

## 1. Projektstruktur erstellen

Zuerst erstellen wir eine saubere Projektstruktur:

```
mkdir -p Astra/ML && mkdir -p Astra/Services && mkdir -p Astra/Models && touch Astra/README.md
PS C:\Astra_curser> mkdir -p Astra/ML && mkdir -p Astra/Services && mkdir -p Astra/Models && touch Astr
In Zeile:1 Zeichen:19
+ mkdir -p Astra/ML && mkdir -p Astra/Services && mkdir -p Astra/Models ...
+ ~~~~~
Das Token "&&" ist in dieser Version kein gültiges Anweisungstrennzeichen.
In Zeile:1 Zeichen:46
+ mkdir -p Astra/ML && mkdir -p Astra/Services && mkdir -p Astra/Models ...
+ ~~~~~
Das Token "&&" ist in dieser Version kein gültiges Anweisungstrennzeichen.
In Zeile:1 Zeichen:71
+ ... tra/ML && mkdir -p Astra/Services && mkdir -p Astra/Models && touch A ...
+ ~~~~~
Das Token "&&" ist in dieser Version kein gültiges Anweisungstrennzeichen.
+ CategoryInfo          : ParserError: (:) [], ParentContainsErrorRecordException
+ FullyQualifiedErrorId : InvalidEndOfLine
PS C:\Astra_curser>
```

Entschuldigung, lassen Sie uns die Befehle für Windows anpassen:

```
mkdir Astra\ML; mkdir Astra\Services; mkdir Astra\Models; New-Item Astra\README.md -ItemType File
PS C:\Astra_curser> mkdir Astra\ML; mkdir Astra\Services; mkdir Astra\Models; New-Item Astra\README.md
PS C:\Astra_curser> mkdir Astra\ML; mkdir Astra\Services; mkdir Astra\Models; New-Item Astra\README.md
PS C:\Astra_curser> mkdir Astra\ML; mkdir Astra\Services; mkdir Astra\Models; New-Item Astra\README.md
-ItemType File
mkdir : Ein Element mit dem angegebenen Namen "C:\Astra_curser\Astra\ML" ist bereits vorhanden.
In Zeile:1 Zeichen:1
+ mkdir Astra\ML; mkdir Astra\Services; mkdir Astra\Models; New-Item As ...
+ ~~~~~
+ CategoryInfo          : ResourceExists: (C:\Astra_curser\Astra\ML:String) [New-Item], IOExcepti
on
+ FullyQualifiedErrorId : DirectoryExist,Microsoft.PowerShell.Commands.NewItemCommand
+ CategoryInfo          : ResourceExists: (C:\Astra_curser\Astra\Models:String) [New-Item], IOExc
eption
+ FullyQualifiedErrorId : DirectoryExist,Microsoft.PowerShell.Commands.NewItemCommand
Verzeichnis: C:\Astra_curser\Astra
Mode                LastWriteTime         Length Name
----                -
-a-----         22.03.2025   14:09              0 README.md
PS C:\Astra_curser>
```

Nun erstellen wir eine README.md mit wichtigen Informationen:

```
README.md+69-1
# Astra - Skateboard Trick Recognition App
Eine iOS-App zur Erkennung von Skateboard-Tricks mit Machine Learning Unterstützung.
## Projektstruktur
...
Astra/
├─ Astra.xcodeproj/      # Xcode Projektdateien
```

```

├── Views/           # SwiftUI Views
├── Models/          # Swift Datenmodelle
├── Services/        # Dienste und Utilities
├── ML/              # Python ML-Modelle und Training
...

```

Jetzt erstellen wir eine requirements.txt für den Python-Teil:

```
requirements.txt(new)+8-1
```

```

tensorflow==2.13.0
scikit-learn==1.3.0
pandas==2.0.3
numpy==1.24.3
matplotlib==3.7.2
seaborn==0.12.2
jupyter==1.0.0
coremltools==7.0

```

## Schritt-für-Schritt Einrichtungsanleitung

### 1. Xcode-Projekt einrichten

1. Öffnen Sie Xcode
1. Gehen Sie zu File > Open
1. Navigieren Sie zum Astra-Ordner und öffnen Sie Astra.xcodeproj
1. In den Projekt-Einstellungen:
  - Wählen Sie Ihr Developer Team
  - Setzen Sie die Bundle ID (z.B. "com.ihername.Astra")
  - Stellen Sie iOS 15.0 als Minimum Deployment Target ein

### 2. Python-Umgebung einrichten (VS Code)

1. Öffnen Sie VS Code
1. Öffnen Sie den Astra/ML Ordner
1. Terminal öffnen (Strg+Shift+) und eingeben: ``powershell  
python -m venv venv\Scripts\activate pip install -r requirements.txt

```

powershell
Apply to README.md

```

```

python -m venv venv
venv\Scripts\activate
pip install -r requirements.txt

```

pleas put that with the how to put everything on to github first, because i gone do that first and then i can do the rest  
Restore checkpoint

Ich werde Ihnen eine detaillierte Anleitung geben, wie Sie das Projekt zuerst auf GitHub einrichten können.

## GitHub Setup Guide

### 1. GitHub Repository erstellen

1. Gehen Sie zu GitHub und loggen Sie sich ein

1. Klicken Sie auf das "+" Symbol oben rechts und wählen Sie "New repository"
1. Füllen Sie die Repository-Details aus:
  - Repository name: Astra
  - Description: "Skateboard Trick Recognition App with ML"
  - Wählen Sie "Public" oder "Private"
  - Aktivieren Sie "Add a README file"
  - Wählen Sie bei "Add .gitignore" die Option "Swift"
  - Wählen Sie bei "Choose a license" die Option "MIT License"
1. Klicken Sie auf "Create repository"

## 2. Git lokal einrichten

Öffnen Sie ein Terminal und führen Sie diese Befehle aus:

```
git --version
```

Pop out terminal

Es scheint, dass Git noch nicht installiert ist. Hier sind die Schritte zur vollständigen Einrichtung:

### 1. Git installieren:

- Laden Sie Git für Windows von [git-scm.com](https://git-scm.com) herunter
- Führen Sie den Installer aus
- Wählen Sie die Standardoptionen während der Installation
- Starten Sie nach der Installation Ihren Computer neu

### 1. Git konfigurieren:

Nach der Installation, öffnen Sie ein neues Terminal und führen Sie diese Befehle aus (ersetzen Sie die Werte in Anführungszeichen):

```
powershell
Apply to README.md
Run
git config --global user.name "Ihr Name"
git config --global user.email "ihre.email@example.com"
```

### 1. Projekt initialisieren:

```
powershell
Apply to README.md
Run
cd C:\Astra_curser
git init
```

### 1. .gitignore erstellen:

Lassen Sie uns eine .gitignore-Datei erstellen:

```
.gitignore(new)+56-1
```

```
# Xcode
.DS_Store
xcuserdata/
*.xcodeproj/*
!*.xcodeproj/project.pbxproj
!*.xcodeproj/xcsheareddata/
!*.xcodeproj/project.xcworkspace/
!*.xcworkspace/contents.xcworkspacedata
**/xcsheareddata/WorkspaceSettings.xcsettings
# Swift
.build/
```

```
DerivedData/  
*.ipa
```

## 1. Code zum Repository hinzufügen:

Nach der Git-Installation führen Sie diese Befehle aus:

```
powershell  
Apply to README.md  
Run
```

```
git add .  
git commit -m "Initial commit"
```

## 1. Mit GitHub verbinden:

- Gehen Sie zu Ihrem GitHub Repository
- Klicken Sie auf den grünen "Code" Button
- Kopieren Sie die HTTPS URL (z.B. <https://github.com/IhrUsername/Astra.git>)
- Im Terminal:

```
powershell  
Apply to README.md  
Run
```

```
git remote add origin https://github.com/IhrUsername/Astra.git  
git branch -M main  
git push -u origin main
```

## 1. GitHub Authentifizierung:

- Beim ersten Push werden Sie nach Ihren GitHub-Anmeldedaten gefragt
- Verwenden Sie Ihren GitHub Benutzernamen und Passwort
- Wenn Sie die Zwei-Faktor-Authentifizierung aktiviert haben, müssen Sie einen Personal Access Token erstellen:
  1. Gehen Sie zu GitHub.com → Settings → Developer Settings → Personal Access Tokens
  1. Generieren Sie einen neuen Token mit repo-Berechtigungen
  1. Verwenden Sie diesen Token anstelle des Passworts

## Nach der GitHub-Einrichtung

Sobald das Repository eingerichtet ist, können Sie mit der regulären Entwicklung beginnen. Typischer Workflow:

## 1. Änderungen vornehmen:

```
powershell  
Apply to README.md  
Run
```

```
git add .  
git commit -m "Initial commit"
```

## 1. Branches für neue Features:

```
powershell  
Apply to README.md  
Run
```

```
git remote add origin https://github.com/IhrUsername/Astra.git  
git branch -M main  
git push -u origin main
```