UNIVERSIDADE FEDERAL DO RIO DE JANEIRO

INSTITUTO DE MATEMÁTICA

ESPECIALIZAÇÃO EM CIÊNCIA DE DADOS

MARCEL DANTAS DE QUINTELA

PREVISÃO DE TRAÇOS DE PERSONALIDADE POR MEIO DO USO DE SMARTPHONES

RIO DE JANEIRO

2022

MARCEL DANTAS DE QUINTELA

PREVISÃO DE TRAÇOS DE PERSONALIDADE POR MEIO DO USO DE SMARTPHONES

Trabalho de conclusão de curso de especialização apresentado ao Instituto de Matemática da Universidade Federal do Rio de Janeiro como parte dos requisitos para obtenção do grau de especialista em Ciência de Dados.

Orientador: Profa. Thais Cristina Oliveira da Fonseca

RIO DE JANEIRO

2021

Q6p Quintela, Marcel Dantas de
Previsão de Traços de Personalidade por meio do uso de Smartphones / Marcel Dantas de Quintela. -- Rio de Janeiro, 2022.
70 f.

Orientadora: Thais Cristina de Oliveira Fonseca.
Trabalho de conclusão de curso (especialização)- Universidade Federal do Rio de Janeiro, Instituto de Matemática, Ciência de Dados, 2022.

1. Big Five, c. Classificação, 3. Machine Learning, 4. Personalidade, 5. Smartphone I. Fonseca, Thais Cristina de Oliveira da, orient. II. Título.

MARCEL DANTAS DE QUINTELA

PREVISÃO DE TRAÇOS DE PERSONALIDADE POR MEIO DO USO DE SMARTPHONES

Trabalho de conclusão de curso de especialização apresentado ao Instituto de Matemática da Universidade Federal do Rio de Janeiro como parte dos requisitos para obtenção do grau de especialista em Ciência de Dados.

Aprovado em _____ de setembro de 2022.

BANCA EXAMINADORA:

_____
Nome do Professor, Titulação (Instituição)

_____
Nome do Professor, Titulação (Instituição)

_____
Nome do Professor, Titulação (Instituição)

Dedicatória

À minha esposa e ao meu filho, por
eles que faço tudo em minha vida.

AGRADECIMENTOS

Meus sinceros agradecimentos a todos aqueles que de alguma forma doaram um pouco de si para que a conclusão deste trabalho se tornasse possível:

A Deus, o que seria de mim sem a fé que eu tenho nele.

A minha esposa Luciana pelo apoio incondicional em todos os momentos e por assumir integralmente os cuidados com nosso pequeno Lucas nos momentos em que tive que voltar minha atenção para a conclusão desta etapa.

Aos meus pais, Antônio e Marielze; meu irmão, Marcílio.

À Ana Maria que não mede esforções em cuidar de todos não e pelo amor incondicional pelo nosso Lucas.

À Rosane e Ronaldo por estarem sempre disponíveis.

Aos professores desta pós-graduação por seus ensinamentos e pela dedicação.

Aos funcionários e colaboradores que nos acompanharam nos sábados de aula e nas demais tarefas administrativas.

Aos colegas da turma, em especial aos amigos Alexandre, Paschoal, Thamirys e Deborah, pelos momentos de companheirismo e horas pós aula que tínhamos de troca de conhecimento e discussões.

“A criação bem-sucedida de inteligência artificial seroa o maior evento na história da humanidade. Infelizmente, pode também ser o último, a menos que aprendamos a evitar os riscos”

Stephen Hawking

RESUMO

O presente estudo buscou classificar por meio de modelos de *machine learning* fatores de personalidade definidos pela teoria do *Big Five* a partir de dados da a frequência e a duração do comportamento real, manifestado pelo uso de aplicativos instalados em *smartphones*, como uma maneira alternativa de traçar os perfis de personalidade além dos instrumentos de autorrelato existentes. A pesquisa foi realizada pelo projeto *PhoneStudy.org*, em 2016, contando com a participação de 137 indivíduos, cujos foram submetidos a um teste de avaliação psicológica para mensurar seus níveis de personalidade e posteriormente tiveram um aplicativo, desenvolvido pelo projeto, instalado em seus *smartphones* para registrar dados durante 60 dias. Após procedimentos de análise exploratória de dados, foram desenhados os modelos *K-Nearest Neighbors* (KNN), *Linear Discriminant Analysis* (LDA) e o *Random Forest* (RF) cuja acurácia média ficou entre 50% e 54%, porém compondo o *Ensemble Voting Classifier* com esses modelos a acurácia subiu para cerca de 70%. Apesar dos resultados satisfatórios, entendemos que uma trabalhos futuros conduzidos com uma amostra maior e com a disponibilização dos dados não agrupados em categorias de uso dos *apps*, mas sim os brutos para que se possa investigar essas associações por meio de outros métodos, possam ajudar na melhoria da classificação desses modelos.

Palavras-chave: *Big Five*, Classificação, *Machine Learning*, Personalidade, *Smartphone*

PREDICTING PERSONALITY TRAITS FROM SMARTPHONES USE

This study sought to classify through machine learning models personality factors by the Big Five theory from data on the frequency and duration of real behavior, manifested using applications installed on smartphones, as an alternative way of tracing personality profiles in addition to existing self-report instruments. The research was managed by the PhoneStudy.org project, in 2016, with the participation of 137 individuals, who underwent a psychological assessment test to measure their personality levels and later had an application, developed by the project, installed on their smartphones. to record data for 60 days. After exploratory data analysis procedures, the *K-Nearest Neighbors* (KNN), *Linear Discriminant Analysis* (LDA) and *Random Forest* (RF) models were drafted, whose average accuracy was between 50% and 54%, but composing the ensemble *Voting Classifier* with these models the accuracy rose to about 70%. Despite the satisfactory results, we believe that future work conducted with a larger sample and with the availability of data not grouped into app usage categories, but the raw data so that these associations can be investigated through other methods, can help in the improvement in the classification of these models

Keywords: Big Five, Classification, Machine Learning, Personality, Smartphone.

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

SUMÁRIO

1. INTRODUÇÃO

Para grande parte das pessoas em todo mundo, os aparelhos de smartphones se tornaram componentes indispensáveis em suas vidas. E com as constantes evoluções em diversos ramos tecnológicos, hoje em dia, os smartphones vão além de meros dispositivos de comunicação. Eles contam com uma gama de funcionalidade que permite aos seus usuários, por exemplo, em alguns segundos enviar instantaneamente fotos ou documentos para outra pessoa em qualquer lugar do globo.

Segundo (TURNER, 2022), em 2021, o número atual de usuários de smartphones no mundo é de 6,648 bilhões, o que significa que 83,40% da população mundial possui um smartphone. Conforme dados de inteligência em tempo real da GSMA[1], existem hoje mais de 10,92 bilhões de conexões móveis em todo o mundo, o que supera a estimativa da população mundial atual de 7,97 bilhões. Esses dados significam que existem 2,94 bilhões de conexões móveis a mais do que pessoas em todo o mundo.

Hoje no Brasil, segundo dados da Anatel para o mês de julho de 2022, são 260,2 milhões de celulares, isso representa que a cada 100 habitantes existem 121,18 celulares habilitados (TELECO, 2022). Segundo o relatório *State of Mobile 2022*, a população brasileira gasta pelo menos um terço das horas acordadas usando apps e navegando na internet. O levantamento mostra uma tenência no aumento da média de horas de uso do smartphone no Brasil, registrando em 2021, 5,4 horas de uso, 0,6 horas a mais que a média mundial. (DATA.AI, 2022)

Em pesquisa realizada em abril de 2022 (MOBILE TIME, 2022), mostra que: WhatsApp, Instagram, Facebook e Youtube, nesta ordem, são os aplicativos com mais frequência e duração de uso no Brasil.

Considerando a quantidade substancial de usuários e a grande variabilidade de aplicações utilizadas, os dados que esses dispositivos coletam incessantemente são uma verdadeira mina de ouro.

---

[1] GMSA: Organização que representa os interesses das operadoras de rede móvel em todo o mundo. (https://www.gsma.com)

Assim como nas caixas-pretas dos aviões, nossos smartphones gera e preserva uma grande quantidade de dados que podem fornecer informações diretas de onde estamos, por onde passamos, o que acessamos, ouvimos, visualizamos, clicamos e muitas outras. Estes dados, podem auxiliar na descoberta de pessoas consideradas desaparecidas ou mesmo indícios para desvendar fraudes e outros crimes.

Além disso, o uso destes dados com auxílio de técnicas de inteligência artificial pode fornecer informações que estão escondidas, embaralhadas nessa grande massa de dados. Padrões de acesso de sites, frequência e duração de uso de determinados aplicativos podem ser *inputs* de equações que podem retornar pistas de diversos padrões comportamentais.

É nesse prisma que este trabalho propõe classificar de acordo com técnicas de *machine learning*, traços de personalidade pautados na teoria dos cinco grandes fatores da Personalidade – *Big Five* – por meio dos padrões de frequência e duração no uso de aplicativos de smartphones.

Assim, em um mundo com cada vez mais serviços automatizados, conhecer um pouco da personalidade de seus clientes e usuários pode ser utilizado para personalizar estes serviços ou mesmo usar essas informações como entrada de processos avaliação de crédito ou de investimentos, por exemplo.

Para isso, foram usados dados de um projeto da *Ludwig-Maximilians-Universität München* (LMU) que investigou as relações entre variáveis psicológicas e os comportamentos registrados via smartphones. Os quais foram coletados entre setembro de 2014 e agosto de 2015 na cidade de Munique na Alemanha.

Além desta introdução este trabalho contará com mais quatro capítulos. Na sequência, um capítulo teórico abordando os cinco grandes fatores de personalidade, trazendo um pouco do desenvolvimento desta teoria assim como a descrição de cada fator de personalidade. O terceiro capítulo, abordará os materiais e métodos utilizados, sendo apresentado as ferramentas, algoritmos, formulações matemáticas bem como técnicas e procedimentos utilizados no tratamento dos dados e modelagem. O quarto capítulo mostrará os resultados, tanto do ponto de vista exploratório quando dos modelos propostos. Por fim, as considerações finais, retomando pontos essenciais, trazendo evidências e possíveis trabalhos futuros.

2. MODELO DOS CINCO GRANDES FATORES DE PERSONALIDADE

O Big Five ou Modelo dos Cinco Grandes Fatores é uma teoria psicológica que permite de traçar e avaliar a personalidade das pessoas. De acordo com o processo de avaliação conduzido por profissionais da psicologia, é possível mapear o estado psicológico de um indivíduo e assim compreender como ele tende a pensar, o que ele poderia sentir e como pode reagir em diferentes contextos e situações.

O Big Five surgiu por meio dos estudos da Teoria dos Traços de Personalidade, a qual afirma que os traços são os principais determinantes do comportamento e servem de base para a consistência de respostas comportamentais em diferentes situações. Esse modelo é dito dimensional, pois assume que os traços podem ser condensados em grandes fatores, e esta é a base para comparar e contrastar indivíduos e grupos. (ENDLER e MAGNUSSON ,1976 apud FONSECA, 2018).

Os primeiros estudos em busca de definir o caráter, conduzidos por Galton em 1934, que mapeou, dentro da língua inglesa, cerca de 9.000 (nove mil) palavras que podiam se referir a traços de um indivíduo. Em 1936, Allport e Odbert, com base no trabalho de Galton, encontraram 18.000 termos que podiam estar relacionados à descrição da personalidade. Cattel, em 1946, utilizando de métodos estatísticos de análise fatorial, concluiu que 16 fatores seriam necessários para descrever a personalidade de um indivíduo. (FONSECA, 2018).

Atualmente, o modelo dos cinco grandes fatores, proposto por McCrae e Costa (1997), é considerado o de maior consenso entre pesquisadores da personalidade, sendo possível a realização de estudos em diferentes países e culturas. (FONSECA, 2018).

Os cinco traços amplos ou fatores de personalidade, que são representados pela sigla em inglês OCEAN:

- Openness to experience → Abertura a Experiências.
- Conscientiousness →Conscienciosidade.
- Extraversion →Extroversão.
- Agreeableness →Amabilidade.
- Neuroticism →Neuroticismo ou Estabilidade Emocional.

Segundo Costa e McCrae (2007):

- Abertura avalia a proatividade e apreciação da experiência por si só; tolerância e exploração do que não é familiar;
- Conscienciosidade, o grau de organização, persistência e motivação no comportamento dirigido para os objetivos. Compara pessoas confiáveis e determinadas com aquelas que são apáticas e descuidadas;
- Extroversão, avalia a quantidade e intensidade de interações interpessoais; nível de atividade; necessidade de estimulação; e capacidade de se sentir alegre;
- Amabilidade, a qualidade da orientação interpessoal ao longo de um contínuo da compaixão ao antagonismo em pensamentos, sentimentos e ações; e
- Neuroticismo avalia o ajustamento versus instabilidade emocional. Identifica indivíduos propensos a perturbações.

Com base nas avaliações de cada fator, Costa e McCrae (2007) apud Fonseca (2018), relaciona, na Tabela 1, algumas características segundo os escores de cada fator de personalidade.

Tabela 1: Os Cinco Grandes Fatores de Personalidade

FATOR	ALTOS ESCORES	BAIXOS ESCORES
Abertura	Curioso, interesses amplos, criativo, original, imaginativo, não tradicional.	Convencional, sensato, interesses limitados, não ligado à arte, não analítico
Conscienciosidade	Organizado, confiável, trabalhador, autodisciplinado, pontual, escrupuloso, asseado, ambicioso, perseverante.	Sem objetivos, não confiável, preguiçoso, descuidado, negligente, relaxado, fraco, hedonista.
Extroversão	Sociável, ativo, falante, gosta de estar com pessoas, otimista, divertido, afetuoso.	Reservado, sóbrio, contraído, indiferente, voltado para tarefas, desinteressado, quieto.
Amabilidade	Generoso, bondoso, confiante, prestativo, clemente, crédulo, honesto	Cínico, rude, desconfiado, não cooperativo, vingativo, inescrupuloso, irritável, manipulador.
Neuroticismo	Preocupado, nervoso, emotivo, inseguro, inadequado, hipocondríaco	Calmo, descontraído, não emotivo, forte, seguro, autoconfiante.

Fonte: Fonseca (2018).

Algumas pesquisas sugerem que influências biológicas e ambientais desempenham um papel na formação de nossas personalidades. Estudos com gêmeos sugerem que tanto a natureza quanto a criação desempenham um papel no desenvolvimento de cada um dos cinco traços de personalidade. (JANG, LIVESLEY e VEMON, 1996).

Estudos de fatores genéticos e ambientais das cinco características de personalidade analisou 123 pares de gêmeos idênticos e 127 pares de gêmeos fraternos. Os resultados sugeriram que a herdabilidade de cada traço de personalidade foi de 53% para extroversão, 41% para amabilidade, 44% para conscienciosidade, 41% para neuroticismo e 61% para abertura. (JANG, LIVESLEY e VEMON, 1996)

Estudos longitudinais também sugerem que esses cinco grandes traços de personalidade tendem a ser relativamente estáveis ao longo da vida adulta. Um estudo de quatro anos com adultos em idade ativa descobriu que a personalidade mudou pouco como resultado de eventos adversos da vida. (CHERRY, 2022).

Estudos mostram que a maturação pode ter um impacto nos cinco traços de personalidade. À medida que as pessoas envelhecem, elas tendem a se tornar menos neuróticas, menos extrovertidas e menos abertas a novas experiências. A amabilidade e a consciência, por outro lado, tendem a aumentar à medida que as pessoas envelhecem. (CHERRY, 2022).

3. MATERIAIS E MÉTODOS

3.1. FERRAMENTAS UTILIZADAS

A principal ferramenta utilizada neste trabalho de conclusão de curso foi o *Google Colab*, o qual foi utilizada a linguagem Python versão 3.7 para todos os procedimentos metodológicos. Dentre as bibliotecas utilizada, destaca-se a *scikit-learn* (PEDREGOSA, VAROQUAUX, *et al.*, 2011), por englobar grande gama de métodos de aprendizado de máquina.

3.2. AJUSTES INICIAIS

3.2.1. Pré-processamento

Um dado oriundo diretamente de aplicações geralmente contém ruídos, valores ausentes e em algumas circunstâncias em formato inutilizável. O pré-processamento transforma os dados brutos em um formato compreensível, usando da adição, exclusão ou transformação dos dados dispostos em um *dataset* antes de aplicar algoritmos de aprendizado de máquina ou mineração de dados (ANUNAYA, 2022). Este processo ajuda na redução do impacto das distorções no *dataset*, e resultando, em muitos casos, na melhoria significativa dos modelos que os utilizam.

O pré-processamento dos dados utilizados neste trabalho ocorreu da seguinte maneira:

a) Verificação de dados ausentes ou faltantes:

Não foram identificados dados faltantes no dataset. Contudo a presença de muitos zeros nas *features* chamou a atenção para possíveis ruídos ou distorções que possam afetar os modelos.

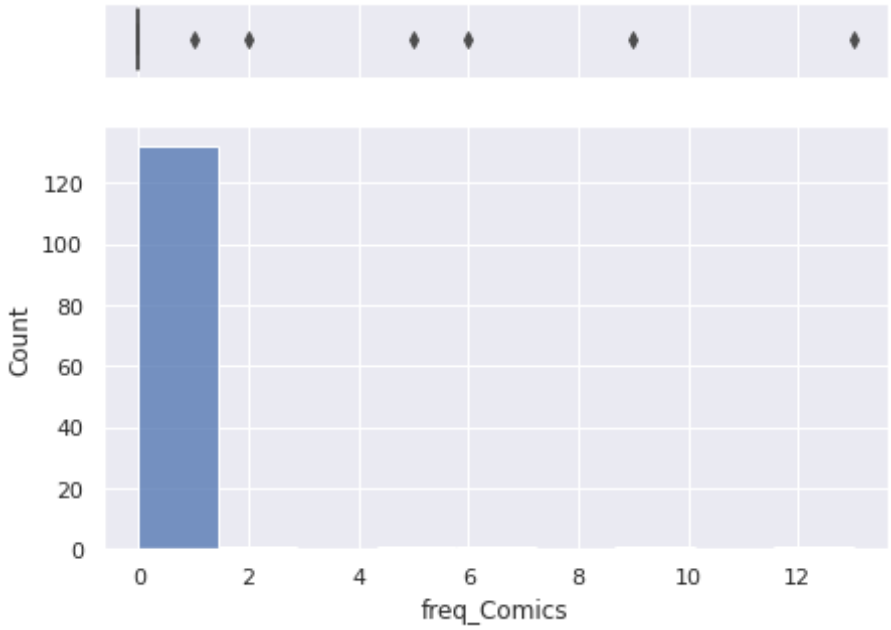
b) Análise de ruídos e presença de Outliers:

Do ponto de vista univariado, cerca de 55% dos casos foram considerados outlier segundo $[abs(Z_{score}) < 3]$. Isso deve estar ocorrendo devido a fatores de similaridade no uso dos aplicativos por parte dos participantes da amostra que podem ter o uso (ou não uso) concentrado em determinado tipo de aplicações.

Tomando as informações da Figura 1, é possível observar que na categoria *Comics* somente 6(seis) usuários utilizam algum aplicativo desta categoria, este isso leva a clara situação de 131(conto e trinta e um) usuários sem frequência e duração de uso destes, levando assim a uma concentração massiva na frequência 0(Zero)

Assim, considerar estes dados como *outliers*, além de reduzir ainda mais o tamanho da amostra, faria com que não consideremos o não uso dessas categorias de aplicativos como uma condição de proximidade com alguma característica de personalidade. Logo, decidiu-se pela permanência de todos os dados no *dataset* de estudo.

Figura 1: Distribuição de Frequência de uso dos Aplicativos da Categoria Comics



3.2.2. Transformação dos Dados

Considerando que as cinco variáveis resposta, resultados do BFSI[2], instrumento de avaliação psicológica utilizado para mapear as características de personalidade descritos no Capítulo 2, tem, originalmente, natureza continua e padronizada em $Zscore$ remete intuitivamente ao uso destas variáveis na construção de modelos de regressão.

[2] Big Five Struktur Inventar - O BFSI é um questionário multidimensional para registrar as Cinco Grandes dimensões estabilidade emocional, extroversão, abertura, consciência e amabilidade. (https://docplayer.org/2016053-Manual-big-five-struktur-inventar-wiener-testsystem-kurzbezeichnung-bfsi-version-22-revision-1.html)

Contudo diante das tentativas frustradas de modelagem em algumas das técnicas de regressão, onde foi constatada a baixa aderência dos dados levando a modelos com baixíssimo poder explicativo, decidiu-se então por categorizar os *targets* em “Baixo” e “Alto”, tomando como base o proposto por Nunes, Hutz e Nunes (2013, p. 125), para a construção de opções de modelos de classificação.

Tendo em vista a natureza de *features*, muitos dele com distribuições, foi realizado o procedimento de transformação logarítmica, e assim, distribuir um pouco mais dos dados. Conforme proposto por Feng, et al., (2019), “a transformação de log, um método amplamente utilizado para lidar com dados distorcidos, é uma das transformações mais populares usadas em pesquisas biomédicas e psicossociais”.

Uma dificuldade dessa transformação foi a incidência de 0(zeros) nos vetores de X onde $X[0,0,0,0,\dots,n]: \log_{10} X_i \to -\infty$, assim a função de transformação foi ajustada para $\log_{10}(X+1)$. Portanto, para uma determinada característica, essa transformação tende a espalhar os valores mais frequentes. Também reduz o impacto de outliers: este é, portanto, um esquema de pré-processamento robusto (PEDREGOSA, VAROQUAUX, *et al.*, 2011).

3.3. LASSO COMO ESCOLHA DAS *FEATURES*

Após experimentações preliminares na modelagem de classificadores dos fatores de personalidade, verificou-se que algumas variáveis explicativas no *dataset* não tinham contribuições significativas na performance dos modelos. E em alguns testes, foi implementado um método *stepwise backward* para medir a se remoção de variáveis neste modelo experimental melhoraria suas métricas sendo comprovada que algumas dessas variáveis não contribuíam para a explicação do modelo, assim como impactavam para a perda desse poder explicativo.

Diante desta situação, foi definido que para cada modelo proposto para classificar os fatores de personalidade teriam suas features escolhidas por um processo de regressão LASSO.

Least Absolute Shrinkage and Selection Operator ou simplesmente LASSO, é uma sigla que provém da língua inglesa que traduzido livremente significa Operador de Retração e Seleção Mínima Absoluta. É uma técnica de regularização usada em métodos de regressão linear com o propósito de encontrar um estimador que possui

menor risco em relação ao estimador de mínimos quadrados. É usada quando temos maior número de features, pois realiza a seleção de recursos automaticamente (KARGIN, 2021).

Para Izbicki e Santos (2020), LASSO consiste em encontrar uma solução β que minimize a soma de seu erro quadrático médio.

> *[...] a ideia é reduzir a variância do estimador de mínimos quadrados. Contudo, ao invés de medir a complexidade de um modelo como uma função do número de parâmetros no lasso essa complexidade é medida pela norma L_1 desse vetor, $\sum_{j=1}^{d}|\beta_j|$. [...] a norma L_1 captura a ideia de que uma pequena mudança nos valores de β não altera demasiadamente a complexidade do modelo resultante, já que suas predições serão praticamente as mesmas. (IZBICKI e SANTOS, 2020, p. 37)*

Na regressão Lasso, a penalidade tem o propósito de forçar que alguns dos coeficientes com pouca contribuição, tenham seus valores equivalentes a zero. Isso significa que o Lasso pode ser uma alternativa no processo de seleção de variáveis, com o propósito de reduzir a complexidade dos modelos (BASIEWICS, 2020).

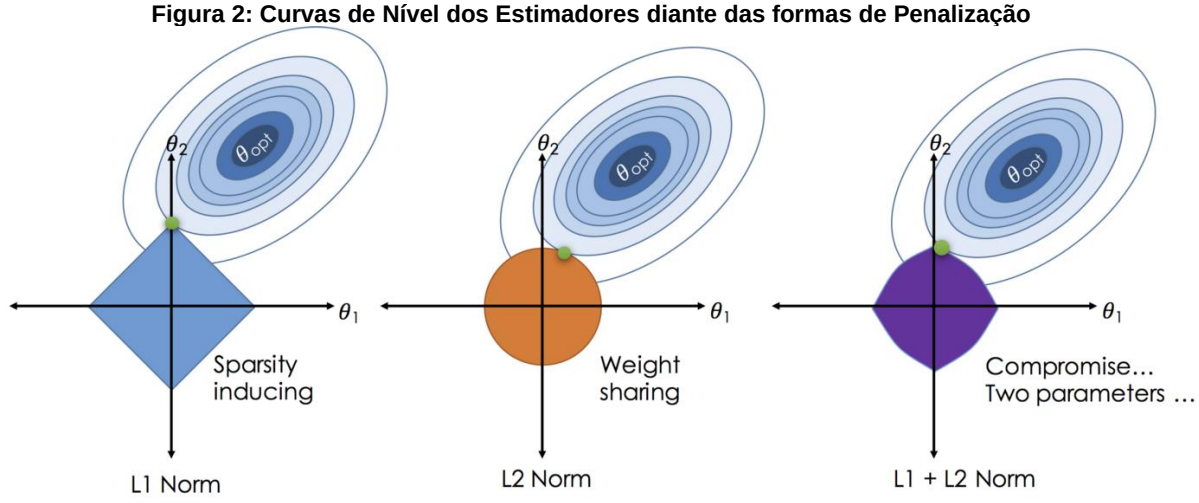
Formalmente o LASSO ($q = 1$) busca:

$$\hat{\beta}_{\lambda,q} = \arg\min_{g}(EQM(g_\beta)) + \lambda\sum_{j=1}^{d}|\beta_j|^q,$$

sendo a norma L_1 a espacialidade em $\beta \rightarrow ||\beta||_{L_1} = \sum_{j=1}^{d}|\beta_j|$.

O parâmetro λ quando igual a 0 (zero), leva a o LASSO a se tornar o mesmo estimador de mínimos quadrados com todos os $\hat{\beta} \neq 0$, tendo um estimador de alta variabilidade e baixo viés. Já quando $\lambda \rightarrow \infty$, os $\hat{\beta_j} = 0$ tendo um estimador com variância próxima de zero e viés muito grande.

Figura 2: Curvas de Nível dos Estimadores diante das formas de Penalização



Fonte: Basiewics (2020)

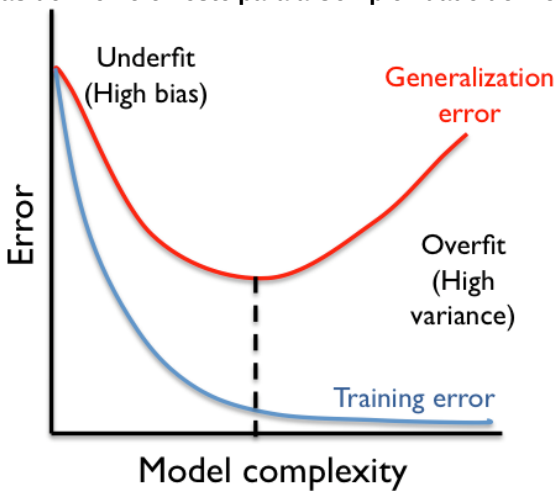
3.4. OVERFITTING

O *overfitting* é uma condição que ocorre quando um determinado modelo de aprendizado de máquina tem um desempenho significativamente melhor para dados de treinamento do que para novos dados. Isso ocorre porque o modelo está memorizando os dados que viu e não consegue generalizar para exemplos não vistos, ou seja, ele não pode fazer previsões precisas sobre novos dados porque não consegue distinguir dados ruidosos de dados essenciais que formam um padrão (AMAZON, 2016).

Algumas situações podem levar ao *overfitting* do modelo, entre elas destacam-se a baixa quantidade de dados de treinamento, modelos muito complexos, ou seja, grande número de *features*, dados ruidosos utilizados no treinamento, presença de features não representativas e modelos com muitos parâmetros de restrição (BRANCO, 2021).

Em suma, um modelo sobreajustado vai apresentar alta variância e baixo viés. Por outro lado, aquele subajustado terá alto viés e baixa variância, conforme ilustrado na Figura 3.

Figura 3: Curvas de Treino e Teste para a Complexidade do Modelo pelo Erro



Fonte: Branco (2021)

Algumas soluções podem ser implementadas para mitigar ou até mesmo evitar o *overfitting*, entre elas estão a simplificação de modelos, como proposto na seção 3.3; a divisão do *dataset;* e a validação cruzada, conforme detalhamento em sequência.

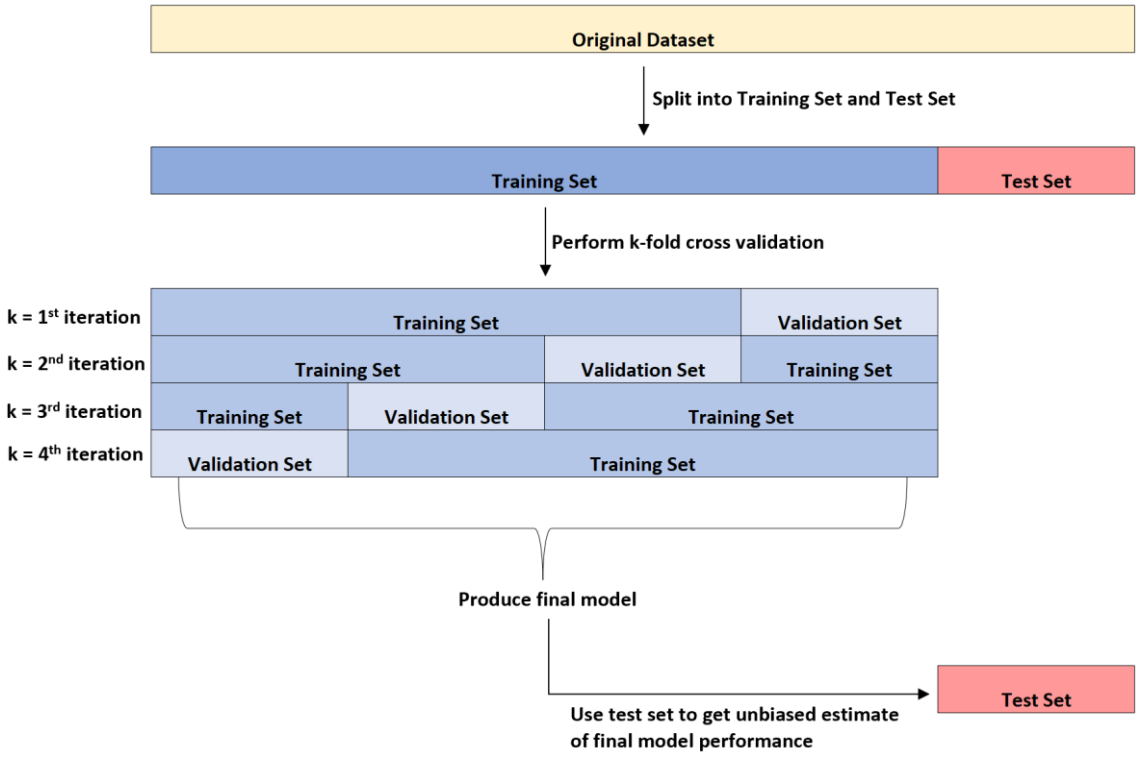
3.4.1. Data Splitting

O *Data Splitting* é o procedimento de particionar os dados disponíveis em dois ou mais subconjuntos distintos. Normalmente, é utilizada uma divisão em duas partes, a maior delas utilizada para treinar e a outra para avaliar ou testar o modelo (GILLIS, 2021).

Os dados devem ser divididos para que se possam ter uma grande quantidade de dados de treinamento. As proporções 80-20 ou 70-30 para treinamento versus dados de teste, são usualmente utilizados em divisões por duas porções. O *dataset* é dividido em dois, quando existe o propósito de utilizar procedimentos de validação cruzada na porção de treinamento.

Entretanto, dependendo do volume de dados é possível que uma proporção exata extra – *validation set* – seja incluída no estudo. Desta forma uma proporção de 70-20-10 para treinamento, validação e teste pode ser considerada nos processos de construção de modelos de *machine learning*. Uma possibilidade de particionamento pode ser vista no esquema da Figura 4.

Figura 4: Particionamento de dados para uso em *Machine Learning*



O esquema da Figura 4, introduz o procedimento de validação cruzada ou *cross validation* detalhado na sessão a seguir.

3.4.2. Validação Cruzada

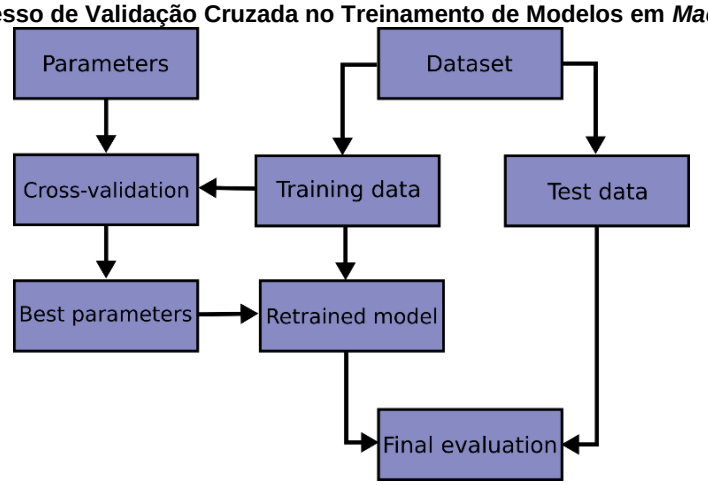
A validação cruzada é uma técnica utilizada na avaliação de desempenho de modelos de aprendizado de máquina. Ela consiste em particionar os dados de treinamento de maneira aleatória em conjuntos menores que serão utilizados como validação de desempenho dos modelos utilizados (AMAZON, 2016).

Existem alguns métodos de se aplicar a validação cruzada, um deles é a *leave-one-out.* e. a qual consiste em selecionar dobras (X, y) em quantidade igual ao número de instâncias do conjunto de dados (STONE, 1973). Assim, o algoritmo de aprendizado é aplicado uma vez para cada instância, usando todas as outras como um conjunto de treinamento e usando a selecionada como um conjunto de teste de item único (SAMMUT e WEBB, 2011).

Outra maneira é dada pelo método *k-fold*, Figura 4, onde o conjunto de treinamento é dividido em k conjuntos menores, diferentes e aproximadamente do mesmo tamanho. Assim, em cada um desses lotes são estimadas medidas de precisão que ao final são agregadas e utilizadas como indicador de desempenho do modelo (PEDREGOSA, VAROQUAUX, *et al.*, 2011)

A validação cruzada, com base no algoritmo *k-fold,* foi utilizada neste trabalho não só com a finalidade de mitigar sobreajustes dos modelos, mas também como maneira de selecionar os melhores parâmetros dos modelos propostos, conforme fluxograma na Figura 5.

Figura 5: Processo de Validação Cruzada no Treinamento de Modelos em *Machine Learning*



Fonte: Pedregosa, Varoquaux, *et al.* (2011)

Os hiperparâmetros foram ajustados por meio do método *GridSearchCV* do *scikit-learn,* o qual consiste na definição de um grid de parâmetros para cada modelo que são utilizados em cada ciclo (lote) da validação cruzada. Ao final desse processo, temos os melhores parâmetros a serem utilizados em um novo procedimento de treinamento de cada modelo proposto.

3.5. TÉCNIAS DE APRENDIZAGEM UTILIZADAS

Com citado neste capítulo, originalmente a natureza das variáveis resposta levou a experimentação de alguns modelos regressão, sejam eles paramétricos ou não. Contudo, tal exploração não apresentou resultados minimamente satisfatórios, trazendo métricas de qualidade do ajustamento longe do que seria considerado minimamente aceitáveis, tanto aos dados de treinamento e, sobretudo, aos dados de teste. Logo, optou-se por utilizar métodos de classificação para os estudos supervisionados deste trabalho.

Assim sendo, torna-se imprescindível citar que a Função de Risco comumente utilizada para o contexto de classificação foi definida como:

$$R(g) \coloneqq \mathbb{E}\left[\mathbb{I}\left(Y \neq g(X)\right)\right] = \mathbb{P}(Y \neq g(X)),$$

conforme Izbicki e Santos (2020), o risco de g é a probalilidade de erro em uma nova observação (X, Y), sendo chamada função de perda 0 – 1.

Por conseguinte, definido o objetivo em ser minimizado o risco, Izbicki e Santos (2020), apresenta a função g que minimiza $R(g)$ como:

$$g(x) = \arg\max_{d \in C} \mathbb{P}(Y = d|x),$$

Considerando o caso binário, onde Y assume dois valores$[c_1 e\ c_2]$,o classificador de *Bayes* pode ser reescrito da seguinte maneira:

$$g(x) = c_1 \Leftrightarrow \mathbb{P}(Y = c_1|x) \geq 0{,}5.$$

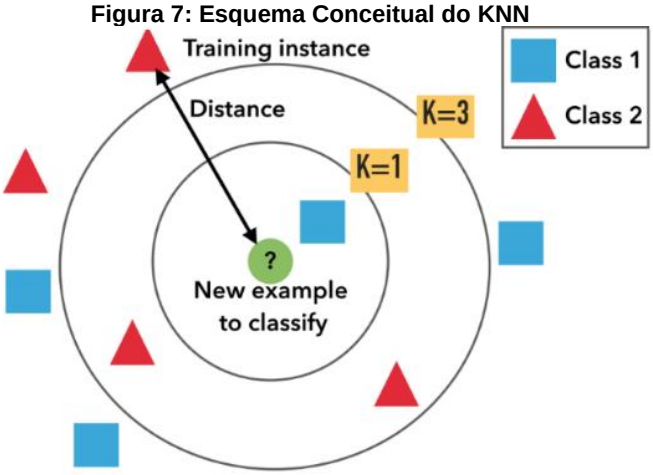
Esta abordagem também é conhecida como classificador *plug-in*.

Diante dos métodos de classificação disponíveis os classificadores abaixo foram os que melhor performaram diante do conjunto de dados.

3.5.1. *K-Nearest Neighbors* (KNN)

A classificação baseada em vizinhança é baseada em instâncias ou aprendizagem não generalizante, ou seja, ela não constrói um modelo interno geral, mas armazena as instâncias dos dados de treinamento. A classificação é calculada a partir de uma votação majoritária simples dos vizinhos mais próximos de cada ponto: a um ponto de consulta é atribuída a classe de dados que tem mais representantes dentro dos vizinhos mais próximos do ponto (PEDREGOSA, VAROQUAUX, *et al.*, 2011).

Figura 7: Esquema Conceitual do KNN



Fonte: APSL (2017)

Cuja formalização proposta por Izbicki e Santos (2020) é dada por:

$$g(x) = \frac{1}{k}\sum_{i \in \mathcal{N}_x} y_i,$$

em que $\mathcal{N}_x$é o conjunto das k onservações mais próximas de x, ou seja:

$$\mathcal{N}_x = \{i \in \{1, \dots, n\} : d(x_i, x) \leq d_x^k\}$$

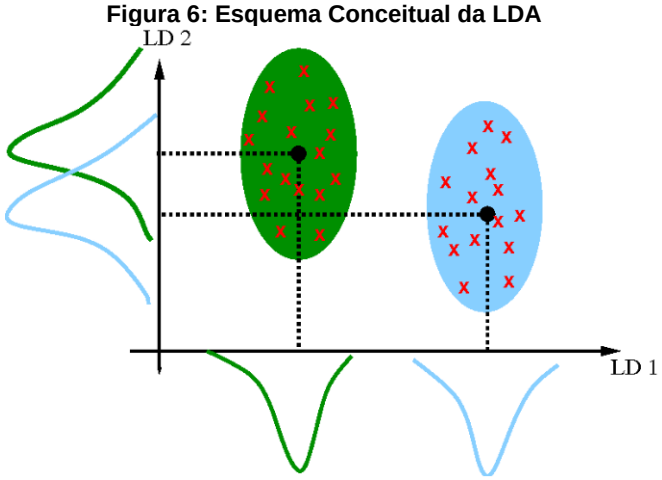
e d_x^k é a distância do $k - ésimo$ vizinho mais próximo de x.

Neste trabalho o valor de k foi definido por validação cruzada dentro do um intervalo de 1 a 7. Para Izbicki e Santos (2020), um valor alto de k leva a um modelo muito simples tendo um viés alto, mas uma variância baixa. Por sua vez, um valor baixo para k leva a um estimador com variância alta, mas viés baixo.

3.5.2. *Linear Discriminant Analysis* (LDA)

A Análise Discriminante Linear (LDA) é uma generalização do discriminante linear de Fisher. Método usado no reconhecimento de padrões e aprendizado de máquina, o qual consiste em encontrar uma combinação linear de *features* que define duas ou mais classes. Este método projeta um conjunto de dados em um espaço de menor dimensão com boa separabilidade de classes para evitar *overfitting* e reduzir custos computacionais. A combinação resultante pode ser usada como classificador linear ou para redução de dimensionalidade (APSL, 2017).

Figura 6: Esquema Conceitual da LDA



Fonte: APSL (2017)

Na análise discriminante, supõe-se que o vetor X, condicional em $Y = c$, possui distribuição normal multivariada. Este classificador tem uma superfície de decisão linear, estimada com base nas densidades condicionais das classes dos dados. O modelo ajusta uma densidade gaussiana para cada classe, assumindo que todas elas usam a mesma matriz de covariância.

Assim, conforme Izbicki e Santos (2020), assume-se que:

$$X = (X_1, \dots, X_d) | \, Y = c \sim Normal\ (\mu_c, \Sigma_c).$$

Modelado como uma distribuição gaussiana multivariada com densidade:

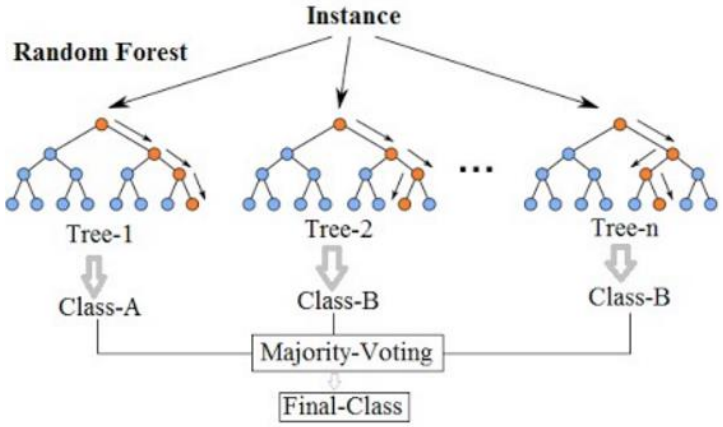
$$P(x \mid y = k) \ = \frac{1}{\sqrt{2\pi|\Sigma_c|}} e^{-(x-\mu_c)^t \Sigma_c^{-1}(x-\mu_c)}.$$

O termo $(x - \mu_c)^t \Sigma_c^{-1}(x - \mu_c)$ corresponde à *Distância Mahalanobis* entre x e a média μ_c, isso significa que o LDA classifica o vetor de X em função das classes de Y em termos desta distância (PEDREGOSA, VAROQUAUX, *et al.*, 2011).

3.5.3. *Random Forest* (RF)

Em florestas aleatórias, cada árvore no conjunto é construída a partir de uma amostra retirada com reposição (amostra *bootstrap*) dos dados de treinamento. Além disso, ao dividir cada nó durante a construção de uma árvore, a melhor divisão é encontrada entre todas as *features* ou um subconjunto aleatório de tamanho $m < \#\, features$.

Figura 7: Esquema Conceitual da *Randon Forest*



Fonte: Amazon (2016)

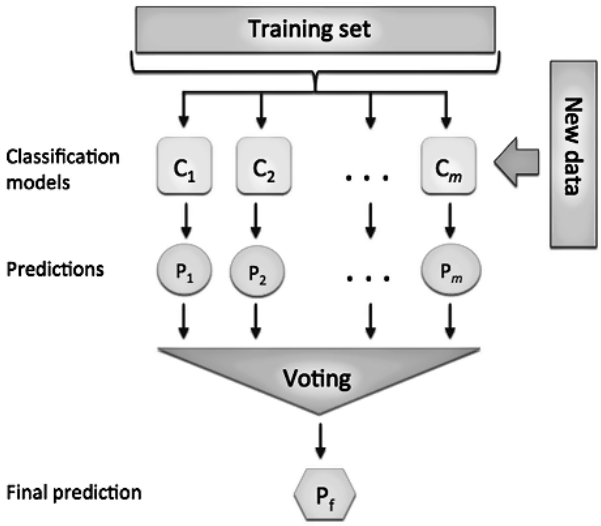
As árvores de decisão individuais normalmente apresentam alta variância e tendem a sofrer *overfitting*. A aleatoriedade presente no método das florestas produz árvores de decisão com erros de previsão um tanto desacoplados e ao usar a média dessas previsões, alguns erros podem ser cancelados. Florestas aleatórias atingem uma variação reduzida combinando diversas árvores, às vezes ao custo de um leve aumento no viés. Na prática, a redução da variância é muitas vezes significativa, resultando em um modelo globalmente melhor (PEDREGOSA, VAROQUAUX, *et al.*, 2011).

O número de estimadores e o máximo de *features* utilizados foram ajustados por meio de validação cruzada. O primeiro é o número de árvores na floresta, quanto maior, melhor, porém a um alto custo de processamento. O último é o tamanho dos subconjuntos aleatórios de recursos a serem considerados ao dividir um nó. Quanto menor, maior a redução da variância, mas também maior o aumento do viés.

3.5.4. Voting Classifier

O *Voting Classifier* combina os classificadores de aprendizado de máquina conceitualmente diferentes e usar uma votação majoritária ou as probabilidades previstas médias para prever os rótulos dos *targets*. Ele pode ser útil para um conjunto de modelos de desempenho igualmente bom para equilibrar suas fraquezas individuais.

Figura 8: Esquema Conceitual do *Voting Classifier*



Fonte: Patil e Patil (2018)

Na votação por maioria, o rótulo de classe previsto para uma determinada amostra é o rótulo de classe que representa a maioria dos rótulos de classe previstos para cada classificador individual.

3.6. MÉTRICAS DE DESEMPENHO

Uma série de métricas pode ser elencada para verificação da performance dos modelos de classificação, contudo o uso de cada uma delas vai depender do objetivo do modelo que se deseja criar.

Considere o problema de diagnosticar um paciente a partir de um exame de imagem, sendo ele com ou sem câncer, e esta classificação é a indicativa para a realização de exames mais caros e complexos para ratificar o diagnóstico inicial. Este é um claro exemplo de que a classificação de um paciente como sadio sendo que ele está com câncer pode custar esta vida e assim priorizar algumas métricas em detrimento de outras (MARIANO e PAZ, 2020).

Um método comum para descrever o desempenho de um modelo de classificação é a matriz de confusão (Tabela 2). Por meio dela é possível visualizar todas os dados distribuídas entre as classes observadas e previstas, codificando o número de previsões corretas e incorretas para cada classe. As células da diagonal principal da matriz de confusão apresentam os casos em que foram corretamente previstos enquanto as demais células ilustram os erros para cada caso possível.

Tabela 2: Representação da Matriz de Confusão Binária

Matriz de Confusão		Classe Predita	
		Positiva	Negativa
Classe Original	Positiva	Verdadeiro Positivo (VP)	Falso Negativo (FN)
	Negativa	Falso Positivo (FP)	Verdadeiro Negativo (FN)

Sob a perspectiva da matriz de confusão e das medidas de qualidade do classificador que ela promove, listamos na Tabela 3 as principais medidas adotadas para avaliar os modelos propostos neste trabalho.

Tabela 3: Métricas de Desempenho Utilizadas

Medida	Fórmula
Acurácia	$\frac{VP + VN}{VP + FP + VN + FN}$
Precisão	$\frac{VP}{VP + FP}$
Recall	$\frac{VP}{VP + FN}$
F1-Score	$2 * \frac{precisão * recall}{precisão + recall}$

A acurácia é amplamente utilizada e tem interpretação direta, ela reflete a concordância entre as classes observadas e previstas. A precisão é definida pela razão entre a quantidade de exemplos classificados corretamente como positivos e o total de exemplos classificados como positivos. Ela ressalta os erros por falso positivo;

O recall ou sensibilidade enfatiza os erros por falso negativo. Ela é razão entre a quantidade de casses classificadas corretamente como positivos e a quantidade de classes que são de fato positivos.

Por fim, o F1-*score.* Ele é uma métrica conjunta da precisão e do recall. F1-*score* alto significa precisão e recall também altos, ou seja, capaz tanto de acertar suas predições quanto a recuperação das classes de interesse. Quanto maior forem estas medidas melhor será o modelo de classificação.

4. RESULTADOS

Este capítulo apresenta os resultados obtidos por meio dos procedimentos apresentados no capítulo anterior. Primeiramente será apresentado o resultado da Análise Exploratório de Dados (EDA) das variáveis que compõe este estudo e logo em seguida, os resultados dos Modelos de Classificação Propostos.

4.1. ANÁLISE EXPLORATÓRIA DE DADOS

4.1.1. Amostra

O estudo contou com a participação de 137 jovens com média de idade de 24 anos (DP=4,71), e consistiu principalmente de alunos e funcionários da *Ludwig-Maximilians-Universität München* (LMU). A amostra foi constituída por 87 mulheres e 50 homens, a maioria tinha pelo menos o ensino médio completo (96%), sendo 31% dos participantes tinham ensino superior completo. (STACHL, HILBERT, *et al.*, 2016)

Em um primeiro momento os participantes foram submetidos a alguns testes psicológicos dentre esses um inventário de personalidade (BFSI), instrumento este que teve seus resultados considerados neste estudo. Posteriormente, um aplicativo de registro de atividades foi instalado nos *smartphones* particulares dos participantes e testado quanto à funcionalidade. O aplicativo registrou as frequências e as durações de uso de aplicativos móveis durante 60 dias em *smartphones* com sistema operacional *Android*.

Os dados coletados passaram inicialmente por um pré-processamento e foram agrupados, majoritariamente segundo a categorização de aplicativos fornecida pela *Google Play Store* para os aplicativos registrados. Resultado desse agrupamento pode ser visto no **Anexo A** – Estatísticas do Uso de Aplicativos do The PhoneStudy *Project*.

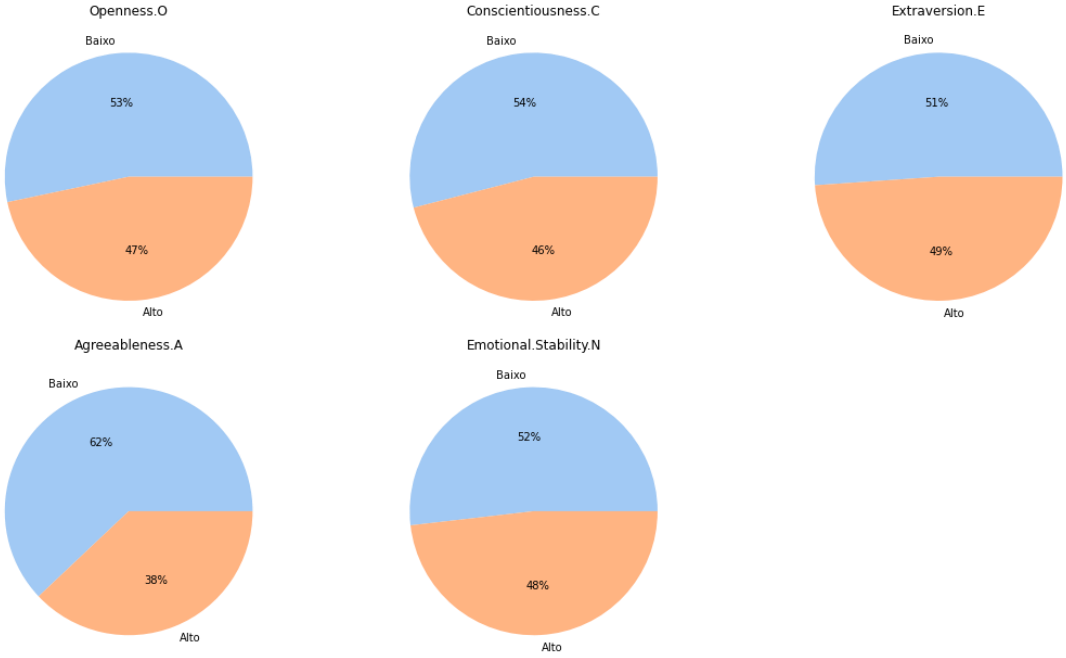
Os dados já pré-processados e categorizados foram coletados do sítio do projeto *The PhoneStudy Project*, contendo 137 (cento e trinta e sete) registros e 94(noventa e quatro) variáveis, destas foram utilizadas 5(cinco) variáveis resposta ou *targets*, uma para cada fator de personalidade, e 52(cinquenta e duas) variáveis explicativas ou *features* que auxiliaram na classificação dos modelos para cada um

dos cinco fatores de personalidade. O *dataset* foi dividido segundo a proporção 7:3 entre os dados de treinamento e teste, respectivamente.

4.1.2. *Targets* de Classificação

Os *targets* utilizados são os escores das medidas dos Cinco Grandes Fatores, rotuladas em “Baixo” e “Alto” conforme tabela proposta em Nunes, Hutz e Nunes (2013, p. 125). Diante desta rotulação, salvo para o fator amabilidade, os demais fatores tem proporções em suas categorias muito próximas (Figura 9), ou seja, maioria dos targets estão balanceados.

Figura 9: Distribuição das Categorias nos *Targets*

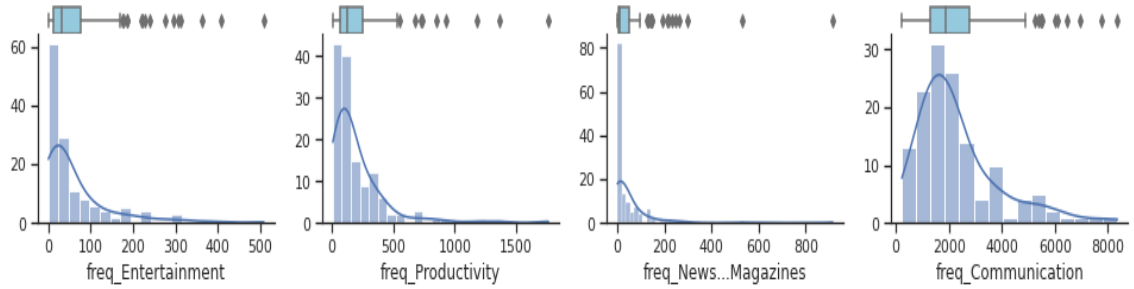


Fonte: Resultados da EDA

4.1.3. Features

Durante o processo de exploração dos dados, foi possível notar que muitas das features tinham comportamento assimétrico positivo, com grande concentração de dados nos valores mínimos. Esta situação pose ser entendida como o baixo uso de determinados aplicativos por parte dos participantes da pesquisa. Assim, aplicativos de comunicação, de uso comum entre os usuários, tem sua distribuição mais esparsa que os aplicativos de entretenimento ou mesmo os de notícias por exemplo (Figura 10).

Figura 10: Distribuições Original de Algumas F*eatures*



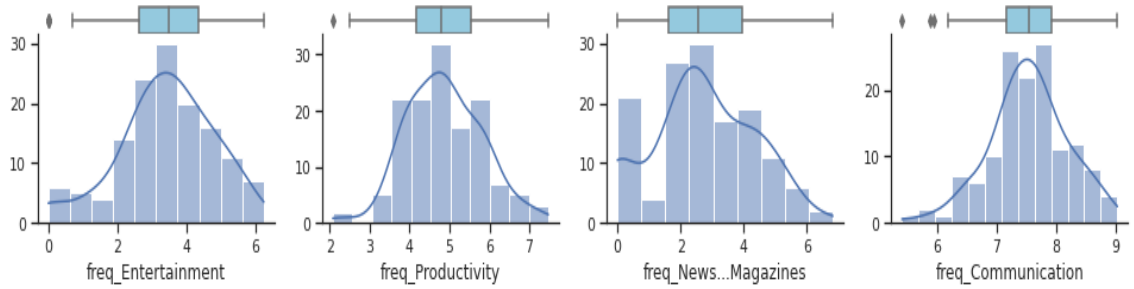
Fonte: Resultados da EDA

A concentração dos dados aos valores mínimos, sugestionava a presença de medidas discrepantes. Fato constatado por meio do critério de $|Zscore| > 3$ que aplicado as features originais, apontou que mais de 60% dos registros eram considerados *outliers*. Isso motivou a busca algum método de transformação de dados que suavizasse a ocorrência de dados atípicos. Foram utilizadas inicialmente a padronização e transformação quantilca com saída para distribuição normal, porém a redução dos outliers não foi muito substancial, cerca de 50% ainda eram classificados como tal.

Após o furor inicial das tentativas frustradas de ajuste de modelos de classificação com estes dados, mudou-se o foco de reduzir os *outliers* para melhorar a distribuição das features. Essa foi a estratégia de melhor impacto nos resultados, inicialmente para o problema dos *outliers*, que reduziram para cerca de 25%, e posteriormente nos resultados dos modelos.

Na Figura 11, podemos observar o resultado da transformação logarítmica na mesma amostra de *features* apresentadas na figura anterior. O ganho que este procedimento trouxe foi essencial ao delineamento, sem perdas de interoperabilidade dos resultados, uma vez que modelos propostos são para classificação de classes.

Figura 11: Distribuições $Log_{10}(X)$ de Algumas *Features*



Fonte: Resultados da EDA

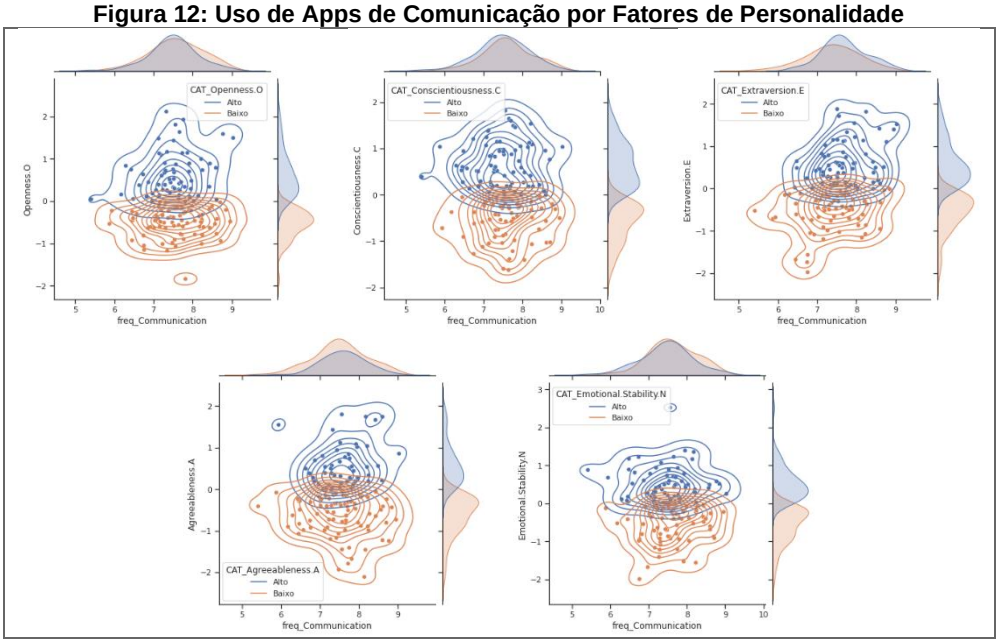
4.1.4. Análise Conjunta

As visões univariadas das features foram de grande valia no entendimento dos seus padrões de comportamento e como estas distribuições poderiam ser uteis no desenho de modelos que pudessem descrevê-las. Contudo a natureza multivariada do problema estabelece que relações, ao menos no $\mathbb{R}^2$devem ser analisadas.

A classificação de dados tem sua motriz pautada em como um conjunto de *features* podem, por meio de seu comportamento e de suas relações, determinar um comportamento previamente determinado (modelos supervisionados) e como esses padrões podem ser uteis na determinação de novos targets quanto se tem disponível exclusivamente o conjunto de *features* (modelos não supervisionados).

A Figura 12 traz uma amostra das relações existentes entre a frequência de usos de apps de comunicação – como WhatsApp (**Anexo A**) – e os fatores de personalidade. A representação completa destas relações pode ser visualizada no **Apêndice D –** Relação entre .

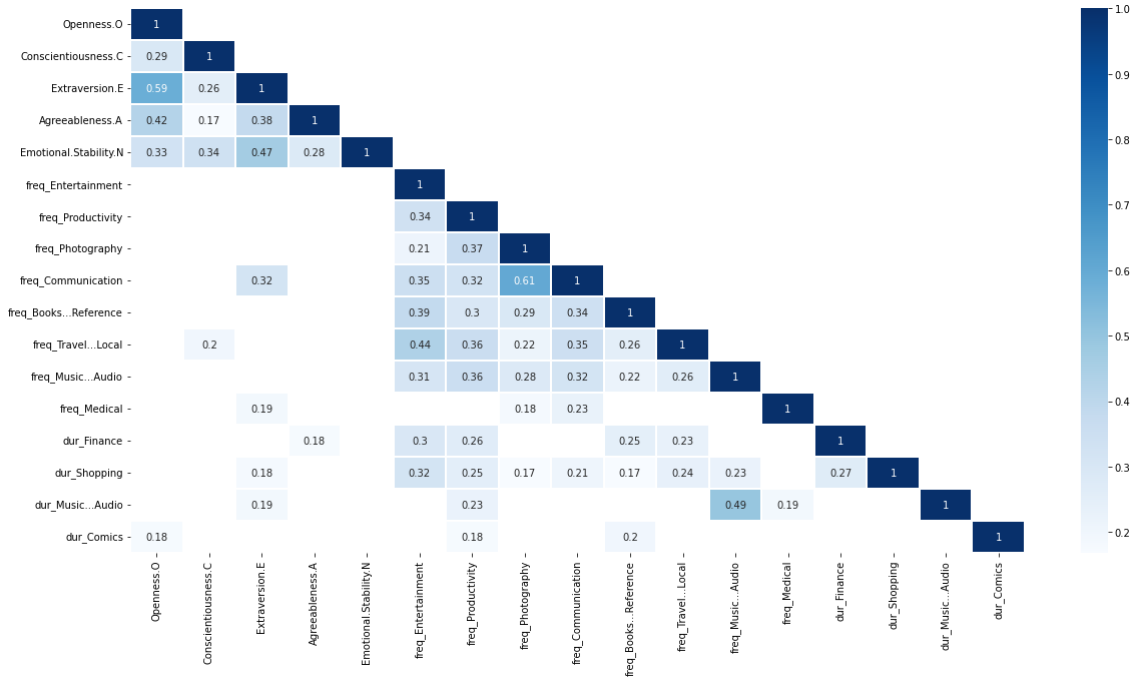
Figura 12: Uso de Apps de Comunicação por Fatores de Personalidade



Fonte: Resultados da EDA

As representações acima denotam a relação entre as variáveis resposta e suas explicativas, sendo possível visualizar relações diretas ou mesmo algumas veladas que com ajuda de algum rótulo se torna mais clara. A magnitude dessa associação é medida por meio das correlações (Figura 13), elas são para alguns modelos o conjunto de entrada responsável por desvendar as associações multidimensionais.

Figura 13: Correlação entre Algumas Variáveis do Estudo de Classificação



Fonte: Resultados da EDA

A matriz de correlação acima, denota a magnitude das relações entre as variáveis selecionadas. Dentro deste recorte, é possível visualizar a relação teórica, como por exemplo, o fator Extroversão com a sua alta intensidade de interações interpessoais e a relação, notável, com os aplicativos de comunicação. Este é um exemplo claro de relação direta, porém muitas dessas relações somente são captadas em métodos mais sofisticados que a visão e a percepção humana. As correlações entre todas as variáveis de frequência e de duração de uso de aplicativos e as *targets* estão disponíveis no **Apêndice E –** Correlação entres *Targets* e *Features* de Frequênciae no **Apêndice F –** Correlação entres *Targets* e *Features* de Duração

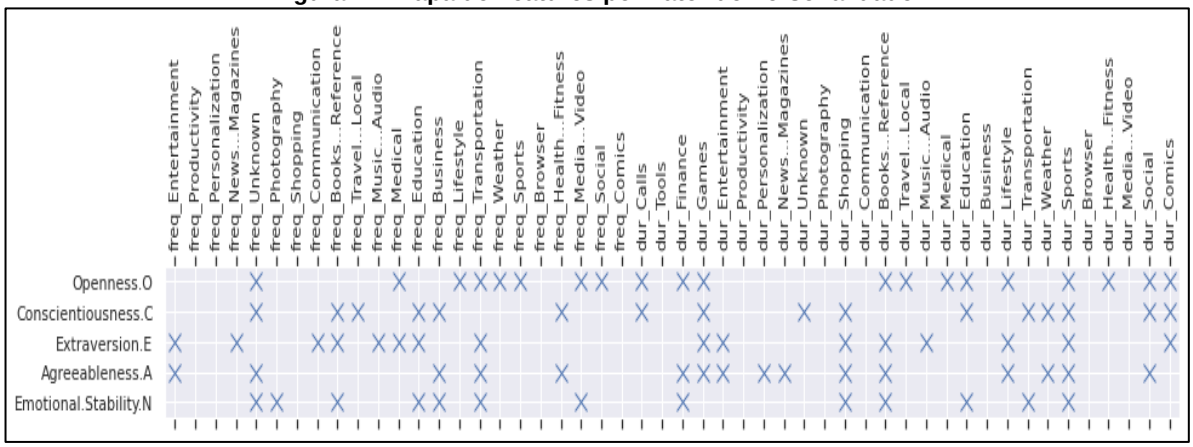
As matrizes de correlação também podem ser usadas no processo de seleção de *features*, grupos de *fratures* que se correlacionam na mesma magnitude (multicolinearidade), pode sugerir que apena uma delas seja suficiente para explicar o *target*.

4.1.5. Features Selecionadas

Diante do processo de penalização descrito na Seção 3.3, foi possível mapear as variáveis mais significativas no processo de classificação dos fatores de

personalidade mediante variáveis de uso de smartphones. O resultado desse processo está representado na Figura 14.

Figura 14: Mapa de Features por Fator de Personalidade



Fonte: Processamento LASSO para escolha de *features* do *PhoneStudy Project*

O procedimento adotado promoveu uma redução significativa nas features em cada modelo. Os modelos gerais de cada fator incialmente foram construídos com todas as 52(cinquenta e duas) *features,* após o LASSO os modelos tiveram entre 13(treze) e 20(vinte) *features*.

4.2. MODELOS DE CLASSIFICAÇÃO PROPOSTOS

Conforme descrito na Seção 3.5, os modelos que com melhores métricas para o conjunto de dados deste trabalho foram o *K-Nearest Neighbors* (KNN), Linear *Discriminant Analysis* (LDA) e o *Random Forest* (RF). Com os resultados destes três classificadores, foi decidido por montar um *ensemble*, um agrupamento dos modelos propostas com a finalidade de potencializar o poder preditivo final. O algoritmo de *Voting Classifier* com critério de escolha “hard” foi escolhido para esta tarefa, cumprindo o objetivo esperado.

O delineamento dos modelos iniciou com a definição de seus parâmetros. Para isso, o método de *GridSerach* foi utilizado sendo definida uma grade para cada parâmetro de cada modelo. E assim, utilizando a amostra de treinamento, um processo de *cross validation (cv=5)* selecionou os melhores parâmetros.

A Tabela 4 mostra os parâmetros escolhidos deste processo. Nela só está descrito os três modelos originais, pois o *Voting Classifier utiliza* os mesmos modelos originais parametrizados com seus melhores parâmetros.

Tabela 4: Hiperparâmetros de cada Modelo

Modelo	Fator	Parâmetros
KNN	Abertura	`'n_neighbors': 5`
	Conscienciosidade	`'n_neighbors': 3`
	Extroversão	`'n_neighbors': 7`
	Amabilidade	`'n_neighbors': 5`
	Neuroticismo	`'n_neighbors': 3`
LDA	Abertura	`'shrinkage': 0.7, 'solver': 'lsqr'`
	Conscienciosidade	`'shrinkage': 0.799999, 'solver': 'lsqr'`
	Extroversão	`'shrinkage': 0.7, 'solver': 'lsqr'`
	Amabilidade	`'shrinkage': 0.799999, 'solver': 'lsqr'`
	Neuroticismo	`'shrinkage': 0.7, 'solver': 'lsqr'`
RF	Abertura	`'max_depth': 4, 'max_features': 'auto', 'n_estimators': 4`
	Conscienciosidade	`'max_depth': 5, 'max_features': 'auto', 'n_estimators': 4`
	Extroversão	`'max_depth': 20, 'max_features': 'auto', 'n_estimators': 4`
	Amabilidade	`'max_depth': 10, 'max_features': 'auto', 'n_estimators': 4`
	Neuroticismo	`'max_depth': 10, 'max_features': 'auto', 'n_estimators': 4`

Com os hiperparâmetros definidos e modelos treinados, partiu-se para verificação de suas performances. A Tabela 5 apresenta as métricas de cada um dos 20(vinte) modelos treinadas, 4(quatro) para cada um dos 5(cinco) fatores de personalidade.

Considerando o amplo contexto da exploração que este trabalho foi conduzido, definiu-se como um bom modelo de classificação seria aquele que tivesse boa acurácia, ou seja, capaz de prever corretamente as verdadeiras classes dos fatores de personalidade; assim como não tivessem grande incidências de Falso Positivos e de Falsos Negativos.

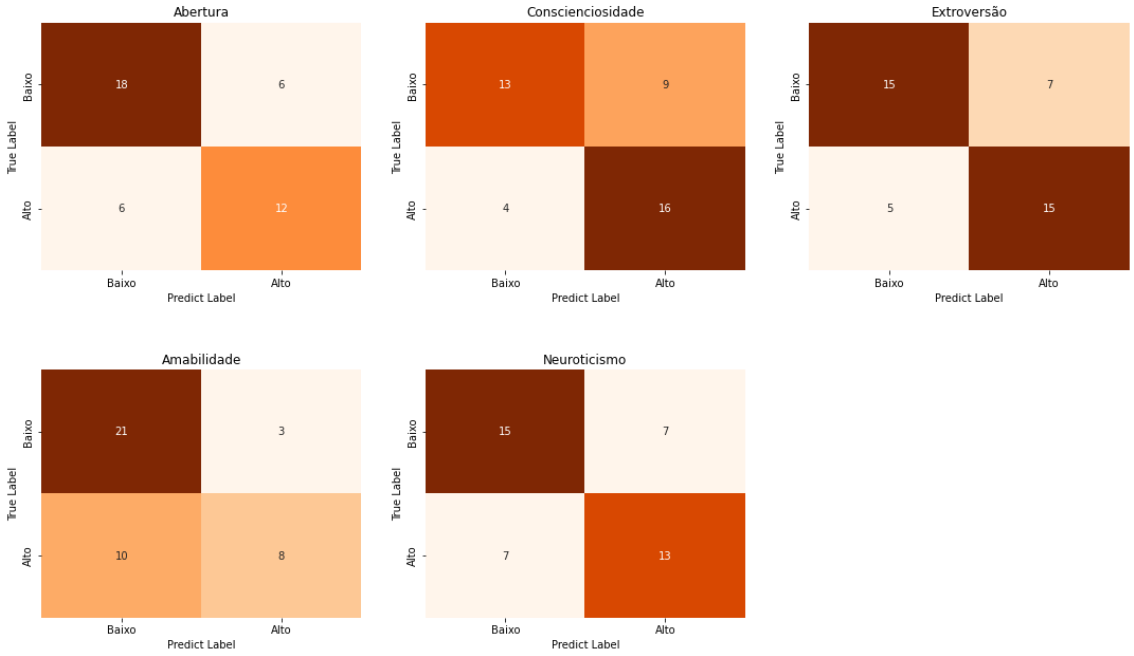
Nota-se que os resultados da junção de modelos do *Voting Classifier* melhor sensivelmente a classificação realizada pelos modelos originais. De modo geral, o *VC* melhorou a acurácia média dos modelos de 60% para 70%, um ganho de aproximadamente 18% na capacidade de classificação.

Tabela 5: Métricas de Desempenho dos Modelos

TARGET	QTD FEATURES	MODELOS	MÉTRICAS			
			Acurácia	Precisão	Recall	F1-score
ABERTURA	20	KNN	0,5952	0,6667	0,5833	0,6222
		LDA	0,6429	0,6800	0,7083	0,6939
		RF	0,5000	0,5652	0,5417	0,5532
		VC	**0,7143**	**0,7500**	**0,7500**	**0,7500**
CONSCIENCIOSIDADE	16	KNN	0,5476	0,6000	0,4091	0,4865
		LDA	0,5952	0,6190	0,5909	0,6047
		RF	0,6190	0,7143	0,4545	0,5556
		VC	**0,6905**	**0,7647**	**0,5909**	**0,6667**
EXTROVERSÃO	16	KNN	0,5714	0,6250	0,4545	0,5263
		LDA	0,6667	0,7000	0,6364	0,6667
		RF	0,5476	0,6364	0,3182	0,4242
		VC	**0,7143**	**0,7500**	**0,6818**	**0,7143**
AMABILIDADE	16	KNN	0,6304	0,6216	0,8846	0,7302
		LDA	0,6190	0,6333	0,7917	0,7037
		RF	0,6429	0,6286	0,9167	0,7458
		VC	**0,6905**	**0,6774**	**0,8750**	**0,7636**
NEUROTICISMO	13	KNN	0,6190	0,6250	0,6818	0,6522
		LDA	0,6190	0,6500	0,5909	0,6190
		RF	0,5000	0,5172	0,6818	0,5882
		VC	**0,6667**	**0,6818**	**0,6818**	**0,6818**

Apesar do ganho na acurácia e demais medidas, uma das maneiras mais tradicionais para se verificar se um classificador binário esta performando bem, é por meio das matrizes de confusão (Figura 15).

Figura 15: Matrizes de confusão do modelo Volting Class para os Cinco Fatores de Personalidade



O exame visual, conforta o analista, permitindo que hipóteses sejam levantadas e testadas em prol do melhor desempenho. Nas matrizes representadas na Figura 15, mesmo com as medidas de desempenho em patamares de 70%, o mapa permite observar além. Nota-se que o modelo que classifica a Amabilidade, classifica muito bem a casse baixa, isso certamente elevou suas métricas, disfarçando o fato deste modelo não ter um desempenho tão bom na classificação da classe alta.

Estas constatações levantam hipóteses sobre a qualidade e tamanho da amostra, seu balanceamento, se o modelo está bem parametrizado ou mesmo se é o mais bem adequado para descrever os dados. Todas estas são consideradas durante a construção, avaliação e reavaliação dos modelos.

Retomando a avaliação do fator Amabilidade, devemos recordar (seção 4.1.3) um leve desequilíbrio em favor da classe baixo. Diferente dos demais fatores, este desbalanceamento reforça a hipótese de que o modelo não classifica bem esta classe por não ter elementos “alto” suficientes para que a aprendizagem ocorra da mesma maneira que nos elementos “baixo”. Certamente este desequilíbrio do *dataset* para este fator será algo a ser considerado em trabalhos futuros.

De modo geral, taxas de acerto de 70% em uma amostra de 131 indivíduos podem ser consideradas satisfatórias, principalmente quando se trata de estudos tão complexos quanto a personalidade humana. Acredita-se que a revisão da maneira com as features foram categorizadas, utilizando elementos de redução de dimensionalidade aos dados brutos, além da classificação teórica utilizada dos apps pelo *Google Play Store*. Isto, associado a amostras maiores tragam melhorias substanciais ao processo de classificação e consequentemente na previsão dos traços de personalidade dos usuários de smartphones.

5. CONSIDERAÇÕES FINAIS

Ao chegar nesta etapa do trabalho podemos pensar o quanto estamos envolvidos pela tecnologia e seja como indivíduo ou sociedade, a dinâmica em que vivemos nos move de maneira cada vez intensa e instantânea por acesso, consumo e produção de informações, serviços, entretenimento. Este *Modus operandi* de grande velocidade nas respostas, impulsionou ao inovador processo de automatização das coisas, onde várias ações são realizadas por máquinas e que hoje em dia exige que sejam cada vez mais personalizadas.

O desenvolvimento de maneiras captação indireta de dados e informações sobre o comportamento humano faz parte do processo de personalização dos processos automáticos, sendo os *smartphones* um grande aliado nessa coleta. E o trabalho realizado pelo *The Phone Study* visa captar traços de personalidade das pessoas por meio desses aparelhos.

Apoiados nesta proposta, apresentamos alguns algoritmos de aprendizagem de máquina podem fazer associação entre a maneira de usar os *smartphones* e os fatores de personalidade descritos na teoria do *Big Five.* Neste processo, salienta-se a importância da Análise Exploratória de Dados (EDA) como primeiros passos do estudo. O que permitiu, por meio de transformação de dados em distribuições mais esparsas, uma melhor captação da variabilidade, assim como na redução de *outliers* e consequentemente na melhoria no treinamento dos modelos de *machine learning*.

A EDA realizada reduziu de 55% para 22% a incidência de *outliers*, proporcionou ainda a otimização de recursos por meio da utilização das variáveis que realmente contribuem no treinamento dos modelos. A escolha dessas variáveis, realizada pelo método LASSO, reduziu, de 52 para 20 features o aprendizado do modelo de classificação para o fator Abertura, 16 a Conscienciosidade, a Extroversão, a Amabilidade e para 16 features o fator Neuroticismo.

Foi possível notar a convergência teórica entre características dos fatores de personalidade e algumas das *features* utilizadas neste trabalho. A Extroversão, por exemplo, caracterizada, principalmente, pela quantidade e pela intensidade nas interações interpessoais mostrou-se amplamente relacionada ao uso de aplicativos de comunicação, entretenimento, música e educação.

Diante dos modelos *K-Nearest Neighbors* (KNN), *Linear Discriminant Analysis* (LDA) e o *Random Forest* (RF) foi possível compor o *Ensemble Voting Classifier,* que elevou, por exemplo, a acurácia média de 59% para 70%, proporcionando uma melhoria média de 18% nas métricas dos modelos isolados.

Apesar dos resultados satisfatórios, entendemos que uma trabalhos futuros conduzidos com uma amostra maior e com a disponibilização dos dados não agrupados em categorias de uso dos apps, mas sim os brutos para que se possa investigar essas associações por meio de outros métodos, possam ajudar na melhoria da classificação desses modelos.

Considerando os fatos expostos neste trabalho, evidenciamos a necessidade pela personalização de serviços automatizados, e a inclusão de elementos de comportamento real captados pelo uso rotineiro dos *smartphones* como uma possibilidade factível no fornecimento de dados a serem usados em *machine learning* com o propósito de fornecer *inputs* da personalidade humana em serviços automatizados.

REFERÊNCIAS

AMAZON. **Amazon Machine Learning:** Developer Guide. [S.l.]: [s.n.], 2016. Disponivel em: <https://docs.aws.amazon.com/machine-learning/latest/dg/machinelearning-dg.pdf>.

ANUNAYA, S. Data Preprocessing in Data Mining -A Hands On Guide. **Analytics Vidhya**, 2022. Disponivel em: <https://www.analyticsvidhya.com/blog/2021/08/data-preprocessing-in-data-mining-a-hands-on-guide/#h2_2>. Acesso em: 01 Jun 2022.

APSL. Using Linear Discriminant Analysis (LDA) for data Explore: Step by Step. **https:** //www.apsl.net/, 2017. Disponivel em: <https://www.apsl.net/blog/2017/07/18/using-linear-discriminant-analysis-lda-data-explore-step-step/>. Acesso em: 10 Ago 2022.

BASIEWICS, A. A. O Big Data e a Estatística. **Pet-Estatística UFPR**, 2020. Disponivel em: <https://pet-estatistica.github.io/site/download/posts/postALTAMIRO.html>. Acesso em: Ago 2022.

BRANCO, H. Overfitting e underfitting em Machine Learning. **ABRACD.org**, 2021. Disponivel em: <https://abracd.org/overfitting-e-underfitting-em-machine-learning/>.

CHERRY, K. The Big Five Personality Dimensions. **verywellmind.com**, 2022. Disponivel em: <https://www.verywellmind.com/the-big-five-personality-dimensions-2795422>. Acesso em: jul 2022.

DATA.AI. Report: State of Mibile 2022. **data.ai**, 01 fev. 2022. Disponivel em: <https://www.data.ai/en/go/state-of-mobile-2022>.

FENG, C. et al. Log-transformation and its implications for data analysis. **Shanghai archives of psychiatry**, v. 26, n. 2, p. 105-109, 06 Set 2019. Disponivel em: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4120293/>.

FONSECA, W. R. D. **Adaptação e Evidências de Validade do Nonvarbal Personality Questionneire**. Dissertação (Mestrado) - Instituto de Psicologia da Universidade de Brasília. Brasília, p. 95. 2018.

GILLIS, A. S. Data Splitting. **Techtarget**, 2021. Disponivel em: <https://www.techtarget.com/searchenterpriseai/definition/data-splitting>. Acesso em: Ago 2022.

IZBICKI, R.; SANTOS, T. M. D. **Aprendizado de máquina:** uma abordagem estatística. 1ª. ed. [S.l.]: [s.n.], 2020. 272 p. ISBN ISBN: 978-65-00-02410-4. Disponivel em: <http://www.rizbicki.ufscar.br/ame/>.

JANG, K. L.; LIVESLEY, W. J.; VEMON, P. A. Heritability of the Big Five Personality Dimensions and Their Facets: A Twin Study. **Journal of Personality**, Set 1996. 577-592. Disponivel em: <https://www.verywellmind.com/the-big-five-personality-dimensions-2795422>.

JAVA T POINT. Machine Learning Tutorial: Data Preprocessing in Machine learning. **JavaTPoint**. Disponivel em: <https://www.javatpoint.com/data-preprocessing-machine-learning>. Acesso em: 25 Maio 2022.

KARGIN, K. Lasso Regression Fundamentals and Modeling in Python. **medium.com**, Maio 2021. Disponivel em: <https://medium.com/analytics-vidhya/lasso-regression-fundamentals-and-modeling-in-python-ad8251a636cd>. Acesso em: Ago 2022.

MARIANO, D.; PAZ, F. J. **Data Mining**. 1ª. ed. Porto Alegre: Sagah, 2020.

MOBILE TIME. Panorama: uso de Apps no Brasil - Junho 2022. **Mobile Time**, 01 jun. 2022. Disponivel em: <https://www.mobiletime.com.br/pesquisas/uso-de-apps-no-brasil-junho-de-2022/>.

NUNES, C. H. S. D. S.; HUTZ, C. S.; NUNES, M. F. O. **Bateria Fatorial de Personalidade (BFP):** manual técnico. 2ª. ed. São Paulo: Pearson Clinical do Brasil, 2013. 240 p.

PATIL, D.; PATIL, J. Malicious URLs Detection Using Decision Tree Classifiers and Majority Voting Technique. **Cybernetics and Information Technologies**, 18, Mar 2018. 11-29.

PEDREGOSA, F. et al. Scikit-learn: Machine Learning in Python. **Journal of Machine Learning Research**, 12, 2011. 2825-2830. Disponivel em: <www.scikit-learn.org>.

SAMMUT, C.; WEBB, G. I. Leave-One-Out Cross-Validation. In: _____ **Encyclopedia of Machine Learning**. Boston: Springer, 2011. ISBN 978-0-387-30164-8.

STONE, M. Cross-validatory choice and assessment of statistical predictions. **Journal of the Royal Statistical Society**, Londres, 36, 5 dez 1973. 111-147. Disponivel em: <http://www.jstor.org/stable/2984809>.

TELECO. Estatísticas de Celulares no Brasil. **Teleco:** Inteligência em Telecomunicações, 01 Jul 2022. Disponivel em: <https://www.teleco.com.br/ncel.asp>.

TURNER, A. How many Smartphones are in the World? **BankMyCell**, 1 mai 2022. Disponivel em: <https://www.bankmycell.com/blog/how-many-phones-are-in-the-world#1579705085743-b3697bdb-9a8f>. Acesso em: 15 jul 2022.

APÊNDICES

APÊNDICE A – CÓDIGO FONTE

```
# -*- coding: utf-8 -*-
"""TCC-8-versaiImpre.ipynb

- Orientadora: *THAIS CRISTINA OLIVEIRA DA FONSECA*
- Aluno: Marcel Dantas de Quintela (marcelquintela@yahoo.com.br)

# Carregando bibliotecas
pip install scikeras[tensorflow]

# IMPORTANDO BIBLIOTECAS
# Install TensorFlow
try:
  # %tensorflow_version only exists in Colab.
#   %tensorflow_version 2.x
except Exception:
  pass

import warnings
import time

# Obtenção e Manipulação de dados
import numpy as np
import pandas as pd
from scipy import stats
from scipy.stats import pearsonr

pd.set_option('display.max_colwidth', None) # definir visualização para tamanho
máximo de caracteres das colunas

# salvar/carregar modelo treinado
import pickle

# Visualização
import matplotlib.pyplot as plt
import matplotlib.gridspec as gridspec
import seaborn as sns

from statsmodels.graphics.gofplots import qqplot
from google.colab import data_table
data_table.enable_dataframe_formatter()

# regressão
import statsmodels.api as sm

# SVM
from sklearn.svm import SVC

# Metricas de analise
from sklearn import metrics
from sklearn import model_selection, metrics, preprocessing
from sklearn.model_selection import train_test_split, GridSearchCV
from sklearn.model_selection import cross_val_score, KFold
from sklearn.model_selection import StratifiedKFold

from sklearn.svm import SVC
from sklearn.discriminant_analysis import LinearDiscriminantAnalysis
from sklearn.neighbors import KNeighborsClassifier
from sklearn.ensemble import RandomForestClassifier, VotingClassifier

from sklearn.metrics import mean_squared_error as MSE# mesma funcao do MSE
from sklearn.metrics import mean_absolute_percentage_error as MAPE
from sklearn.metrics import r2_score
from statsmodels.graphics.gofplots import qqplot

```

```
# preprocessing
from sklearn import preprocessing
from sklearn.preprocessing import StandardScaler
from sklearn.preprocessing import RobustScaler
from collections import Counter

# Treinamento Redes Neurais
import tensorflow as tf
from tensorflow import keras
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Dropout
from tensorflow.keras import layers
from tensorflow.keras import callbacks
from tensorflow.keras.callbacks import LearningRateScheduler
from scikeras.wrappers import KerasClassifier

print("Pacotes carregados!")

# Configuração do ambiente de trabalho
# montar ambiente - requer autorização do dono do espaço
from google.colab import drive
drive.mount('/content/drive')
#drive.mount('/content/drive', force_remount=True)

# Definição do ambiente de trabalho
import os
os.chdir('/content/drive/MyDrive/Colab Notebooks/Ciencia_de_Dados/TCC')
!pwd

warnings.filterwarnings("ignore")

def timer(start,end):
  hours, rem = divmod(end-start, 3600)
  minutes, seconds = divmod(rem, 60)
  print("Tempo de execução:
        {:0>2}:{:0>2}:{:05.2f}".format(int(hours),int(minutes),seconds))

"""#Leitura dos Dados"""

#lendo csv
data = pd.read_csv('../../Data/data.csv', encoding='UTF-8',sep=";",decimal=',')
data = data.rename({'mydatatest.avg_duration_calls': 'dur_Calls'}, axis=1)

#vericicar se existem valores nulos
if data.isnull().values.any():
    print('Existem valores faltantes no dataset')
else:
    print('Não existem valores faltantes no dataset')

"""# Variáveis do Estudo

## Variáveis Resposta

Cada um dos Fatores da Big Five é considerado variável resposta. Estas variáveis
são de natureza contínua e estão padronizadas (z), porém serão convertidas
Categóricas Ordinal de 5 pontos [Muito Baixo; Baixo, Médio, Alto, Muito Alto] de
acordo com critério de corte para os percentis.
"""

#Variaveis resposta
resp = list(data.iloc[:,6:11])

resp = [resp[i] for i in [2,3,1,4,0]]

# Função de conversão para Categórica
def conditions(x):
  clas = stats.norm.ppf(sum([.29,.7])/2)  # Classificação Percentilica tirada de
BFP Tab.65 pag. 125
```

```
    if   x > clas: return 1 # usando a metade das classes centrais para definir
    else:          return 0

  func= np.vectorize(conditions)

  # Criação de DF com as respostas Categoricas
  data_c = data[resp].apply(func)
  data_c.columns = 'CAT_'+data_c.columns
  cat = list(data_c)
  # ordenar OCEAN
  #cat = [cat[i] for i in [2,3,1,4,0]]

  data = pd.concat([data, data_c], axis=1)
  del data_c

  """## FEATURES (Preditoras)"""

  # Features do estudo
  all = list(data.iloc[:,11:-5])
  #data[all].describe().T

  """## Análise Exploratória

  ### Funções e ajustes
  """

  avnd =
  ['#FF5800','#CE056A','#C80000','#890078','#FFB414','#008C95','#4B912A','#006EC6']
  avnd_dark =
  ['#DC4600','#A50646','#9E120E','#5A1455','#E6A61C','#005F62','#05732A','#004B7D']
  avnd_gray =
  ['#E5E5E5','#CCCCCC','#B2B2B2','#999999','#7F7F7F','#666666','#4C4C4C','#333333']

  freq = list(data.iloc[:,11:34])
  dur = list(data.iloc[:,34:-5])

  # tranformação quantilica
  def transf (x):
    a =
  pd.DataFrame(preprocessing.quantile_transform(x,output_distribution='normal'))
    a.columns = x.columns
    return (a)

  def transf_1 (x):
    return np.log(x+1)

  def transf_2 (x):
    return np.log(x)

  data_new = pd.concat([data[resp], transf_1(data[all]), data[cat]], axis=1)

  for i in range(5):
    data_new[cat[i]+'_'] = data_new[cat[i]].map( {0:'Baixo',
                                                  1:'Alto'} ).astype('category')

  D = data.iloc[:,2:-1].describe().T.style.format("{0:.2f}").background_gradient(
      subset=['mean','std'], cmap='Blues')

  D.set_caption("Resumo Estatístico das Variáveis Numéricas")\
    .set_table_styles([{'selector': 'caption',
                        'props':'caption-side: top; font-size:1.5em'}],
                      overwrite=False)

  """### Target"""

  fig = plt.figure(figsize = (15,10))
  j = 0
  for i in cat:
```

```
201       plt.subplot(2, 3, j+1)
202       a = data[i].value_counts()
203       a.index = ['Baixo','Alto']
204       cols = list(map(lambda x: x.replace('CAT_',''),cat))
205       plt.pie(list(a), labels = a.index, colors = ['#FF5800','#CCCCCC'],
206               autopct='%.0f%%')
207       plt.title(cols[j])
208       j += 1
209
210   #fig.text(x=0.5, y=0.94, s="Matrizes de Confusão", fontsize=26, ha="center",
211   transform=fig.transFigure)
212   #fig.text(x=0.5, y=0.89, s= "base de teste", fontsize=18, ha="center",
213   transform=fig.transFigure)
214   #fig.suptitle('Distibuição das Vozes')
215   fig.tight_layout()
216   fig.subplots_adjust(top=0.84, wspace=0.2)
217
218   """### Features"""
219
220   from matplotlib import colors
221   def plot_features (dt,x):
222       sns.set_theme(style="ticks")
223       fig = plt.figure(figsize=(25, 15))
224       outer = gridspec.GridSpec(5, 6, wspace=0.2, hspace=0.3)
225       for i in range(len(x)):
226           inner = gridspec.GridSpecFromSubplotSpec(2, 1,
227                                                    subplot_spec=outer[i],
228                                                    wspace=0.1, hspace=-0.02,
229                                                    width_ratios=[1],
230                                                    height_ratios=[0.15, 0.85])
231           # subplot 1
232           ax = plt.Subplot(fig, inner[0])
233           sns.boxplot(dt[x[i]], color='#CCCCCC', ax=ax)
234           ax.axis('off')
235           fig.add_subplot(ax)
236
237           #subplot 2'#DC4600'
238           ax = plt.Subplot(fig, inner[1])
239           sns.histplot(data=dt, x=x[i], kde=True,color='#DC4600', ax=ax)
240           sns.despine(top=True, right=True, left=False, bottom=False, ax=ax)
241           ax.set(ylabel='')
242           fig.add_subplot(ax)
243
244       fig.show()
245
246   plot_features(data_new, resp)
247
248   plot_features(data, [freq[i] for i in [0,1,3,7]])
249
250   plot_features(data_new, [freq[i] for i in [0,1,3,7]])
251
252   plot_features(transf_1(data),dur)
253
254   """### Conjunta - Relação"""
255
256   filename = '/content/drive/MyDrive/Colab Notebooks/Ciencia_de_Dados/TCC/img/'
257   for i in np.arange(0,50,5):
258     g = sns.pairplot(data_new[resp+all], y_vars=resp, x_vars=all[i:i+5])
259     #g = sns.jointplot(transf(data[resp+all]), x_vars=resp, y_vars=all[i:i+5],
260     #                  kind="scatter")
261     #g.map_lower(sns.kdeplot)
262     #g.map_upper(sns.kdeplot)
263     g.map_offdiag(sns.kdeplot)
264     g.map_diag(sns.kdeplot)
265     plt.savefig(filename+'output'+str(i)+'.jpg')
266     plt.show()
267     #data[all].iloc[:,i:i+5]
268
```

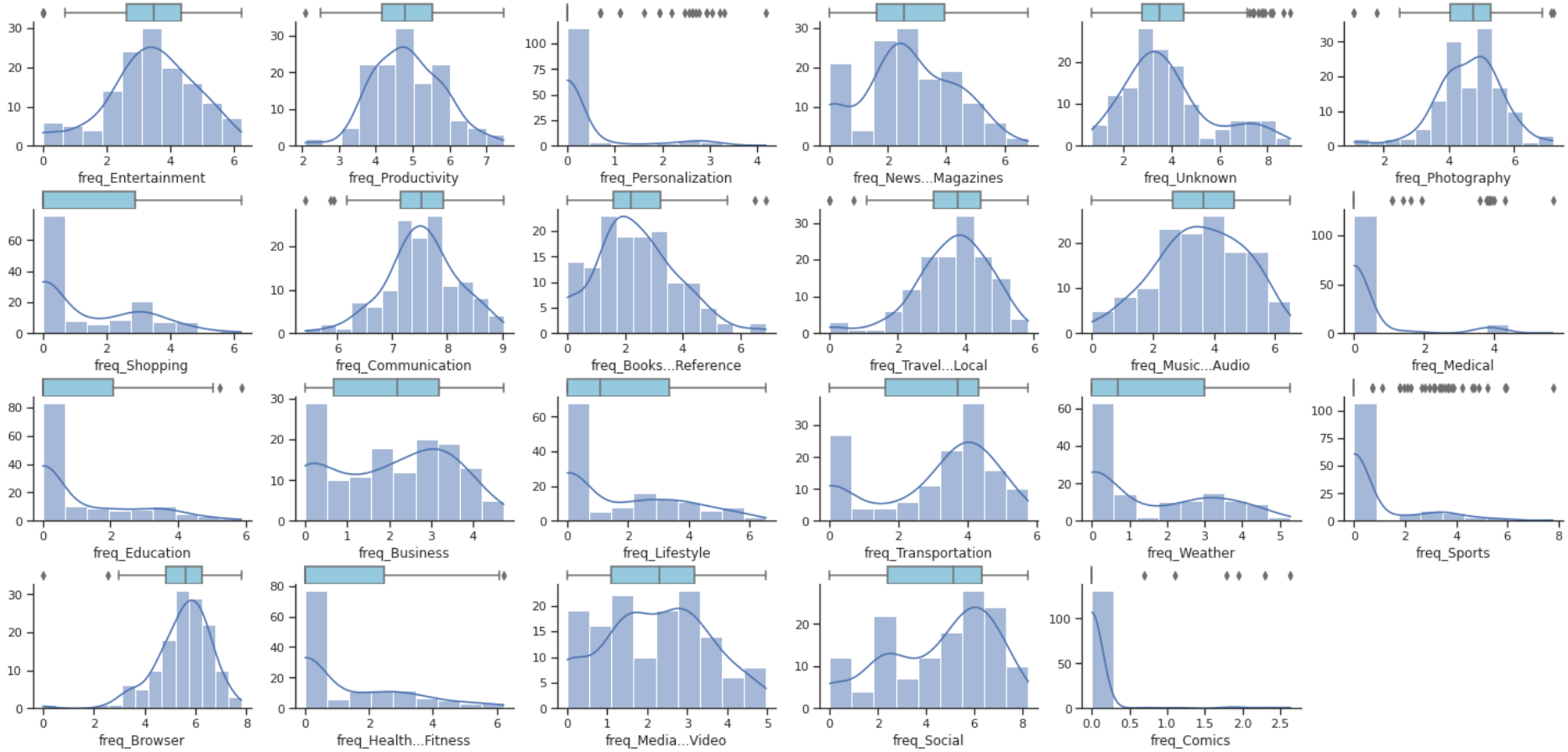
```
# correlação entre as Target e Features
#c1 = data_new[resp + [freq[i] for i in [0,1,5,7,8,9,10,11]] + [dur[i] for i in
[2,10,14,26]] ]
c1 = data_new[resp + dur]

# set the significance threshold
psig = 0.05
# get p-values
pvals1 = c1.corr(method=lambda x, y: pearsonr(x, y)[1]) - np.eye(*c1.corr().shape)

plt.figure(figsize=(20, 10))
sns.heatmap(c1.corr()[pvals1<psig].abs(), annot=True,cmap="Blues",
            linewidth=1, mask=np.triu(c1.corr()[pvals1<psig].abs(), k=1))
plt.show()

"""### Outliers"""

# Outliers consideram do z >3

# Standard scale
dt = abs(stats.zscore(data_new[all]))
#dt = data[all]
dt_n = dt[(dt<3).all(axis=1)]
x= (dt.shape[0]-dt_n.shape[0])/dt.shape[0]*100
#print("%.2f das features foram consideradas Outliers" %(x))
print(f'{x:.2f}'+'% dos dados em todas features foram consideradas Outlires
considerando o Standar Scale')

# caso especial para decidir pela permanência de todosoos dados

#sns.set(style="darkgrid")

f, (ax_box, ax_hist) = plt.subplots(2, sharex=True, gridspec_kw={"height_ratios":
(.15, .85)}, figsize=(7, 5))

# assigning a graph to each ax
sns.boxplot(data[freq[7]], ax=ax_box)
sns.histplot(data=data, x=freq[7], ax=ax_hist)

# Remove x axis name for the boxplot
ax_box.set(xlabel='')
plt.show()

"""# Dividindo o Dataset"""

#split train and test
#X =
pd.DataFrame(preprocessing.quantile_transform(data_new[all],output_distribution='no
rmal'),columns=all)
X = data_new[all]
y = data_new[cat]
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.3,
random_state = 73)

"""# Seleção das features

### Selecionando Feaqtures via Reg Lasso

Baseado nas respostas (y) continua e em features (X_i) zscore.
"""

from sklearn.feature_selection import chi2
from sklearn.feature_selection import SelectKBest

from sklearn.feature_selection import SelectFromModel
from sklearn.linear_model import Lasso
from sklearn.pipeline import Pipeline

```

```
337 ## Mesmo o Alpha best sendo definido escolheu-se usar um valor proximo de 0
338 # menor o a penalização seja ... verificar a relação entre alpha e apenalização dos
339 coeficientes
340 ## quanto maior o alpha(penalidade) menor as variáveis selecionadas |||--- Checar
341 essa proposição
342
343 def features_ (X,y):
344   sel = SelectFromModel(Lasso(alpha=0.06, random_state=25)) # regressão Lasso para
345 selecionar
346   #sel = SelectKBest(chi2, k=20).fit(X_train,y_train) # seleção por quiquadrado
347   sel.fit(X,y)
348   a = list(X.columns[sel.get_support()])
349   b = list(X)
350   print('*'*60)
351   print('Features para ', y.name)
352   print('*'*60)
353   print('N° Features Inicial     :', len(b))
354   print('N° Features Excluídas   :',len(b)-len(a))
355   print('N° Features Selecionadas:', len(a))
356   print('-'*60)
357   print('Features Selecionadas:')
358   print(list(a))
359   return(a)
360
361 # Dicionário das features por Respostas
362 Modelos = [{'Resposta': "" ,'Features': list(), 'Modelo':['','','','',''']},
363            {'Resposta': "" ,'Features': list(), 'Modelo':['','','','',''']},
364            {'Resposta': "" ,'Features': list(), 'Modelo':['','','','',''']},
365            {'Resposta': "" ,'Features': list(), 'Modelo':['','','','',''']},
366            {'Resposta': "" ,'Features': list(), 'Modelo':['','','','',''']}]
367
368 X = data_new[all]
369 #X = stats.zscore(data[all])
370 #X = pd.DataFrame(preprocessing.normalize(data[all]),columns=all)
371 #X =
372 pd.DataFrame(preprocessing.quantile_transform(data[all],output_distribution='normal
373 '),columns=all)
374 for i,_ in enumerate(resp):
375   y = data[resp[i]]
376   x = features_(X,y)
377   Modelos[i]['Resposta'] = cat[i]
378   Modelos[i]['Features'] = x
379
380 ## Para visualização do quadro de regressoes por variável
381 df = pd.DataFrame(columns=cat, index=all)
382
383 for i in range(5):
384   df.loc[Modelos[i]['Features'],Modelos[i]['Resposta']] = 1
385 df = df.fillna(0)
386
387 df.columns = list(map(lambda x: x.replace('CAT_',''),cat)) # remover o termo CAT_
388
389 plt.figure(figsize=(15, 15))
390 plt.spy(df.T, marker="x")
391 plt.xticks(list(range(50)),list(df.index), rotation=90)
392 plt.yticks(list(range(5)),list(df.columns))
393 plt.show()
394
395 import collections
396 for i in range(5):
397   print(cat[i], collections.Counter(y_test[cat[i]]))
398
399 """# Modelos de classificação"""
400
401 import pickle # para salvar modelo
402 filename = '/content/drive/MyDrive/Colab Notebooks/Ciencia_de_Dados/TCC/Models/'
403
404 def plt_CM (model, f, r, t, labels=['Baixo','Alto']):
```

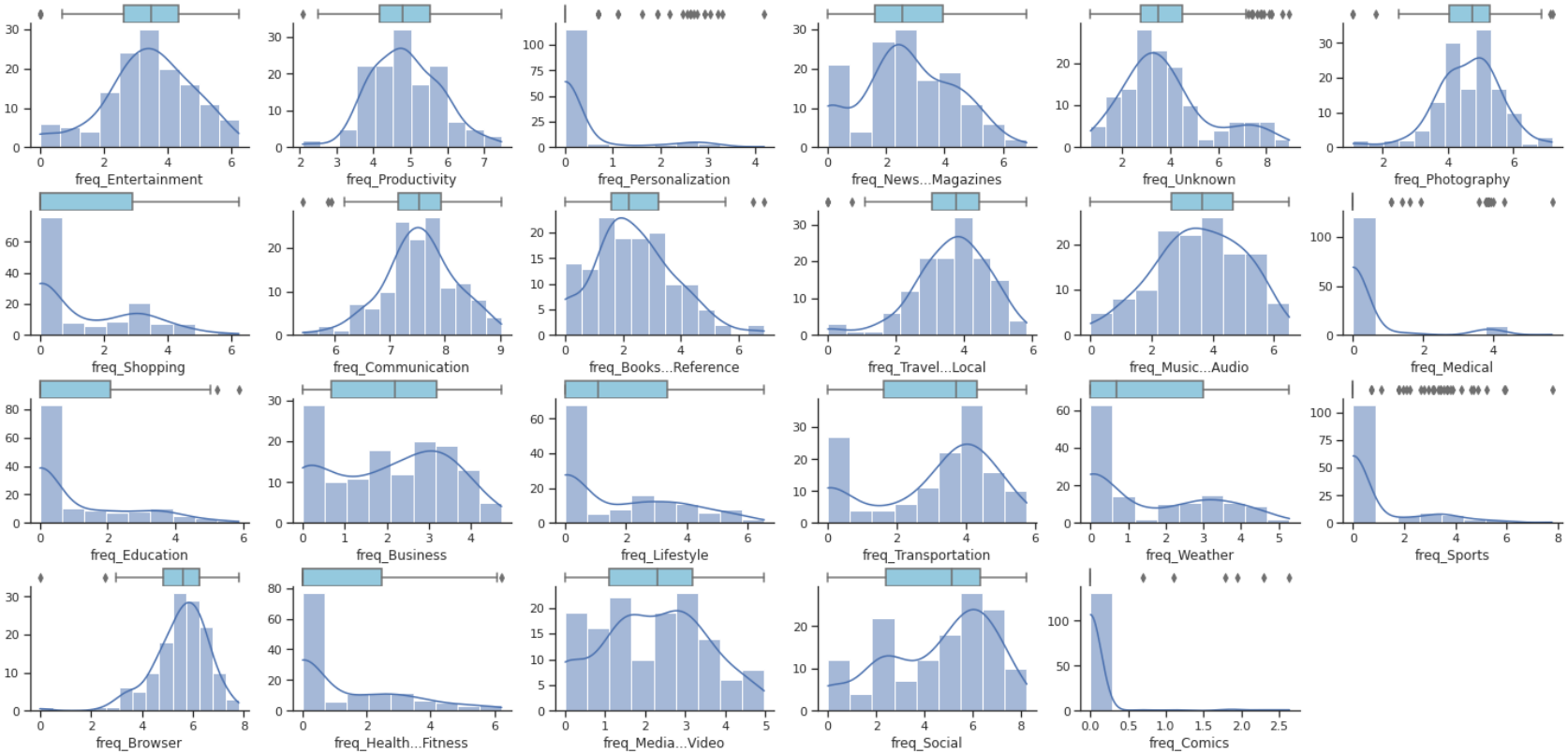
```
405     # labes in array [ ]
406     #gráfico com base na matriz de confusão simples de metrics
407     ConfM=metrics.confusion_matrix(y_test[r],model.predict(X_test[f]))
408     # Auxiliar nos gráficos
409
410     sns.heatmap(pd.DataFrame(ConfM, index=labels, columns=labels), cmap='Oranges',
411   annot=True,fmt='d' , cbar=False)
412     plt.title(t)
413     plt.xlabel('Predict Label')
414     plt.ylabel('True Label')
415     plt.yticks(rotation=0)
416     #plt.show()
417
418   # Modelos de classificação
419   cls = ['KNC','LDA','RF','VotingCls','VotingCls_' ]
420   #clf1 = SVC(random_state = 45)
421   clf1 = KNeighborsClassifier(weights='distance')
422   clf2 = LinearDiscriminantAnalysis()
423   clf3 = RandomForestClassifier(random_state = 45)
424
425   eclf = VotingClassifier(
426       estimators=[('knc', clf1), ('ld', clf2), ('rf', clf3)],
427       voting='hard')
428
429   eclf_ = VotingClassifier(
430       estimators=[('knc', clf1), ('ld', clf2), ('rf', clf3)],
431       voting='soft')
432
433   #params1 = {'kernel':['linear'],'C':[0.1, 2, 5]}
434   params1 = {'n_neighbors':np.arange(1,9,2)}
435
436   params2 = {'solver':['lsqr'], 'shrinkage':np.arange(.7, .8, 0.1)}
437
438   params3 = {'n_estimators': [4,5,10,20],
439              'max_features': ['auto'],
440              'max_depth' : [4,5,10,20]}
441
442   params = {#'svc__kernel':['linear'],
443             #'svc__C':[0.1],
444             'knc__n_neighbors':np.arange(1,9,2),
445             'ld__solver':['lsqr'],
446             'ld__shrinkage':np.arange(.7, .8, 0.1),
447             'rf__n_estimators': [4,5,10,20],
448             'rf__max_features': ['auto'],
449             'rf__max_depth' : [4,5,10,20]}
450
451   grid=['','','','','']
452
453   grid[0] = GridSearchCV(estimator=clf1, param_grid=params1, cv=3,
454                          scoring = 'precision', verbose = 4 ,n_jobs = -1)
455
456   grid[1] = GridSearchCV(estimator=clf2, param_grid=params2, cv=3,
457                       scoring = 'precision', verbose = 4 ,n_jobs = -1)
458
459   grid[2] = GridSearchCV(estimator=clf3, param_grid=params3, cv=3,
460                       scoring = 'precision', verbose = 4 ,n_jobs = -1)
461
462   grid[3] = GridSearchCV(estimator=eclf, param_grid=params, cv=3,
463                       scoring = 'precision', verbose = 4 ,n_jobs = -1)
464
465   grid[4] = GridSearchCV(estimator=eclf_, param_grid=params, cv=3,
466                       scoring = 'precision', verbose = 4 ,n_jobs = -1)
467
468   warnings.filterwarnings("ignore")
469
470   for i in range(5):
471     start = time.time()
472     f = Modelos[i]['Features']
```

```
473     r = cat[i]
474     for j in range(len(grid)) :
475       name = str(Modelos[i]['Resposta']+"_"+cls[j])
476       model = grid[j].fit(X_train[f],y_train[r] )
477       Modelos[i]['Modelo'][j] = name
478       pickle.dump(model, open(filename+name+'.sav', 'wb'))
479     timer(start,time.time())
480
481   for i in range(5):
482     fig = plt.figure(figsize=(15,10))
483     f = Modelos[i]['Features']
484     r = Modelos[i]['Resposta']
485     for j in range(4):
486       name = Modelos[i]['Modelo'][j]
487       loaded_model = pickle.load(open(filename+name+'.sav', 'rb'))
488       print(name)
489       print(loaded_model.best_params_)
490       plt.subplot(2, 2, j+1)
491       plt_CM(loaded_model,f,r,cls[j])
492     fig.text(x=0.1, y=0.94, s=Modelos[i]['Resposta'], fontsize=18, ha="center")
493     plt.show()
494
495   fig = plt.figure(figsize=(15,10))
496   j = 0
497
498   labels = ['Baixo','Alto']
499
500   for i in range(5):
501     f = Modelos[i]['Features']
502     r = Modelos[i]['Resposta']
503     name = Modelos[i]['Modelo'][3]# fixar o esemble
504     model = pickle.load(open(filename+name+'.sav', 'rb'))
505
506     ConfM=metrics.confusion_matrix(y_test[r],model.predict(X_test[f]))
507
508     plt.subplot(2, 3, j+1)
509     j += 1
510     sns.heatmap(pd.DataFrame(ConfM, index=labels, columns=labels), cmap='Oranges',
511   annot=True, fmt='d', cbar=False)
512     plt.title(r)
513     plt.xlabel('Predict Label')
514     plt.ylabel('True Label')
515
516   fig.text(x=0.5, y=0.94, s="Matrizes de Confusão", fontsize=26, ha="center",
517   transform=fig.transFigure)
518   fig.text(x=0.5, y=0.89, s= "base de teste", fontsize=18, ha="center",
519   transform=fig.transFigure)
520   fig.tight_layout()
521   fig.subplots_adjust(top=0.84, wspace=0.2)
522   plt.show()
```

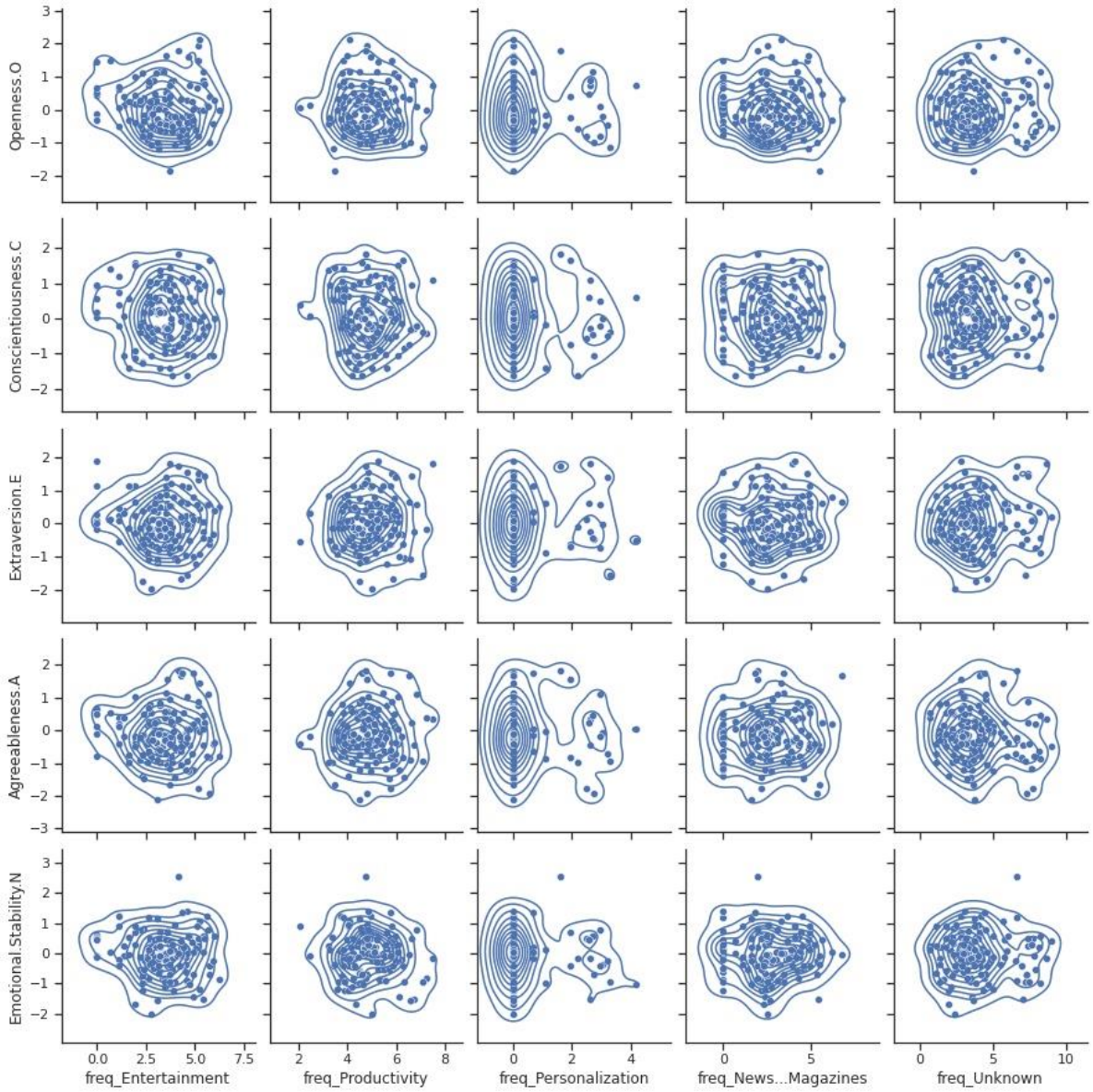
APÊNDICE B – *FEATURES* DE FREQUÊNCIA APÓS TRANSFORMAÇÃO LOGARÍTMICA

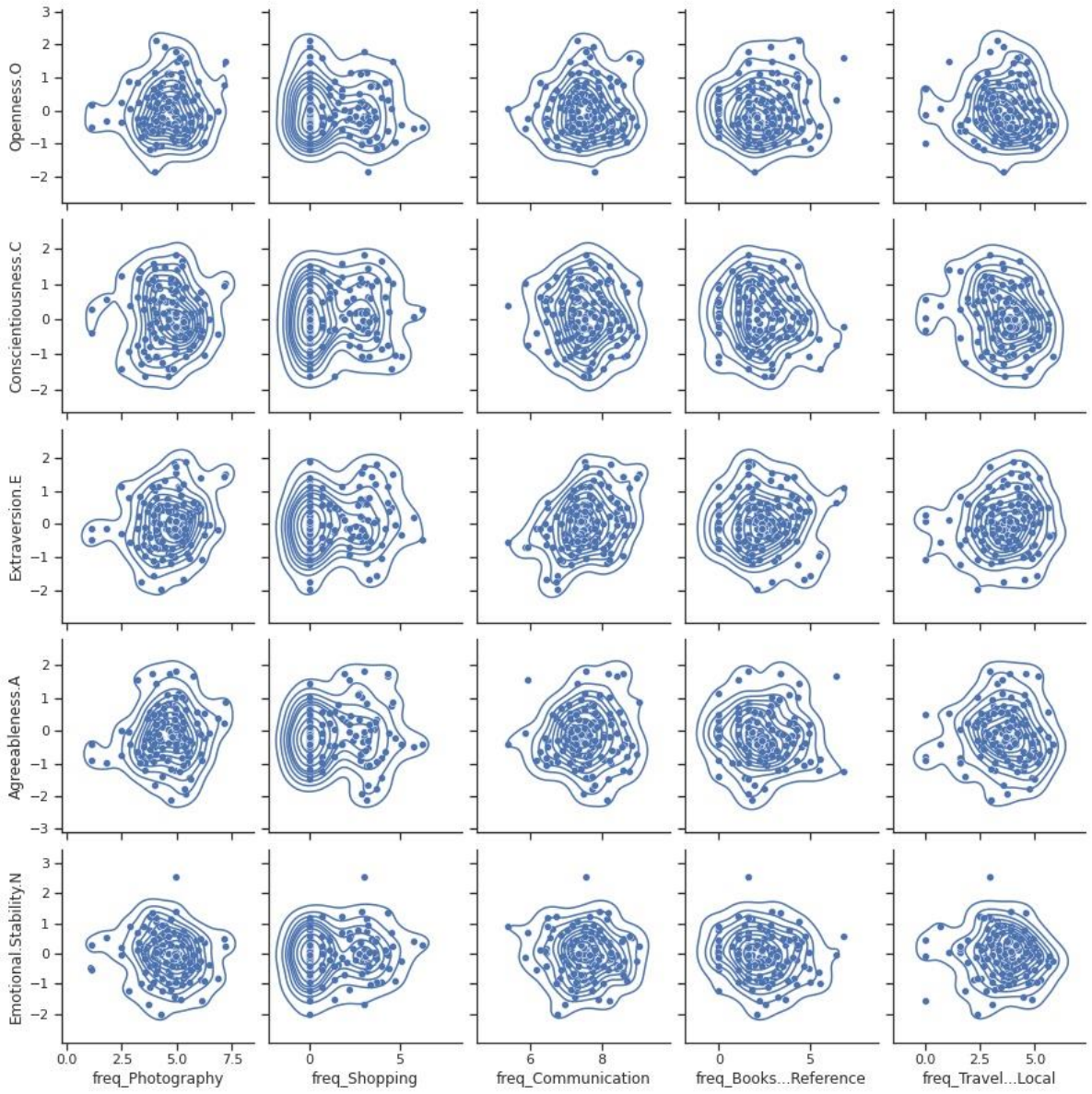


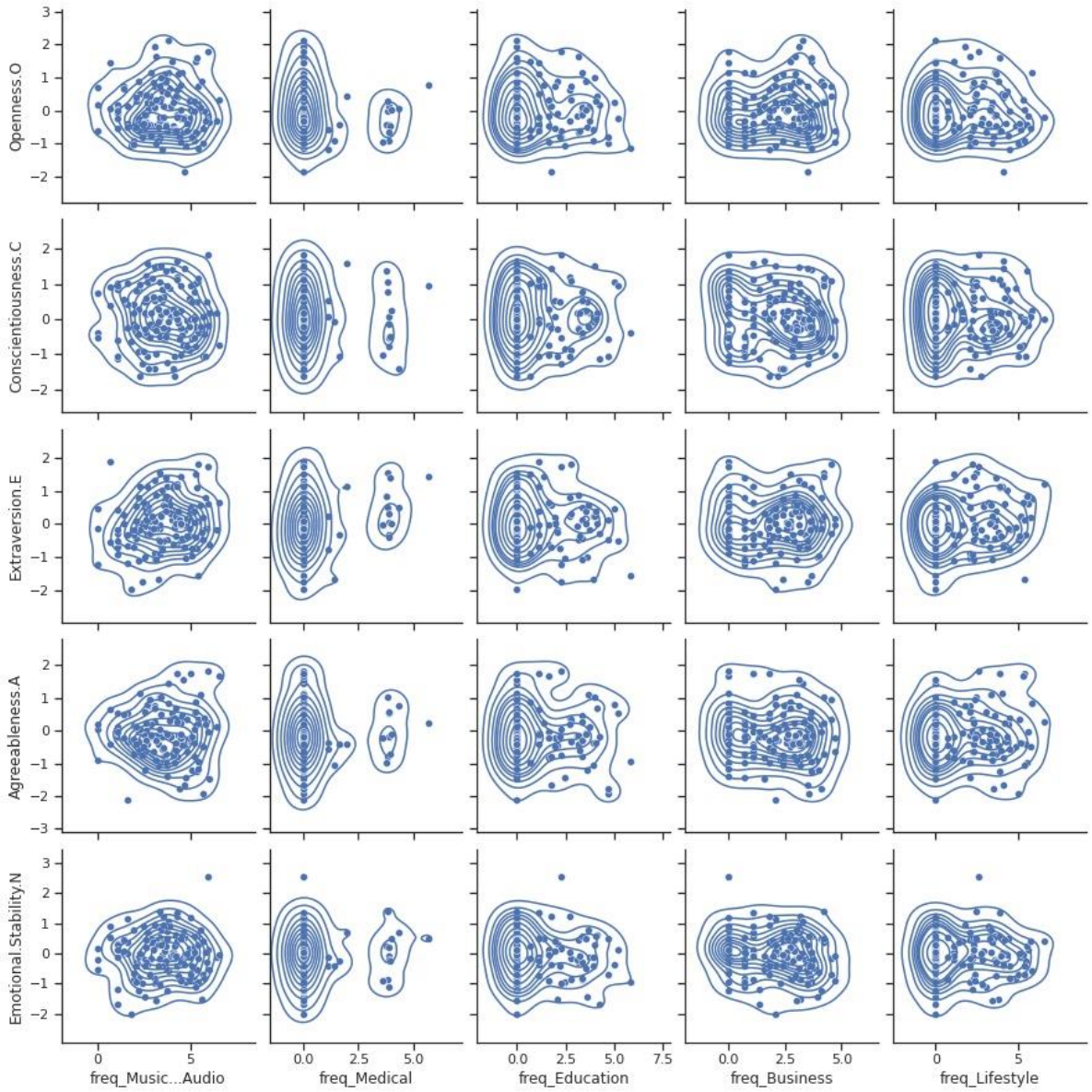
APÊNDICE C – *FEATURES* DE DURAÇÃO APÓS TRANSFORMAÇÃO LOGARÍTMICA

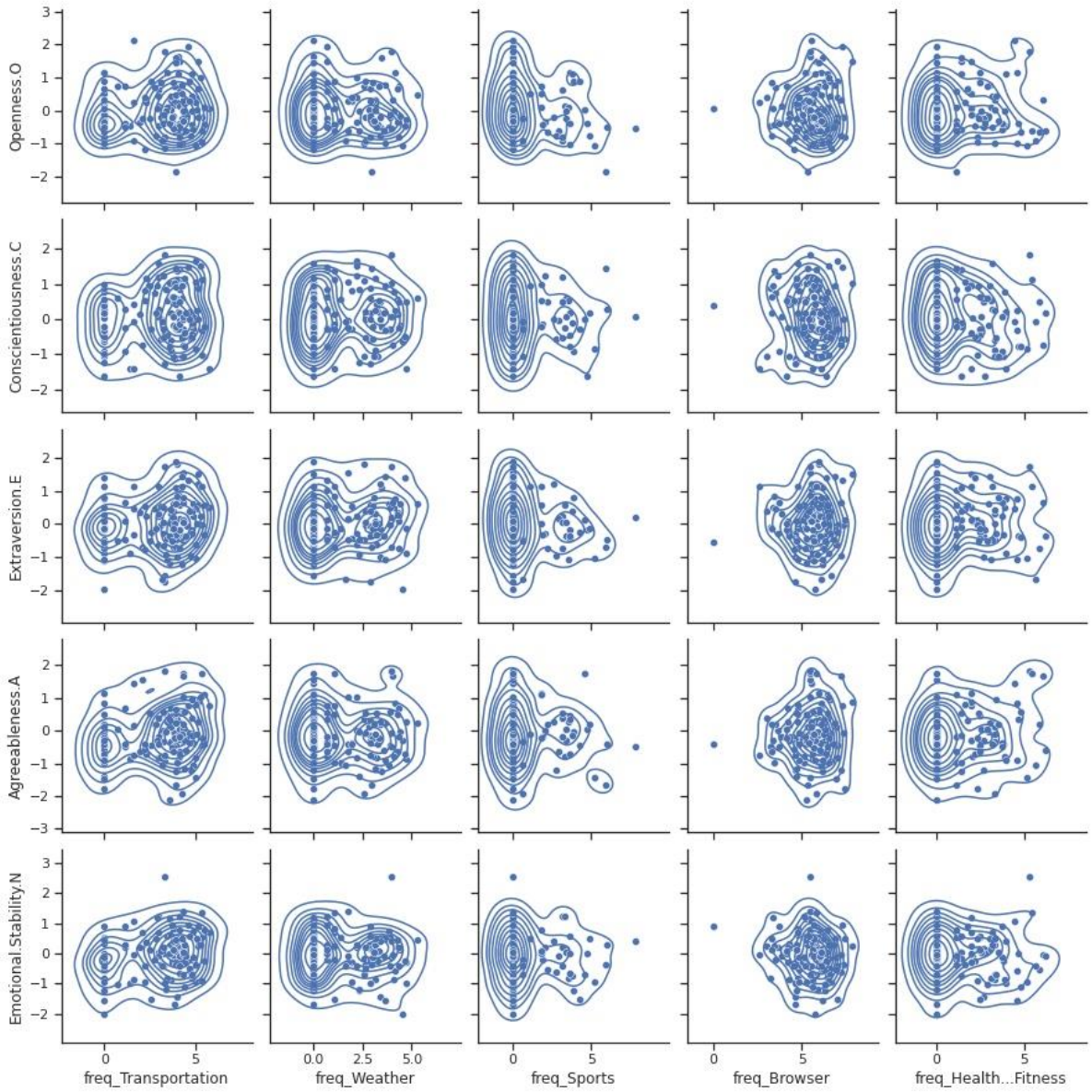


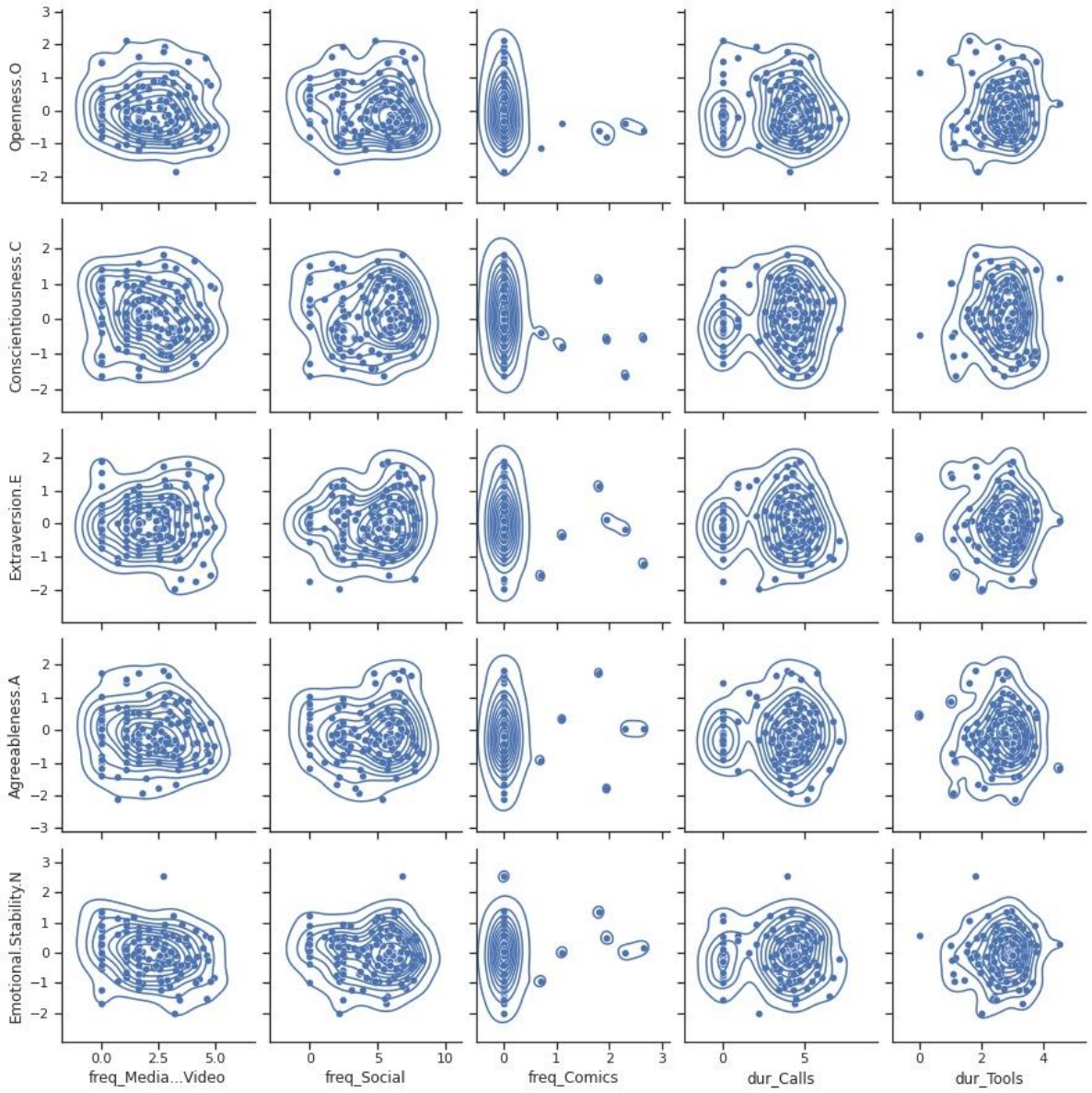
APÊNDICE D – RELAÇÃO ENTRE *TARGETS E FEATURES*

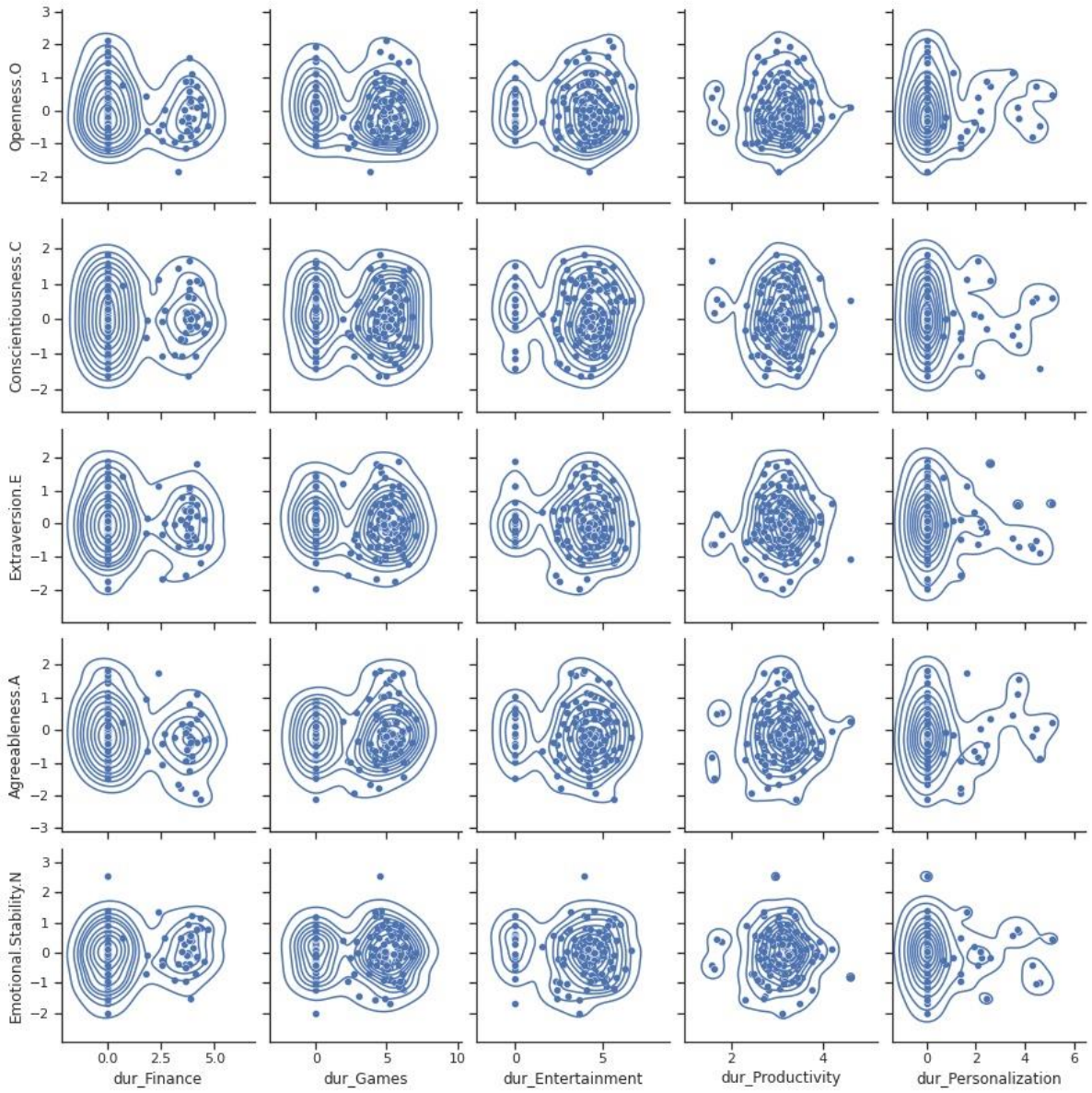


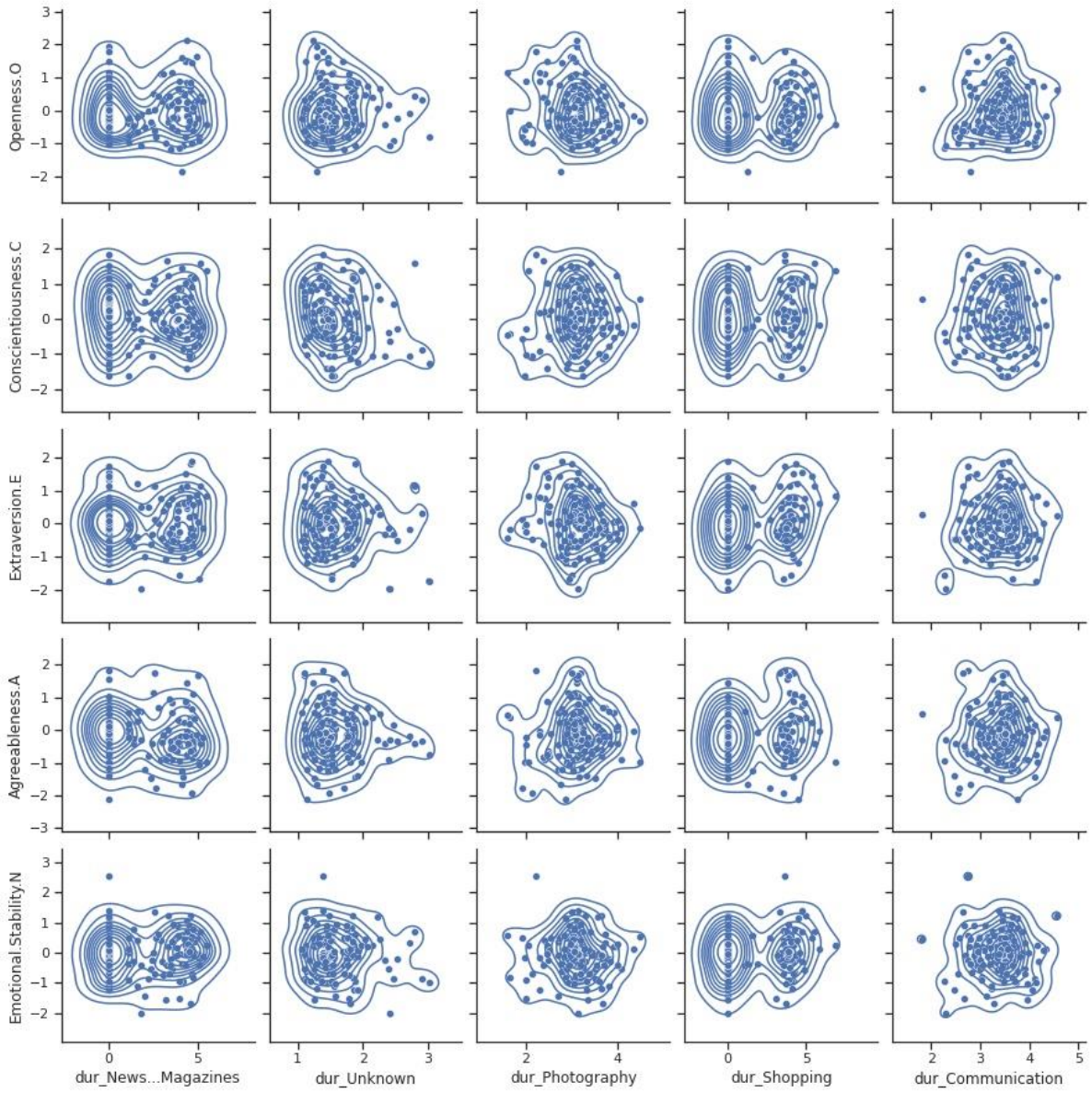

Openness.O
Conscientiousness.C
Extraversion.E
Agreeableness.A
Emotional.Stability.N
freq_Photography
freq_Shopping
freq_Communication
freq_Books...Reference
freq_Travel...Local

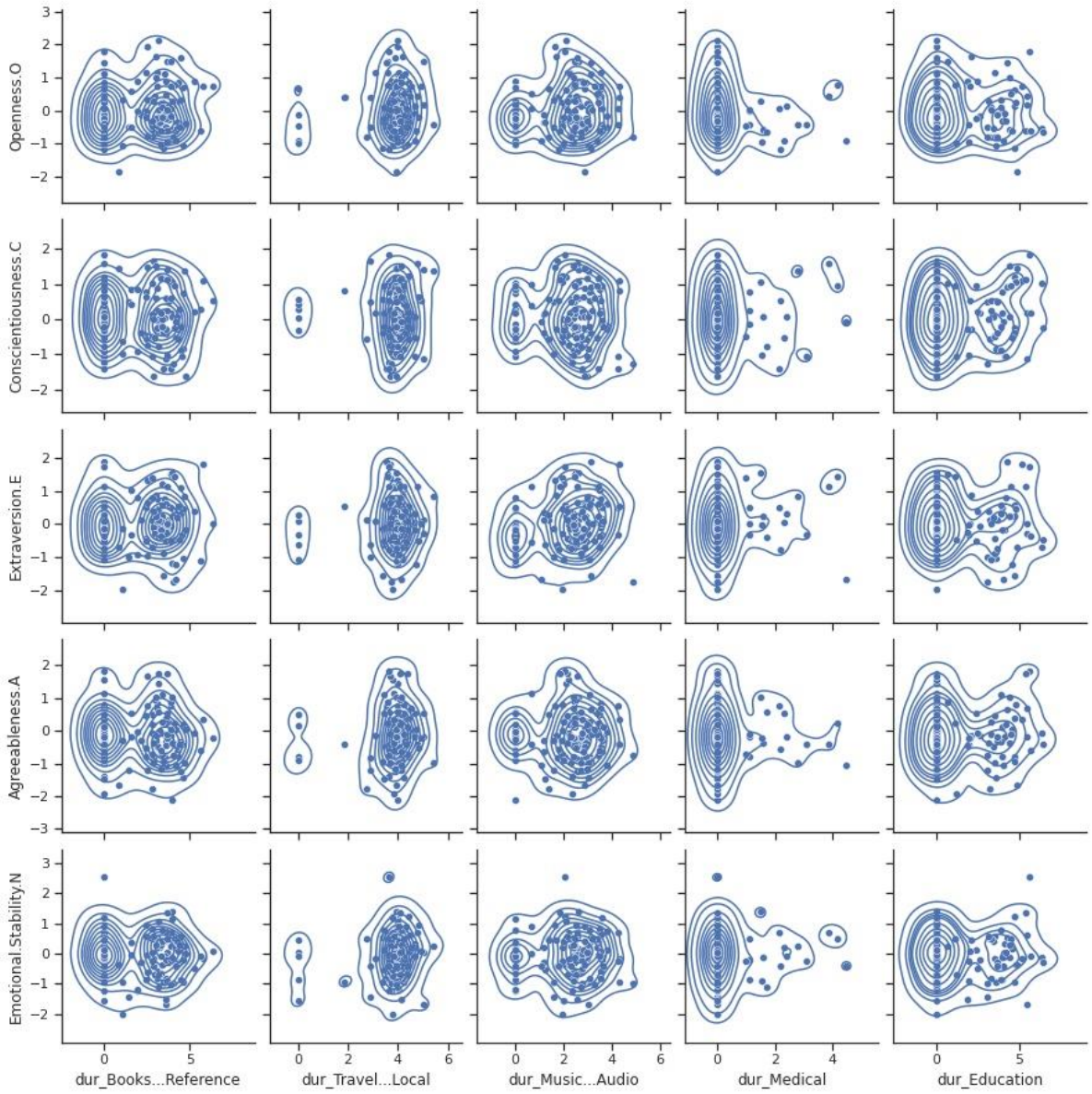

Openness.O
Conscientiousness.C
Extraversion.E
Agreeableness.A
Emotional.Stability.N
freq_Music...Audio
freq_Medical
freq_Education
freq_Business
freq_Lifestyle

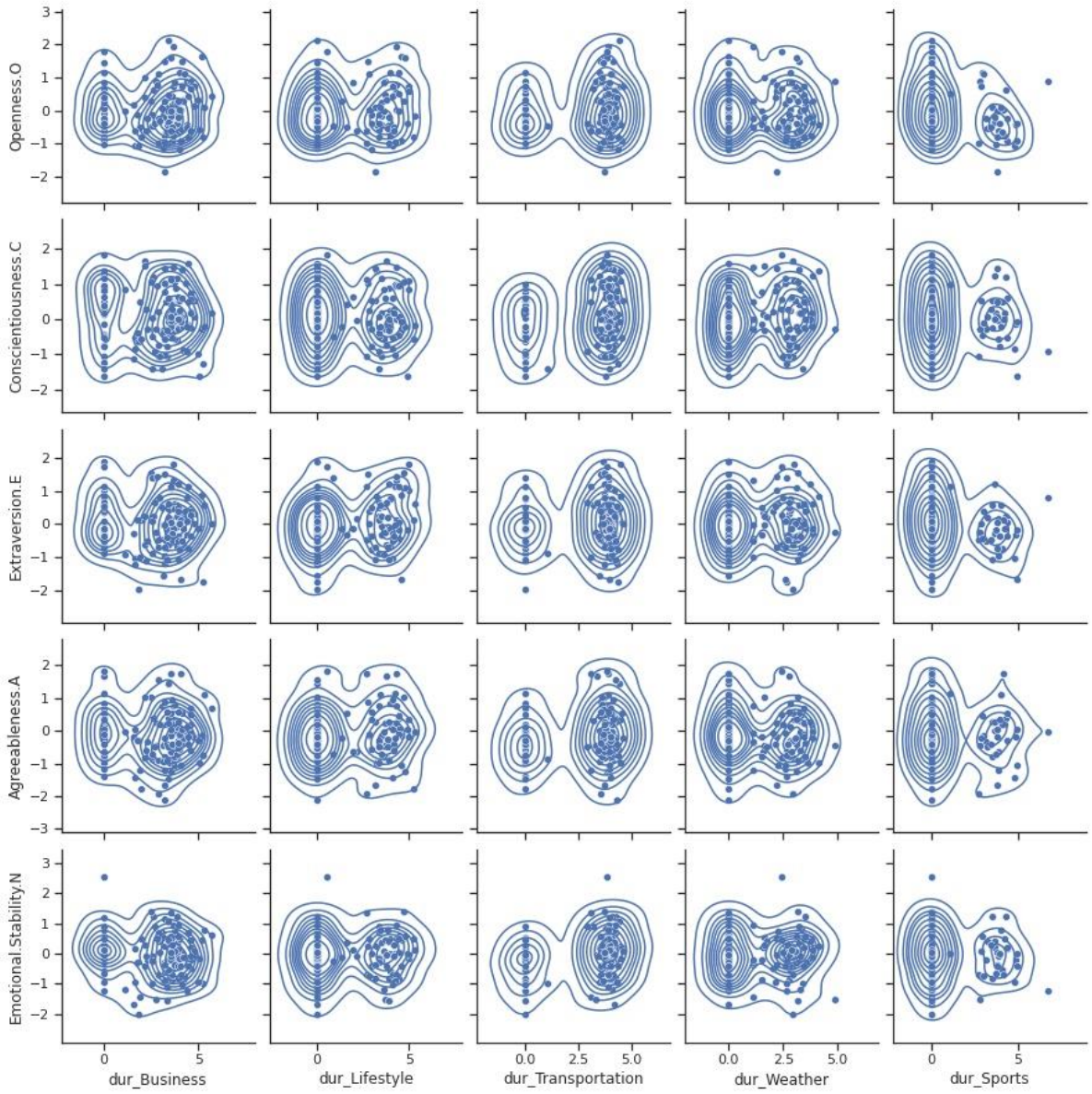

Openness.O
Conscientiousness.C
Extraversion.E
Agreeableness.A
Emotional.Stability.N
freq_Transportation
freq_Weather
freq_Sports
freq_Browser
freq_Health...Fitness

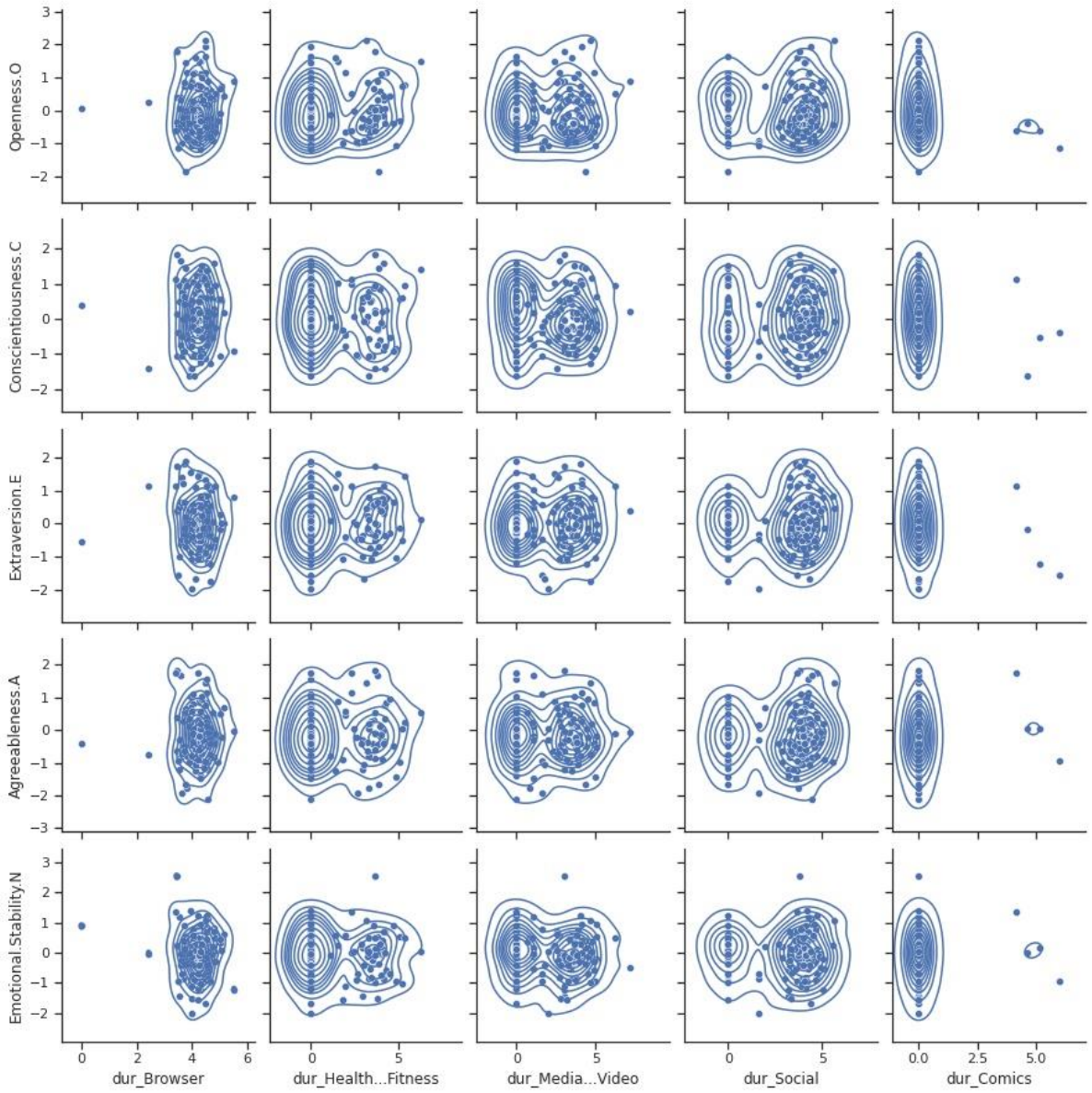

Openness.O
Conscientiousness.C
Extraversion.E
Agreeableness.A
Emotional.Stability.N
freq_Media...Video
freq_Social
freq_Comics
dur_Calls
dur_Tools


Openness.O
Conscientiousness.C
Extraversion.E
Agreeableness.A
Emotional.Stability.N
dur_Finance
dur_Games
dur_Entertainment
dur_Productivity
dur_Personalization

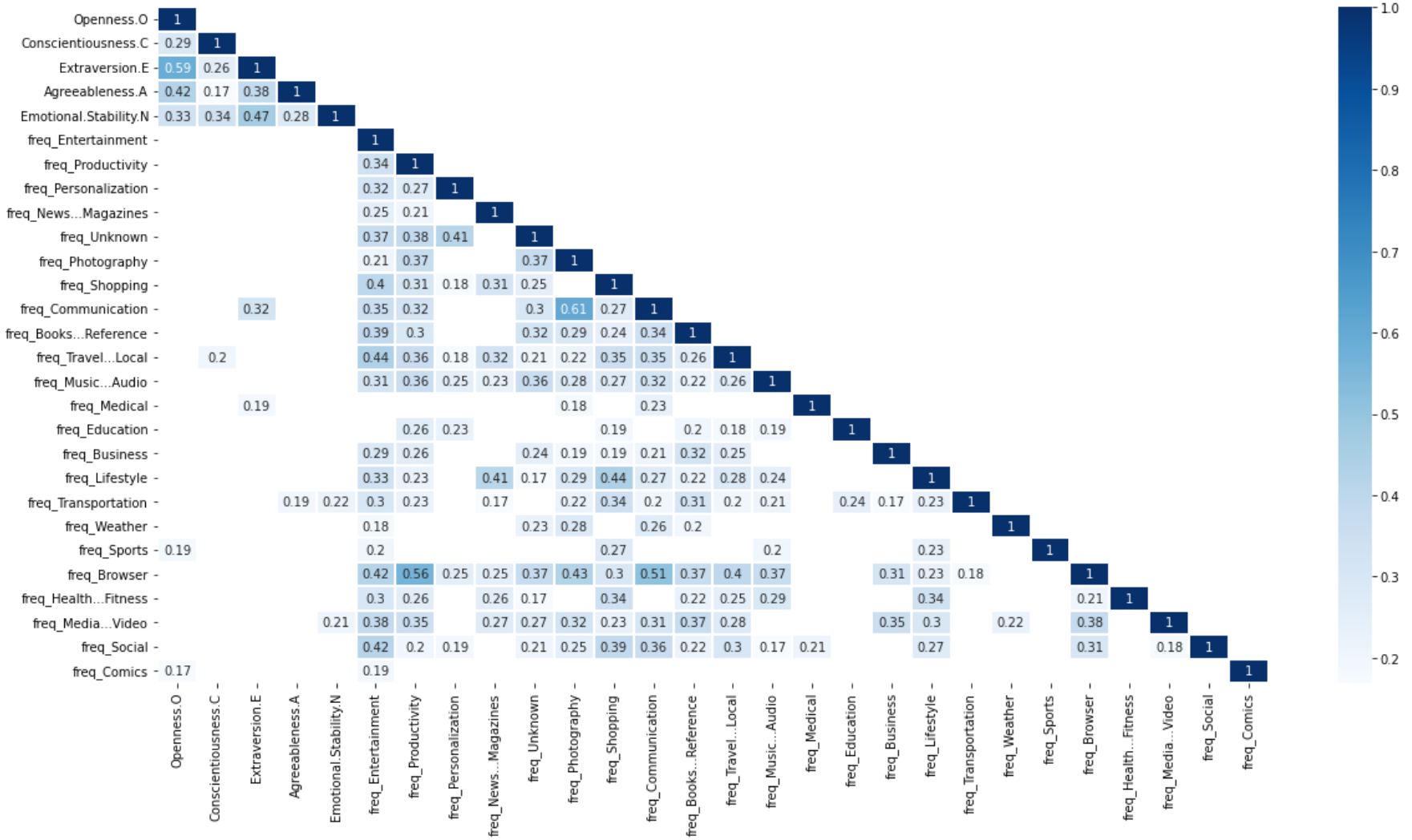

Openness.O
Conscientiousness.C
Extraversion.E
Agreeableness.A
Emotional.Stability.N
dur_News...Magazines
dur_Unknown
dur_Photography
dur_Shopping
dur_Communication


Openness.O
Conscientiousness.C
Extraversion.E
Agreeableness.A
Emotional.Stability.N
dur_Books...Reference
dur_Travel...Local
dur_Music...Audio
dur_Medical
dur_Education

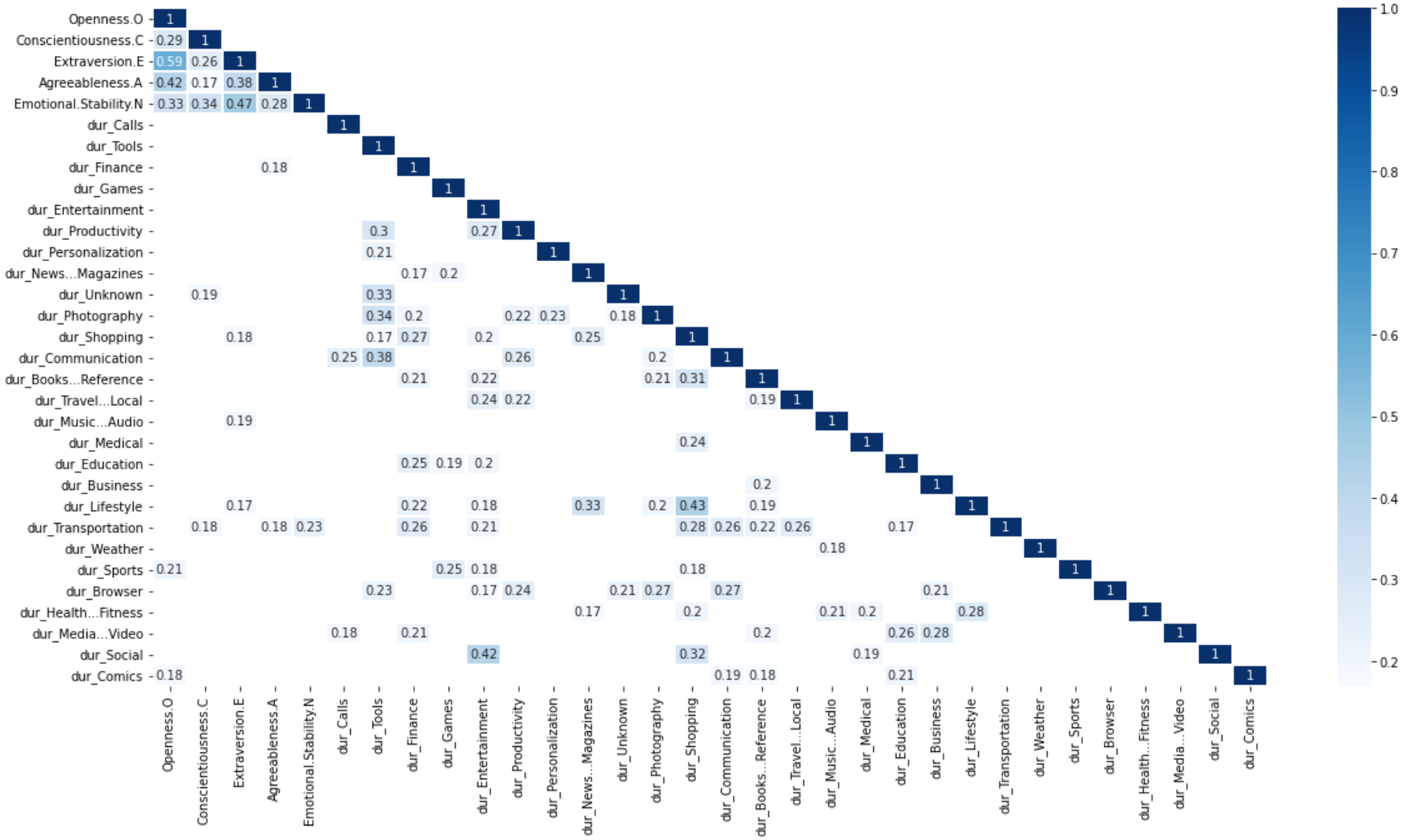

Openness.O
Conscientiousness.C
Extraversion.E
Agreeableness.A
Emotional.Stability.N
dur_Business
dur_Lifestyle
dur_Transportation
dur_Weather
dur_Sports


Openness.O
Conscientiousness.C
Extraversion.E
Agreeableness.A
Emotional.Stability.N
dur_Browser
dur_Health...Fitness
dur_Media...Video
dur_Social
dur_Comics

APÊNDICE E – CORRELAÇÃO ENTRES *TARGETS* E *FEATURES* DE FREQUÊNCIA



APÊNDICE F – CORRELAÇÃO ENTRES *TARGETS* E *FEATURES* DE DURAÇÃO



ANEXOS

ANEXO A – ESTATÍSTICAS DO USO DE APLICATIVOS DO *THE PHONESTUDY PROJECT*

Category		Frequency		Duration		Apps [a]	Users	Most Frequently Used Apps
		M	SD	M (s)	SD (s)			
1	Communication	38,48	25,87	31,14	13,6	62	137	WhatsApp, Mail, Contacts, Dialer, SMS/MMS
2	Social	7,26	10,87	50,52	47,26	80	125	Facebook, Instagram, Snapchat, Twitter, Weibo
3	Tools	6,8	8,07	16,2	9	225	137	Google Search, Clock, Google Play Store, Calculator, S Voice
4	Browser	6,44	6,17	71,56	29,93	6	136	Internet, Firefox, Opera, Dolphin Browser, UC Browser
5	Calls	4,14	4,07	103,73	162,45	1	137	Phone
6	Productivity	3,45	4,21	23,22	10,21	134	137	Settings, S Planner, Calendar, ColorNote, Google Drive
7	Photography	2,69	3,23	23,57	11,46	47	137	Gallery, Camera, SnapApp, Album, PicsArt
8	Games	2,52	5,39	153,33	196,64	229	100	Clash of Clans, Quizduell, Candy Crush Saga, Farm Heroes Saga, Trials Frontier
9	Music & Audio	1,4	1,92	15,37	17,53	78	134	Spotify, Music Player, Google Play Music, MP3-Player, SoundCloud
10	Entertainment	1,08	1,46	94,01	108,69	98	131	YouTube, 9GAG, PlayerPro, appinio, PS4-Magazin
11	Travel & Local	1,01	1,01	54,61	24,91	85	134	Maps, MVV Companion, TripAdvisor, BlaBlaCar, Airbnb
12	Transportation	0,9	1,02	39,47	25,77	40	110	MVG Fahrinfo, DB Navigator, fi, MeinFernbus, Uber
13	News & Mag.	0,77	1,4	36,66	51,08	52	118	FOCUS Online, reddit sync, SPIEGEL ONLINE, Flipboard, SZ.de
14	Lifestyle	0,54	1,17	24,79	40	72	72	Tinder, Sleep, Chefkoch, eBay Kleinanzeigen, PAYBACK
15	Sports	0,52	3,52	18,18	77,43	38	33	kicker, Comunio, Kicktipp, Score, Sportschau
16	Books & Ref.	0,45	0,82	31,24	63,9	71	123	Munpia, dict.cc plus, dict.cc, Wikipedia, LEO
17	Health & Fitness	0,42	1,25	20,52	54,88	59	60	SleepBot, Strava, Fitbit, Freeletics, MyFitnessPal
18	Media & Video	0,32	0,46	38,65	125,42	47	118	Video-Player, Google Play Movies, VLC, Video anzeigen, ZDF
19	Shopping	0,3	0,92	33,2	91,04	45	68	eBay, mydealz, Amazon, brands4friends, Shpock
20	Business	0,28	0,37	36,41	44,99	40	108	Eigene Dateien, AnyConnect, POLARIS Office Viewer 5, Polaris Viewer 4.1, OfficeSuite
21	Education	0,22	0,67	32,85	90,41	67	54	UnlockYourBrain, AnkiDroid, TUM Campus App, Duolingo, Web Opac
22	Finance	0,22	0,75	11,37	22,43	34	39	Sparkasse, Banking 4A, Wstenrot, YNAB, Banking
23	Weather	0,18	0,24	10,65	16,66	17	74	Weather, wetter.com, WetterOnline, WetterApp, Wetter-Widget
24	Medical	0,1	0,47	2,15	10,35	10	17	Lady Pill Reminder, PillReminder, Pillreminder, iPhysikum, Remember Your Pill
25	Personalization	0,03	0,12	4,35	19,08	14	22	Dokumente, Backgrounds, Zedge, Flatastico, HD Widgets
26	Comics	0	0,03	5,33	38,01	6	6	xkcd Browser, NICHTLUSTIG, Marvel Unlimited, xkcdViewer, xkcd - Now

Fonte: The PhoneStudy Project **(STACHL, HILBERT, *et al.*, 2016)**

Nota:

[a] número total de aplicativos na categoria em todos os participantes no conjunto de dados.