

Курс: “Machine Learning and Artificial Intelligence Using Swift”

Підготувала: Шевченко Дар’я Валеріївна, ІПЗ, 3 курс

Розділи:

1. Вступ до машинного навчання та штучного інтелекту
2. Класифікація зображень за допомогою вже існуючих моделей
3. Моделі навчання класифікації зображень
4. Класифікація звуку
5. Розпізнавання облич
6. Виявлення тексту та штрих-коду
7. Класифікація зображень за допомогою Turi Create
8. Текст і мова
9. Аналіз настроїв тексту
10. Розпізнавання малюнків
11. Розпізнавання діяльності за допомогою Core Motion Framework
12. Передача стилю за допомогою Turi Create
13. Генерування речень за допомогою моделі GPT2
14. Запитання та відповіді за допомогою BERT
15. Рекомендації
16. Прогнозування цін за допомогою регресії
17. Доповнена реальність з використанням штучного інтелекту
18. Висновки

1. Вступ

Що таке штучний інтелект (AI)?

Це широка галузь інформатики, яка займається створенням розумних машин, здатних виконувати завдання, які зазвичай вимагають людського інтелекту, та вчитися на розумній поведінці людини.

Приклад ШІ:

- Deep blue computer від IBM
- Автономні автомобілі.
- Текстові повідомлення
- IBM Watson

Що таке машинне навчання (ML)?

Це тип штучного інтелекту (AI), який дозволяє програмним додаткам ставати більш точними в прогнозуванні результатів, не будучи явно запрограмованим для цього.

Чому Swift?

- Сучасний → є нові типи з сучасним синтаксисом та зручні фреймворки
- Безпечно → можливість використовувати опціонали
- Швидкий → компілятор LLVM, строго типізований, наступник мови C і Objective-C

- Відкритий код → кросплатформний
- Swift для TensorFlow → TensorFlow — це потужний інструмент для машинного навчання

Чому машинне навчання та AI?

CoreML – фреймворк машинного навчання, який допомагає створювати дуже портативні моделі. Функції включають комп’ютерний зір і програми природної мови.

Create ML – Machine Learning Framework і App, призначені для створення та оцінки моделей CoreML

Turi Create — спрощує розробку користувачьких моделей машинного навчання. Не потрібно бути експертом з машинного навчання, щоб додати рекомендації, виявлення об’єктів, класифікацію зображень, подібність зображень або класифікацію діяльності до вашої програми.

- **Easy-to-use:** Focus on tasks instead of algorithms
- **Visual:** Built-in, streaming visualizations to explore your data
- **Flexible:** Supports text, images, audio, video and sensor data
- **Fast and Scalable:** Work with large datasets on a single machine
- **Ready To Deploy:** Export models to Core ML for use in iOS, macOS, watchOS, and tvOS apps

Turi дозволяє створювати:

ML Task	Description
Recommender	Personalize choices for users
Image Classification	Label images
Drawing Classification	Recognize Pencil/Touch Drawings and Gestures
Sound Classification	Classify sounds
Object Detection	Recognize objects within images
One Shot Object Detection	Recognize 2D objects within images using a single example
Style Transfer	Stylize images
Activity Classification	Detect an activity using sensors
Image Similarity	Find similar images
Classifiers	Predict a label
Regression	Predict numeric values
Clustering	Group similar datapoints together
Text Classifier	Analyze sentiment of messages

2. On-Device Machine Learning vs Remote Machine Learning

У *on-device machine learning* ми навчаємо модель ізольовано від фактичного пристрою. Далі ми представляємо основну модель ML й інтегруємо в наш додаток для iOS.

Переваги:

- завжди доступний, навіть без підключення до Інтернету
- Попереджає використування даних Wi-Fi для отримання моделі, вони вже є на пристрії
- зберігає конфіденційність

Недоліки:

- важко оновлювати моделі
- збільшує розмір програми

Remote Machine Learning просто означає, що модель насправді не на пристройі, а доступ до неї здійснюється через перевірене підключення до мережі. Приклад - Firebase, ML Kit

Що make deep learning?

Artificial intelligence > Machine Learning > Deep Learning

Deep Learning - підмножина машинного навчання, у якій штучні нейронні мережі адаптуються та навчаються на величезних обсягах даних

3. Image Classification - Training Models

Що make image classification?

Класифікація зображень відноситься до маркування зображень до попередньо визначених класів

Image Classification Tasks

- Попередня обробка зображень
- Виявлення об'єкта
- Вилучення та навчання характеристик
- Класифікація об'єкта

Остаточний проект - програма для малювання, яка класифікує зображення

Кроки:

1 - розробка додатку

2 - завантажити набір даних

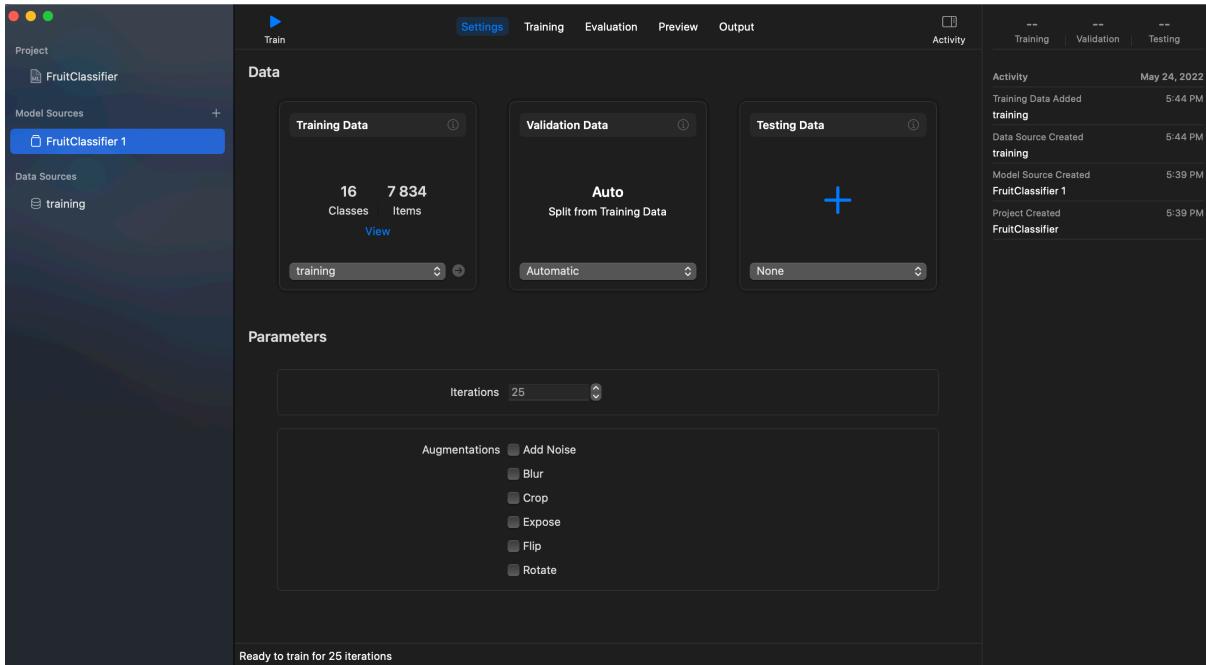
3 - навчальна модель класифікації зображень

4 - інтеграція моделі в проект

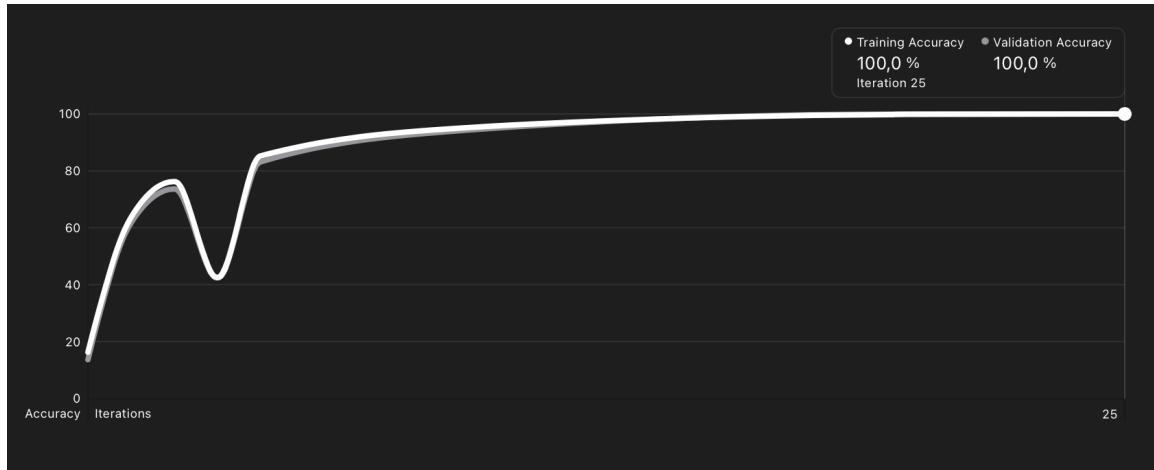
Як працювати з CreateML

Дані - <https://www.kaggle.com/datasets/moltean/fruits>

Завантаження даних до CreateML



Результат тренувань



Тестування

The screenshot shows the ModelIO application interface. At the top, there are tabs: Train, Settings, Training, Evaluation (which is highlighted in blue), Preview, Output, and Activity. On the left, there's a sidebar with sections for Training, Validation, and Testing. Under Testing, there is a list item for 'test' with the timestamp 'May 24, 2022 at 5:52 PM'. The main area displays the evaluation results for the 'test' model. The table has columns for Class, Item Count, Precision, and Recall. The data is as follows:

Class	Item Count	Precision	Recall
Apple Crimson Snow	148	100%	100%
Apple Golden 1	160	100%	100%
Apple Golden 2	164	100%	100%
Apple Golden 3	161	98%	90%
Apple Granny Smith	164	100%	98%
Apple Red 2	0	0%	0%
Apple Red 3	0	0%	0%
Banana	0	0%	0%
Banana Lady Finger	152	100%	100%
Banana Red	166	100%	98%

Готова модель

The screenshot shows the ModelHub application interface. At the top, it displays the model name 'FruitClassifier 1' with a file icon, and download options for 'Get', 'Xcode', and 'Share'. Below this, it shows the model type as 'Image Classifier', size as '247 KB', document type as 'Core ML Model', and availability as 'macOS 10.14+ | iOS 12.0+ | tvOS 12.0+'. There are two tabs at the bottom: 'General' (which is selected) and 'Predictions'. The 'General' tab contains sections for 'Metadata' and 'Class Labels'. The 'Metadata' section includes 'Description' (with the note '--') and 'Author' (Daria Shevchenko). The 'Class Labels' section lists 16 labels: Apple Braeburn, Apple Crimson Snow, and Apple Golden 1.

Що make Vision Framework?

Computer Vision - технологія, яка дозволяє створювати програми, де комп'ютери «бачають» і розуміють зміст цифрових зображень і відео

Vision Framework (частина iOS) використовують для:

- Виявлення орієнтирів
- Виявлення тексту
- Розпізнавання штрих-коду
- Фіксування зображення

Використання Vision Classifier в проекті

```
import CoreML
import Vision
import UIKit

class VisionClassifier {
    private let model: VNCoreMLModel
    private var completion: (String) -> Void = { _ in }

    private lazy var request: [VNCoreMLRequest] = {
        let request = VNCoreMLRequest(model: model) { (request, error) in
            guard let results = request.results as? [VNClassificationObservation] else {
                return
            }

            if !results.isEmpty {
                if let result = results.first {
                    self.completion(result.identifier)
                }
            }
        }
        request.imageCropAndScaleOption = .centerCrop
        return [request]
    }()

    init?(mlModel: MLModel) {
        if let model = try? VNCoreMLModel(for: mlModel) {
            self.model = model
        } else {
            return nil
        }
    }
}
```

Функція для класифікації зображень

```
func classify(_ image: UIImage, completion: @escaping (String) -> Void) {
    self.completion = completion

    DispatchQueue.global().async {
        guard let cgImage = image.cgImage else {
            return
        }

        let handler = VNIImageRequestHandler(cgImage: cgImage, options: [:])

        do {
            try handler.perform(self.requests)
        } catch {
            print(error.localizedDescription)
        }
    }
}
```

4. Sound Classification y CoreML

Приклад використання моделі для класифікації звуків

```
import Foundation
import AVFoundation
import SoundAnalysis

class AudioClassifier: NSObject, SNResultsObserving {

    private let model: MLModel
    private let request: SNClassifySoundRequest
    private var results: [(String,Double)] = []
    private var completion: (String?) -> () = { _ in }

    init?(model: MLModel) {

        guard let request = try? SNClassifySoundRequest(mlModel: model) else {
            return nil
        }

        self.model = model
        self.request = request
    }

    func request(_ request: SNRequest, didProduce result: SNResult) {

        guard let results = result as? SNClassificationResult,
              let result = results.classifications.first
        else { return }

        if result.confidence > 0.8 {
            // add the result to an array
            self.results.append((result.identifier, result.confidence))
        }
    }

    func requestDidComplete(_ request: SNRequest) {

        self.results.sort {
            return $0.1 > $1.1
        }

        guard let result = self.results.first else { return }

        self.completion(result.0)
    }
}
```

```
func classify(audioFile: URL, completion: @escaping (String?) -> Void) {

    self.completion = completion

    guard let analyzer = try? SNAudioFileAnalyzer(url: audioFile),
          let _ = try? analyzer.add(self.request, withObserver: self) else {
        return
    }

    analyzer.analyze()

}
```

5. Що таке Face Detection?

Face Detection - це технологія, що дозволяє комп'ютеру розпізнавати обличчя людей на фото або відео.

Що важливо при створенні моделі?

- Якість зображень
- Кількість зображень
- Різноманітність зображень

Де використовується така технологія?

- Відстеження обличчя - дозволяє виявити обличчя в кадрі відео

Може використовуватися рітейлерами для підрахунку кількості відвідувачів і відстеження їх руху. На основі результату вони можуть налаштовувати свої дисплеї

Можна використовувати, щоб переконатися, що користувач звертає увагу. Запобігання ДТП через сонних водіїв

- Аналіз обличчя - дозволяє виявити вираз обличчя

Визначення статі людини

Визначення емоційного стану людини (щасливий, сумний, злий)

Програми для камери можуть вибрати найкраще зображення з кількох зображень

- Розпізнавання облич - система, яка ідентифікує або верифікує людину за цифровим зображенням або відео

Позначення фотографії

Перевірка в аеропортах або пристрой для розблокування

Камери дверного дзвінка

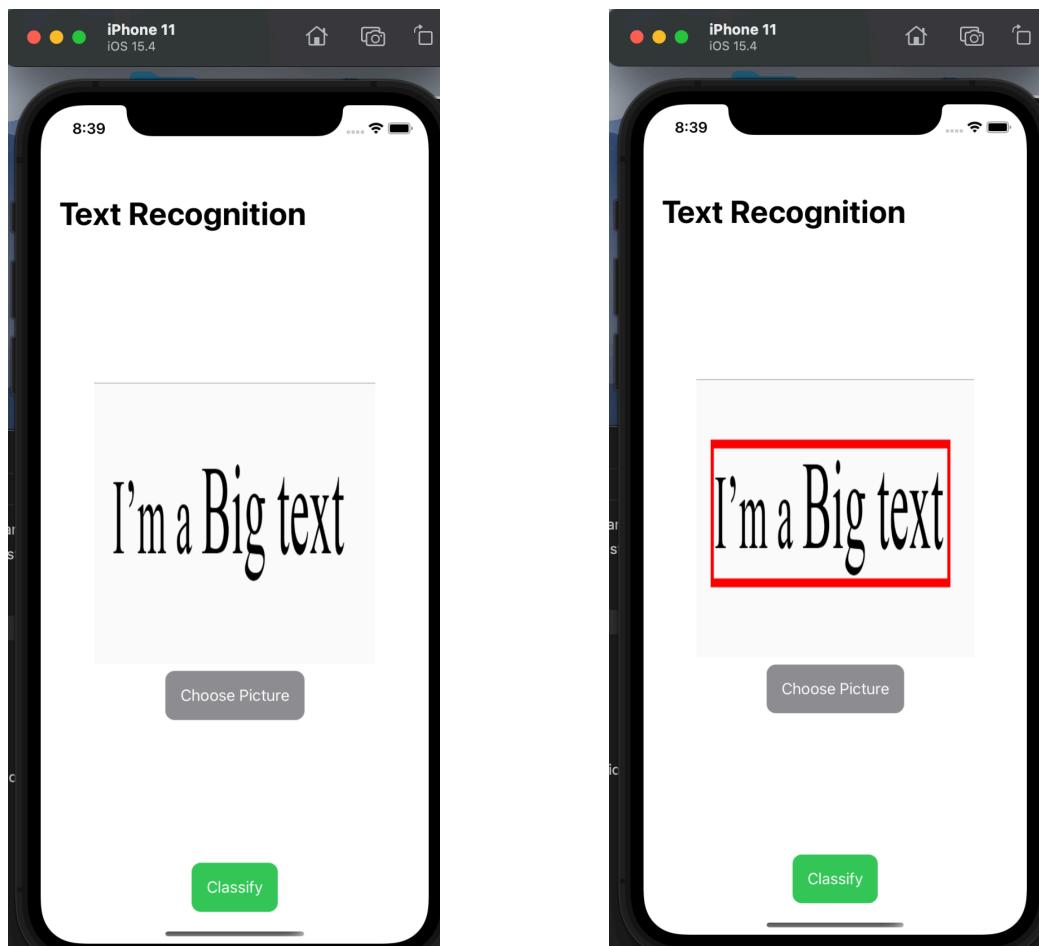
Система соціального кредиту

6. Text and Barcode Detection

Іплементування моделі в програму, задля додавання функціоналу розпізнання тексту:

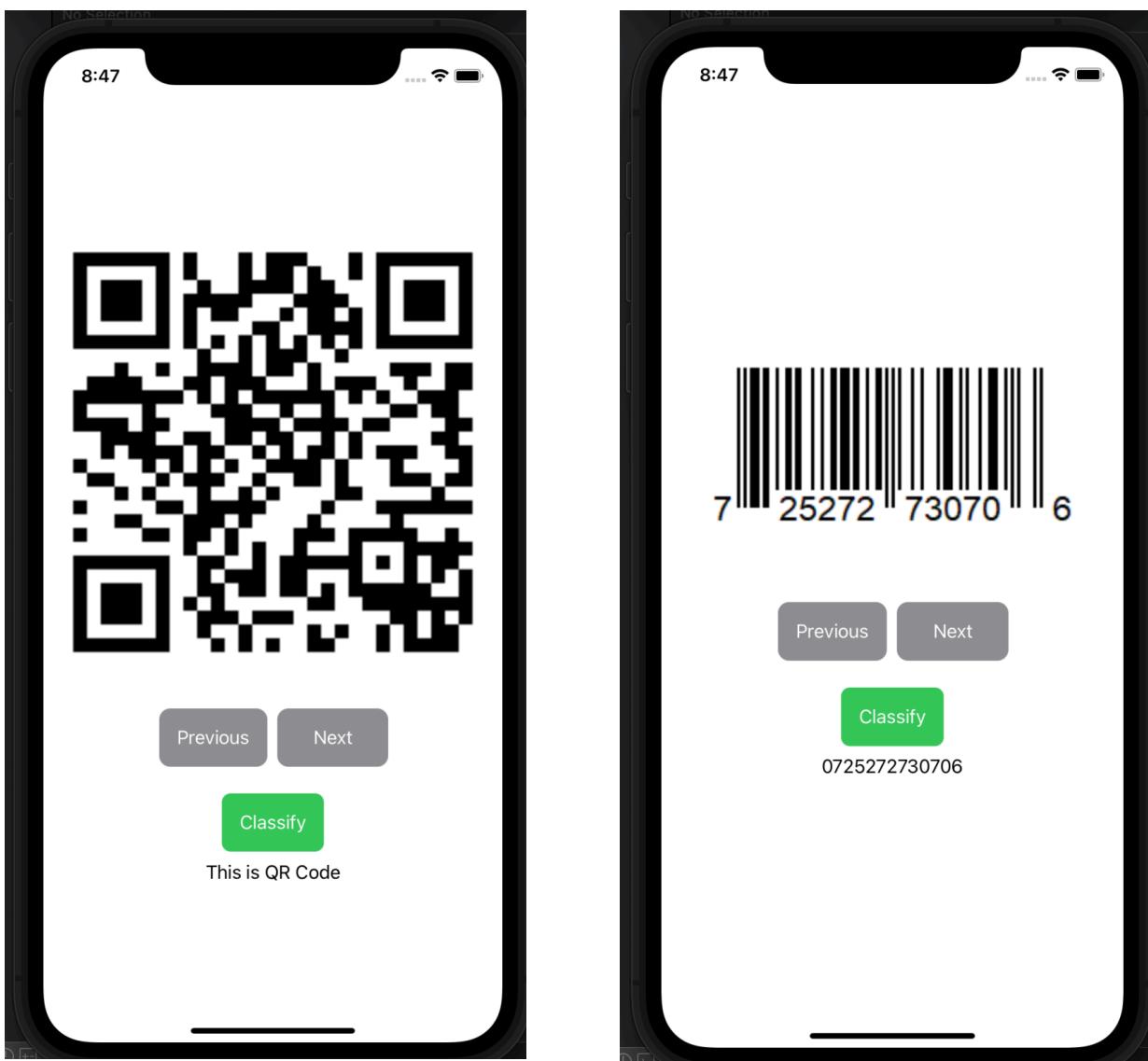
```
private func performTextClassification() {  
  
    guard let img = image,  
        let cgImage = img.cgImage,  
        let orientation = self.sourceType == .camera ? CGImagePropertyOrientation.right : CGImagePropertyOrientation(rawValue:  
            UInt32(img.imageOrientation.rawValue))  
    else {  
        return  
    }  
  
    let request = VNRecognizeTextRequest { (request, error) in  
  
        if let observations = request.results as? [VNRecognizedTextObservation] {  
  
            DispatchQueue.global().async {  
                if let result = img.drawOnImage(observations: observations) {  
                    DispatchQueue.main.async {  
                        self.image = result  
                    }  
                }  
            }  
        }  
    }  
  
    let handler = VNImageRequestHandler(cgImage: cgImage, orientation: orientation, options: [:])  
  
    do {  
        try handler.perform([request])  
    } catch {  
        print(error.localizedDescription)  
    }  
}
```

Фінальний додаток



Код виявлення штрих-коду та впровадження його в проект

```
private func performBarcodeDetection(completion: @escaping ([VNBarcodeObservation]?) -> Void) {  
  
    guard let image = UIImage(named: photos[currentIndex]),  
          let orientation = CGImagePropertyOrientation(rawValue:  
              UInt32(image.imageOrientation.rawValue)),  
          let cgImage = image.cgImage else {  
            return completion(nil)  
    }  
  
    let request = VNImageRequestHandler(cgImage: cgImage, orientation: orientation, options: [:])  
  
    do {  
        try handler.perform([request])  
    } catch {  
        print(error.localizedDescription)  
    }  
  
}
```



7. Image Classification з допомогою TuriCreate

Використання python для завантаження зображень

```
1 import os
2 import requests
3
4 def make_directories(directory_name):
5     if not os.path.exists(directory_name):
6         os.makedirs(directory_name)
7         print('Directory has been created!')
8     else:
9         print('Directory already exists!')
10
11 def download_and_save_images(url, directory_name):
12     list = requests.get(url).text.split('\r\n')[:50]
13
14     for url in list:
15         filename = os.path.basename(url)
16         print(filename)
17         try:
18             with open(f"{directory_name}/{filename}", "wb") as file_object:
19                 file_object.write(requests.get(url, timeout=1).content)
20         except:
21             print(f"Error downloading {filename}")
22             continue
23
24 cat_url = "http://image-net.org/api/text/imagenet.synset.geturls?wnid=n02123045"
25 dog_url = "http://image-net.org/api/text/imagenet.synset.geturls?wnid=n02111277"
26
27 current_directory = os.getcwd()
28 cats_directory = current_directory + "/train/cats"
29 dogs_directory = current_directory + "/train/dogs"
30
31 make_directories(cats_directory)
32 make_directories(dogs_directory)
33
34 download_and_save_images(cat_url, cats_directory)
35 download_and_save_images(dog_url, dogs_directory)
36
```

Python код для створення моделі за допомогою turicreate

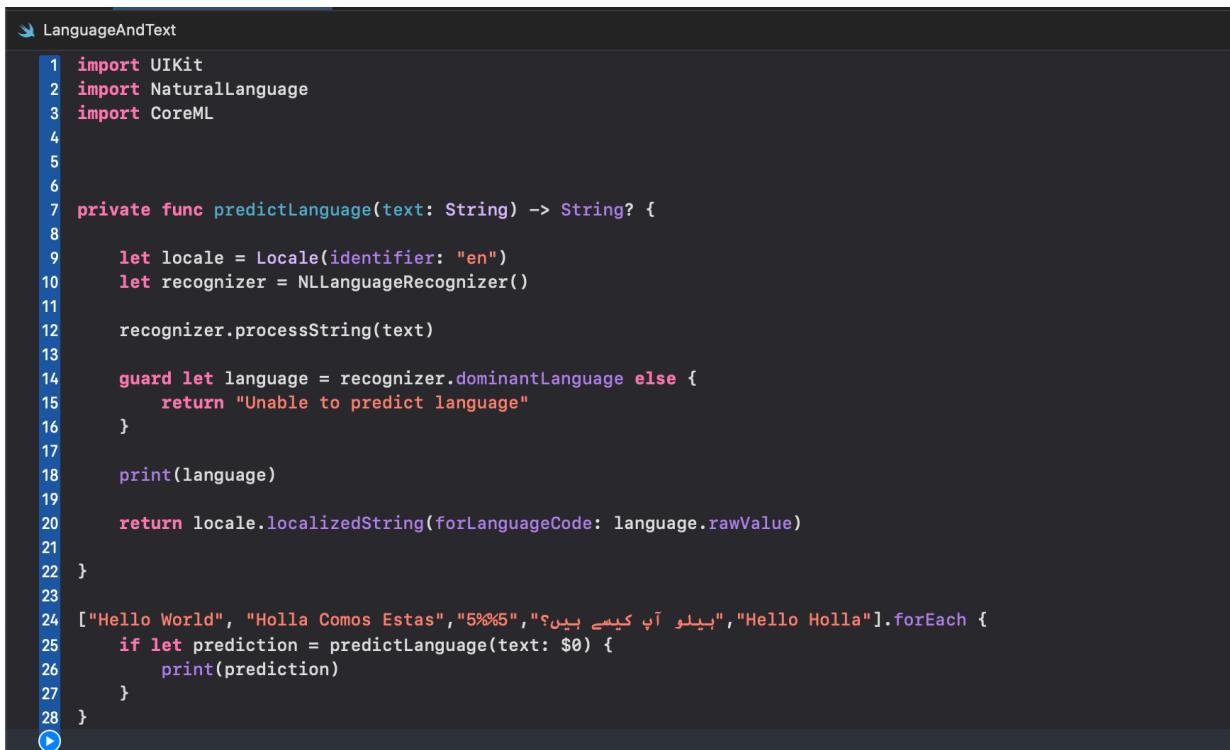
```
1 import turicreate as tc
2
3 data = tc.image_analysis.load_images("train", with_path=True)
4
5 data['label'] = data['path'].apply(lambda path: 'dog' if '/dogs' in path else 'cat')
6
7 (train_data, test_data) = data.random_split(0.8)
8
9 model = tc.image_classifier.create(data, target='label')
10
11 predictions = model.predict(data)
12
13 model.export_coreml('CatDogClassifier.mlmodel')
14
15
```

8. Текст та Natural Language

Як його можна використовувати?

- Ідентифікація мови – визначення мови тексту
- Розпізнавання іменованих об'єктів – визначення елементів тексту, таких як люди, місця та організації
- Аналіз настроїв - визначення позитивних чи негативних настроїв у тексті
- Користувальські текстові класифікатори

Код для визначення мови фраз



```
LanguageAndText
1 import UIKit
2 import NaturalLanguage
3 import CoreML
4
5
6
7 private func predictLanguage(text: String) -> String? {
8
9     let locale = Locale(identifier: "en")
10    let recognizer = NLLanguageRecognizer()
11
12    recognizer.processString(text)
13
14    guard let language = recognizer.dominantLanguage else {
15        return "Unable to predict language"
16    }
17
18    print(language)
19
20    return locale.localizedString(forLanguageCode: language.rawValue)
21
22 }
23
24 ["Hello World", "Holla Comos Estas", "5%5", "بیلوبیلیو آپ کیسے بین؟", "Hello Holla"].forEach {
25     if let prediction = predictLanguage(text: $0) {
26         print(prediction)
27     }
28 }
```

Код для з'ясування особових імен, топонімів, організаційних назв



```
private func printNamedEntities(text: String) {
    let tagger = NSLinguisticTagger(tagSchemes: [.nameType], options: 0)
    tagger.string = text

    let range = NSRange(location: 0, length: text.count)

    let options: NSLinguisticTagger.Options = [.omitPunctuation, .omitWhitespace, .joinNames]

    let tags: [NSLinguisticTag] = [.personalName, .placeName, .organizationName]

    tagger.enumerateTags(in: range, unit: .word, scheme: .nameType, options: options) { tag, tokenRange, stop in
        if let tag = tag, tags.contains(tag) {
            let name = (text as NSString).substring(with: tokenRange)
            print("\(name) is a \(tag.rawValue)")
        }
    }
}

["My name is John and I love to visit Australia", "My name is Mary and I work for Google", "Yesterday, I went to Costa Rica and also met with James who works at Exxon", "Holla Comos Estas"].forEach { text in
    printNamedEntities(text: text)
}
```

9. Sentiment Analysis

Ми використовуємо sentiment analysis для виявлення емоцій тексту

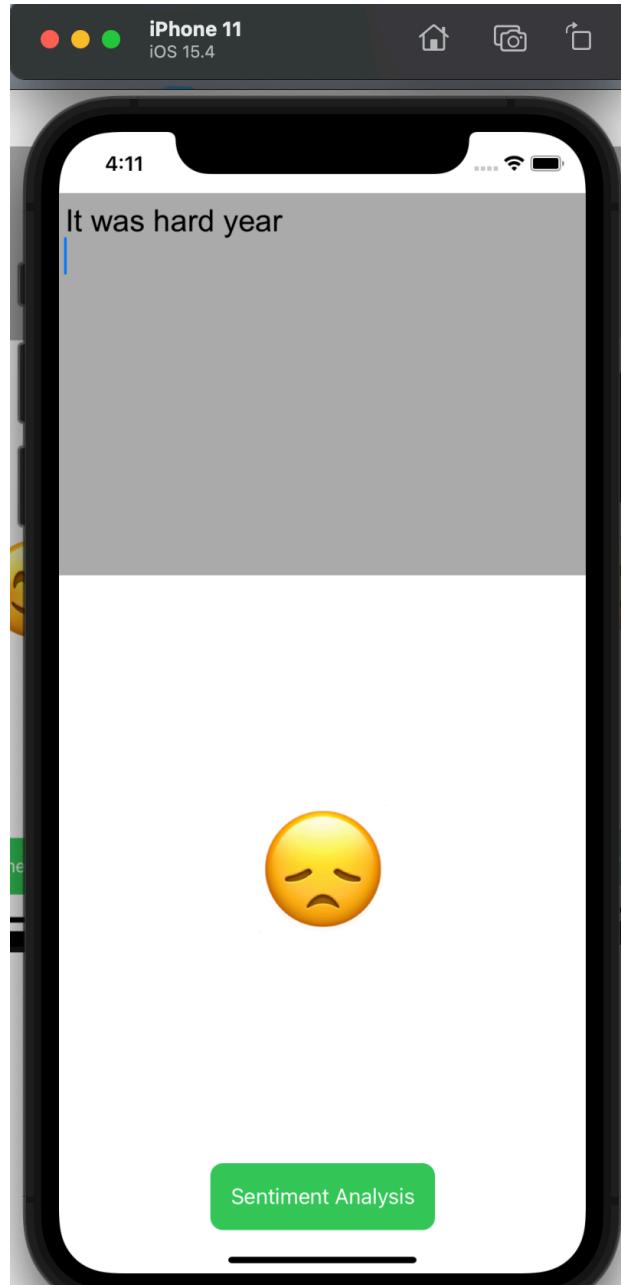
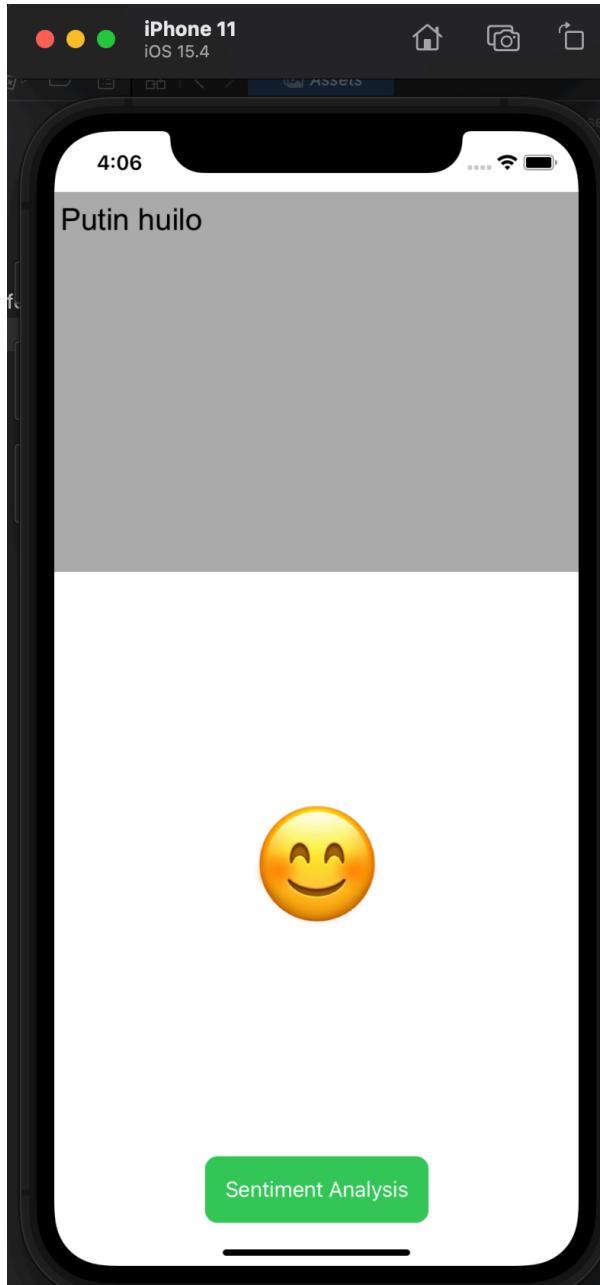
Виконання аналізу даних за допомогою pandas та робота з csv файлами

```
1 import pandas as pd
2
3 df = pd.read_csv('~/Desktop/epinions3.csv')
4
5 def_neg = df[df['class'] == 'Neg']
6 def_pos = df[df['class'] == 'Pos']
7
8 for row in range(def_neg.shape[0]):
9     def_neg.iloc[row].to_csv(f'~/Desktop/train/negative/negative{row}.csv', index = False)
10
11 for row in range(def_pos.shape[0]):
12     def_pos.iloc[row].to_csv(f'~/Desktop/train/positive/positive{row}.csv', index = False)
13
```

Створення моделі за допомогою playground

```
1 import Cocoa
2 import CreateML
3 import Foundation
4
5 let filePath = "/Users/azamsharp/Desktop/epinions3.csv"
6 let outputFileName = "/Users/azamsharp/Desktop/SentimentAnalysisClassifierModel.mlmodel"
7
8 let dataURL = URL(fileURLWithPath: filePath)
9 let data = try MLDataTable(contentsOf: dataURL)
10
11 let (trainingData,testingData) = data.randomSplit(by: 0.8)
12
13 let sentimentClassifier = try MLTextClassifier(trainingData: trainingData, textColumn: "text", labelColumn: "class")
14
15 let metadata = MLModelMetadata(author: "Shevchenko Daria", shortDescription: "Sentiment Analysis Model", version: "1.0")
16
17 try sentimentClassifier.write(to: URL(fileURLWithPath: outputFileName), metadata: metadata)
18
19
```

Імплементування моделі та перевірка роботи аналізу “настрою” тексту в додатку



10. Drawing Recognition

Набір даних, який ми можемо використовувати для програм для малювання – quick draw dataset

Імплементування моделі в проект

```
@IBAction func classify() {  
  
    let image = canvas.uiImage()  
    guard let resizedImage = image.resizeTo(size: CGSize(width: 299, height: 299)),  
          let buffer = resizedImage.toBuffer()  
    else {  
        return  
    }  
  
    let output = try? model.prediction(image: buffer)  
  
    if let output = output {  
        self.title = output.classLabel  
    }  
  
}
```

11. Activity Recognition за допомогою Core Motion Framework

Ми використовуємо CMMotionActivityManager для відслідковування активності

```
private let tracker = CMMotionActivityManager()  
@State private var activityName: String = "Not Tracking"
```

```
private func startTracking(handler: @escaping (CMMotionActivity?) -> Void)  
throws {  
  
    if !CMMotionActivityManager.isActivityAvailable() {  
        throw TrackingError.notAvailable  
    } else if CMMotionActivityManager.authorizationStatus() == .denied {  
        throw TrackingError.notAuthorized  
    }  
  
    self.tracker.startActivityUpdates(to: .main, withHandler: handler)  
}
```

```

import CoreMotion

extension CMMotionActivity {

    var name: String {
        if walking { return "Walking" }
        if running { return "Running" }
        if cycling { return "Cycling" }
        if automotive { return "Driving" }
        if stationary { return "Stationary" }
        return "Unknown"
    }
}

```

12. Style Transfer за допомогою Turi Create

Тренування та створення моделі

```

1 import turicreate as tc
2
3 tc.config.set_num_gpus(0)
4
5 styles = tc.load_images('style/')
6 content = tc.load_images('content/')
7
8 model = tc.style_transfer.create(styles, content, max_iterations=6000)
9
10 test_images = tc.load_images('test/')
11
12 stylized_images = model.stylize(test_images, max_size=1024)
13
14 stylized_images.explore()
15
16 model.export_coreml('MyCustomStyleTransfer.mlmodel')
17
18

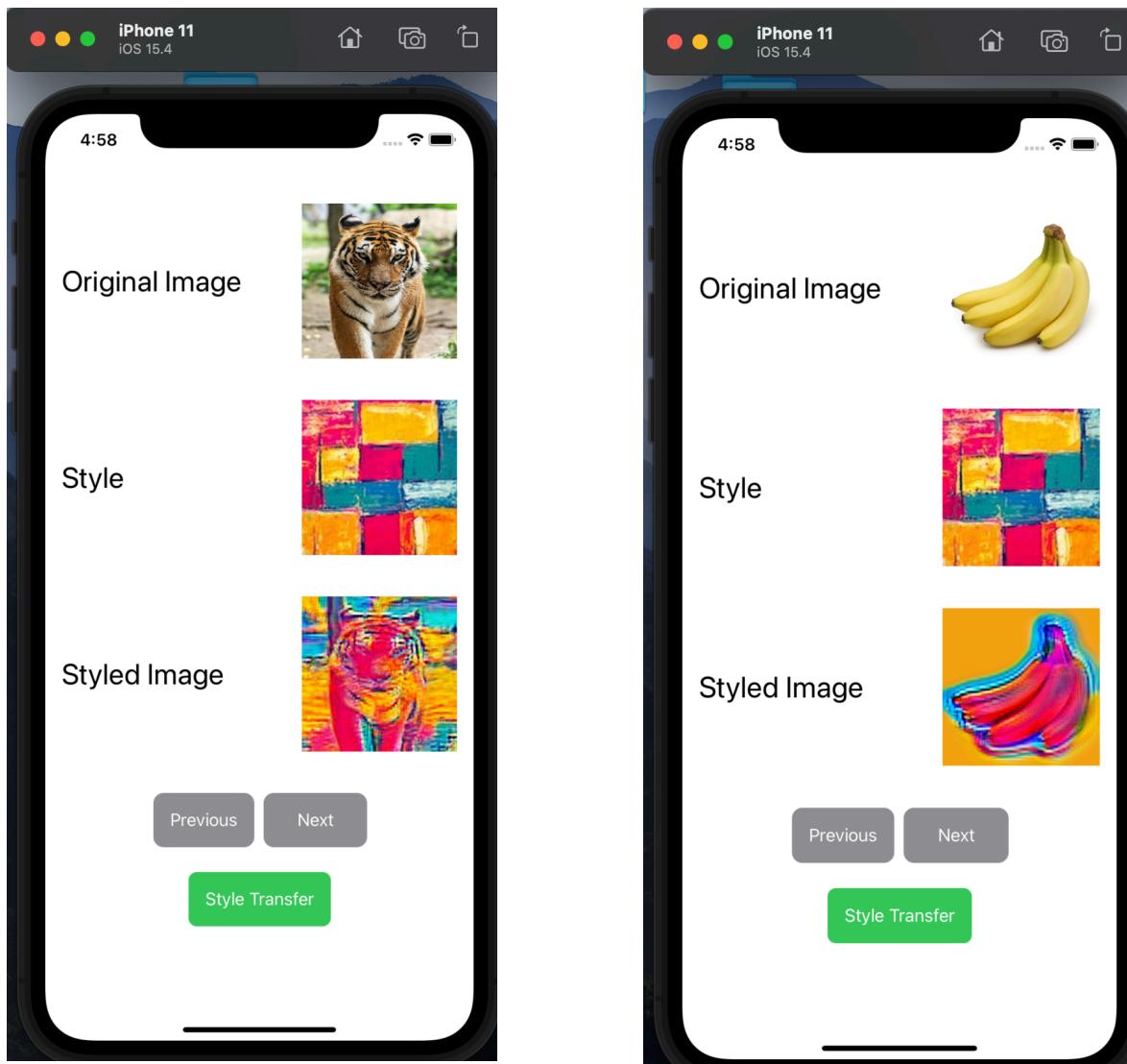
```

Використання моделі в проєкті для стилізації зображень

```

24     private func performStyleTransfer() {
25
26         let currentImageName = photos[currentIndex]
27
28         guard let img = UIImage(named: currentImageName),
29             let resizedImage = img.resizeTo(size: CGSize(width: 256, height: 256)),
30             let buffer = resizedImage.toBuffer() else {
31             return
32         }
33
34         guard let styleArray = try? MLMultiArray(shape: [1] as [NSNumber], dataType: .double) else {
35             return
36         }
37
38         styleArray[0] = 1.0
39
40         let output = try? model.prediction(image: buffer, index: styleArray)
41
42         if let output = output {
43             let stylizedBuffer = output.stylizedImage
44             self.styledImage = UIImage.imageFromCVPixelBuffer(pixelBuffer: stylizedBuffer)
45         }
46
47     }
48

```



13. Sentence Generation Using GPT2 Model

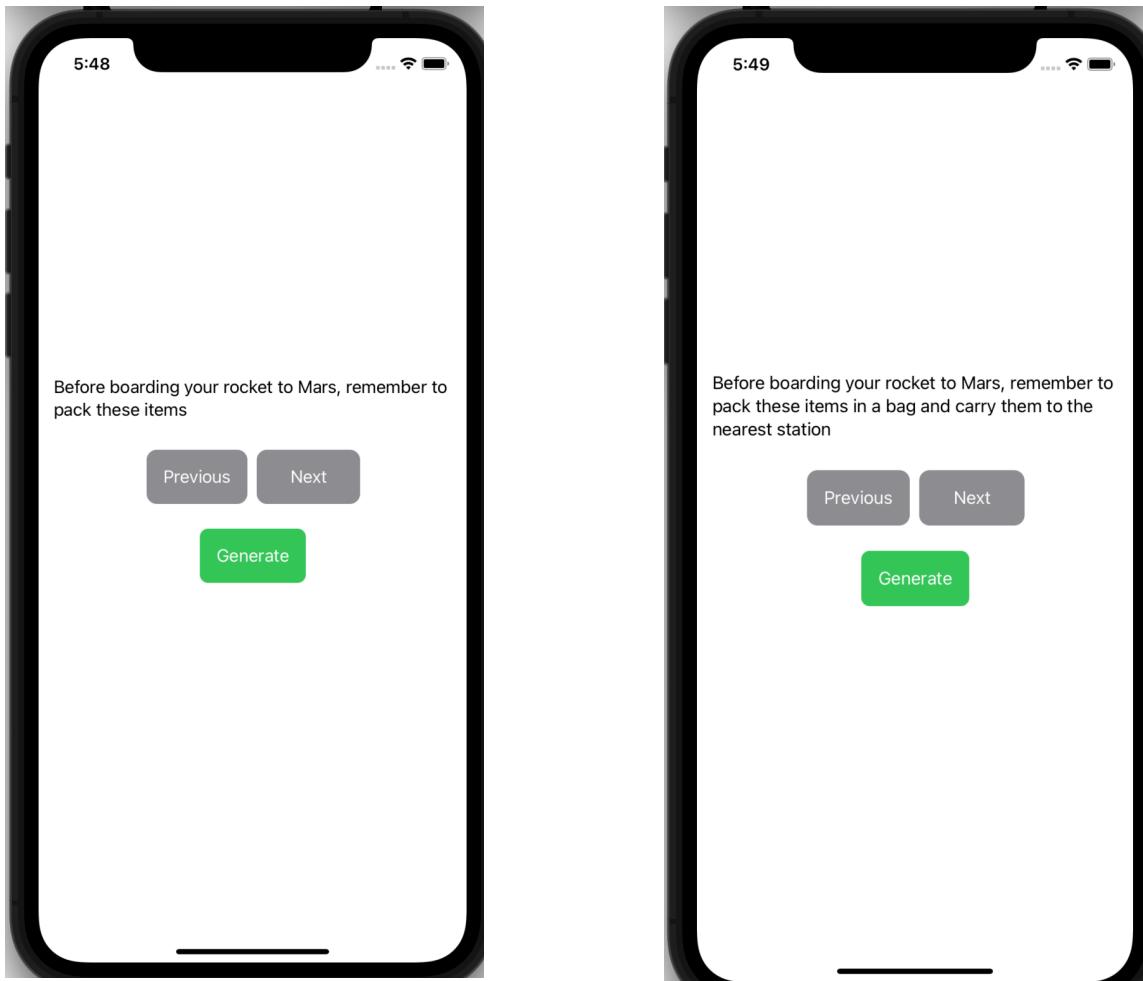
Ми використовуємо *GPT2 model*

<https://github.com/huggingface/swift-coreml-transformers>

Імлементування моделі у проект для генерації продовження тексту

```
Button("Generate") {
    // generate the text
    let text = self.prompts[self.currentIndex]

    DispatchQueue.global(qos: .userInitiated).async {
        _ = self.model.generate(text: text) { completion, _ in
            DispatchQueue.main.async {
                self.predictionLabel = completion
            }
        }
    }
}
```



14. Questions And Answers Using BERT

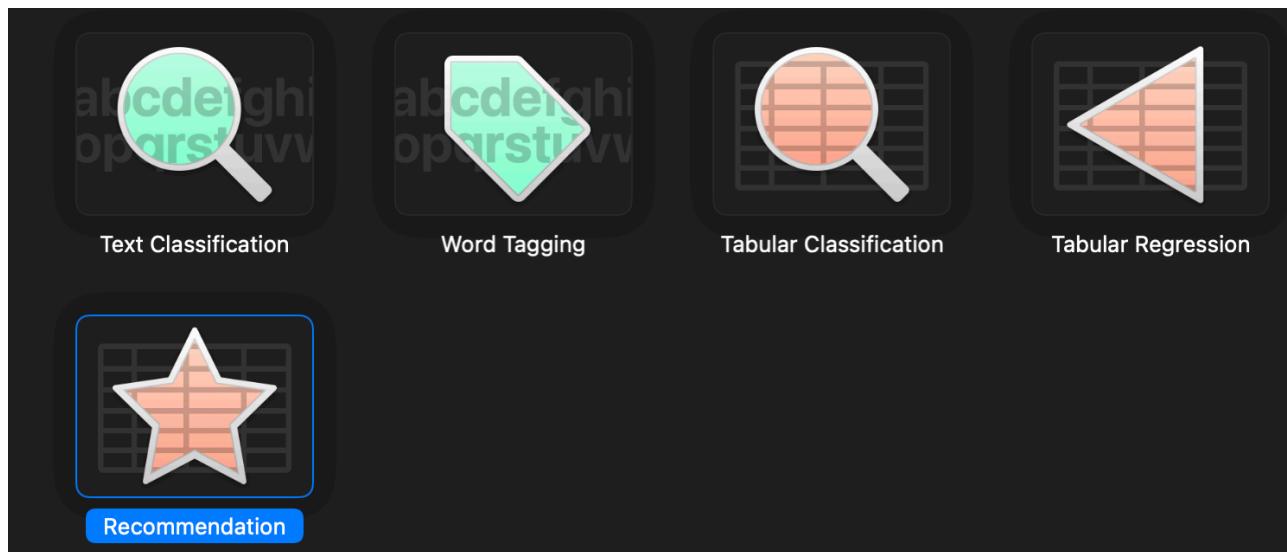
Bert - фреймворк, що дозволяє розпізнавати текст та знаходити відповідь на питання у залежності від контексту наданого матеріалу

```
let model = BERT()
```

```
Button("Answer") {  
    DispatchQueue.global(qos: .userInitiated).async {  
        let answer = self.model.findAnswer(for: self.question, in: self.context)  
        self.answerLabel = String(answer)  
    }  
}
```

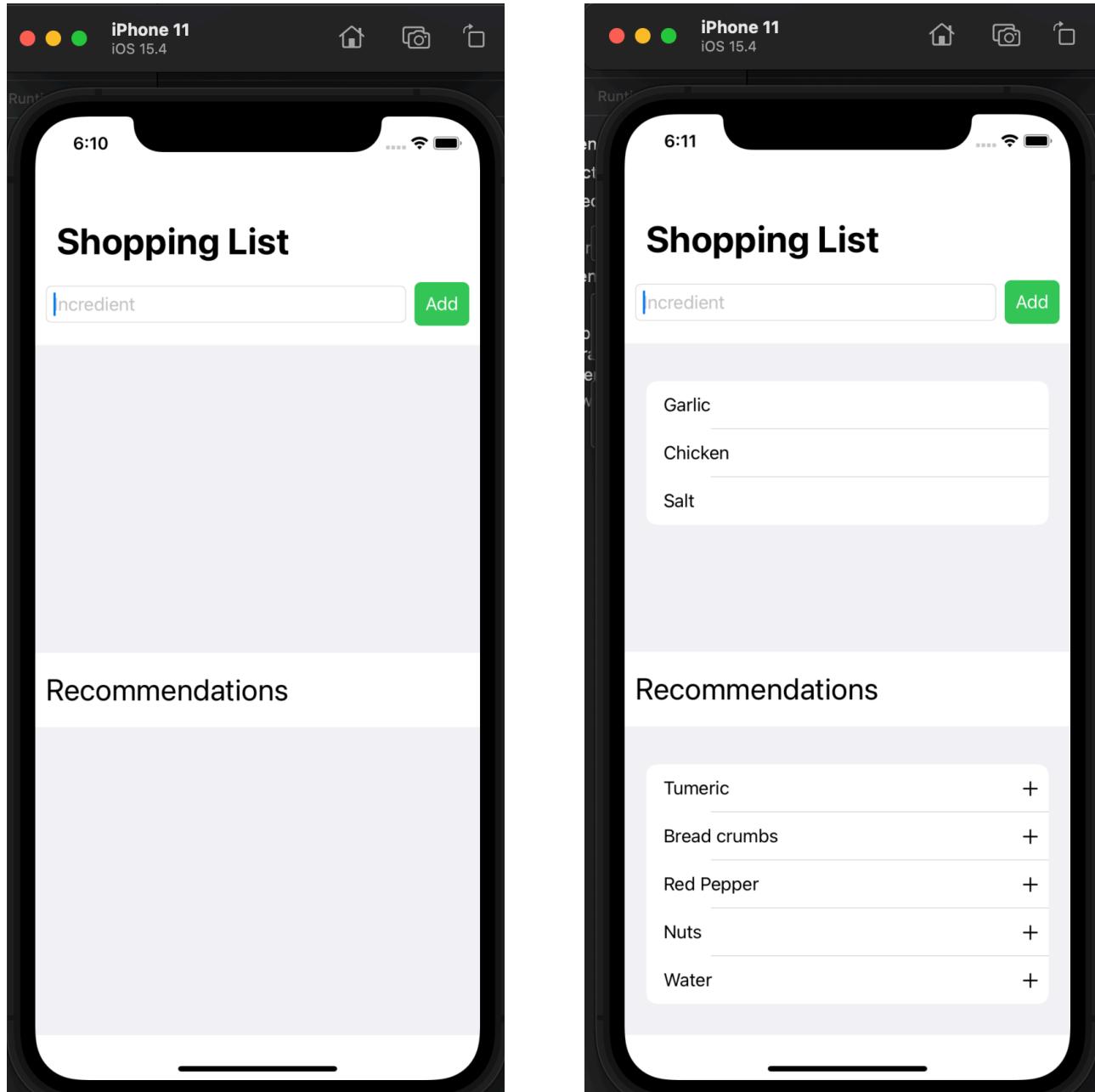
15. Recommending Ingredients

Ми можемо створити модель за допомогою “Recommendation” в CreateML та таблиці з даними



Імплементування моделі у проект, де формується список рекомендацій до покупок у залежності від вже обраних продуктів

```
private func performRecommendations() {  
  
    let items = Dictionary(uniqueKeysWithValues: self.ingredients.map { ($0, 5.0) })  
  
    if let output = try? model.prediction(items: items, k: 5, restrict_: [], exclude:  
        self.ingredients) {  
  
        self.recommendations = output.recommendations  
    }  
  
}
```



16. Прогнозування ціни за допомогою регресії

Що таке регресія?

Регресія – це метод моделювання цільового значення на основі незалежних предикторів

https://developer.apple.com/documentation/createlml/creating_a_model_from_tabular_data

Тренування набору даних у csv форматі

```
import Cocoa
import CoreML
import CreateML

let outputFileName = "/Users/azamsharp/Desktop/MartianPricesModel.mlmodel"

guard let csvFile = Bundle.main.url(forResource: "Mars", withExtension: "csv"),
    let dataTable = try? MLDataTable(contentsOf: csvFile) else {
    fatalError("Unable to load data!")
}

let regressorColumns = ["price", "solarPanels", "greenhouses", "size"]
let regressorTable = dataTable[regressorColumns]

let (regressorTestingTable, regressorTrainingTable) = regressorTable.randomSplit(by: 0.2)

let regressor = try MLLinearRegressor(trainingData: regressorTrainingTable, targetColumn: "price")

let metadata = MLModelMetadata(author: "Shevchenko Daria", shortDescription: "Prediction of Martian Houses", version: "1.0")

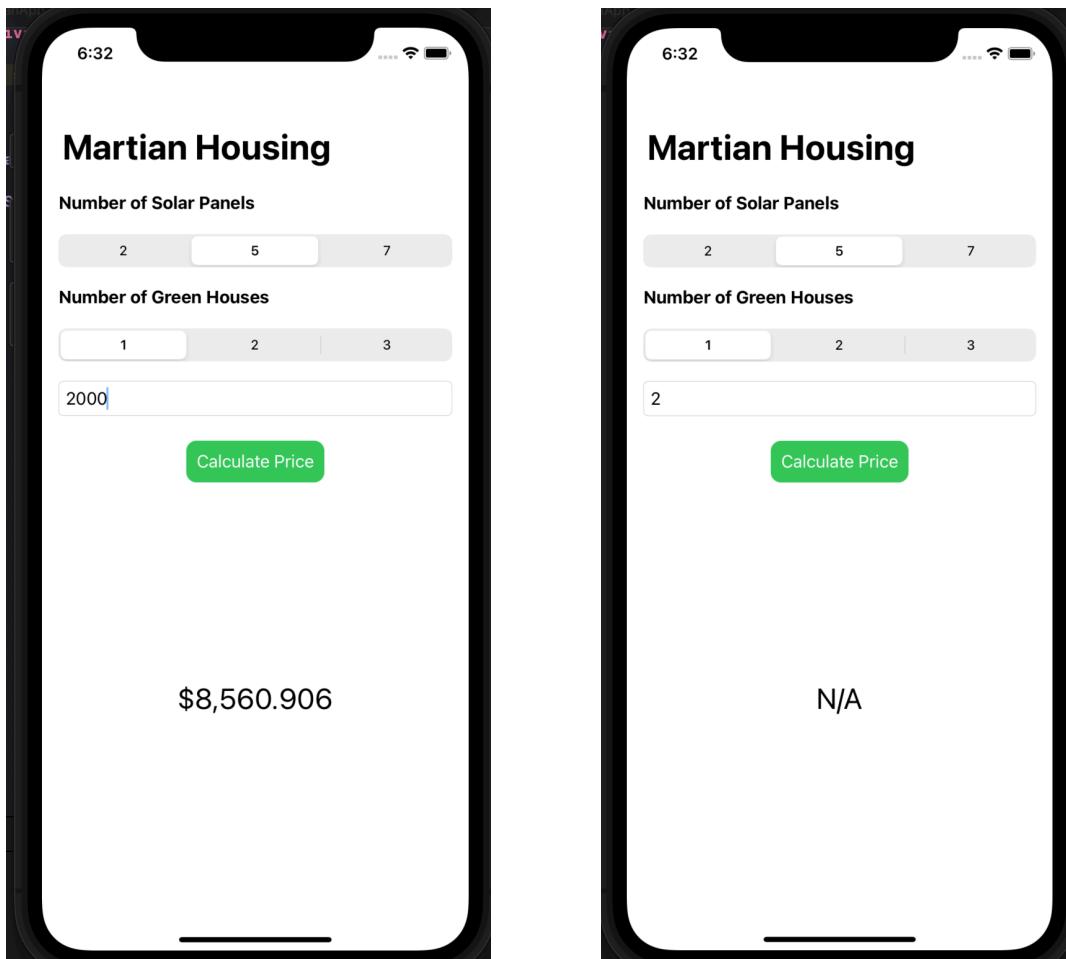
try regressor.write(toFile: outputFileName, metadata: metadata)
```

Використання моделі

```
let output = try? self.model.prediction(solarPanels: self.numberOfSolarPanels, greenhouses: self.numberOfGreenHouses, size: Double(self.size) ?? 200)

if let output = output {
    self.predictedPrice = output.price < 0 ? "N/A" : "$\(output.price.withCommas())"
}
```

Додаток, що використовує регресійну модель для того, щоб визначати ціну будинку (n/a - не існує такої ціни)



17. Augmented Reality With AI

Що таке Augmented Reality?

Augmented Reality є технологією, що дозволяє поєднувати реальний та віртуальний світ Framework, які використовують для створення додатків під iOS - ARKit, Unity

Використання UIKit у програмі для класифікації зображень на фотокамері

```
func view(_ view: ARSKView, nodeFor anchor: ARAnchor) -> SKNode? {

    let labelNode = SKLabelNode(text: self.classification)
    labelNode.horizontalAlignmentMode = .center
    labelNode.verticalAlignmentMode = .center
    return labelNode;
}

override func touchesBegan(_ touches: Set<UITouch>, with event: UIEvent?) {

    self.currentFrame = sceneView.session.currentFrame
    classifyImage()
}

private func classifyImage() {

    guard let orientation = CGImagePropertyOrientation(rawValue: UInt32(UIDevice.current.orientation.rawValue)) else {
        return
    }

    let handler = VNImageRequestHandler(cvPixelBuffer: self.currentFrame.capturedImage, orientation: orientation, options: [:])

    DispatchQueue.global().async {

        do {
            try handler.perform([self.classificationRequest])
        } catch {
            print(error.localizedDescription)
        }
    }
}
```

```
private func processClassifications(_ classifications: [VNClassificationObservation]) {

    if let observation = classifications.first {

        self.classification = observation.identifier

        // Create a transform with a translation of 0.2 meters in front of the camera
        var translation = matrix_identity_float4x4
        translation.columns.3.z = -0.2
        let transform = simd_mul(self.currentFrame.camera.transform, translation)

        // Add a new anchor to the session
        let anchor = ARAnchor(transform: transform)
        sceneView.session.add(anchor: anchor)

    }
}
```

```
lazy var classificationRequest: VNCoreMLRequest = {
    do {
        let model = try VNCoreMLModel(for: MobileNetV2().model)           △ 'init()' is deprecated: Use init(configuration:) instead and handle
        let request = VNCoreMLRequest(model: model) { request, error in
            // process classifications
            guard let classifications = request.results as? [VNClassificationObservation], error == nil else {
                return
            }

            self.processClassifications(classifications)
        }

        request.imageCropAndScaleOption = .centerCrop
        return request
    } catch {
        print(error.localizedDescription)
        fatalError("Unable to initialize request for Machine Learning Model")
    }
}()
```

18. Висновки

Машинне навчання є дуже зручним та гнучким інструментом для створення багатьох програм. Фреймфорк CreateML дозволяє створювати моделі на основі багатьох типів даних: аудіо, текстів та фото, та зручно імплементувати їх у проекти. ARKit допомагає у створенні додатків з доповненою реальністю, що в поєднанні з ML робить його дуже корисним інструментом для розширення функціоналу застосунків.