Personal Automobile Mileage Based Auto Insurance

Introduction

In the past decade, insurance companies have been experimenting with new ways to rate personal automobile, PA, insurance policies. These new products allow PA policies to be rated by buyer driving behavior, colloquially called User Based Insurance, UBI. This can be a win-win for the companies and insureds since for the companies, premium is based on risk and for the insureds, safe driving habits can be rewarded. Acceptance by the driving public tends be muted by a catch-22 due to the Big Brother phenomena. Insureds want cheaper premiums, yet do not want an invasion of their privacy by whomever they perceive to be Big Brother at the moment.

Is there a way to modify how real-time exposure is determined to be more in line with driver behavior? Right now, in the midst of the Covid-19 pandemic many companies are offering rebate discounts due to the drastic change in driver habits. For example Progressive is offering a 20% rebate for insureds for April and May, 2020. This accomplishes several things. First, it provides insureds with some much needed cash. Second, Progressive gets some positive customer service points. In reality, it will cost Progressive only about 3.3% of full annual premiums if they don't extend the program.

In this document, I am proposing a modified PA Pay as You Drive program similar to that used by Worker's Compensation, WC. Generally, across most insurance lines of business, insureds pay monthly premiums based some predefined unit of exposure. For WC, premiums are based on their estimated annual payroll. At the end of the year, they are audited to state what their actual payroll was. Over time this has been modified for a pay-as-you-go plan where premiums can be paid weekly based on weekly payroll. This structure allows insureds to pay premiums commensurate with their current business economic conditions: payroll decreases/increases, insurance premiums decrease/increase. These plans are typically handled by the insured's payroll company who acts as their insurance agent or broker.

Proposed Policy Premium Management

Personal Automobile policies are typically based on uniform monthly installments for a 6 or 12 month term. Can the monthly premiums be variable instead of uniform month to month? I believe it is possible. My proposal is an entry point for full implementation of UBI by insurance companies. It would allow insureds to be in control of their own premium discounts without the feeling of "Big Brother" watching over them.

In its most simplistic form, insureds would pay their insurance for the full term and at the end of the term, submit their actual mileage. In its most aggressive form, at the end each month insureds would enter their mileage for the month. Based on the miles actually driven vs what the premium was based on, they would receive a credit, either in the form of a check or credit on a future premium installments. Because miles driven is an important risk factor, any improvement in a company's ability to capture this information will decrease the moral hazard of insureds underestimating actual mileage, thus leading to premiums reflective of the actual risk.

At the beginning of the policy period, the policy would be rated as usual with a base mileage determined. A discount table would be setup based on the percentage of base miles actually driven. In

my model and the current model used by most insurance companies, the moral hazard of underestimating annual miles driven still exists at policy inception. I propose that in my model, it can be reduced in the first renewal period, or maybe even eliminated; details will be provided later in this paper.

In the scenario where the insured provides the actual miles at the end of the first policy period, multiple outcomes are possible. If:

- Actual miles are less than the base miles, the insured can receive a credit.
- Actual miles are equal to the base miles, no credit is available. This is the current model, where no insured receives a credit for driving fewer miles.
- Actual miles are greater than base miles. This is similar to the above scenario, with the exception that in the next renewal period the base miles will increase to reflect the miles driven in the previous period.

With the more aggressive monthly reporting model, more programming on the company is needed to account for monthly credits. Policies will still be written annually or semi-annually, however the insured can receive credits on a monthly basis. Due to timing issues, there will be a delay of the credit(s). There are at least two different scenarios.

- First, the insured pays full premium for the policy period. Since, the premium is paid in full, the insured can receive monthly credits if applicable. Re-rating as described previously will apply.
- Second, the insured pays monthly installments. The cleanest solution would be to have the credits applied to the the premium installments. Due to timing considerations, the premium credits will skip a month. For example, since, January's credit will be determined when the February payment is due, January's credit will be applied to the March premium. These initial 2 months premium can be classified as a premium deposit for the term, thereby reducing moral hazard and proving incentive for the insured to submit the mileage.

Of course insureds will not always update the miles driven on the last day of the monthly cycle, so companies will need to develop their own way to prorate the miles driven should miles be submitted mid-month, quarterly, etc.

Reasons for an automated process

During this time of Covid-19, or any other shock, use of a simple phone app to update miles driven would be an automated way of achieving proper milage based rating. The current, manual way is for the customer call up their agent or company and change the rating from Drive to Work > 15 miles to Pleasure use, then back again when the pandemic is over. This is time consuming and wastes the resources of the insured, agent and company. Of course the insured has no incentive for the rating change to revert back once the crisis is over, so procedures must be put in place either on the company or agency side to ensure proper risk premiums.

It is estimated that 30% of insurance companies do not offer any UBI; this plan could ease them into this market without alienating any customers due to privacy concerns. Since this app would be

completely controlled by the insured and only collect mileage, it should alleviate privacy concerns. Over time, more data could be collected as insureds get used to this new idea.

There are at least two ways to institute this program, and it is predicated on the company offering what I call the standard classification system. Drivers are generally classified by Pleasure use, Drive to work <15 miles, Drive to work >15 miles, Business use, etc.

In the first method, drivers would be classified as currently practiced by many rating algorithms. An assumption by the company can be made for how many base miles are inherent in each classification. Changes in driving mileage credits can then be applied to insureds who drive less than the base miles. The credits can vary by classification, since an insured with a Pleasure use classification who is now stuck at home, has a much different change in risk profile compared to the one with a classification of Drive to Work >15 miles or Business use

An alternate method is to modify the base rate structure to remove all the previously stated classifications and have just one base rate. Other classification factors such as age or sex can still be used. This method would remove the case where an insured needs to be reclassified due to understating the miles driven; also as stated earlier, the moral hazard of underestimating miles driven is removed. For an installment policy, the first two months premiums can be marketed as a premium down payment, especially for new business, since subsequent premiums will reflect actual miles driven.

In the below table are two different ways of charging premiums for an insured expecting to receive a 20% credit for driving fewer than the base miles. $100 per month base premium for 6 months is assumed; if the premiums were $100 per month, base premium for the term is $600 and adjusted for a 20% credit (80% of final premiums) would be $480. The first scenario has the insured receiving a credit in the 7th month, whereas the premiums for months 2-4 are adjusted to ensure the final 6 month premium is correct in the second scenario. I have not shown any scenarios where insureds go over their allotted miles in a given month, since there are many scenarios. For example, in month 6, June, assume the insured is due to receive no credit and pay a premium of $100, in scenario #1 the premium would increase from $80 to $100, a $20 (+25%) increase, in scenario #2, the premium would rise from $70 to $100 a $30 (+43%) increase.

Now, think of a case where the insured drives a substantial amount of miles in the last month so that for the entire year his credit goes from 20% to 0%. Can the insurer collect the credits given after the policy period is over? For example in scenario #1, instead of receiving a credit of $40 at the end of the term, the insured would be required to pay $160 for excessive driving.

Scenario 1

	Jan	Feb	Mar	Apr	May	June	July	Total
premium paid at beginning of month	$100	$100	$80	$80	$80	$80	-$40	$480
credit earned at end of month	$20	$20	$20	$20	$20	$20		
credit owed to insured	$20	$40	$40	$40	$40	$40	$0	

Scenario 2

	Jan	Feb	Mar	Apr	May	June	July	Total
premium paid at beginning of month	$100	$100	$70	$70	$70	$70		$480
credit earned at end of month	$20	$20	$20	$20	$20	$20		
credit owed to insured	$20	$40	$30	$20	$10	$0	$0	

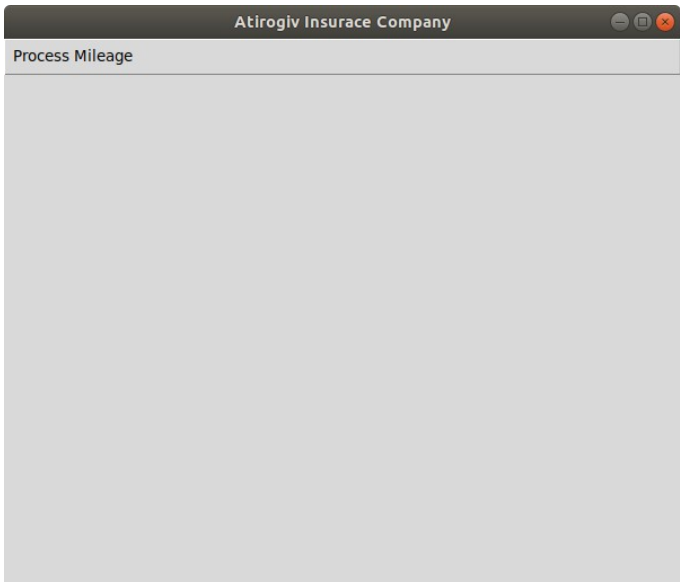
Other rating considerations to consider which may also be addressed, are territories and coverages. Typically differences in territories can be as simple as urban, suburban and rural. In a mileage based approach, instead of population density affecting insured losses, the types of miles driven can play a factor. Population density related losses have their roots in physics, where the more items in an area, the greater chance of collision between any of them. Miles driven in a gridlocked metropolitan area are different than those on the highway going to work and those on a desolated rural back road.

All of the coverages, Liability and Physical Damage are to some extent affected by miles driven. I believe that comprehensive coverage is muted somewhat when looking at the miles driven. That is because many of the perils occur regardless of milage, such as theft and weather related losses; while others are attributable to mileage such as glass damage and animal collision. Based on this, rather than a credit table applicable to all coverages, it might be worthwhile to have an independent table for each coverage.

APP

The App is simple and can be added to a company's existing app or can be a standalone. Right now since it is proof of concept, it is very simplistic. Since the insurance company needs to verify the mileage being submitted is for the correct vehicle and is the actual miles, I thought of two methods, one

of which is built into the current version of the App. Below is the 1st screen of the App, the user presses the “Process Mileage” button to begin.



The insurance company sends the insured a small QR code sticker that the insured can stick on his dashboard. The QR code contains the VIN # of the vehicle associated with the policy. The insured scans (takes a picture of) the QR code to begin the process. The alternative way is to take a picture of the VIN. This alternative way, is less desirable because it makes the insured take a picture in what could be inclement weather (VINs are readable from the outside of the vehicle) and the programming effort involved to convert the VIN to actual numbers requires some effort; computers don’t read numbers from a photo as easily as you and I, allowing for some possible error.

Photo of QR Code

QR Code translated to VIN #

After obtaining the VIN for the vehicle, the insured then takes a picture of the odometer and also physically enters the current mileage, as a check. As stated before, computers can't easily read numbers from a photo. In my example I took a photo of just the mileage, in practice, methods can be used to determine which set of numbers on the dash are the total miles as opposed to the trip miles, temperature, etc. In most of today's cars, digits are presented in seven segments: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 . What makes this difficult is that the number 1 can be misinterpreted as a ":" since it is made up of two vertical segments. For this reason the model must be trained to interpret the graphical representation of a number to a real number. Most digital numbers are fairly consistent in appearance. For older vehicles, those with rolling wheels for odometer numbers, a new set of training data will need to be developed. Since the average age of a car is about 12 years old and digital odometers became popular around 2000, this would be a small segment of the market.

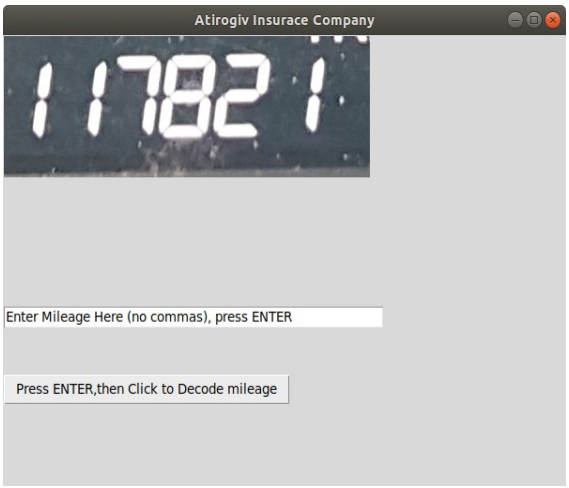
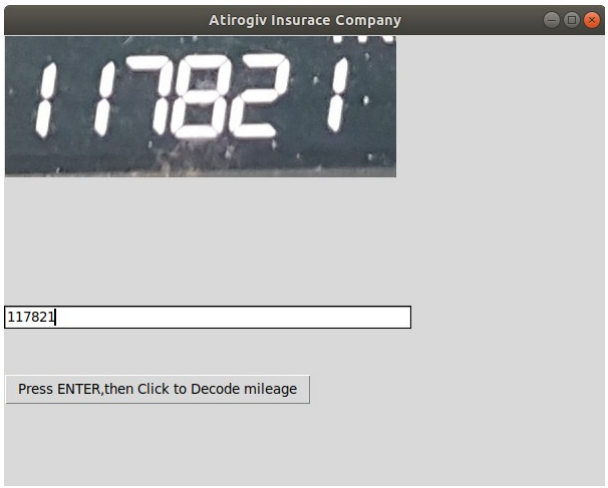
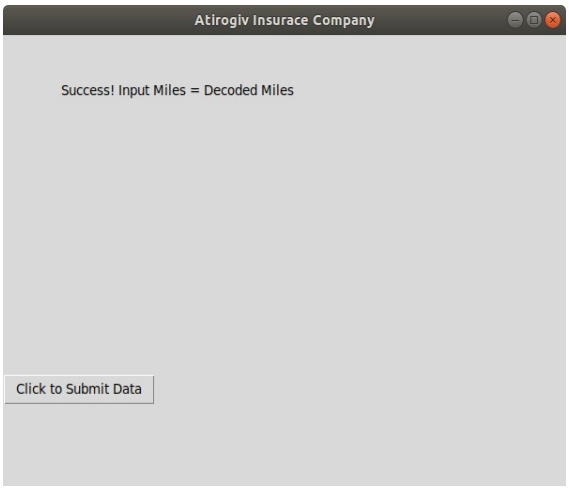


Photo of Odometer

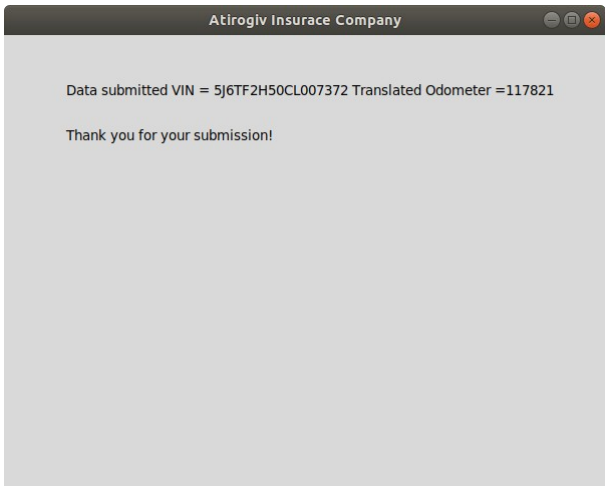


Mileage physically entered as a verification

Once the data is collected, the data would be submitted to the insurance company.



Converted miles is compared to those entered by the user



Simple submission screen

Of course the App can have many more features, but these are just the basics. From an insurance company point of view, the information needs to be trustworthy. From an insured point of view, simplicity and privacy issues are of utmost importance.

- For insurance companies, since the app will take control of the camera, the insured can't upload any picture they choose, modify the timestamp, etc. Thereby ensuring the photo was captured at that moment.

- For the insured, the app will be easy to use, require the insured open the app and take a picture of the QR code to begin. Due to privacy concerns while it would be possible to associate a policy with a given VIN, it is more desirable for the insured to enter their policy on the App, then have the App check that the VIN is actually on the policy.

Future APP Improvements

Since this is a proof of concept App all of the programming effort was directed to verifying that a QR code could be encoded with the VIN and that the odometer could be translated to readable text. In the future, it should be possible to take one photo and obtain all the information assuming that QR code is on the dashboard near the odometer. From here, two techniques are possible to obtain the VIN and Mileage.

- Method 1 would have the insured use some kind of indicating mechanism to highlight the QR code and the odometer.
- Method 2 would be entirely automated. Depending on how the photo taking parameters, i.e. only take a photo of the QR and odometer or the entire dashboard, will dictate the level of programming necessary for this. While it is a Catch-22, it is possible to use the VIN to determine the vehicle make, model, etc which in turn would tell what kind of dashboard layout

the vehicle has. A database of dashboard layouts would have to be developed to accomplish this. This could be used as a check to verify the dashboard being shown is actually the vehicle the insured says it is.

Regulatory Issues

I foresee several possible regulatory issues. How will insureds who do not own a smartphone obtain coverage though a program such as this? According to the latest PEW Research, 81% of US adults own a smartphone, from this the number of households should be a bit higher. If a household does not own a smart device there are multiple workarounds, two of which are:

- If the company uses telematic devices for other programs, the insured would use this, thus allowing the company to possibly collect other information (but not used in rating).
- Since inexpensive tablets and telematic devices are similarly priced, it might be worthwhile for a company to give these insureds a tablet to take the photos.

Another issue alluded to earlier, is how to handle insureds who drive more than their allotted miles? Do the mileage factors need to be capped at 1.00? Can they go higher? What about collecting premiums after the policy period? In WC, businesses are used to the audit process and pay premiums after the policy term. This concept, might be unpalatable to the general public.

Other Ideas

One last thought, is it possible to have this be a stand-alone company that collects the information from insureds, then uploads them to insurance companies for a transaction fee? Due to the costs involved in designing an App, it might be beyond the financial capabilities for smaller insurance companies, and therefore a centralized clearinghouse might be the solution.

App Code

In the following pages is the code for App written in Python. Several modules are required and the trained data set for converting the seven segment numbers into readable text.


```
# Python code for the App

from Tkinter import *
from PIL import Image, ImageTk
from pytesseract import image_to_string
import cv2 as cv
from pyzbar.pyzbar import decode

root = Tk()
root.geometry("600x450")

# This imports the QR code as an example of what will be scanned


def showQR():
    global load, QRFrame
    hide_all_frames()
#   this is the sample QR code
    QRFrame = Frame(root, width=600, height=450).pack(fill="both", expand=1)
    load = Image.open("QR-VIN.png")
    load = load.resize((100, 100), Image.ANTIALIAS)
    render = ImageTk.PhotoImage(load)
    img = Label(QRFrame, image=render)
    img.image = render
    img.place(x=0, y=0)
#   this is a button to simulate the QR code has been scanned and to translate it
    QRButton = Button(QRFrame, text="Click to decode QR code", command=DecodeQR)
    QRButton.place(relx=0.0, rely=0.75)

# this decodes the QR code that was scanned


def DecodeQR():
    global QR
    QR = decode(load)
    hide_all_frames()
    QRtext = 'Translated QR for VIN # is '+(QR[0].data.decode('ascii'))
    vin_text = Label(QRFrame, text=QRtext)
    vin_text.place(relx=0.0, rely=0.6)
#   This button simulates ending the VIN capture and capturing the mileage
    GoToMilesButton = Button(QRFrame, text="Click to get mileage", command=showMiles)
    GoToMilesButton.place(relx=0.0, rely=0.75)

#   this binds the entry by the user to global variable MUST HIT ENTER!!!!!
```

```
def captureMiles(en):
    global inMiles
    inMiles = inputMiles.get()

# Take photo of mileage

def showMiles():
    hide_all_frames()
    global odom, MilesFrame, inMiles, inputMiles
    MilesFrame = Frame(root, width=600, height=450).pack(fill="both", expand=1)
    load = Image.open("/home/charlie/vinnie/DashCrop.jpg")
    render = ImageTk.PhotoImage(load)
    img = Label(MilesFrame, image=render)
    img.image = render
    img.place(relx=0, rely=.0)

    inputMiles = Entry(MilesFrame, width=50)
    inputMiles.place(relx=0.0, rely=0.6)
    inputMiles.insert(0, "Enter Mileage Here (no commas), press ENTER")
    inputMiles.bind('<Return>', captureMiles)

#   this calls the function to translate the captured image to a text string
    GoDecodeMilesButton = Button(MilesFrame,
                          text="Press ENTER,then Click to Decode mileage",
                          command=DecodeMiles)
    GoDecodeMilesButton.place(relx=0.0, rely=0.75)

#   This converts the image to a string of numbers

def DecodeMiles():
    global odom
    hide_all_frames()
    img = cv.imread("DashCrop.jpg")
    Mileage = image_to_string(
        img,
        lang='ssd',
        config='--oem 3 --psm 6 tessedit_char_whitelist=0123456789')

    odom = ("Translated Odometer =" + Mileage)

#   Quality check to see if the entered #'s match the translated #'s
    if inMiles == Mileage:
        milesText = "Success! Input Miles = Decoded Miles"
    else:
        milesText = "Error! Input Miles <> Decoded Miles"
```

```
    text = Label(MilesFrame, text=milesText)
    text.place(relx=0.1, rely=0.1)

#   this calls the final button to submit the data
    SubmitButton = Button(MilesFrame, text="Click to Submit Data", command=submit)
    SubmitButton.place(relx=0.0, rely=0.75)


# Submit all data to company
def submit():
    hide_all_frames()
    submitFrame = Frame(root, width=600, height=450).pack(fill="both", expand=1)
    vin_text = str("VIN = "+(QR[0].data.decode('ascii')))
    submit_text = ("Data submitted " + vin_text + " " + odom)
    text = Label(submitFrame, text=submit_text)
    text2 = Label(submitFrame, text="Thank you for your submission!")
    text.place(relx=0.1, rely=0.1)
    text2.place(relx=0.1, rely=0.2)

# Hide all previuos Frames
def hide_all_frames():
    for widget in root.winfo_children():
        widget.destroy()


my_menu = Menu(root)
root.config(menu=my_menu)

photos_menu = Menu(my_menu)
my_menu.add_cascade(label="Process Mileage", menu=photos_menu)

photos_menu.add_command(label="Begin Process", command=showQR)

photos_menu.add_separator()
photos_menu.add_command(labe="Exit", command=root.quit)

# Create Frames
QRFrame = Frame(root, width=600, height=450)
MilesFrame = Frame(root, width=600, height=450)
submitFrame = Frame(root, width=600, height=450)

root.title("Atirogiv Insurace Company")

root.mainloop()
```