter.visibility.configapi.params.RuleParam

import com.twitter.visibility.configapi.params.RuleParams.\_

import com.twitter.visibility.models.TweetSafetyLabelType

import com.twitter.visibility.rules.Condition.\_

import com.twitter.visibility.rules.Condition.{True => TrueCondition}

import com.twitter.visibility.rules.Reason.\_

import com.twitter.visibility.rules.RuleActionSourceBuilder.TweetSafetyLabelSourceBuilder

object AbusiveTweetLabelRule

extends NonAuthorWithTweetLabelRule(

Drop(Unspecified),

TweetSafetyLabelType.Abusive

)

with DoesLogVerdict

object AbusiveNonFollowerTweetLabelRule

extends NonFollowerWithTweetLabelRule(

Drop(Toxicity),

TweetSafetyLabelType.Abusive

)

object AbusiveUqfNonFollowerTweetLabelRule

extends NonFollowerWithUqfTweetLabelRule(

Drop(Toxicity),

TweetSafetyLabelType.Abusive

)

object AbusiveHighRecallTweetLabelRule

extends NonAuthorWithTweetLabelRule(

Drop(Unspecified),

TweetSafetyLabelType.AbusiveHighRecall

)

object AbusiveHighRecallNonFollowerTweetLabelRule

extends NonFollowerWithTweetLabelRule(

Interstitial(PossiblyUndesirable),

TweetSafetyLabelType.AbusiveHighRecall

)

object AutomationTweetLabelRule

extends NonFollowerWithTweetLabelRule(

Drop(Unspecified),

TweetSafetyLabelType.Automation

)

object BystanderAbusiveTweetLabelRule

extends NonAuthorWithTweetLabelRule(

Drop(Unspecified),

TweetSafetyLabelType.BystanderAbusive

)

object BystanderAbusiveNonFollowerTweetLabelRule

extends NonFollowerWithTweetLabelRule(

Drop(Unspecified),

TweetSafetyLabelType.BystanderAbusive

)

abstract class DuplicateContentTweetLabelRule(action: Action)

extends NonAuthorWithTweetLabelRule(

action,

TweetSafetyLabelType.DuplicateContent

)

object DuplicateContentTweetLabelDropRule

extends DuplicateContentTweetLabelRule(Drop(TweetLabelDuplicateContent))

object DuplicateContentTweetLabelTombstoneRule

extends DuplicateContentTweetLabelRule(Tombstone(Epitaph.Unavailable))

object DuplicateMentionTweetLabelRule

extends NonFollowerWithTweetLabelRule(

Drop(Unspecified),

TweetSafetyLabelType.DuplicateMention

)

object DuplicateMentionUqfTweetLabelRule

extends NonFollowerWithUqfTweetLabelRule(

Drop(TweetLabelDuplicateMention),

TweetSafetyLabelType.DuplicateMention

)

object GoreAndViolenceTweetLabelRule

extends ConditionWithTweetLabelRule(

Drop(Unspecified),

And(

NonAuthorViewer,

TweetComposedBefore(TweetSafetyLabelType.GoreAndViolence.DeprecatedAt)

),

TweetSafetyLabelType.GoreAndViolence

)

object LiveLowQualityTweetLabelRule

extends NonAuthorWithTweetLabelRule(

Drop(Unspecified),

TweetSafetyLabelType.LiveLowQuality

)

object LowQualityMentionTweetLabelRule

extends RuleWithConstantAction(

Drop(LowQualityMention),

And(

TweetHasLabelForPerspectivalUser(TweetSafetyLabelType.LowQualityMention),

ViewerHasUqfEnabled

)

)

abstract class NsfwCardImageTweetLabelBaseRule(

override val action: Action,

val additionalCondition: Condition = TrueCondition,

) extends RuleWithConstantAction(

action,

And(

additionalCondition,

TweetHasLabel(TweetSafetyLabelType.NsfwCardImage)

)

)

object NsfwCardImageTweetLabelRule

extends NsfwCardImageTweetLabelBaseRule(

action = Drop(Nsfw),

additionalCondition = NonAuthorViewer,

)

object NsfwCardImageAllUsersTweetLabelRule

extends NsfwCardImageTweetLabelBaseRule(

action = Interstitial(Nsfw)

)

object NsfwCardImageAvoidAllUsersTweetLabelRule

extends NsfwCardImageTweetLabelBaseRule(

action = Avoid(Some(AvoidReason.ContainsNsfwMedia)),

) {

override def enabled: Seq[RuleParam[Boolean]] = Seq(EnableAvoidNsfwRulesParam)

override val fallbackActionBuilder: Option[ActionBuilder[\_ <: Action]] = Some(

new ConstantActionBuilder(Avoid(Some(MightNotBeSuitableForAds))))

}

object NsfwCardImageAvoidAdPlacementAllUsersTweetLabelRule

extends NsfwCardImageTweetLabelBaseRule(

action = Avoid(Some(AvoidReason.ContainsNsfwMedia)),

) {

override def enabled: Seq[RuleParam[Boolean]] = Seq(EnableAvoidNsfwRulesParam)

}

object SearchAvoidTweetNsfwAdminRule

extends RuleWithConstantAction(

Avoid(Some(AvoidReason.ContainsNsfwMedia)),

TweetHasNsfwAdminAuthor

) {

override def enabled: Seq[RuleParam[Boolean]] = Seq(EnableAvoidNsfwRulesParam)

}

object SearchAvoidTweetNsfwUserRule

extends RuleWithConstantAction(

Avoid(Some(AvoidReason.ContainsNsfwMedia)),

TweetHasNsfwUserAuthor

) {

override def enabled: Seq[RuleParam[Boolean]] = Seq(EnableAvoidNsfwRulesParam)

}

object NsfwCardImageAllUsersTweetLabelDropRule

extends NsfwCardImageTweetLabelBaseRule(

action = Drop(Nsfw),

)

object HighProactiveTosScoreTweetLabelDropRule

extends NonAuthorWithTweetLabelRule(

Drop(Unspecified),

TweetSafetyLabelType.HighProactiveTosScore

)

object HighProactiveTosScoreTweetLabelDropSearchRule

extends NonAuthorAndNonFollowerWithTweetLabelRule(

Drop(Unspecified),

TweetSafetyLabelType.HighProactiveTosScore

)

object NsfwHighPrecisionTweetLabelRule

extends NonAuthorWithTweetLabelRule(

Drop(Nsfw),

TweetSafetyLabelType.NsfwHighPrecision

)

object NsfwHighPrecisionAllUsersTweetLabelDropRule

extends TweetHasLabelRule(

Drop(Nsfw),

TweetSafetyLabelType.NsfwHighPrecision

)

object NsfwHighPrecisionInnerQuotedTweetLabelRule

extends ConditionWithTweetLabelRule(

Drop(Nsfw),

And(IsQuotedInnerTweet, NonAuthorViewer),

TweetSafetyLabelType.NsfwHighPrecision

) {

override def enabled: Seq[RuleParam[Boolean]] = Seq(EnableNsfwHpQuotedTweetDropRuleParam)

}

object NsfwHighPrecisionTombstoneInnerQuotedTweetLabelRule

extends ConditionWithTweetLabelRule(

Tombstone(Epitaph.Unavailable),

And(IsQuotedInnerTweet, NonAuthorViewer),

TweetSafetyLabelType.NsfwHighPrecision

) {

override def enabled: Seq[RuleParam[Boolean]] = Seq(EnableNsfwHpQuotedTweetTombstoneRuleParam)

}

object GoreAndViolenceHighPrecisionTweetLabelRule

extends NonAuthorWithTweetLabelRule(

Drop(Nsfw),

TweetSafetyLabelType.GoreAndViolenceHighPrecision

)

object NsfwReportedHeuristicsTweetLabelRule

extends NonAuthorWithTweetLabelRule(

Drop(Nsfw),

TweetSafetyLabelType.NsfwReportedHeuristics

)

object GoreAndViolenceReportedHeuristicsTweetLabelRule

extends NonAuthorWithTweetLabelRule(

Drop(Nsfw),

TweetSafetyLabelType.GoreAndViolenceReportedHeuristics

)

object NsfwHighPrecisionInterstitialAllUsersTweetLabelRule

extends TweetHasLabelRule(

Interstitial(Nsfw),

TweetSafetyLabelType.NsfwHighPrecision

)

with DoesLogVerdict

object GoreAndViolenceHighPrecisionAvoidAllUsersTweetLabelRule

extends TweetHasLabelRule(

Avoid(Some(AvoidReason.ContainsNsfwMedia)),

TweetSafetyLabelType.GoreAndViolenceHighPrecision

) {

override def enabled: Seq[RuleParam[Boolean]] = Seq(EnableAvoidNsfwRulesParam)

override val fallbackActionBuilder: Option[ActionBuilder[\_ <: Action]] = Some(

new ConstantActionBuilder(Avoid(Some(MightNotBeSuitableForAds))))

}

object GoreAndViolenceHighPrecisionAllUsersTweetLabelRule

extends TweetHasLabelRule(

Interstitial(Nsfw),

TweetSafetyLabelType.GoreAndViolenceHighPrecision

)

with DoesLogVerdict {

override def actionSourceBuilder: Option[RuleActionSourceBuilder] = Some(

TweetSafetyLabelSourceBuilder(TweetSafetyLabelType.GoreAndViolenceHighPrecision)

)

}

object NsfwReportedHeuristicsAvoidAllUsersTweetLabelRule

extends TweetHasLabelRule(

Avoid(Some(AvoidReason.ContainsNsfwMedia)),

TweetSafetyLabelType.NsfwReportedHeuristics

) {

override def enabled: Seq[RuleParam[Boolean]] = Seq(EnableAvoidNsfwRulesParam)

override val fallbackActionBuilder: Option[ActionBuilder[\_ <: Action]] = Some(

new ConstantActionBuilder(Avoid(Some(MightNotBeSuitableForAds))))

}

object NsfwReportedHeuristicsAvoidAdPlacementAllUsersTweetLabelRule

extends TweetHasLabelRule(

Avoid(Some(AvoidReason.ContainsNsfwMedia)),

TweetSafetyLabelType.NsfwReportedHeuristics

) {

override def enabled: Seq[RuleParam[Boolean]] = Seq(EnableAvoidNsfwRulesParam)

override val fallbackActionBuilder: Option[ActionBuilder[\_ <: Action]] = Some(

new ConstantActionBuilder(Avoid(Some(MightNotBeSuitableForAds))))

}

object NsfwReportedHeuristicsAllUsersTweetLabelRule

extends TweetHasLabelRule(

Interstitial(Nsfw),

TweetSafetyLabelType.NsfwReportedHeuristics

)

object GoreAndViolenceReportedHeuristicsAllUsersTweetLabelRule

extends TweetHasLabelRule(

Interstitial(Nsfw),

TweetSafetyLabelType.GoreAndViolenceReportedHeuristics

)

object GoreAndViolenceReportedHeuristicsAvoidAllUsersTweetLabelRule

extends TweetHasLabelRule(

Avoid(Some(AvoidReason.ContainsNsfwMedia)),

TweetSafetyLabelType.GoreAndViolenceReportedHeuristics

) {

override def enabled: Seq[RuleParam[Boolean]] = Seq(EnableAvoidNsfwRulesParam)

override val fallbackActionBuilder: Option[ActionBuilder[\_ <: Action]] = Some(

new ConstantActionBuilder(Avoid(Some(MightNotBeSuitableForAds))))

}

object GoreAndViolenceReportedHeuristicsAvoidAdPlacementAllUsersTweetLabelRule

extends TweetHasLabelRule(

Avoid(Some(AvoidReason.ContainsNsfwMedia)),

TweetSafetyLabelType.GoreAndViolenceReportedHeuristics

) {

override def enabled: Seq[RuleParam[Boolean]] = Seq(EnableAvoidNsfwRulesParam)

override val fallbackActionBuilder: Option[ActionBuilder[\_ <: Action]] = Some(

new ConstantActionBuilder(Avoid(Some(MightNotBeSuitableForAds))))

}

object GoreAndViolenceHighPrecisionAllUsersTweetLabelDropRule

extends TweetHasLabelRule(

Drop(Nsfw),

TweetSafetyLabelType.GoreAndViolenceHighPrecision

)

object NsfwReportedHeuristicsAllUsersTweetLabelDropRule

extends TweetHasLabelRule(

Drop(Nsfw),

TweetSafetyLabelType.NsfwReportedHeuristics

)

object GoreAndViolenceReportedHeuristicsAllUsersTweetLabelDropRule

extends TweetHasLabelRule(

Drop(Nsfw),

TweetSafetyLabelType.GoreAndViolenceReportedHeuristics

)

object NsfwHighRecallTweetLabelRule

extends NonAuthorWithTweetLabelRule(

Drop(Nsfw),

TweetSafetyLabelType.NsfwHighRecall

)

object NsfwHighRecallAllUsersTweetLabelDropRule

extends TweetHasLabelRule(

Drop(Nsfw),

TweetSafetyLabelType.NsfwHighRecall

)

abstract class PdnaTweetLabelRule(

override val action: Action,

val additionalCondition: Condition)

extends ConditionWithTweetLabelRule(

action,

And(NonAuthorViewer, additionalCondition),

TweetSafetyLabelType.Pdna

)

object PdnaTweetLabelRule extends PdnaTweetLabelRule(Drop(PdnaTweet), Condition.True)

object PdnaTweetLabelTombstoneRule

extends PdnaTweetLabelRule(Tombstone(Epitaph.Unavailable), Condition.True)

object PdnaQuotedTweetLabelTombstoneRule

extends PdnaTweetLabelRule(Tombstone(Epitaph.Unavailable), Condition.IsQuotedInnerTweet) {

override def enabled: Seq[RuleParam[Boolean]] = Seq(EnablePdnaQuotedTweetTombstoneRuleParam)

}

object PdnaAllUsersTweetLabelRule

extends TweetHasLabelRule(

Drop(Unspecified),

TweetSafetyLabelType.Pdna

)

object SearchBlacklistTweetLabelRule

extends NonAuthorWithTweetLabelRule(

Drop(Unspecified),

TweetSafetyLabelType.SearchBlacklist

)

object SearchBlacklistHighRecallTweetLabelDropRule

extends NonAuthorAndNonFollowerWithTweetLabelRule(

Drop(Unspecified),

TweetSafetyLabelType.SearchBlacklistHighRecall

)

abstract class SpamTweetLabelRule(

override val action: Action,

val additionalCondition: Condition)

extends ConditionWithTweetLabelRule(

action,

And(NonAuthorViewer, additionalCondition),

TweetSafetyLabelType.Spam

)

with DoesLogVerdict

object SpamTweetLabelRule extends SpamTweetLabelRule(Drop(TweetLabeledSpam), Condition.True)

object SpamTweetLabelTombstoneRule

extends SpamTweetLabelRule(Tombstone(Epitaph.Unavailable), Condition.True)

object SpamQuotedTweetLabelTombstoneRule

extends SpamTweetLabelRule(Tombstone(Epitaph.Unavailable), Condition.IsQuotedInnerTweet) {

override def enabled: Seq[RuleParam[Boolean]] = Seq(EnableSpamQuotedTweetTombstoneRuleParam)

}

object SpamAllUsersTweetLabelRule

extends TweetHasLabelRule(

Drop(Unspecified),

TweetSafetyLabelType.Spam

)

abstract class BounceTweetLabelRule(override val action: Action)

extends NonAuthorWithTweetLabelRule(

action,

TweetSafetyLabelType.Bounce

)

object BounceTweetLabelRule extends BounceTweetLabelRule(Drop(Bounce))

object BounceTweetLabelTombstoneRule extends BounceTweetLabelRule(Tombstone(Epitaph.Bounced))

abstract class BounceOuterTweetLabelRule(override val action: Action)

extends ConditionWithTweetLabelRule(

action,

And(Not(Condition.IsQuotedInnerTweet), NonAuthorViewer),

TweetSafetyLabelType.Bounce

)

object BounceOuterTweetTombstoneRule extends BounceOuterTweetLabelRule(Tombstone(Epitaph.Bounced))

object BounceQuotedTweetTombstoneRule

extends ConditionWithTweetLabelRule(

Tombstone(Epitaph.Bounced),

Condition.IsQuotedInnerTweet,

TweetSafetyLabelType.Bounce

)

object BounceAllUsersTweetLabelRule

extends TweetHasLabelRule(

Drop(Bounce),

TweetSafetyLabelType.Bounce

)

abstract class SpamHighRecallTweetLabelRule(action: Action)

extends NonAuthorWithTweetLabelRule(

action,

TweetSafetyLabelType.SpamHighRecall

)

object SpamHighRecallTweetLabelDropRule

extends SpamHighRecallTweetLabelRule(Drop(SpamHighRecallTweet))

object SpamHighRecallTweetLabelTombstoneRule

extends SpamHighRecallTweetLabelRule(Tombstone(Epitaph.Unavailable))

object UntrustedUrlAllViewersTweetLabelRule

extends TweetHasLabelRule(

Drop(Unspecified),

TweetSafetyLabelType.UntrustedUrl

)

object DownrankSpamReplyAllViewersTweetLabelRule

extends TweetHasLabelRule(

Drop(Unspecified),

TweetSafetyLabelType.DownrankSpamReply

) {

override def enabled: Seq[RuleParam[Boolean]] =

Seq(EnableDownrankSpamReplySectioningRuleParam)

}

object UntrustedUrlTweetLabelRule

extends NonAuthorWithTweetLabelRule(

Drop(Unspecified),

TweetSafetyLabelType.UntrustedUrl

)

object DownrankSpamReplyTweetLabelRule

extends NonAuthorWithTweetLabelRule(

Drop(Unspecified),

TweetSafetyLabelType.DownrankSpamReply

) {

override def enabled: Seq[RuleParam[Boolean]] =

Seq(EnableDownrankSpamReplySectioningRuleParam)

}

object UntrustedUrlUqfNonFollowerTweetLabelRule

extends NonFollowerWithUqfTweetLabelRule(

Drop(UntrustedUrl),

TweetSafetyLabelType.UntrustedUrl

)

object DownrankSpamReplyUqfNonFollowerTweetLabelRule

extends NonFollowerWithUqfTweetLabelRule(

Drop(SpamReplyDownRank),

TweetSafetyLabelType.DownrankSpamReply

) {

override def enabled: Seq[RuleParam[Boolean]] =

Seq(EnableDownrankSpamReplySectioningRuleParam)

}

object NsfaHighRecallTweetLabelRule

extends RuleWithConstantAction(

Drop(Unspecified),

And(

NonAuthorViewer,

TweetHasLabel(TweetSafetyLabelType.NsfaHighRecall)

)

)

object NsfaHighRecallTweetLabelInterstitialRule

extends RuleWithConstantAction(

Interstitial(Unspecified),

And(

NonAuthorViewer,

TweetHasLabel(TweetSafetyLabelType.NsfaHighRecall)

)

)

object NsfwVideoTweetLabelDropRule

extends NonAuthorWithTweetLabelRule(

Drop(Nsfw),

TweetSafetyLabelType.NsfwVideo

) {

override def enabled: Seq[RuleParam[Boolean]] = Seq(EnableNsfwTextSectioningRuleParam)

}

object NsfwTextTweetLabelDropRule

extends NonAuthorWithTweetLabelRule(

Drop(Nsfw),

TweetSafetyLabelType.NsfwText

)

object NsfwVideoAllUsersTweetLabelDropRule

extends TweetHasLabelRule(

Drop(Nsfw),

TweetSafetyLabelType.NsfwVideo

)

object NsfwTextAllUsersTweetLabelDropRule

extends TweetHasLabelRule(

Drop(Nsfw),

TweetSafetyLabelType.NsfwText

) {

override def enabled: Seq[RuleParam[Boolean]] = Seq(EnableNsfwTextSectioningRuleParam)

}

abstract class BaseLowQualityTweetLabelRule(action: Action)

extends RuleWithConstantAction(

action,

And(

TweetHasLabel(TweetSafetyLabelType.LowQuality),

TweetComposedBefore(PublicInterest.PolicyConfig.LowQualityProxyLabelStart),

NonAuthorViewer

)

)

with DoesLogVerdict {

override def actionSourceBuilder: Option[RuleActionSourceBuilder] = Some(

TweetSafetyLabelSourceBuilder(TweetSafetyLabelType.LowQuality))

}

object LowQualityTweetLabelDropRule extends BaseLowQualityTweetLabelRule(Drop(LowQualityTweet))

object LowQualityTweetLabelTombstoneRule

extends BaseLowQualityTweetLabelRule(Tombstone(Epitaph.Unavailable))

abstract class SafetyCrisisLevelDropRule(level: Int, condition: Condition = TrueCondition)

extends ConditionWithTweetLabelRule(

Drop(Unspecified),

And(

NonAuthorViewer,

condition,

TweetHasSafetyLabelWithScoreEqInt(TweetSafetyLabelType.SafetyCrisis, level)

),

TweetSafetyLabelType.SafetyCrisis

)

object SafetyCrisisAnyLevelDropRule

extends NonAuthorWithTweetLabelRule(

Drop(Unspecified),

TweetSafetyLabelType.SafetyCrisis

)

object SafetyCrisisLevel2DropRule extends SafetyCrisisLevelDropRule(2, Not(ViewerDoesFollowAuthor))

object SafetyCrisisLevel3DropRule extends SafetyCrisisLevelDropRule(3, Not(ViewerDoesFollowAuthor))

object SafetyCrisisLevel4DropRule extends SafetyCrisisLevelDropRule(4)

abstract class SafetyCrisisLevelSectionRule(level: Int)

extends ConditionWithNotInnerCircleOfFriendsRule(

ConversationSectionAbusiveQuality,

And(

TweetHasLabel(TweetSafetyLabelType.SafetyCrisis),

TweetHasSafetyLabelWithScoreEqInt(TweetSafetyLabelType.SafetyCrisis, level))

) {

override def actionSourceBuilder: Option[RuleActionSourceBuilder] = Some(

TweetSafetyLabelSourceBuilder(TweetSafetyLabelType.SafetyCrisis))

}

object SafetyCrisisLevel3SectionRule

extends SafetyCrisisLevelSectionRule(3)

with DoesLogVerdictDecidered {

override def verdictLogDeciderKey = DeciderKey.EnableDownlevelRuleVerdictLogging

}

object SafetyCrisisLevel4SectionRule

extends SafetyCrisisLevelSectionRule(4)

with DoesLogVerdictDecidered {

override def verdictLogDeciderKey = DeciderKey.EnableDownlevelRuleVerdictLogging

}

object DoNotAmplifyDropRule

extends NonFollowerWithTweetLabelRule(Drop(Unspecified), TweetSafetyLabelType.DoNotAmplify)

object DoNotAmplifyAllViewersDropRule

extends TweetHasLabelRule(Drop(Unspecified), TweetSafetyLabelType.DoNotAmplify)

object DoNotAmplifySectionRule

extends ConditionWithNotInnerCircleOfFriendsRule(

ConversationSectionAbusiveQuality,

TweetHasLabel(TweetSafetyLabelType.DoNotAmplify))

object HighPSpammyScoreAllViewerDropRule

extends TweetHasLabelRule(Drop(Unspecified), TweetSafetyLabelType.HighPSpammyTweetScore)

object HighPSpammyTweetScoreSearchTweetLabelDropRule

extends RuleWithConstantAction(

action = Drop(Unspecified),

condition = And(

LoggedOutOrViewerNotFollowingAuthor,

TweetHasLabelWithScoreAboveThreshold(

TweetSafetyLabelType.HighPSpammyTweetScore,

ModelScoreThresholds.HighPSpammyTweetScoreThreshold)

)

)

with DoesLogVerdictDecidered {

override def enabled: Seq[RuleParam[Boolean]] = Seq(

EnableHighPSpammyTweetScoreSearchTweetLabelDropRuleParam)

override def actionSourceBuilder: Option[RuleActionSourceBuilder] = Some(

TweetSafetyLabelSourceBuilder(TweetSafetyLabelType.HighPSpammyTweetScore))

override def verdictLogDeciderKey: DeciderKey.Value =

DeciderKey.EnableSpammyTweetRuleVerdictLogging

}

object AdsManagerDenyListAllUsersTweetLabelRule

extends TweetHasLabelRule(

Drop(Unspecified),

TweetSafetyLabelType.AdsManagerDenyList

)

abstract class SmyteSpamTweetLabelRule(action: Action)

extends NonAuthorWithTweetLabelRule(

action,

TweetSafetyLabelType.SmyteSpamTweet

) {

override def enabled: Seq[RuleParam[Boolean]] = Seq(EnableSmyteSpamTweetRuleParam)

}

object SmyteSpamTweetLabelDropRule extends SmyteSpamTweetLabelRule(Drop(TweetLabeledSpam))

object SmyteSpamTweetLabelTombstoneRule

extends SmyteSpamTweetLabelRule(Tombstone(Epitaph.Unavailable))

object SmyteSpamTweetLabelDropSearchRule extends SmyteSpamTweetLabelRule(Drop(Unspecified))

object HighSpammyTweetContentScoreSearchLatestTweetLabelDropRule

extends RuleWithConstantAction(

action = Drop(Unspecified),

condition = And(

Not(IsTweetInTweetLevelStcmHoldback),

LoggedOutOrViewerNotFollowingAuthor,

TweetHasLabelWithScoreAboveThresholdWithParam(

TweetSafetyLabelType.HighSpammyTweetContentScore,

HighSpammyTweetContentScoreSearchLatestProdTweetLabelDropRuleThresholdParam)

)

)

with DoesLogVerdictDecidered {

override def actionSourceBuilder: Option[RuleActionSourceBuilder] = Some(

TweetSafetyLabelSourceBuilder(TweetSafetyLabelType.HighSpammyTweetContentScore))

override def verdictLogDeciderKey: DeciderKey.Value =

DeciderKey.EnableSpammyTweetRuleVerdictLogging

}

object HighSpammyTweetContentScoreSearchTopTweetLabelDropRule

extends RuleWithConstantAction(

action = Drop(Unspecified),

condition = And(

Not(IsTweetInTweetLevelStcmHoldback),

LoggedOutOrViewerNotFollowingAuthor,

TweetHasLabelWithScoreAboveThresholdWithParam(

TweetSafetyLabelType.HighSpammyTweetContentScore,

HighSpammyTweetContentScoreSearchTopProdTweetLabelDropRuleThresholdParam)

)

)

with DoesLogVerdictDecidered {

override def actionSourceBuilder: Option[RuleActionSourceBuilder] = Some(

TweetSafetyLabelSourceBuilder(TweetSafetyLabelType.HighSpammyTweetContentScore))

override def verdictLogDeciderKey: DeciderKey.Value =

DeciderKey.EnableSpammyTweetRuleVerdictLogging

}

object HighSpammyTweetContentScoreTrendsTopTweetLabelDropRule

extends RuleWithConstantAction(

action = Drop(Unspecified),

condition = And(

Not(IsTweetInTweetLevelStcmHoldback),

LoggedOutOrViewerNotFollowingAuthor,

IsTrendClickSourceSearchResult,

TweetHasLabelWithScoreAboveThresholdWithParam(

TweetSafetyLabelType.HighSpammyTweetContentScore,

HighSpammyTweetContentScoreTrendTopTweetLabelDropRuleThresholdParam)

)

)

with DoesLogVerdictDecidered {

override def actionSourceBuilder: Option[RuleActionSourceBuilder] = Some(

TweetSafetyLabelSourceBuilder(TweetSafetyLabelType.HighSpammyTweetContentScore))

override def verdictLogDeciderKey: DeciderKey.Value =

DeciderKey.EnableSpammyTweetRuleVerdictLogging

}

object HighSpammyTweetContentScoreTrendsLatestTweetLabelDropRule

extends RuleWithConstantAction(

action = Drop(Unspecified),

condition = And(

Not(IsTweetInTweetLevelStcmHoldback),

LoggedOutOrViewerNotFollowingAuthor,

IsTrendClickSourceSearchResult,

TweetHasLabelWithScoreAboveThresholdWithParam(

TweetSafetyLabelType.HighSpammyTweetContentScore,

HighSpammyTweetContentScoreTrendLatestTweetLabelDropRuleThresholdParam)

)

)

with DoesLogVerdictDecidered {

override def actionSourceBuilder: Option[RuleActionSourceBuilder] = Some(

TweetSafetyLabelSourceBuilder(TweetSafetyLabelType.HighSpammyTweetContentScore))

override def verdictLogDeciderKey: DeciderKey.Value =

DeciderKey.EnableSpammyTweetRuleVerdictLogging

}

object GoreAndViolenceTopicHighRecallTweetLabelRule

extends NonAuthorWithTweetLabelRule(

Drop(Unspecified),

TweetSafetyLabelType.GoreAndViolenceTopicHighRecall

) {

override def enabled: Seq[RuleParam[Boolean]] = Seq(

EnableGoreAndViolenceTopicHighRecallTweetLabelRule)

}

object CopypastaSpamAllViewersTweetLabelRule

extends TweetHasLabelRule(

Drop(Unspecified),

TweetSafetyLabelType.CopypastaSpam

) {

override def enabled: Seq[RuleParam[Boolean]] =

Seq(EnableCopypastaSpamSearchDropRule)

}

object CopypastaSpamAllViewersSearchTweetLabelRule

extends TweetHasLabelRule(

Drop(Unspecified),

TweetSafetyLabelType.CopypastaSpam

) {

override def enabled: Seq[RuleParam[Boolean]] =

Seq(EnableCopypastaSpamSearchDropRule)

}

object CopypastaSpamNonFollowerSearchTweetLabelRule

extends NonFollowerWithTweetLabelRule(

Drop(Unspecified),

TweetSafetyLabelType.CopypastaSpam

) {

override def enabled: Seq[RuleParam[Boolean]] =

Seq(EnableCopypastaSpamSearchDropRule)

}

object CopypastaSpamAbusiveQualityTweetLabelRule

extends ConditionWithNotInnerCircleOfFriendsRule(

ConversationSectionAbusiveQuality,

TweetHasLabel(TweetSafetyLabelType.CopypastaSpam)

)

with DoesLogVerdictDecidered {

override def enabled: Seq[RuleParam[Boolean]] = Seq(

EnableCopypastaSpamDownrankConvosAbusiveQualityRule)

override def actionSourceBuilder: Option[RuleActionSourceBuilder] = Some(

TweetSafetyLabelSourceBuilder(TweetSafetyLabelType.CopypastaSpam))

override def verdictLogDeciderKey = DeciderKey.EnableDownlevelRuleVerdictLogging

}

object DynamicProductAdLimitedEngagementTweetLabelRule

extends TweetHasLabelRule(

LimitedEngagements(LimitedEngagementReason.DynamicProductAd),

TweetSafetyLabelType.DynamicProductAd)

object SkipTweetDetailLimitedEngagementTweetLabelRule

extends AlwaysActRule(LimitedEngagements(LimitedEngagementReason.SkipTweetDetail)) {

override def enabled: Seq[RuleParam[Boolean]] = Seq(

SkipTweetDetailLimitedEngagementRuleEnabledParam)

}

object DynamicProductAdDropTweetLabelRule

extends TweetHasLabelRule(Drop(Unspecified), TweetSafetyLabelType.DynamicProductAd)

object NsfwTextHighPrecisionTweetLabelDropRule

extends RuleWithConstantAction(

Drop(Reason.Nsfw),

And(

NonAuthorViewer,

Or(

TweetHasLabel(TweetSafetyLabelType.ExperimentalSensitiveIllegal2),

TweetHasLabel(TweetSafetyLabelType.NsfwTextHighPrecision)

)

)

)

with DoesLogVerdict {

override def enabled: Seq[RuleParam[Boolean]] = Seq(EnableNsfwTextHighPrecisionDropRuleParam)

override def actionSourceBuilder: Option[RuleActionSourceBuilder] = Some(

TweetSafetyLabelSourceBuilder(TweetSafetyLabelType.NsfwTextHighPrecision))

}

object ExperimentalNudgeLabelRule

extends TweetHasLabelRule(

TweetVisibilityNudge(TweetVisibilityNudgeReason.ExperimentalNudgeSafetyLabelReason),

TweetSafetyLabelType.ExperimentalNudge) {

override def enabled: Seq[RuleParam[Boolean]] = Seq(EnableExperimentalNudgeEnabledParam)

}

object NsfwTextTweetLabelAvoidRule

extends RuleWithConstantAction(

Avoid(),

Or(

TweetHasLabel(TweetSafetyLabelType.ExperimentalSensitiveIllegal2),

TweetHasLabel(TweetSafetyLabelType.NsfwTextHighPrecision)

)

) {

override def actionSourceBuilder: Option[RuleActionSourceBuilder] = Some(

TweetSafetyLabelSourceBuilder(TweetSafetyLabelType.NsfwTextHighPrecision))

}

object DoNotAmplifyTweetLabelAvoidRule

extends TweetHasLabelRule(

Avoid(),

TweetSafetyLabelType.DoNotAmplify

) {

override val fallbackActionBuilder: Option[ActionBuilder[\_ <: Action]] = Some(

new ConstantActionBuilder(Avoid(Some(MightNotBeSuitableForAds))))

}

object NsfaHighPrecisionTweetLabelAvoidRule

extends TweetHasLabelRule(

Avoid(),

TweetSafetyLabelType.NsfaHighPrecision

) {

override val fallbackActionBuilder: Option[ActionBuilder[\_ <: Action]] = Some(

new ConstantActionBuilder(Avoid(Some(MightNotBeSuitableForAds))))

}

object NsfwHighPrecisionTweetLabelAvoidRule

extends TweetHasLabelRule(

Avoid(Some(AvoidReason.ContainsNsfwMedia)),

TweetSafetyLabelType.NsfwHighPrecision

) {

override val fallbackActionBuilder: Option[ActionBuilder[\_ <: Action]] = Some(

new ConstantActionBuilder(Avoid(Some(MightNotBeSuitableForAds))))

}

object NsfwHighRecallTweetLabelAvoidRule

extends TweetHasLabelRule(

Avoid(Some(AvoidReason.ContainsNsfwMedia)),

TweetSafetyLabelType.NsfwHighRecall

) {

override val fallbackActionBuilder: Option[ActionBuilder[\_ <: Action]] = Some(

new ConstantActionBuilder(Avoid(Some(MightNotBeSuitableForAds))))

}